IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received June 12, 2020, accepted June 23, 2020, date of publication June 29, 2020, date of current version March 19, 2021.

Digital Object Identifier 10.1109/ACCESS.2020.3005452

Kapur's Entropy for Underwater Multilevel Thresholding Image Segmentation Based on Whale Optimization Algorithm

ZHEPING YAN, JINZHONG ZHANG, ZEWEN YANG, AND JIALING TANG
College of Intelligent Systems Science and Engineering, Harbin Engineering University, Harbin 150001, China
Corresponding author: Jinzhong Zhang (zhangjinzhongz@126.com)

This work was supported in part by the National Nature Science Foundation of China under Grant 51679057, and in part by the Province Science Fund for Distinguished Young Scholars under Grant J2016JQ0052.

ABSTRACT Multilevel thresholding is an effective and indispensable technology for image segmentation that has attracted extensive attention in recent years. However, the multilevel thresholding method has some disadvantages, such as a large computational complexity and low segmentation accuracy. Therefore, this paper proposes a whale optimization algorithm (WOA) based on Kapur's entropy method to solve the image segmentation problem. The WOA can effectively balance exploration and exploitation to avoid falling into premature convergence and obtain the global optimal solution. To verify the segmentation performance of the WOA, a series of experiments on underwater images from the experimental pool of Harbin Engineering University are conducted, and the segmentation results are compared with those of the BA, the FPA, MFO, the MSA, PSO and WWO by maximizing the fitness value of Kapur's entropy method. The fitness value, peak signal-to-noise ratio (PSNR), structure similarity index (SSIM), execution time and Wilcoxon's rank-sum test are used to evaluate the overall performance of each algorithm. The experimental results reveal that the WOA is superior to the other comparison algorithms and has a higher segmentation accuracy, better segmentation effect and stronger robustness. In addition, the feasibility and efficiency of the WOA are verified.

INDEX TERMS Multilevel thresholding, whale optimization algorithm, image segmentation, Kapur's entropy.

I. INTRODUCTION

In recent years, unmanned underwater vehicles (UUVs) with vision systems have been widely used to gather image information for analysis and research, in which underwater image segmentation is difficult to accomplish in machine vision. The quality of image segmentation directly affects the accuracy of target feature extraction and target detection [1]–[3]. The three-dimensional model of a UUV equipped with a vision system is given in Fig 1. Image segmentation is a fundamental and important technology in image processing and robotic vision, which is a key operation from image processing to image analysis, as well as one of the key target feature extraction, recognition and tracking. The purpose is to segment a given original image into several independent and special-quality parts with respect to feature, colour, texture,

FIGURE 1. Three-dimensional model equipped with a vision system.

histogram, grayscale and edge, and then to extract the target of interest. The grey values of pixels in same area are approximately the same, whereas, the grey values of pixels in different areas are significantly different. The quality of image segmentation directly affects the stability and reliability of the

The associate editor coordinating the review of this manuscript and approving it for publication was Huazhu Fu.

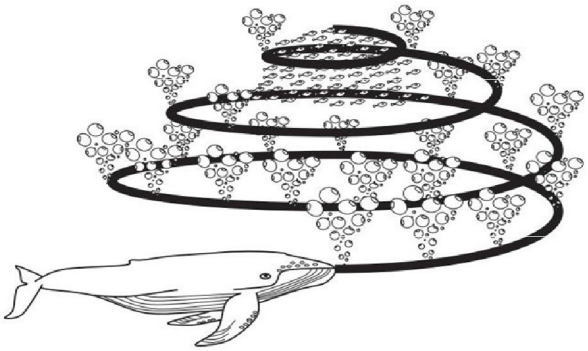

FIGURE 2. Bubble-net feeding behaviour of humpback whales.

feature extraction and the object recognition. The common methods of image segmentation include the thresholding-based method, region-based method, edge-based method, clustering-based method and graph-based method [4]–[8]. The thresholding-based method has been widely used to solve the image segmentation problem. Bi-level thresholding and multilevel thresholding are applied to process the simple images and complex images, respectively [9], [10]. Multilevel thresholding can find accurate target regions and obtain the segmentation effects. Various techniques have been proposed to determine the appropriate thresholds [11], such as Otsu's method, Kapur's entropy, minimum cross entropy, fuzzy entropy, Tsallis entropy, Shannon entropy and Renyi entropy. Kapur's entropy is an effective and feasible image processing technology and is used to carry out image segmentation. As the thresholds increase, the computational complexity increases exponentially. To overcome this shortcoming, many meta-heuristic algorithms are used to solve the multi-threshold image segmentation, such as the bat algorithm (BA) [12], flower pollination algorithm (FPA) [13], moth flame optimization (MFO) [14], moth swarm algorithm (MSA) [15], particle swarm optimization (PSO) [16], and water wave optimization (WWO) [17].

Zhou *et al.* applied the moth swarm algorithm based on Kapur's entropy method to solve the multi-threshold image segmentation problem, and the proposed method has proved the robustness and effectiveness according to numerical experimental results and image segmentation results [18]. Aziz *et al.* designed a whale optimization algorithm and moth flame optimization algorithm to improve the multi-threshold search ability, and the methods provided a good balance between exploration and exploitation in all segmentation images and obtained a better segmentation effect [19]. Quadfel *et al.* proposed the social spider algorithm and flower pollination algorithm to optimize the objective function value and Kapur's method to perform multi-threshold image segmentation [20]. Díaz-Cortés *et al.* tried to solve the multi-level thresholding image segmentation problem using a dragonfly algorithm, and the experimental results indicated that the proposed algorithm obtained better optimization performance [21]. Sambandam *et al.* provided a self-adaptive dragonfly algorithm based on Kapur's entropy

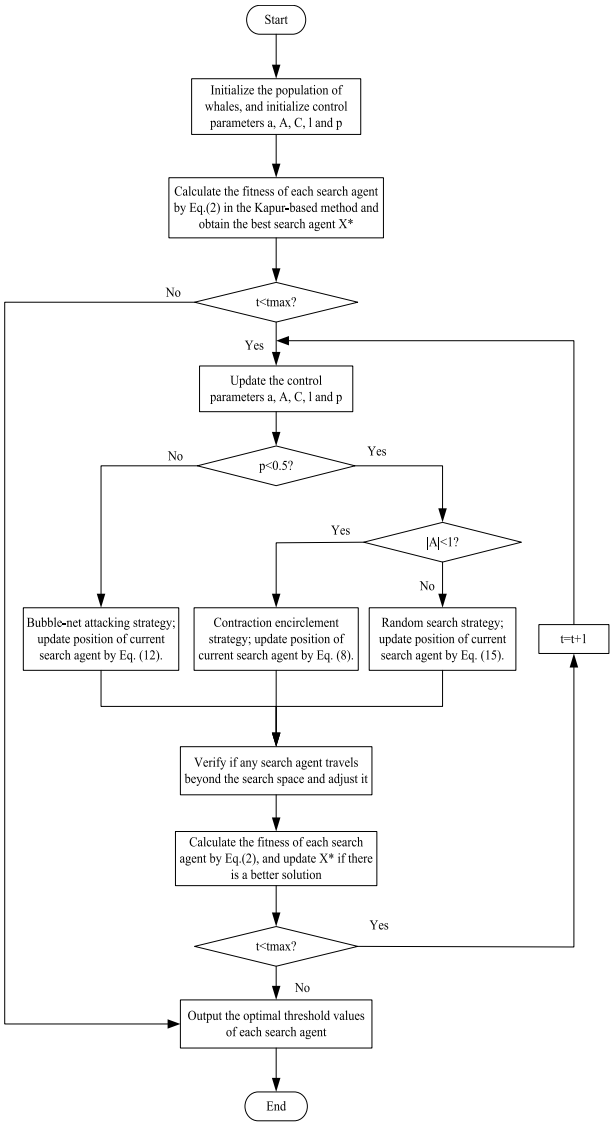

FIGURE 3. Flowchart of the WOA for multilevel thresholding.

method for image segmentation, and the results indicated that the algorithm achieved higher segmentation accuracy and strong robustness [22]. Sun *et al.* proposed a hybrid optimization algorithm comprising the gravitational search algorithm and genetic algorithm to solve the image segmentation problem. The hybrid algorithm achieved complementary advantages and found the global optimal solution [23]. Shen *et al.* achieved multi-threshold image segmentation through a modified flower pollination algorithm, and the proposed algorithm had a strong global search ability and obtained the best segmentation effect [24]. Gao *et al.* introduced an improved artificial bee colony algorithm to solve the multi-threshold image segmentation problem, and the algorithm had high segmentation accuracy and fast convergence efficiency [25]. Pare *et al.* conducted a study on the firefly algorithm and Lévy flight strategy to conduct image segmentation, and the results showed that the proposed algorithm obtained the best threshold values and segmented

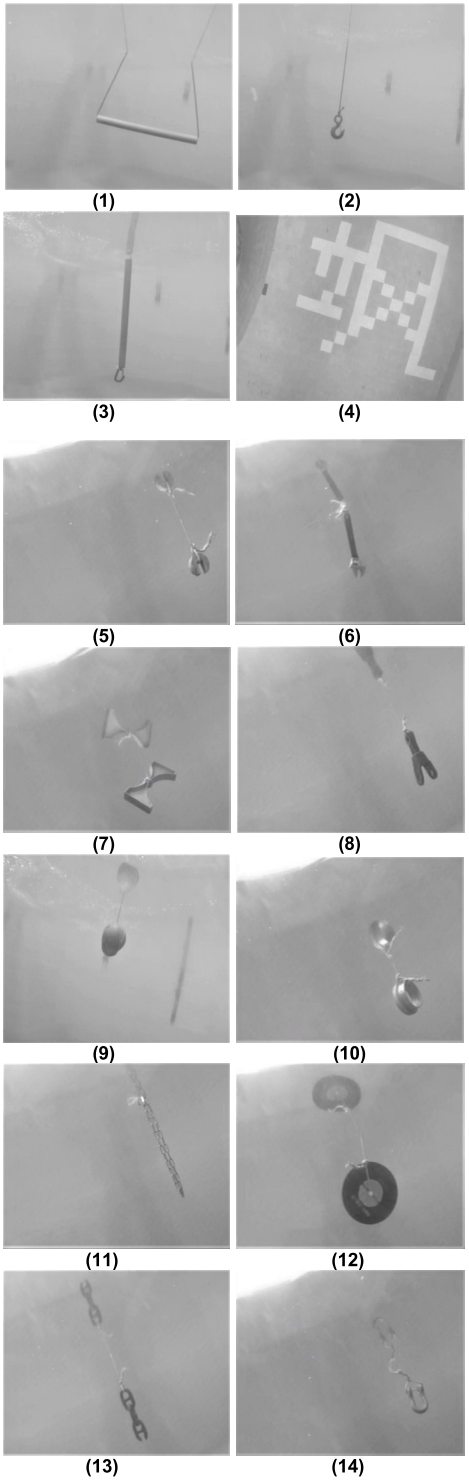

FIGURE 4. Original test images.

images [26]. Satapathy *et al.* combined the bat algorithm with the chaotic strategy for image segmentation, and the results verified the stability and feasibility of that combination [27]. Akay *et al.* conducted a study on the particle swarm optimization algorithm and artificial bee colony algorithm for image segmentation, and the overall optimization performance of

TABLE 1. Initial parameters of all algorithms.

Algorithm	Parameter	Values
BA [12]	Pulse frequency range f	[0,2]
	Echo loudness A	0.25
	Decreasing coefficient γ	0.5
FPA [13]	Switch probability ρ	0.8
MFO [14]	Constant b	1
	Random number t	[-1,1]
	Random number r	[-2,-1]
MSA [15]	Random number θ	[-2,1]
	Random number ε_2	[0,1]
	Random number ε_3	[0,1]
	Random number r_1	[0,1]
	Random number r_2	[0,1]
PSO [16]	Constant inertia ω	0.3
	First acceleration coefficient c_1	1.4962
	Second acceleration coefficient c_2	1.4962
WWO [17]	Wavelength λ	0.5
	Wave height $h_{\max}$	6
	Wavelength reduction coefficient α	1.0026
	Breaking coefficient β	[0.001,0.25]
	Maximum number $k_{\max}$ of breaking directions	$\min(12, D/2)$
WOA [38]	Random number r_1	[0,1]
	Random number r_2	[0,1]
	Convergence factor α	[0,2]
	Constant coefficient b	1
	Random number l	[-1,1]

the proposed algorithms was significantly better than that of other algorithms [28]. Bao *et al.* proposed the colour image multilevel thresholding method based on the Harris hawks optimization algorithm for image segmentation, and the image segmentation effect of the proposed algorithm was the best [29]. Jia *et al.* applied a modified moth flame algorithm to solve the multilevel thresholding image segmentation problem, and the results indicated that the proposed algorithm found the optimal objective fitness values and achieved the best segmentation effect [30]. Bohat *et al.* tried to combine the TH heuristic with swarm intelligence algorithms for colour image segmentation, and the proposed algorithms had superior segmentation performance [31]. Lu *et al.* employed and validated a neutrosophic c-means clustering method, which was effective and able to solve the image segmentation problem [32]. Li *et al.* presented the MapReduce-based fast fuzzy c-means algorithm to conduct large-scale underwater image segmentation, and the results indicated that the proposed algorithm obtained the best solution [33]. Elaziz *et al.* created a multi-objective multi-verse optimizer

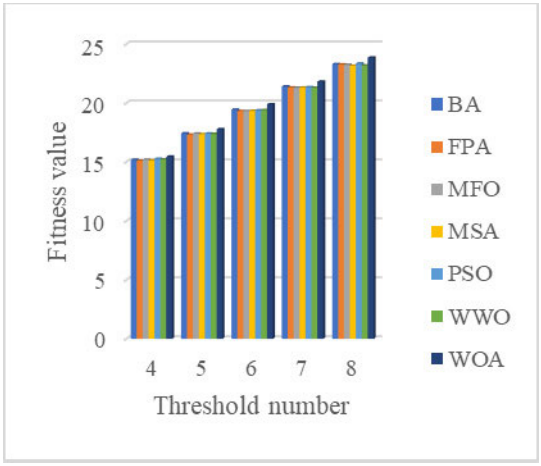

FIGURE 5. The best fitness values of the algorithms over all images.

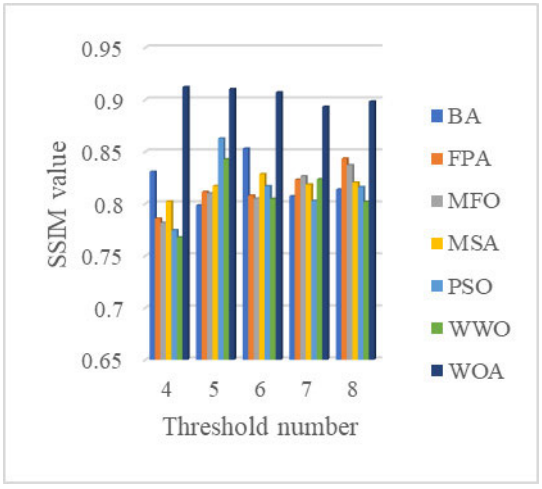

FIGURE 7. The average of SSIM of the algorithms over all images.

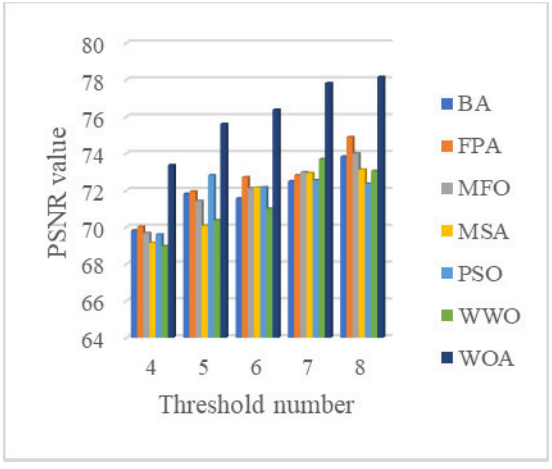

FIGURE 6. The average of PSNR of the algorithms over all images.

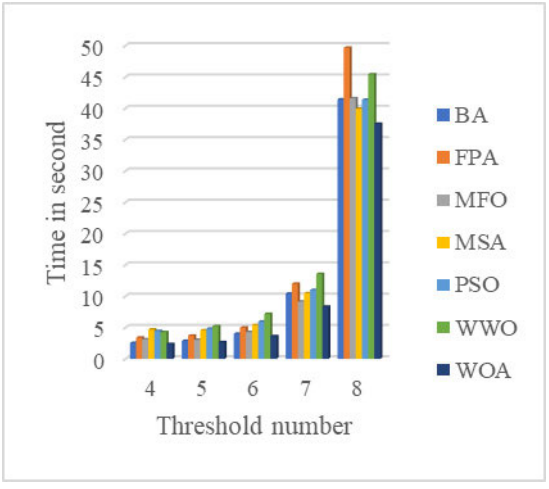

FIGURE 8. The average time of the algorithms over all images.

to solve the image segmentation problem and obtained the optimal fitness value by maximizing the Kapur and Otsu objective functions [34]. Oliva *et al.* proposed a BNMTH method for multilevel image segmentation, and the results indicated that the BNMTH method had better segmentation performance [35]. Rodríguez-Esparza *et al.* proposed the Harris hawks optimization algorithm based on the minimum cross-entropy to solve the image segmentation problem, which was found to be an effective and feasible method [36]. Alwerfali *et al.* combined the salp swarm algorithm with fuzzy entropy for multilevel image thresholding, and the results indicated that the segmentation performance of the proposed algorithm is significantly better than that of other algorithms [37].

The whale optimization algorithm (WOA) is mainly used to simulate the bubble-net attacking behaviour of humpback whales in nature for global search and contains three important operations: encircling of the prey, the bubble-net attacking method and the search for prey [38]. The WOA has faster convergence speed and higher calculation accuracy. The WOA based on Kapur's entropy method is used to solve the underwater multilevel thresholding image segmentation problem. The proposed algorithm can effectively avoid premature convergence and balance exploration and exploitation to obtain the optimal fitness value. Fourteen underwater images are used as test objects to verify the optimization performance of the WOA in solving the image segmentation problem, it is compared in this paper with other algorithms, i.e., the BA [12], the FPA [13], MFO [14], the MSA [15], PSO [16], and WWO [17]. The experimental results reveal that the WOA not only achieves a better segmentation effect and strong robustness, but also is an effective and feasible segmentation method. This research work will lay a good foundation for image extraction and image recognition.

This article is divided into the following sections: Section 2 introduces the multilevel thresholding. Section 3 reviews the WOA. In Section 4, the proposed WOA-based multilevel threshold method is described in detail. The experimental results and analysis are presented in Section 5. Finally, the conclusions and future research directions are discussed in Section 6.

II. MULTILEVEL THRESHOLDING

Image threshold segmentation is mainly divided into two categories: a bi-level thresholding method and a multilevel thresholding method. The bi-level thresholding method involves a threshold value, which divides the image into a foreground and background to process simple images. The multilevel thresholding method is a crucial and unsupervised image processing technology that can not only solve complex image segmentation but also achieve better segmentation results.

Kapur's entropy method is a nonparametric threshold technique, which is applied to classify the image into the multiple classes by comparing the entropy of histogram, and a higher entropy value indicates more homogeneous classes. The proposed method has attracted the attention of many scholars and has been proved to be superior than other thresholding-based methods. The Kapur's entropy method has the following unique advantages: low required number of computations, easy implementation, strong stability, fast processing speed, and high segmentation accuracy. The entropy of a given image indicates the compactness and separateness among distinctive classes. Kapur's entropy can effectively obtain the optimal threshold values by maximizing the objective function and has been widely applied to accomplish image segmentation [39]. Assume that n threshold values from the optimal threshold values $[t_1, t_2, \ldots, t_n]$ are used to divide the image into various classes. The probability p_i can be defined as:

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

where i denotes the grey level, h_i denotes the number of pixels, N denotes the total number of pixels, and L denotes the number of levels.

Kapur's entropy can be defined as:

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

where

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

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

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

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

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

III. WOA

The WOA is a novel swarm intelligence optimization algorithm based on the bubble-net attacking behaviour of humpback whales, which mainly simulates encircling of the prey, the bubble-net attacking method and the search for prey to perform an efficient global search. In the WOA, each humpback whale denotes a candidate solution. The search mechanism of the WOA is used to screen the candidate solutions to obtain the global optimal solution. The model of the bubble-net feeding behaviour is given in Fig. 2.

A. ENCIRCLING PREY

A humpback whale can find the position of prey and quickly encircle them. The position of the optimal solution is unknown, and it is assumed that the current position of the humpback whale is the target prey or a suboptimal solution. After determining the position of the optimal humpback whale, other whales will update their positions according to the optimal position. The position update can be defined as:

$$\vec{D} = \left| \vec{C} \cdot \vec{X^*}(t) - \vec{X}(t) \right| \tag{7}$$

$$\vec{X}(t+1) = \vec{X^*}(t) - \vec{A} \cdot \vec{D} \tag{8}$$

where t denotes the current iteration, X^* denotes the position vector of the optimal solution, X denotes the current position vector, $||$ denotes the absolute value, and $\cdot$ denotes an element-wise multiplication. $\vec{A}$, $\vec{C}$ denote coefficient vectors and can be defined as:

$$\vec{A} = 2\vec{a} \cdot \vec{r} - \vec{a} \tag{9}$$

$$\vec{C} = 2 \cdot \vec{r} \tag{10}$$

where $\vec{r}$ denotes a random vector in [0, 1] and $\vec{a}$ denotes a linear decrease from 2 to 0.

B. BUBBLE-NET ATTACKING STRATEGY (EXPLOITATION PHASE)

The bubble-net attacking strategy can be divided into the shrinking encircling mechanism and the spiral position updating. The shrinking encircling mechanism is used to reduce the distance between the global optimal position and the current optimal position according to a random vector $\vec{A}$ and control variable a. The humpback whales swim towards their prey according to the updated spiral position and capture the prey by calculating the distance between the whale position and the prey position. The position updating can be defined as:

$$\vec{D'} = \left| \vec{X^*}(t) - \vec{X}(t) \right| \tag{11}$$

$$\vec{X'}(t+1) = \vec{D'} \cdot e^{bl} \cos(2\pi l) + \vec{X^*}(t) \tag{12}$$

where $\vec{D'}$ denotes the distance between whale and prey, l denotes a random number in $[-1, 1]$, and b denotes a constant for defining the shape of the logarithmic spiral.

Humpback whales update their positions to capture prey, and there is a probability of 50% that either the shrinking encircling mechanism or the logarithmic spiral position

updating is carried out. The position updating can be defined as:

$$\overrightarrow{X}(t+1) = \begin{cases} \overrightarrow{X^*}(t) - \overrightarrow{A} \cdot \overrightarrow{D} & if\ p < 0.5 \\ \overrightarrow{D'} \cdot e^{bl} \cdot \cos(2\pi l) + \overrightarrow{X^*}(t) & if\ p > 0.5 \end{cases} \quad (13)$$

where p denotes a random number in [0, 1].

C. SEARCH FOR PREY (EXPLORATION PHASE)

To avoid falling into the local optimum, the WOA performs a random search strategy to find the prey by adjusting the vector $\overrightarrow{A}$. If $|\overrightarrow{A}| > 1$, the WOA has a strong exploration ability to obtain the global optimal solution. The position updating can be defined as:

$$\overrightarrow{D} = \left| \overrightarrow{C} \cdot \overrightarrow{X_{rand}} - \overrightarrow{X} \right| \quad (14)$$

$$\overrightarrow{X}(t+1) = \overrightarrow{X_{rand}} - \overrightarrow{A} \cdot \overrightarrow{D} \quad (15)$$

where $\overrightarrow{X_{rand}}$ denotes a random position vector (a random whale) selected from the current population.

The WOA can balance the exploration ability and the exploitation ability to find the global optimal solution. To better describe the solution process, the pseudo-code of the WOA is shown in Algorithm 1.

Algorithm 1 WOA Algorithm

Begin
Step 1. Initialize the whale population $X_i(i = 1, 2, \ldots, n)$
Step 2. Calculate the fitness of each search agent
Obtain the best search agent X^*
Step 3. while ($t < t_{\max}$) do
for each search agent
Update a, A, C, l, and p
if1 ($p < 0.5$)
if2 ($|A| < 1$)
Update the position of the current search agent by Eq. (8)
else if2 ($|A| \geq 1$)
Select a random search agent (X_{rand})
Update the position of the current search agent by Eq. (15)
end if2
else if1 ($p \geq 0.5$)
Update the position of the current search agent by Eq. (12)
end if1
end for
Check if any search agent goes beyond the search space and amend it
Calculate the fitness of each search agent
Update X^* if there is a better solution

$$t = t + 1$$

end while
return X^*
End

IV. WOA-BASED MULTILEVEL THRESHOLD METHOD

For each whale or search agent, its position denotes the threshold value of a given image segmentation. A whale changes its position to obtain the optimal solution by setting the threshold level. The WOA based on image segmentation is shown in Algorithm 2. The flowchart of the WOA for multilevel thresholding is shown in Fig. 3.

Algorithm 2 WOA Based on Image Segmentation for Kapur Entropy

Begin
Step 1. Initialize the whale population $X_i(i = 1, 2, \ldots, n)$
Step 2. Calculate the fitness of each search agent by Eq. (2) for the Kapur-based method
Obtain the best search agent X^*
Step 3. while ($t < t_{\max}$) do
for each search agent
Update a, A, C, l, and p
if1 ($p < 0.5$)
if2 ($|A| < 1$)
Update position of the current search agent by Eq. (8)
else if2 ($|A| \geq 1$)
Select a random search agent (X_{rand})
Update position of the current search agent by Eq. (15)
end if2
else if1 ($p \geq 0.5$)
Update position of the current search agent by Eq. (12)
end if1
end for
Check if any search agent goes beyond the search space and amend it
Calculate the fitness of each search agent by Eq. (2) for the Kapur-based method
Update X^* if there is a better solution

$$t = t + 1$$

end while
return the bset search agent X^*, which denotes the optimal threshold values of segmentation
End

A. TIME COMPLEXITY

The proposed algorithm contains five major steps: initialization, encircling the prey (exploration phase), attacking the prey (exploitation phase), updating the whales' positions and halting judgement. If the population size is N, the maximum number of iterations is T, and the dimensionality of the problem is D, then the time complexity of the WOA is given as follows: the initialization step contains a double loop (N and D times); thus, the time complexity is $O(N * D)$. For the exploration phase, there is a triple loop (N, D and T times), the time complexity of which is $O(N * D * T)$. For the exploration phase, there is a triple

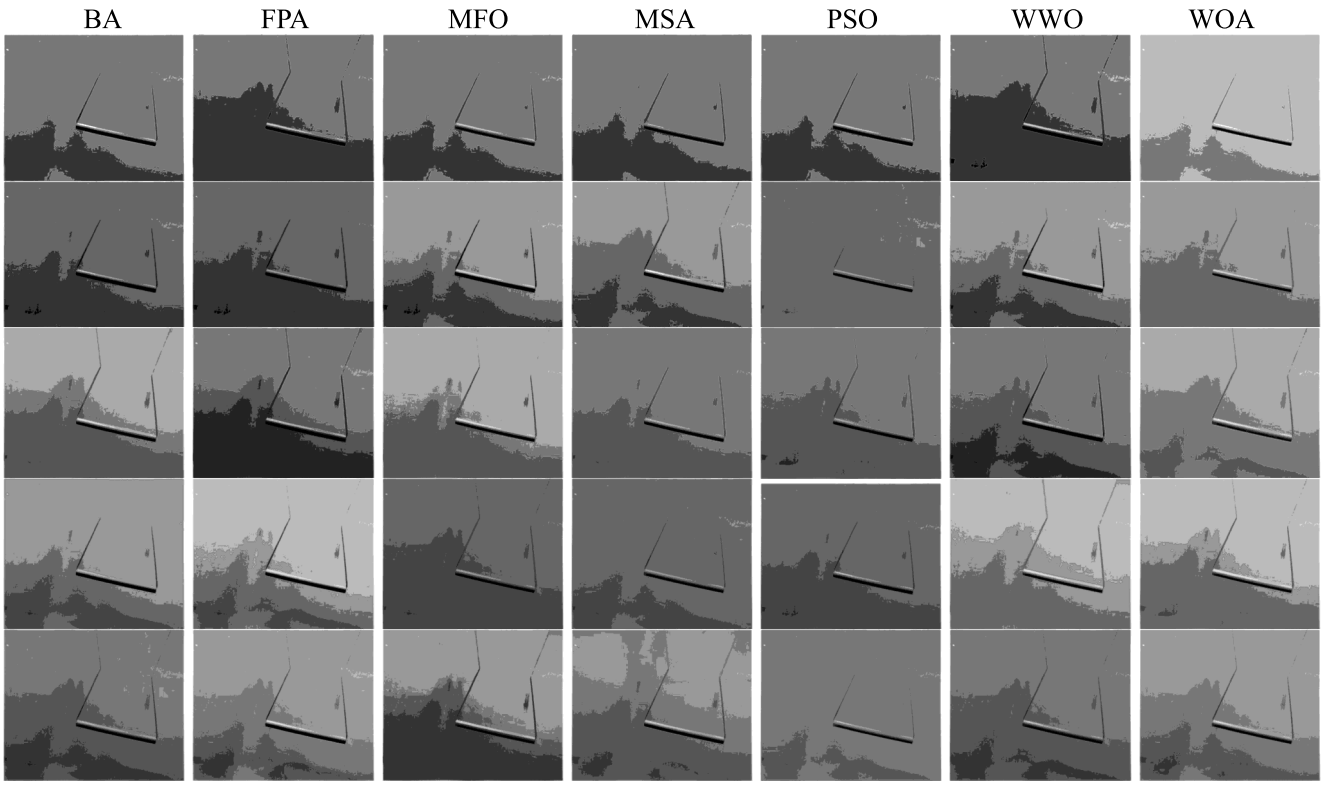

FIGURE 9. Segmented images of Test 1.

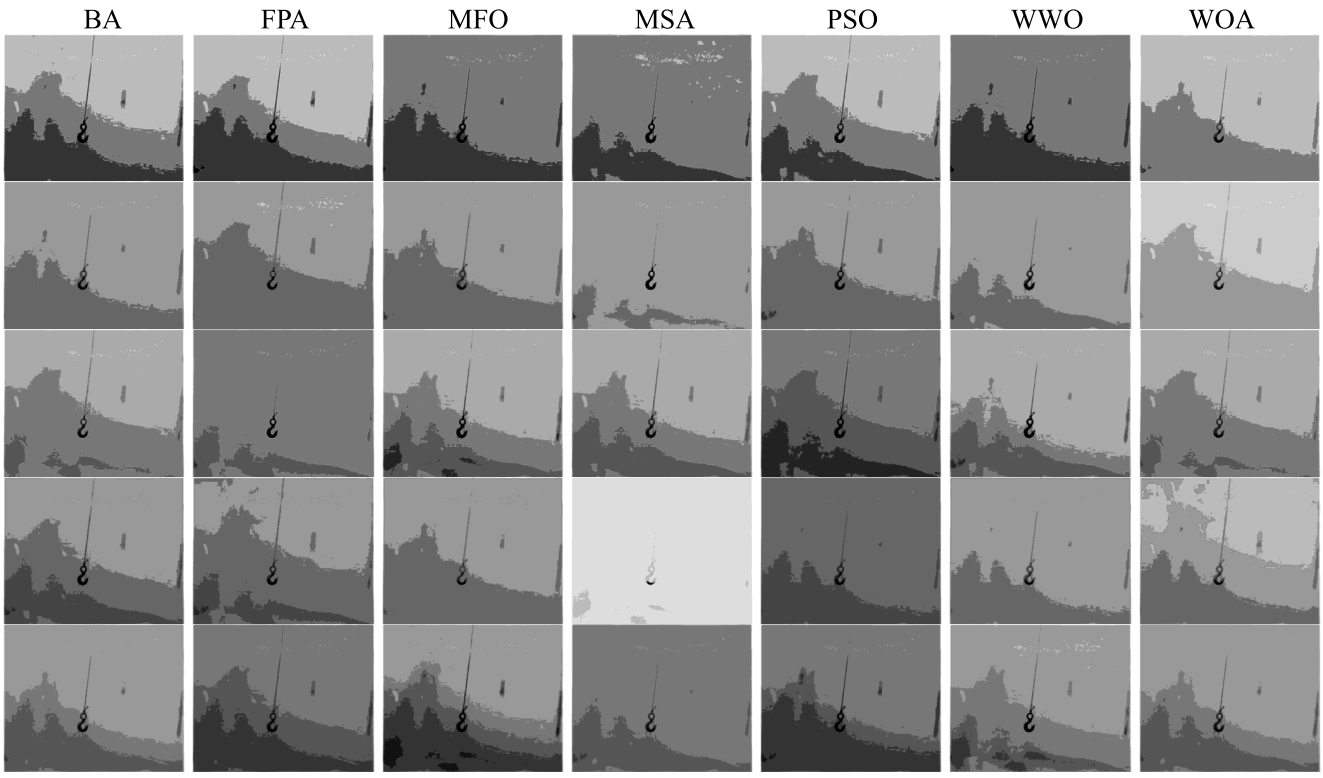

FIGURE 10. Segmented images of Test 2.

loop (N, D and T times), the time complexity of which is $O(N * D * T)$. For the fourth step, there is a triple loop (N, D and T times), the time complexity of which is $O(N * D * T)$. To halt the judgement, the time complexity is $O(1)$. Thus, the total time complexity of the WOA is $O(N * D * T)$.

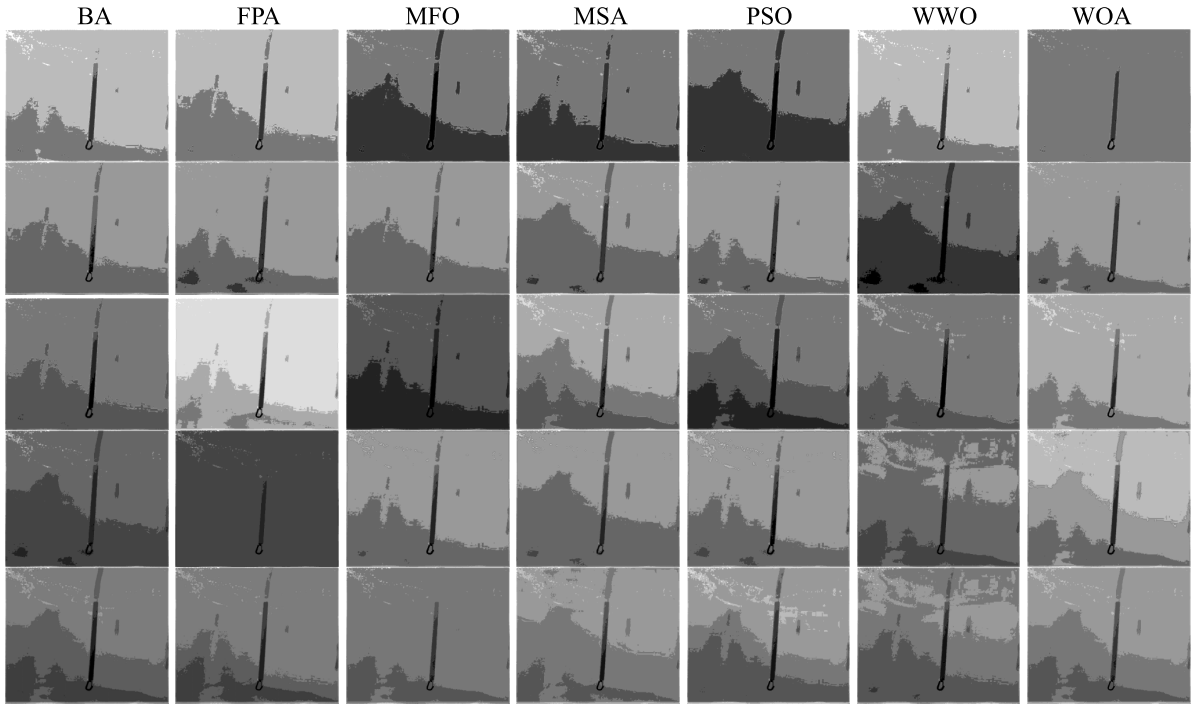

FIGURE 11. Segmented images of Test 3.

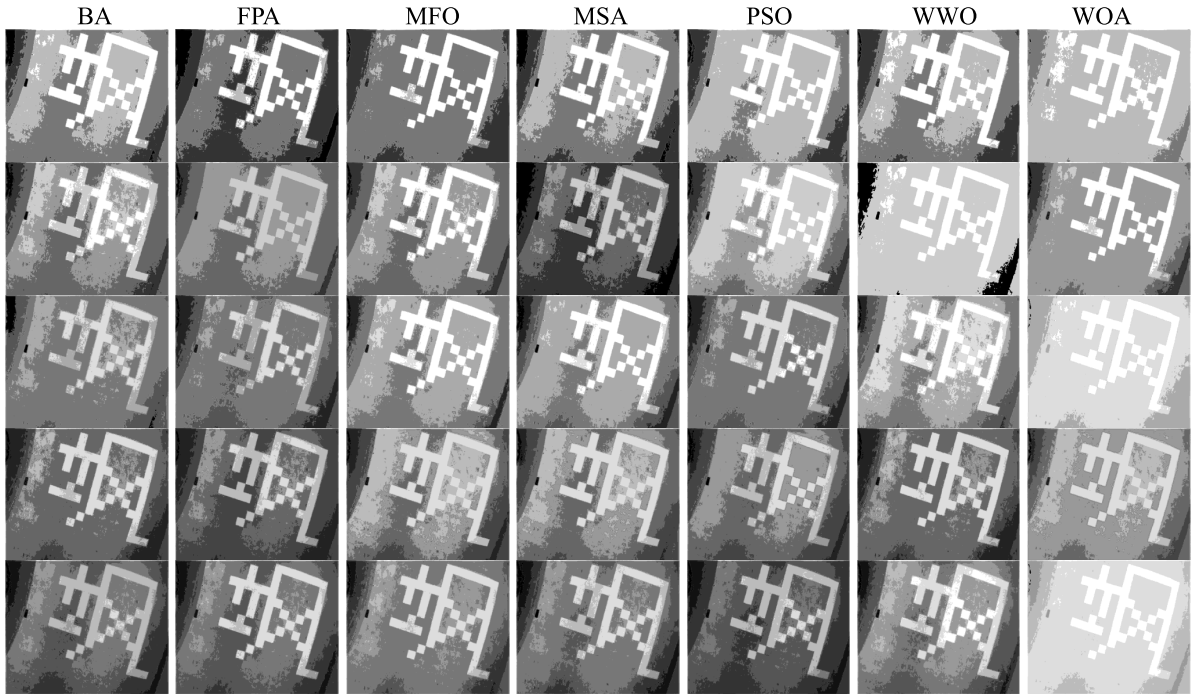

FIGURE 12. Segmented images of Test 4.

B. SPACE COMPLEXITY

The space complexity of an algorithm is defined as the storage space consumed by the algorithm. The WOA adopts N search agents to calculate the space complexity, and the dimensionality of the solved problem is D. Therefore, the total space complexity of the WOA is $O(N * D)$; hence, the space efficiency of the WOA is good.

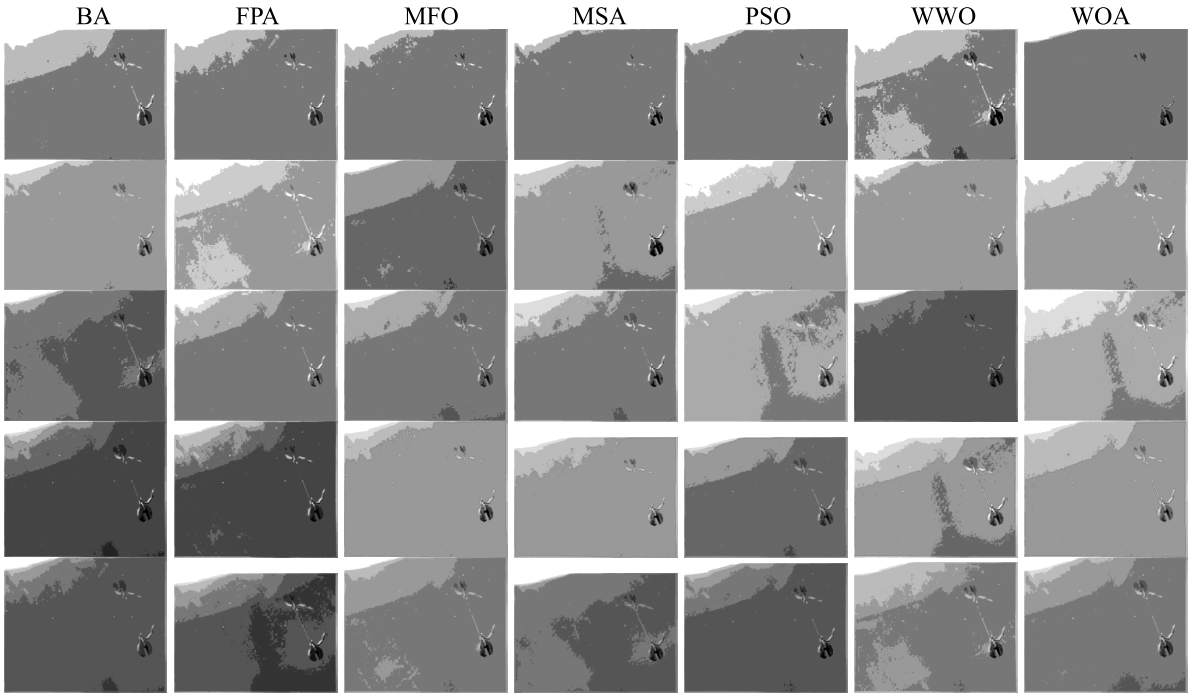

FIGURE 13. Segmented images of Test 5.

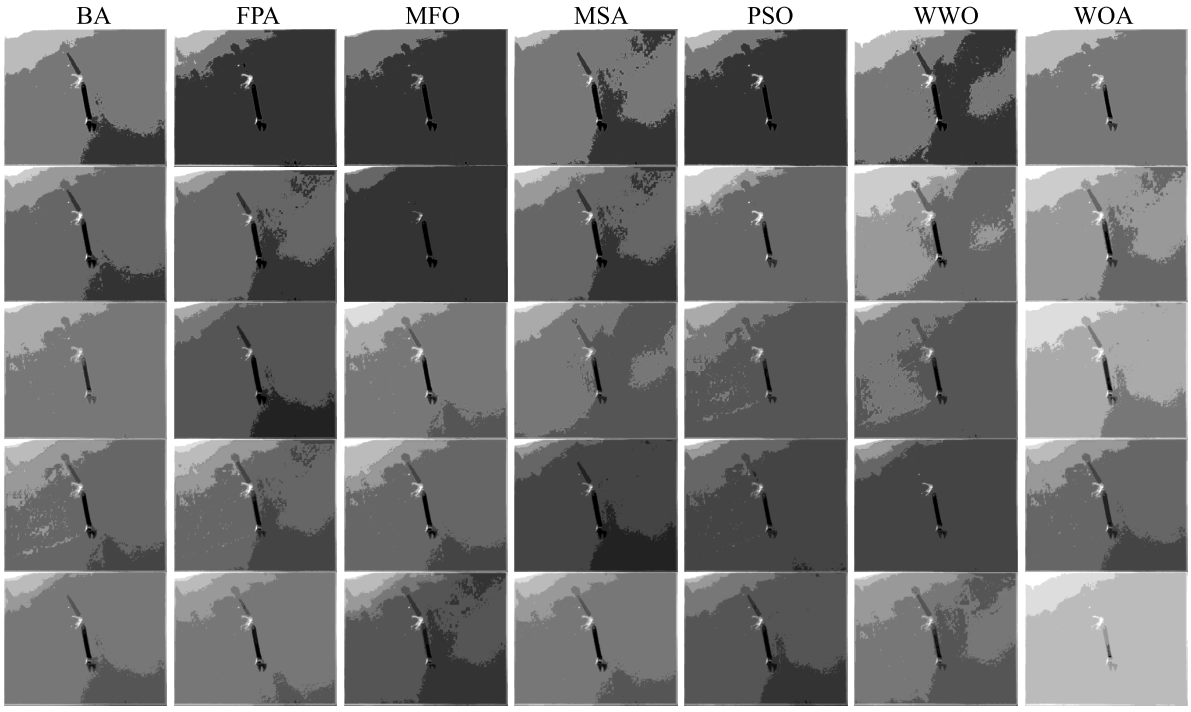

FIGURE 14. Segmented images of Test 6.

V. EXPERIMENTAL RESULTS AND ANALYSIS

A. EXPERIMENTAL SETUP

All the algorithms are programmed in MATLAB R2018b, and the numerical experiment is set up on an i7-8750H 2.2 GHz processor with 8 GB of memory.

B. TEST IMAGES

The computer vision system and the image processing technology are inseparable, and they are used to solve reconstruction and modeling, tracking and segmentation, and mitigating unnecessary image artifacts. The processing of the

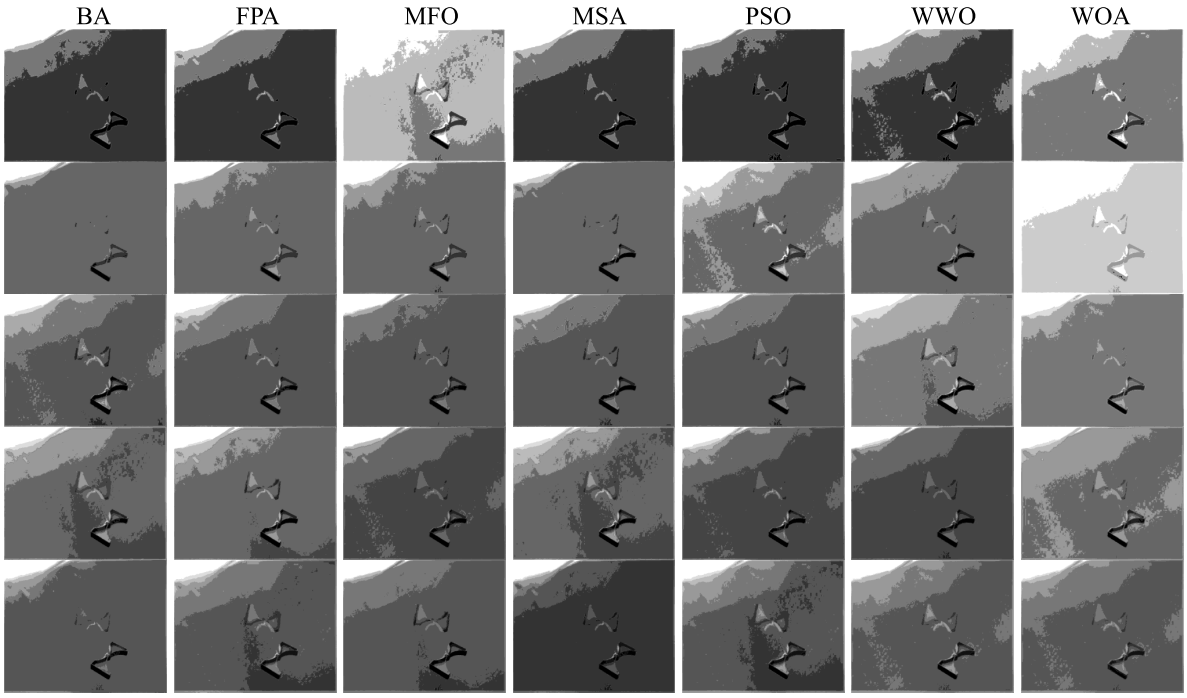

FIGURE 15. Segmented images of Test 7.

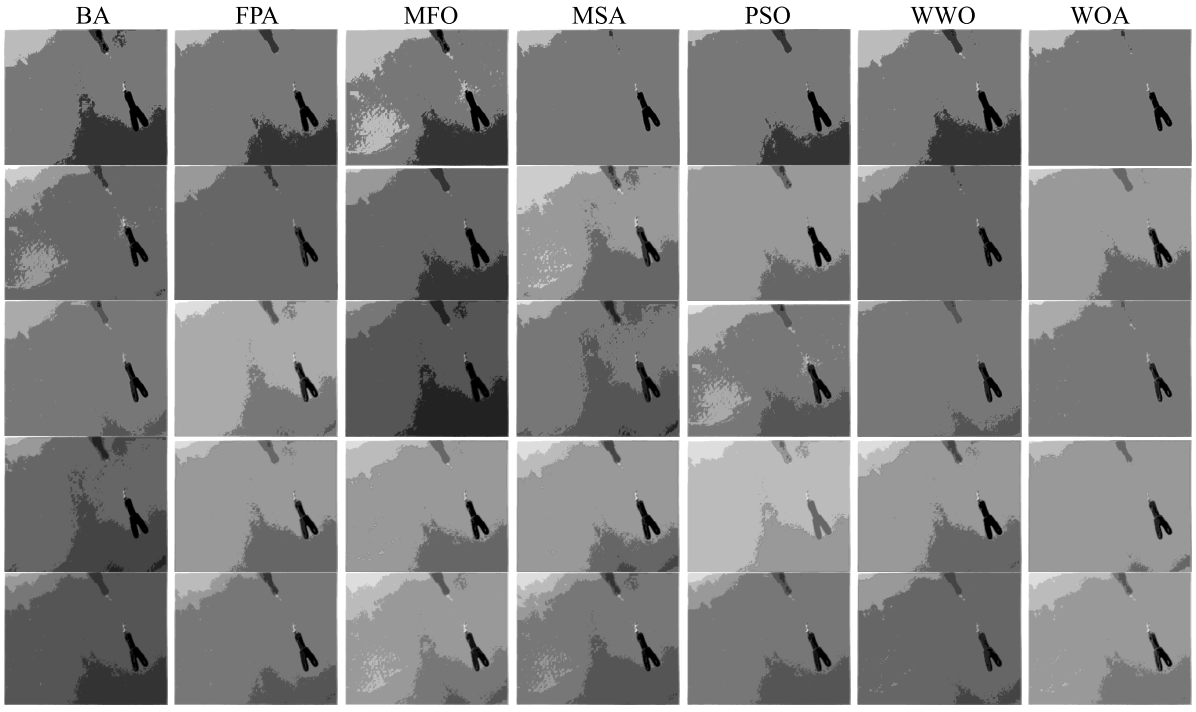

FIGURE 16. Segmented images of Test 8.

underwater image is an important key point to the underwater robot's dynamic perception of the environment. The underwater image segmentation is not only the premise of the visual image analysis, understanding and recognition technology, but also one of the classic topics of underwater robot vision. Videos and underwater images are captured by an underwater camera (Tornado) in the experimental pool at Harbin Engineering University. The video processing is regarded as the processing operation of the video frames, and the underwater video can be used as research object of

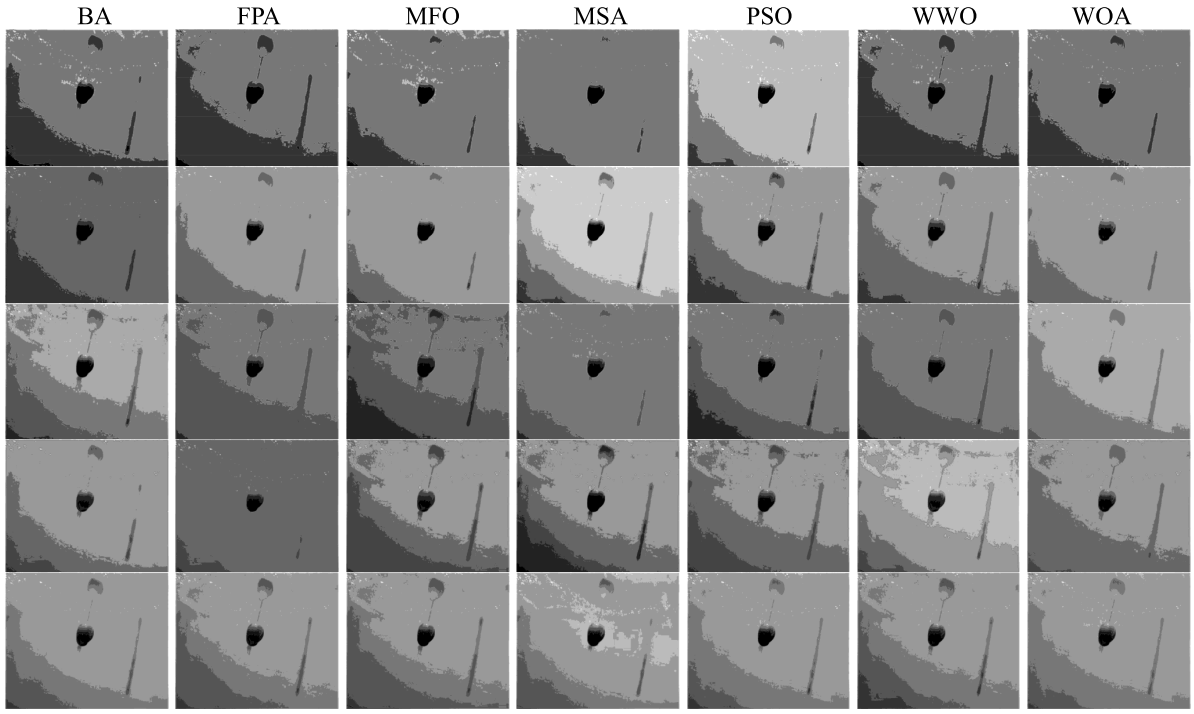

FIGURE 17. Segmented images of Test 9.

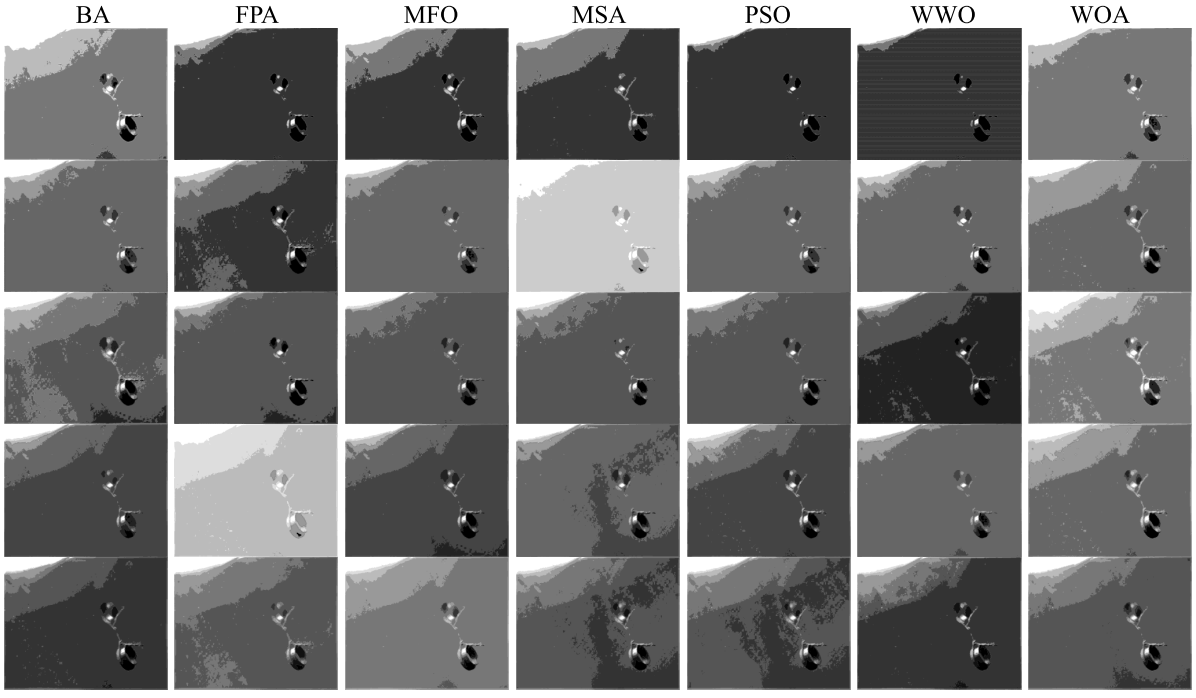

FIGURE 18. Segmented images of Test 10.

the underwater image. Due to the fluctuation of the water medium and the effects of the light scattering, refraction and absorption in the underwater environment, the pixels and contrast of the underwater images are relatively low. The image segmentation is the basic step of the image processing, the collection of the underwater images information affects the completion of UUV underwater task to a certain extent. Therefore, the WOA based on Kapur's entropy method has simplicity and stability in achieving the image segmentation. The experiments are conducted on fourteen test images that

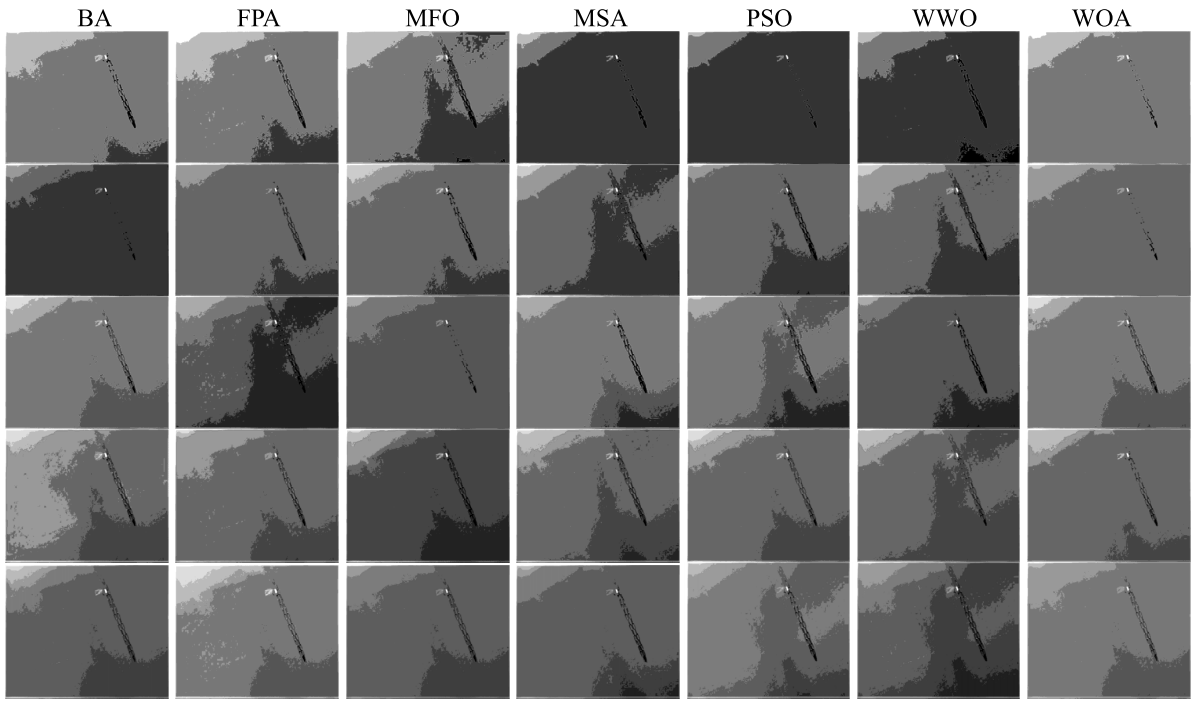

FIGURE 19. Segmented images of Test 11.

were carefully selected from the experimental pool of Harbin Engineering University and are shown in Fig. 4.

C. PARAMETER SETTING

To verify the effectiveness and feasibility of the WOA, the proposed algorithm is applied to accomplish underwater multilevel thresholding image segmentation, and the WOA is compared with other optimization algorithms, such as the BA, the FPA, MFO, the MSA, PSO and WWO. The control parameters of the comparison algorithm are important empirical values and derived from the original articles; they are shown in Table 1.

D. SEGMENTED IMAGE QUALITY MEASUREMENTS

Five important measures are used as performance evaluation indicators to detect the segmented image and further verify the overall superiority of the WOA:

1) FITNESS VALUE

The amount of information of the segmented image depends on the size of the fitness value. A larger fitness value means that the segmented image contains more information.

2) PEAK SIGNAL-TO-NOISE RATIO (PSNR)

The PSNR is used to detect the difference between the segmented image and the original image based on the intensity value of the image. The larger the PSNR value is, the better the image segmentation effect. Due to the difference in visual acuity of the human eye, a segmented image of a higher PSNR value may be worse than a segmented image of a lower PSNR value. The PSNR can be defined as [40]:

$$PSNR = 10\log_{10}\left(\frac{255^2}{MSE}\right) \tag{16}$$

where MSE denotes the mean squared error and can be defined as:

$$MSE = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}[I\,(i,j) - K\,(i,j)]^2 \tag{17}$$

where $I(i,j)$ and $K(i,j)$ denote the original and segmented images with size $M \times N$, respectively.

3) STRUCTURE SIMILARITY INDEX (SSIM)

The SSIM is used to detect the similarity between the segmented image and the original image according to brightness, contrast and structural similarity. As the SSIM value approaches one, the image segmentation effect improves. The SSIM can be defined as [41]:

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \tag{18}$$

where μ_x and μ_y denote the mean intensity of the original image and that of the segmented image, respectively. σ_x^2 and σ_y^2 denote the standard deviation of the original image and that of the segmented image, respectively. σ_{xy} denotes the covariance between the original image and the segmented image. c_1 and c_1 are constants.

TABLE 2. The optimal fitness values of each algorithm.

Images	k	Fitness values							
		BA	FPA	MFO	MSA	PSO	WWO	WOA	Rank
Test 1	4	15.0722	15.2359	14.9729	14.7392	15.2711	14.9306	15.2931	1
	5	17.2723	17.1296	17.0925	17.0524	16.9145	17.2535	17.3343	1
	6	19.1686	18.9847	18.3896	18.4839	18.6567	18.9171	19.1916	1
	7	20.6582	20.6650	20.7622	20.5686	20.4822	20.7884	21.0367	1
	8	22.7917	22.4747	22.3337	22.6088	22.9337	22.1950	23.4738	1
Test 2	4	15.0840	14.5255	15.5920	14.8018	14.7088	15.3131	15.5552	2
	5	17.2299	17.3487	17.3152	17.0468	17.0589	17.0966	17.3629	1
	6	18.8481	18.8530	18.7867	19.0072	18.8868	18.6512	19.1291	1
	7	20.7701	20.5415	20.6043	20.2197	20.5224	20.3012	21.2052	1
	8	22.3523	22.4408	22.3081	22.1635	22.3186	22.4934	22.5310	1
Test 3	4	15.0079	15.1209	15.2381	14.9610	15.3760	15.3638	15.5228	1
	5	16.8840	17.4398	17.1547	17.5093	17.4047	17.1968	17.6715	1
	6	19.3484	18.4441	19.1090	19.2999	19.0334	19.3095	19.4854	1
	7	21.4444	21.0465	21.5620	21.1234	21.3922	21.2509	21.6513	1
	8	23.4581	23.1860	23.2525	23.1838	23.3794	23.3554	23.5792	1
Test 4	4	14.1299	14.1304	14.0499	14.0723	14.1624	14.1514	14.2871	1
	5	15.9046	15.8587	16.2985	15.8336	16.0421	16.2106	16.7338	1
	6	17.7294	17.6296	17.9322	18.0394	17.8603	17.7842	18.5243	1
	7	19.6227	19.3862	19.4639	19.5819	19.3082	19.4982	19.6653	1
	8	21.2471	21.1342	21.3398	21.3111	21.0989	21.1785	22.1502	1
Test 5	4	15.1355	14.8892	15.6235	15.0966	15.4092	15.1042	15.7992	1
	5	17.8999	17.3727	17.6953	17.2736	17.8433	17.6893	17.9984	1
	6	19.7514	19.7821	19.5973	19.3915	19.8163	19.8753	20.0295	1
	7	21.8159	21.5011	21.5816	21.7759	21.9349	21.8652	21.9364	1
	8	23.5855	23.6579	23.8503	23.6995	23.7428	23.3712	23.9417	1
Test 6	4	16.1616	16.0158	15.6919	15.6826	16.1138	15.8484	16.1904	1
	5	18.0434	17.6869	17.5958	18.1396	18.0464	17.8026	18.2128	1
	6	19.6831	19.6557	19.9505	20.0103	19.8316	19.6198	20.0400	1
	7	22.2304	22.1412	21.7112	21.6523	21.8450	21.9476	22.2647	1
	8	23.8550	23.9282	23.5100	24.0698	24.3232	23.6632	24.5015	1
Test 7	4	14.6607	14.6623	14.3539	14.7395	14.7228	14.6734	14.8990	1
	5	16.8840	16.8619	16.8613	17.0000	16.9133	17.0436	17.1759	1
	6	19.3046	19.4686	19.2965	18.8889	19.2734	19.2272	19.4785	1
	7	21.0757	20.8985	21.2709	21.0786	21.2736	21.0659	21.5306	1
	8	23.3970	23.4366	23.4924	23.1987	23.2310	23.3482	24.7256	1
Test 8	4	15.5464	15.7907	15.6518	15.9751	15.6980	16.1812	16.1928	1
	5	18.0826	17.8217	18.2499	18.1272	17.7864	17.9369	18.4448	1
	6	20.2513	20.1098	20.1231	20.2280	20.4068	19.9121	20.4873	1
	7	22.1075	22.4825	22.1994	22.6045	22.1076	22.0994	23.6696	1
	8	24.0093	24.2594	24.1425	24.2566	24.3474	24.0268	24.4749	1
Test 9	4	15.4074	15.3022	15.2221	15.2049	15.3901	15.2440	15.5017	1
	5	17.3670	17.7573	17.5453	17.2648	17.7686	17.5846	17.9300	1
	6	19.8240	19.3799	18.9553	19.8081	19.8054	19.8002	21.4986	1
	7	21.4274	21.5375	21.2935	21.5857	21.4274	21.3211	21.6422	1
	8	23.3681	23.6221	23.1561	23.0468	23.6150	23.6815	23.7222	1
Test 10	4	14.8705	14.0559	14.8658	14.6081	14.8217	14.7869	14.8718	1
	5	17.4287	17.1170	17.2052	17.2915	17.1015	17.2146	17.8025	1

TABLE 2. *(Continued.)* **The optimal fitness values of each algorithm.**

	6	18.9790	18.8752	19.4507	19.0912	19.5845	19.4999	20.0293	1
	7	21.7741	21.1941	21.4333	21.2682	21.7802	21.2488	22.4819	1
	8	23.2561	23.0256	23.0224	23.3970	23.1613	23.0298	25.1283	1
Test 11	4	15.1904	15.2322	15.0381	15.2213	15.2430	14.8924	15.2764	1
	5	17.4178	16.9314	17.6395	17.6321	17.3344	17.1427	18.0862	1
	6	19.4661	19.4432	19.4860	19.0648	19.2662	19.5469	19.5522	1
	7	21.2827	21.3893	21.3414	21.4553	21.5794	21.3049	21.6415	1
	8	23.2607	23.0877	23.4219	22.9227	23.2142	23.1584	23.4861	1
Test 12	4	16.1782	15.8087	16.3431	16.3011	16.3520	16.1844	16.3521	1
	5	18.2569	18.1546	18.5073	18.3468	18.8298	18.6576	18.5397	3
	6	20.6802	20.8579	20.6660	20.2556	20.4901	20.5062	21.3620	1
	7	22.8486	22.8364	22.4722	22.8491	22.3102	22.5684	22.8856	1
	8	24.4701	24.8927	24.8742	24.2823	24.8531	24.0143	25.0135	1
Test 13	4	15.5692	15.3250	15.4428	15.7674	15.7984	15.7487	15.8076	1
	5	18.0590	17.7891	17.6018	17.8231	18.0083	17.8068	18.1791	1
	6	19.9128	20.0285	19.5029	20.1492	19.9370	19.8311	20.2943	1
	7	21.6954	22.1542	21.6492	21.8035	21.7813	22.0701	22.2164	1
	8	24.1869	23.8509	23.7434	23.4545	23.6996	24.2034	24.2277	1
Test 14	4	14.5151	14.3826	14.4190	14.6327	14.7478	14.5891	14.7554	1
	5	17.1700	17.2275	16.9477	16.7385	16.7179	16.6738	17.3624	1
	6	19.2043	18.9097	19.1916	19.0640	18.5847	19.0652	19.3021	1
	7	20.8962	20.6646	20.7948	20.7755	21.0201	20.8674	21.4820	1
	8	22.9104	22.6920	22.9554	22.6753	22.9475	22.8579	22.9630	1

4) EXECUTION TIME

The execution time is based on each comparison algorithm after 30 independent runs, which can reflect the convergence speed. The shorter the time is, the higher the efficiency of the algorithm.

5) WILCOXON'S RANK-SUM TEST

Wilcoxon's rank-sum test is used to detect whether there is a significant difference between two algorithms [42]. If $p < 0.05$, then there is a significant difference. If $p > 0.05$, then there is no significant difference.

E. RESULTS AND ANALYSIS

For all algorithms, the size of the population is 30, the maximum number of iterations is 100, and the number of independent runs is 30. The threshold numbers are set to 4, 5, 6, 7 and 8. The experimental results of the WOA based on Kapur's entropy are compared to those of the BA, the FPA, MFO, the MSA, PSO and WWO in Tables 2-7 and Figs. 5-8 according to the optimal fitness value, the best threshold value, the PSNR value, the SSIM value, the average execution time and the *p-value* of Wilcoxon's rank-sum test.

In computational science and mathematics management, optimization is the process of selecting the best solution from the available alternatives. In other words, the purpose of optimization is to obtain the global optimal solution in a given search space by maximizing or minimizing the objective function. In Table 2, for different thresholds, the WOA can obtain more effective and feasible results. The optimal fitness values of the WOA are significantly better than those of the other optimization algorithms. For all algorithms, the optimal fitness values increase as the threshold levels increase, which indicates that an image segmented with a higher threshold level can result in a better segmentation effect. To verify the superiority of the WOA, the ranking is based on the optimal fitness value. The ranking of the WOA is the highest, which indicates that the algorithm obtains the best solution among those of all algorithms. For test images 1, 3, 4, 5, 6, 7, 8, 9, 10, 11, 13 and 14, the WOA has a strong global search ability, avoiding falling into a local optimum. Compared with those of the other algorithms, the optimal fitness values of the WOA are the best, and all its rankings are first, which indicates that the WOA can successfully handle underwater test images. For test image 2, the optimal fitness value of the WOA is worse than the fitness value of the FPA at $K = 4$, but the WOA is better than the BA, the FPA, MFO, the MSA, PSO and WWO at $K = 5, 6, 7, 8$. For test image 12, the optimal fitness value of the WOA is worse than the fitness value of the PSO and WWO at $K = 5$, but the WOA is better than the BA, the FPA, MFO and the MSA at $K = 5, 6, 7, 8$. In Fig. 5, the average fitness values increase as the threshold

TABLE 3. The best threshold values of each algorithm.

Images	k	Best threshold values						
		BA	FPA	MFO	MSA	PSO	WWO	WOA
Test 1	4	145,165,196,206	143,180,198,223	130,165,196,220	137,168,208,222	141,166,197,214	126,142,164,204	149,182,195,214
	5	149,171,195,208, 229	149,175,199,218, 234	149,164,175,196, 227	144,160,181,196, 208	131,148,194,214, 232	125,138,176,197, 220	147,162,176,195, 218
	6	131,147,173,183, 201,226	142,172,183,195, 211,232	130,141,171,177, 199,232	115,135,158,182, 196,211	130,150,178,205, 219,232	146,161,178,200, 209,224	133,146,172,199, 211,228
	7	136,148,162,173, 202,223,234	134,148,158,170, 178,197,231	113,126,148,179, 198,211,227,	122,138,146,163, 195,216,222	120,129,150,173, 180,196,221	122,136,144,170, 185,199,217	132,149,164,172, 196,206,220
	8	125,148,162,180, 194,210,223,234	130,145,155,164, 179,195,208,233	122,145,172,179, 185,196,215,224	120,146,162,184, 196,217,222,241	120,136,144,166, 176,198,215,229	119,141,148,161, 187,201,225,233	124,148,176,186, 195,207,217,233
Test 2	4	140,166,184,201	116,139,184,217	142,170,196,219	138,162,194,223	128,161,182,199	120,145,174,198	143,170,197,219
	5	122,134,170,197, 211	124,140,180,195, 222	116,144,173,203, 224	131,143,156,197, 215	128,146,176,198, 207	123,144,163,206, 217	114,143,177,198, 224
	6	115,142,149,184, 195,224	130,144,158,199, 218,238	130,151,163,178, 198,222	123,137,162,178, 204,240	114,131,138,176, 196,224	137,146,161,171, 198,211	125,139,148,169, 201,222
	7	131,145,162,179, 197,214,239	119,146,158,185, 198,213,238	113,126,144,175, 198,204,239	114,121,140,167, 186,203,229	120,140,166,202, 209,222,241	112,122,142,166, 204,221,240	103,113,146,159, 190,203,230
	8	120,130,144,165, 174,203,210,226	123,140,165,179, 199,209,222,240	134,153,167,176, 184,199,223,236	115,132,144,164, 175,195,207,213	138,146,170,177, 197,208,221,237	130,142,154,160, 178,195,207,224	103,111,129,146, 164,197,207,240
Test 3	4	110,146,166,199	117,147,172,219	147,176,201,222	133,169,197,213	147,181,211,235	119,141,167,199	119,147,205,241
	5	124,134,172,208, 235	120,152,168,212, 237	121,139,173,215, 242	114,150,180,196, 210	121,149,166,218, 240	152,181,197,215, 232	111,149,164,203, 235
	6	129,147,171,208, 217,239	113,130,142,155, 170,212	136,170,196,216, 231,237	111,128,144,163, 195,238	137,162,180,200, 207,238	135,146,164,195, 222,232	121,133,163,178, 196,235
	7	119,151,179,197, 203,220,243	117,142,196,210, 219,237,245	117,140,148,170, 198,201,228	106,131,150,181, 205,212,230	125,140,151,170, 196,218,22	115,140,160,189, 205,222,235	110,142,169,184, 202,221,234
	8	128,145,161,180, 195,209,216,236	109,118,133,149, 180,199,221,240	113,123,144,161, 199,209,221,232	111,129,141,158, 185,197,218,242	116,133,147,170, 180,193,213,235	120,139,149,171, 189,205,219,234	110,132,141,168, 194,206,230,241
Test 4	4	131,149,173,188	142,169,187,202	122,158,187,199	134,156,175,195	114,148,184,228	122,153,172,185	126,151,165,191
	5	128,146,170,180, 205	126,138,171,197, 220	125,142,168,182, 201	146,171,183,203, 219	125,146,159,171, 201	122,144,155,186, 199	148,158,174,186, 227
	6	126,136,148,160, 187,195	123,146,163,185, 205,217	123,138,146,169, 185,199	128,142,155,168, 188,201	123,145,158,184, 199,213	121,132,149,166, 177,202	109,110,154,178, 228,237
	7	123,149,159,179, 185,201,228	122,149,171,180, 191,207,212	122,137,158,168, 176,193,207	120,128,144,159, 177,192,232	127,138,160,174, 194,205,222	124,150,158,178, 185,196,208	125,133,153,169, 178,193,235
	8	125,140,161,172, 183,198,206,219	118,137,155,171, 182,190,202,221	122,132,149,158, 177,186, 202,220	127,143,156,165, 181,191,216,231	124,144,166,179, 181,197,214,236	123,135,146,163, 173,183,206,220	110,148,155,175, 184, 199,227,236
Test 5	4	108,142,158,205	112,141,218,238	110,137,180,213	120,131,188,205	114,134,191,213	127,145,165,192	98,140,176,202
	5	96,115,142,186, 214	90,115,131,165, 186	108,143,168,214, 238	122,138,152,183, 214	90,114,141,170, 183	86,113,141,187, 203	90,110,143,172, 191
	6	98,138,163,192, 215,238	95,110,130,171, 189,202	96,112,147,172, 201,226	108,120,145,173, 184,221	94,115,137,157, 188,211	109,136,178,207, 221,239	93,118,137,154, 173,193
	7	117,146,170,183, 196,213,230	114,137,168,178, 186,212,233	98,118,133,140, 182,202,220	100,116,126,140, 177,204,231	90,119,144,170, 201,212,235	94,117,132,154, 170,184,219	88,107,120,140, 170,201,212
	8	96,106,146,176, 188,209,228,242	97,137,158,171, 182,196,224,240	93,118,133,143, 167,193,213,248	99,123,141,150, 162,200,214,239	114,129,143,171, 186,208,230,244	101,119,135,143, 165,177,204,216	89,107,134,149, 169,187,208,221
Test 6	4	129,148,174,213	109,130,172,219	132,171,217,235	132,153,198,216	126,171,195,216	132,157,176,202	133,171,188,214
	5	122,145,172,198, 211	132,152,189,205, 235	131,189,215,230, 244	129,152,176,198, 224	108,131,171,179, 211	116,132,159,170, 219	104,127,153,172, 205
	6	89,107,132,169, 209,239	89,129,167,175, 198,215	110,125,143,169, 191,227	92,133,159,181, 207,223	103,123,166,188, 222,241	105,132,164,196, 211,242	95,114,130,150, 171,214
	7	115,129,143,166, 188,213,247	89,126,147,168, 189,209,214	97,116,127,144, 169,183,215,	126,150,179,200, 213,234,246	109,134,167,197, 208,225,246	89,112,130,179, 193,209,230	106,122,152,167, 185,204,238
	8	88,117,128,149, 167,197,216,234	101,121,133,141, 170,193,215,230	99,119,154,171, 180,191,224,236	109,123,132,144, 170,189,210,222	108,133,150,173, 198,209,226,239	94,106,125,155, 169,201,230,245	103,124,130,175, 193,217,226,242
Test 7	4	112,128,171,191	128,165,187,208	107,140,153,171	131,165,191,219	142,172,217,245	140,160,179,213	136,170,205,239
	5	108,134,194,219, 247	106,137,172,208, 221	106,140,173,194, 206	118,139,188,201, 230	107,137,159,175, 190	116,138,168,212, 232	87,132,170,188, 241
	6	126,143,160,178, 209,236	114,137,164,186, 200,246	102,140,165,189, 209,241	116,128,142,171, 190,202	115,141,167,190, 220,232	106,121,137,175, 199,227	120,142,168,189, 216,234
	7	113,132,153,166, 189,205,238	118,139,148,170, 187,218,241	115,137,160,187, 212,229,237	106,127,160,178, 204,227,232	103,133,171,185, 203,220,237	110,126,139,158, 192,210,235	117,141,164,193, 207,221,235
	8	113,128,141,177, 185,205,224,245	107,132,151,163, 184,197,205,231	100,119,140,160, 181,204,227,237	104,113,141,169, 187,194,218,222	109,130,153,163, 176,198,211,237	109,126,142,160, 175,206,233,247	104,121,140,166, 200,211,231,244
Test 8	4	135,160,183,235	117,156,192,223	135,158,172,218	113,132,191,215	128,155,200,229	111,132,188,219	128,158,180,217
	5	108,138,172,192, 211	100,136,183,223, 234	123,156,181,209, 232	97,134,162,175, 227	114,133,156,209, 234	103,121,158,199, 220	105,132,182,200, 228
	6	100,130,146,179, 219,231	93,110,141,161, 192,213	117,160,180,201, 219,238	98,141,164,190, 209,233	98,130,156,172, 202,223	98,129,148,201, 217,232	93,109,132,178, 209,227
	7	120,142,163,187, 204,214,233	96,114,130,159, 190,205,224	105,115,125,157, 177,204,210	103,127,143,156, 178,193,231	92,110,126,143, 191,213,220	114,121,138,159, 183,195,216	99,131,145,177, 202,223,240
	8	93,102,110,137, 157,177,203,220	96,120,139,153, 178,188,212,241	89,109,123,145, 160,174,213,233	104,120,139,161, 174,186,211,227	95,113,135,157, 180,195,212,226	95,121,142,154, 176,193,202,230	104,127,139,157, 178,194,209,224
Test 9	4	145,171,195,220	140,178,209,238	137,160,194,207	134,155,207,242	131,148,166,197	136,177,196,206	128,165,198,237
	5	128,165,199,213, 244	125,140,168,204, 222	129,144,160,206, 236	133,147,161,176, 202	124,156,174,197, 227	117,149,177,199, 224	113,132,163,198, 220
	6	130,145,166,183, 199,230	119,138,180,201, 225,235	119,162,184,203, 214,226	116,138,158,196, 221,241	131,157,173,200, 213,235	125,144,175,204, 219,235	142,156,197,204, 212,228
	7	107,137,153,171, 204,224,241	114,136,152,198, 213,224,233	109,135,170,181, 199,227,236	138,159,168,181, 197,219,227	120,137,163,183, 199,205,221	107,134,148,161, 184,198,238	112,131,145,179, 199,226,234
	8	110,122,140,159, 175,202,213,236	109,127,143,167, 179,199,221,230	119,132,144,170, 181,201,218,226	114,125,134,159, 171,189,206,238	119,128,144,156, 176,197,216,230	112,129,152,167, 181,200,237,248	110,124,137,155, 176,196,226,239
Test 10	4	129,143,167,197	138,180,198,229	110,142,179,214	124,163,198,244	139,194,212,235	138,195,220,238	137,166,185,226

TABLE 3. ***(Continued.)*****The best threshold values of each algorithm.**

	5	109,140,174,197, 220	137,159,179,196, 230	110,136,179,210, 246	123,138,162,196, 230	107,137,176,193, 241	120,142,176,189, 214	104,141,179,193, 138
	6	126,146,158,177, 207,240	136,145,176,190, 218,245	116,138,171,207, 223,246	105,129,139,161, 177,198	124,140,172,202, 222,237	133,161,187,202, 225,243	119,139,178,192, 218,239
	7	108,133,164,184, 206,224,238	123,133,141,162, 179,210,236	125,146,164,184, 198,225,231	118,139,152,181, 206,226,237	120,140,162,176, 188,215,229	110,123,142,179, 212,233,239	115,139,162,179, 201,223,237
	8	121,141,162,181, 197,230,239,241	100,122,134,160, 179,207,230,241	111,122,136,143, 164,188,216,239	115,136,153,167, 183,203,221,236	115,133,154,165, 184,203,218,246	112,142,163,170, 179,206,235,248	123,132,139,178, 192,216,227,232
Test 11	4	134,144,176,216	129,150,173,202	137,159,181,228	122,134,180,207	123,183,210,227	142,174,202,227	134,183,200,214
	5	121,177,199,216, 231	120,147,191,222, 237	133,147,178,196, 229	130,161,183,201, 216	135,155,198,215, 230	121,129,179,212, 234	132,158,174,202, 222
	6	122,131,153,181, 199,227	133,162,172,191, 224,237	119,133,176,204, 227,238	133,144,154,191, 217,239	127,148,161,192, 214,232	114,135,154,187, 200,218	111,132,155,186, 202,226
	7	119,136,155,169, 191,200,225	117,135,152,173, 206,213,227	130,153,176,183, 200,214,221	119,141,158,177, 191,212,229	116,130,154,178, 195,202,236	115,133,147,181, 191,221,230	116,134,161,181, 194,218,221
	8	110,134,154,174, 189,197,204,236	115,124,133,153, 172,179,214,227	119,132,154,176, 202,220,231,243	122,141,151,180, 202,212,224,247	121,137,148,164, 185,201,229,241	127,148,164,173, 190,216,229,235	115,126,134,189, 217,224,237,241
Test 12	4	94,137,176,223	85,146,193,214	134,172,194,215	128,148,177,205	101,128,175,219	120,173,195,210	102,145,173,207
	5	87,123,133,184, 224	79,120,180,209, 230	110,128,163,189, 213	88,122,136,194, 215	99,124,142,172, 205	88,125,137,173, 216	100,149,174,191, 211
	6	93,109,125,151, 179,191	86,101,124,149, 193,219	99,130,143,158, 179,209	90,109,118,152, 166,184	100,119,134,159, 174,199	90,100,138,156, 181,209	98,127,172,197, 208,226
	7	89,117,139,151, 171,181,197	82,113,151,170, 196,225,230	86,115,151,159, 173,186,215	76,128,152,167, 187,214,235	67,89,125,140, 166,184,218	83,107,123,173, 181,205,221	98,134,152,166, 193,223,238
	8	86,132,151,160, 175,193,223,235	84,105,136,146, 175,187,210,226	77,94,134,148, 166,187,216,239	80,98,128,174, 195,201,217,228	96,108,129,152, 184,205,226,230	91,105,137,146, 194,215,237,241	93,114,132,145, 166,180,227,235
Test 13	4	134,160,191,233	125,188,212,235	118,129,180,205	133,162,195,231	128,158,197,226	136,158,202,233	126,163,196,223
	5	116,140,180,200, 236	126,139,164,202, 234	138,151,167,204, 224	109,126,138,180, 208	114,132,161,176, 201	127,162,179,217, 232	123,162,180,207, 224
	6	109,121,160,187, 210,224	104,130,161,183, 211,237	103,140,176,189, 196,229	107,135,168,186, 201,232	94,140,157,177, 208,228	99,140,181,204, 224,238	105,127,159,183, 203,219
	7	110,123,134,153, 182,209,228	103,116,127,158, 178,202,217	97,111,134,163, 186,220,230	112,136,161,169, 188,203,219	111,128,139,157, 177,188,201	101,124,135,163, 181,214,227	111,137,148,186, 205,222,231
	8	101,121,140,163, 184,203,217,229	109,124,139,154, 163,197,205,217	104,123,138,179, 192,207,223,238	109,120,136,150, 157,181,219,226	97,106,139,151, 182,196,218,235	106,130,160,179, 194,210,226,236	114,140,176,183, 195,218,229,233
Test 14	4	124,152,205,223	123,166,221,239	110,142,156,191	116,150,176,207	116,163,186,206	120,161,188,240	117,156,177,216
	5	120,160,182,206, 232	124,161,184,219, 239	119,158,181,208, 221	121,147,175,202, 241	100,123,157,175, 208	102,121,167,195, 217	116,122,161,176, 236
	6	108,123,153,191, 213,234	119,162,173,206, 224,239	121,137,165,195, 221,246	126,155,169,194, 210,229	114,149,167,181, 211,231	118,135,158,182, 208,233	106,122,155,186, 204,233
	7	110,142,152,176, 192,218,227	120,139,146,185, 216,221,230	123,131,166,181, 200,216,229	126,140,152,168, 177,198,236	119,145,159,186, 200,229,239	107,140,164,179, 194,214,220	124,157,178,195, 206,221,230
	8	125,140,159,177, 186,212,223,235	100,120,152,162, 188,208,237,241	111,124,163,177, 199,225,231,245	112,140,151,159, 176,193,237,241	125,133,154,178, 186,211,231,248	120,142,159,171, 203,218,223,241	120,140,158,179, 191,201,232,246

levels increase, and the fitness values of the WOA are the best among those of all algorithms. To summarize, the WOA has strong exploration and exploitation abilities and thus can achieve a higher calculation accuracy and better segmentation effect than can the other algorithms.

The threshold value plays an important role in image segmentation, as it not only determines the quality of the segmented image but also affects the value of each evaluation index. In this paper, the segmentation thresholds are determined according to Kapur's entropy. An algorithm with a strong search ability can obtain better threshold values and thus can improve the accuracy and quality of the segmented images. Therefore, it is necessary to analyse the optimization performance of each algorithm in solving the multilevel thresholding image segmentation problem. Compared with other algorithms, the WOA has stronger robustness and a better search ability; hence, it can effectively avoid premature convergence and obtain the global optimal solution. For example, the BA affects the calculation accuracy by adjusting the pulse frequency range f, echo loudness A, and decreasing coefficient γ. The FPA uses the switch probability ρ to perform search optimization. MFO seeks the optimal solution by adjusting the control parameters b, t and r. The MSA adjusts the random numbers θ, ε_2, ε_3, r_1 and r_2. PSO uses the constant inertia ω and the acceleration coefficients c_1 and c_2 to update the position and then performs a global search to obtain the best solution. WWO uses the wavelength λ and wave height $h_{\max}$ to avoid premature convergence. Although these factors balance the global search and local search in solving the image segmentation problem, the factors are not universal, especially for complex optimization problems. As the number of thresholds increases, the computational complexity increases, and the accuracy of the segmented images increases. The WOA balances the global search and the local search by simulating the encircling of prey, the bubble-net attacking method and the search for prey and avoids falling into a local optimum. The WOA improves the calculation accuracy to a certain extent and obtains the best threshold values, which indicates that the WOA is effective and useful in solving the image segmentation problem.

For multilevel thresholding image segmentation, the larger the number of segmented thresholds, the higher the accuracy of the segmented image is. To better verify the segmentation performance of each algorithm, the PSNR value is regarded as an important evaluation index. The PSNR values of the segmented images obtained by the BA, the FPA,

TABLE 4. The PSNR of each algorithm.

Images	k	PSNR values							
		BA	FPA	MFO	MSA	PSO	WWO	WOA	Rank
Test 1	4	68.6177	68.8211	70.0823	69.4128	69.0502	67.5338	70.5899	1
	5	67.5338	67.5338	67.5338	68.6539	69.9713	68.2907	70.7230	1
	6	69.9713	68.9382	70.0823	69.7644	70.0823	68.3998	73.6002	1
	7	69.5149	69.6649	70.5899	71.1675	69.8551	71.1675	71.6086	1
	8	70.7230	70.0823	71.1675	71.6086	71.2460	71.9003	71.3845	2
Test 2	4	72.9849	78.8241	72.5897	73.2348	74.8827	72.1896	77.1751	2
	5	76.6182	75.9869	78.8241	74.1675	74.8827	76.2845	79.8129	1
	6	79.2467	74.3609	74.3609	76.2845	79.4791	73.3766	79.4791	1
	7	74.1675	77.6123	79.8129	79.4791	77.1751	80.1746	81.2722	1
	8	77.1751	76.2845	73.7040	80.6048	73.2348	74.3609	79.2467	2
Test 3	4	81.1504	79.1312	66.2595	69.1304	66.2595	78.1740	78.1740	3
	5	72.9235	77.3280	76.4657	79.7318	76.4657	62.9510	80.6044	1
	6	70.3676	80.0982	68.2845	76.4657	68.0398	68.5276	80.6044	1
	7	78.1740	79.1312	79.1312	83.1835	72.1534	79.4760	80.6044	2
	8	70.5735	81.9179	80.0982	80.0644	79.2344	77.3280	80.0982	2
Test 4	4	67.3600	61.9829	75.6931	65.6211	72.3413	75.6931	84.9243	1
	5	69.8368	72.3413	73.2764	60.5183	73.2764	59.8295	75.6931	1
	6	66.5061	74.8415	74.8415	69.8368	74.8415	76.4368	79.8328	1
	7	74.8415	75.6931	75.6931	73.2764	71.0995	74.0952	77.1001	1
	8	73.2764	78.9037	75.6931	71.0995	74.0952	74.8415	81.5739	1
Test 5	4	77.4508	76.1954	77.1293	74.4780	76.0225	71.6986	76.6975	3
	5	81.5913	84.1620	77.4508	73.3188	84.1620	85.5114	84.1620	1
	6	80.9154	81.8211	81.5913	77.4508	82.3033	77.3260	82.8060	1
	7	75.2472	76.0225	80.9154	80.0918	84.1620	82.3033	84.8314	1
	8	81.5913	81.2334	82.8060	80.3644	76.0225	79.7174	84.2437	1
Test 6	4	68.7697	67.6558	68.0532	68.0532	69.1628	68.0532	72.1056	1
	5	69.7209	68.0532	68.4393	68.7697	72.2313	70.8446	73.6066	1
	6	72.1056	78.5530	71.8112	83.7902	74.1395	73.1790	78.7727	2
	7	71.0001	72.8868	70.8446	69.1628	72.1056	71.5032	87.5530	1
	8	88.3920	75.2902	75.9923	72.1056	72.2313	80.3347	87.5530	2
Test 7	4	66.6612	67.7667	72.8124	67.2797	65.2297	65.8152	72.2560	1
	5	72.5689	73.0060	73.0060	69.1462	72.8124	69.5113	85.2715	1
	6	68.0656	70.1372	74.5468	68.7844	69.7707	68.7844	73.0060	2
	7	70.5119	69.1462	69.7707	73.0060	73.9617	69.3254	71.8726	3
	8	70.5119	72.8124	69.0003	70.5119	72.2560	72.2560	75.5736	1
Test 8	4	65.4623	66.8191	65.4623	67.0097	66.3646	66.3646	67.1445	1
	5	67.4246	69.1764	66.5970	69.8398	66.9573	67.9625	68.4864	3
	6	69.1764	71.1101	66.8191	69.6488	69.6488	69.6488	71.1101	1
	7	66.6968	70.0625	67.9625	68.4864	68.2039	66.9573	71.5207	1
	8	68.2039	70.0625	73.5581	68.2039	70.3751	66.6647	71.1101	2
Test 9	4	65.6366	67.3263	67.9404	68.6606	68.9433	68.2136	69.1520	1
	5	69.1520	69.4035	69.0802	68.7218	69.4726	70.3427	71.0124	1
	6	69.0137	70.0777	70.0777	70.4608	68.9433	69.4035	70.4608	1
	7	71.3015	70.8081	73.0470	67.5542	69.9214	74.2466	74.2466	1
	8	72.0930	73.0470	70.0777	70.8081	70.0777	71.3015	72.0930	2
Test 10	4	68.1808	66.5070	66.7064	69.1202	66.3430	66.5070	73.2368	1
	5	74.1994	66.7064	73.2368	69.2587	75.4670	69.7323	82.6015	1

TABLE 4. *(Continued.)* The PSNR of each algorithm.

	6	68.8796	66.9621	70.4255	70.2605	69.1202	67.4497	76.5998	1
	7	75.0131	69.2587	68.9925	70.1122	69.7323	73.2368	82.6015	1
	8	69.5256	80.4901	72.3485	70.6281	70.6281	71.7102	81.1903	1
Test 11	4	72.3726	74.7603	69.4574	72.3726	77.5234	63.2389	78.1058	1
	5	78.6704	78.2232	73.0222	74.3576	71.4489	73.4846	78.6704	1
	6	78.1058	73.0222	79.6319	73.0222	75.6297	71.4489	81.9079	1
	7	79.6319	80.6345	74.3576	79.6319	81.1701	81.1701	81.4648	1
	8	84.3182	81.4648	79.6319	78.1058	78.6704	75.6297	90.6585	1
Test 12	4	61.5760	62.1555	59.4365	60.0812	58.3759	60.3208	61.3596	3
	5	61.3917	63.3598	61.0428	61.9402	61.4223	61.9402	62.0135	2
	6	61.6110	62.1002	61.4223	61.7545	61.3917	61.7545	63.6124	1
	7	61.8628	62.5811	62.1002	64.2639	61.4530	62.4126	73.8039	1
	8	62.1002	62.2648	63.9108	63.1198	61.5146	61.7016	63.3598	2
Test 13	4	67.7237	68.4176	68.3316	67.8623	68.1863	67.3564	69.0028	1
	5	69.2107	68.3316	67.1008	68.5559	69.4209	68.2723	70.0480	1
	6	70.0480	71.1130	71.5164	70.3627	70.7886	73.0226	78.6382	1
	7	69.7378	71.5164	74.6986	69.6235	69.7378	72.1503	72.8815	2
	8	72.1503	70.0480	71.1130	70.0480	70.3627	70.5704	74.6986	1
Test 14	4	73.5448	73.9165	75.5417	75.7899	75.7899	74.5209	77.2970	1
	5	74.5209	73.5448	74.9622	74.1851	81.4523	80.3018	75.7899	3
	6	77.8100	74.9622	74.1851	72.1924	76.2105	75.3351	78.8516	1
	7	77.2970	74.5209	73.9165	72.1924	74.9622	73.5448	78.1890	1
	8	73.0576	74.5209	76.9808	76.5108	73.0576	74.5209	81.4523	1

MFO, the MSA, PSO, WWO and WOA based on Kapur's entropy are given in Table 4. The PSNR values are applied to evaluate the distortion degree of each segmented image. A larger PSNR value indicates a smaller distortion degree. For different threshold levels, the PSNR values of the WOA are better than those of the other algorithms, which indicates that the WOA has certain advantages in solving the image segmentation problem. In Table 4, as the threshold level increases, the PSNR values of each algorithm increases accordingly. To better compare the PSNR values, a ranking is carried out based on the sizes of the PSNR values. The higher the ranking is, the better the PSNR value. For each segmented image, the number of threshold levels is 4, 5, 6, 7 and 8. In other words, each algorithm has 70 PSNR values. For the WOA, the number of first-place rankings is 51, the number of second-place rankings is 13, and the number of third-place rankings is 6. The average PSNR values of the algorithms over all images are given in Fig. 6. The WOA has a higher average PSNR, and the difference between the WOA and the other algorithms becomes increasingly obvious, which indicates that the overall optimization performance of the WOA based on Kapur's entropy is better than that of other algorithms. The experimental results reveal that the WOA not only is suitable for solving the function optimization problem but also has strong practicability in image segmentation.

The SSIM values, which are based on the brightness, contrast and structural information, are used to evaluate the similarity between the original image and the segmented image. The SSIM values of the segmented images obtained by the BA, the FPA, MFO, the MSA, PSO, WWO and the WOA based on Kapur's entropy are given in Table 5. As the threshold level increases, the distortion of the segmented images decreases, and the segmented images become closer to the original image. Similarly, the ranking is based on the size of the SSIM value. A higher ranking indicates that the WOA contains more image segmentation information. Each algorithm has 70 SSIM values. For the WOA, the number of first-place rankings is 57, the number of second-place rankings is 9, and the number of third-place rankings is 4. The SSIM values of the WOA are obviously superior to those of the other algorithms; hence, the WOA obtains better threshold values and global optimal solutions. Compared with the other algorithms, the WOA has better optimization performance and higher similarity. The average SSIM values of the algorithms over all images are given in Fig. 7. The SSIM values of the WOA are superior to those of the other algorithms, and the difference between the WOA and the comparison algorithms is significant. The experimental results reveal that the WOA has a higher calculation accuracy and certain advantages in obtaining optimal or suboptimal SSIM values and has the ability to solve complex images.

TABLE 5. The SSIM of each algorithm.

Images	k	SSIM values							
		BA	FPA	MFO	MSA	PSO	WWO	WOA	Rank
Test 1	4	0.8220	0.7615	0.8262	0.8231	0.8209	0.7457	0.9274	1
	5	0.7087	0.6992	0.8109	0.8164	0.8213	0.8151	0.8866	1
	6	0.8636	0.7055	0.8585	0.8511	0.8325	0.7755	0.8939	1
	7	0.8701	0.8723	0.7744	0.8173	0.7920	0.8994	0.9067	1
	8	0.8087	0.8742	0.7898	0.8067	0.8169	0.8272	0.8890	1
Test 2	4	0.8339	0.9287	0.8131	0.8368	0.8642	0.8149	0.9395	1
	5	0.9091	0.8879	0.9094	0.9288	0.8970	0.9242	0.9473	1
	6	0.9223	0.8886	0.8854	0.9042	0.9419	0.8914	0.9129	3
	7	0.8636	0.8539	0.9203	0.9082	0.8196	0.9272	0.9543	1
	8	0.9122	0.8101	0.8087	0.8884	0.7985	0.8884	0.9143	1
Test 3	4	0.9109	0.9173	0.7653	0.7982	0.7533	0.9171	0.9110	3
	5	0.8880	0.8933	0.8938	0.8657	0.8976	0.6437	0.9063	1
	6	0.8410	0.9139	0.6419	0.8779	0.7594	0.8493	0.9253	1
	7	0.7601	0.6817	0.9042	0.8966	0.8919	0.7888	0.9099	1
	8	0.8033	0.9281	0.8838	0.8811	0.8579	0.8183	0.9345	1
Test 4	4	0.7906	0.6669	0.7957	0.7517	0.8188	0.7738	0.8729	1
	5	0.7165	0.8128	0.7602	0.5697	0.7788	0.8345	0.8477	1
	6	0.8033	0.7656	0.8125	0.8026	0.7983	0.7713	0.8988	1
	7	0.7285	0.6807	0.7510	0.7828	0.7651	0.7309	0.8335	1
	8	0.7324	0.7463	0.7954	0.7473	0.6899	0.7621	0.9059	1
Test 5	4	0.8535	0.9169	0.9202	0.9271	0.9325	0.8392	0.9358	1
	5	0.9547	0.8689	0.8615	0.9154	0.9335	0.9552	0.9349	3
	6	0.8065	0.8913	0.9193	0.9119	0.8814	0.7983	0.9259	1
	7	0.7408	0.7334	0.9557	0.9512	0.8982	0.8977	0.9562	1
	8	0.8353	0.7468	0.9151	0.8455	0.8454	0.8836	0.9202	1
Test 6	4	0.8754	0.6845	0.6900	0.8196	0.6888	0.7666	0.9306	1
	5	0.8194	0.7618	0.5741	0.7627	0.8619	0.8477	0.8838	1
	6	0.9016	0.8061	0.8971	0.8280	0.7941	0.8017	0.9250	1
	7	0.8446	0.8195	0.8671	0.6766	0.7425	0.7432	0.8634	2
	8	0.9054	0.9143	0.7720	0.9125	0.8090	0.8677	0.9513	1
Test 7	4	0.6881	0.7112	0.8103	0.7120	0.6785	0.6871	0.8966	1
	5	0.8752	0.8668	0.8631	0.8689	0.8040	0.8709	0.9064	1
	6	0.7878	0.8195	0.8215	0.8081	0.8158	0.8845	0.9289	1
	7	0.8174	0.8644	0.7502	0.7519	0.7470	0.7600	0.8528	2
	8	0.8451	0.8102	0.8241	0.6882	0.7762	0.8326	0.8409	2
Test 8	4	0.8316	0.8599	0.7897	0.9243	0.8658	0.8498	0.9262	2
	5	0.8127	0.8608	0.7918	0.8677	0.9198	0.8602	0.9202	1
	6	0.9121	0.9251	0.7016	0.8236	0.8482	0.9076	0.9308	1
	7	0.7922	0.9259	0.9210	0.9107	0.7896	0.9164	0.9524	1
	8	0.7896	0.9030	0.9044	0.8632	0.8978	0.7821	0.9378	1
Test 9	4	0.7971	0.7762	0.8463	0.8891	0.8499	0.7729	0.9171	1
	5	0.7706	0.9179	0.9290	0.8987	0.8749	0.8718	0.9254	2
	6	0.8535	0.8208	0.7338	0.8880	0.8304	0.8533	0.9284	1
	7	0.8847	0.8511	0.8323	0.8045	0.8380	0.8778	0.9015	1
	8	0.9212	0.8925	0.8759	0.8743	0.9118	0.8786	0.9155	2
Test 10	4	0.8999	0.6799	0.6993	0.7150	0.6757	0.6762	0.9297	1
	5	0.8678	0.6125	0.8752	0.8680	0.8727	0.8644	0.9294	1

TABLE 5. *(Continued.)* The SSIM of each algorithm.

	6	0.7642	0.7900	0.8066	0.8068	0.8066	0.5508	0.8963	1
	7	0.7641	0.8892	0.7434	0.8112	0.7535	0.9000	0.9347	1
	8	0.6974	0.8418	0.9303	0.7575	0.7302	0.6950	0.9534	1
Test 11	4	0.8945	0.8609	0.7926	0.6744	0.6781	0.6567	0.9326	1
	5	0.5786	0.8244	0.8194	0.7102	0.7862	0.7638	0.8697	1
	6	0.8992	0.6324	0.8030	0.8709	0.7985	0.7429	0.8947	2
	7	0.8300	0.8534	0.6769	0.8143	0.8517	0.7973	0.8675	1
	8	0.8042	0.8875	0.8043	0.7966	0.8314	0.6952	0.9092	1
Test 12	4	0.8719	0.8488	0.6384	0.8238	0.6192	0.6535	0.8869	1
	5	0.7725	0.8463	0.7969	0.9003	0.8754	0.8944	0.9071	1
	6	0.8798	0.9032	0.8261	0.8360	0.8409	0.8422	0.9037	1
	7	0.8547	0.8189	0.7998	0.8115	0.7831	0.8624	0.8969	1
	8	0.7795	0.8634	0.8560	0.8217	0.8620	0.8623	0.8671	1
Test 13	4	0.8383	0.6573	0.8111	0.8282	0.8608	0.8588	0.9253	1
	5	0.8565	0.8662	0.8044	0.7586	0.8589	0.7561	0.9544	1
	6	0.8754	0.8759	0.7624	0.8053	0.8845	0.7908	0.8814	2
	7	0.8639	0.9271	0.9048	0.8060	0.9148	0.9054	0.9383	1
	8	0.8743	0.9045	0.8342	0.9222	0.8863	0.7798	0.9100	2
Test 14	4	0.7209	0.7301	0.7445	0.7059	0.7368	0.7347	0.8377	1
	5	0.6471	0.6396	0.6497	0.7087	0.8964	0.8975	0.9235	1
	6	0.8311	0.5723	0.8000	0.5843	0.6044	0.8077	0.8530	1
	7	0.6874	0.7513	0.7681	0.7156	0.6519	0.5240	0.7376	3
	8	0.6829	0.6845	0.7261	0.6795	0.7107	0.6528	0.7264	1

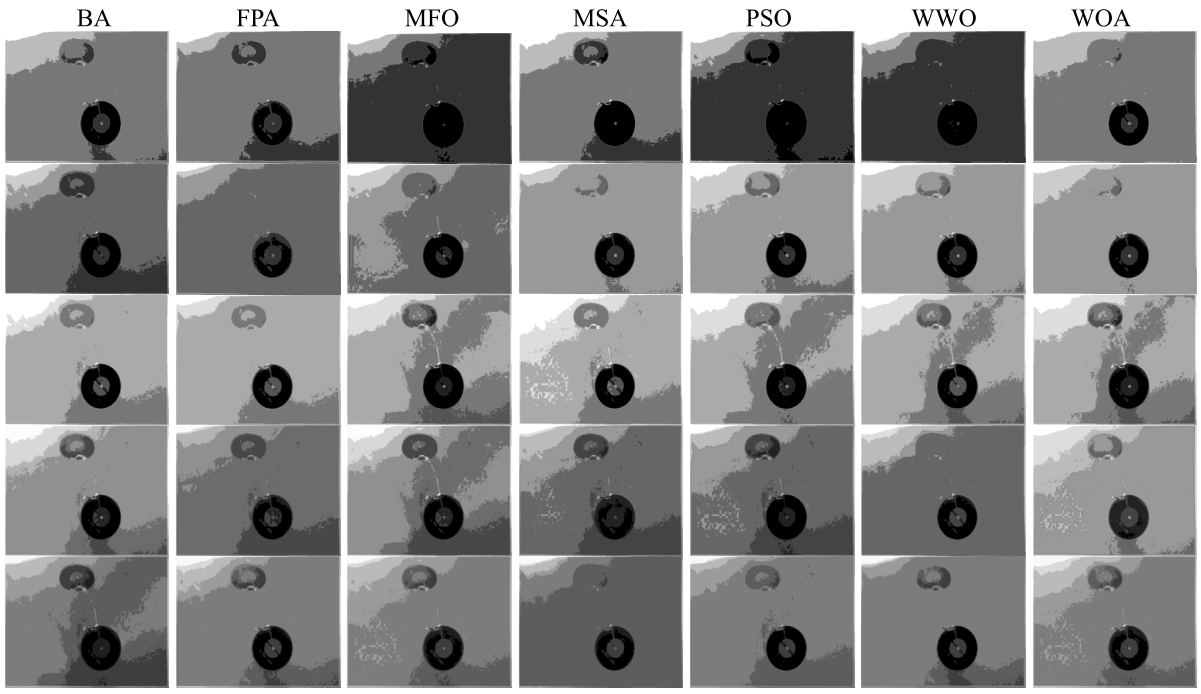

FIGURE 20. Segmented images of Test 12.

The average execution time of each algorithm is given in Table 6. As the threshold level increases, the execution time of each algorithm increases accordingly due to the computational complexity of the algorithms, which indicates that the algorithms consume more time at a higher threshold level. Compared with the other algorithms, the WOA has

TABLE 6. The average execution time of each algorithm.

Images	k	Execution time (in second)							
		BA	FPA	MFO	MSA	PSO	WWO	WOA	Rank
Test 1	4	3.3342	3.4914	2.1015	5.2185	3.5700	5.1318	2.6658	2
	5	4.4329	2.7139	2.5635	6.0095	4.2126	6.6766	3.5115	3
	6	4.3345	4.1562	4.1026	5.5584	5.6782	9.9858	4.0788	1
	7	10.934	11.773	11.497	12.574	12.789	21.984	10.607	1
	8	47.472	47.681	48.683	49.519	50.976	71.162	46.396	1
Test 2	4	3.2732	3.4127	2.0197	5.2195	5.1846	5.1245	2.4169	2
	5	3.6121	3.9565	3.6333	5.7510	5.8348	6.3590	3.2133	1
	6	5.3053	5.5262	5.2427	7.1463	7.3330	8.6719	4.9558	1
	7	11.414	11.634	11.235	13.125	13.866	16.462	11.049	1
	8	70.764	39.880	35.846	39.274	43.388	43.805	31.234	1
Test 3	4	2.1269	2.2723	5.5898	5.1138	3.5602	5.5528	2.9241	3
	5	2.3078	2.4516	2.2508	5.4540	3.8582	5.6108	2.3066	2
	6	2.8282	3.0556	4.2145	4.3491	4.3321	6.7587	4.0745	3
	7	4.8885	4.9869	4.7606	6.0449	6.0520	9.3607	4.1043	1
	8	35.529	12.104	32.687	13.400	13.304	26.564	18.591	4
Test 4	4	3.5956	3.7460	5.6696	9.7696	3.4535	5.4774	4.9235	4
	5	5.4786	5.8768	3.4170	4.7840	4.8916	8.6549	4.3903	2
	6	16.055	14.135	10.759	11.109	11.758	19.699	10.390	1
	7	77.007	79.873	52.610	52.496	50.546	67.195	51.828	2
	8	341.83	467.58	364.47	347.93	344.54	367.16	345.95	3
Test 5	4	3.4939	3.5692	3.2400	5.2026	5.5881	5.4329	1.8581	1
	5	3.5914	3.7407	3.6797	5.3831	5.9754	6.4557	3.0664	1
	6	2.6044	4.1190	3.7737	5.6897	6.3085	7.4741	3.5943	2
	7	4.6557	4.8921	4.5795	6.5229	7.3631	9.1417	2.8914	1
	8	6.9315	6.9835	4.3649	8.3816	9.4764	9.9192	6.9397	3
Test 6	4	2.2529	3.7102	2.1635	3.5746	5.7280	3.4702	1.9663	1
	5	2.3948	3.8142	2.3649	3.6963	3.9066	4.1244	2.0371	1
	6	2.8227	4.3351	2.5621	3.9627	6.5899	4.8309	2.4222	1
	7	3.4369	3.4369	4.2676	4.7063	7.5493	6.1147	3.2746	1
	8	5.6373	6.9138	8.3635	6.7032	7.3005	8.9073	5.5695	1
Test 7	4	2.1607	3.5758	3.4552	3.5734	3.5166	3.4813	1.8405	1
	5	2.3447	3.8646	3.5980	5.4632	3.8661	4.2872	2.0375	1
	6	2.7627	4.4731	4.1965	6.2795	4.3674	4.9357	2.7097	1
	7	4.0316	6.4636	5.0992	8.0257	5.7709	6.6730	3.9676	1
	8	8.5966	13.851	10.586	14.566	9.6952	12.210	8.7047	2
Test 8	4	2.2308	5.0117	3.3171	5.1853	5.6690	3.3941	2.8275	2
	5	2.3905	3.8172	3.7483	5.5916	5.9452	4.0139	3.0899	2
	6	2.5164	4.0638	3.4876	4.7601	6.8685	4.7553	2.4522	1
	7	3.2393	5.0841	4.3031	6.7760	7.8004	5.9063	3.0523	1
	8	5.2404	8.2737	10.087	6.2333	11.481	9.6968	5.1996	1
Test 9	4	2.2370	3.1011	2.0249	3.3835	3.4251	5.1257	2.0367	2
	5	2.0905	3.8550	2.2895	3.6307	3.7300	6.2974	2.0901	1
	6	3.1401	5.1121	2.8980	4.2647	4.4301	8.4456	2.9904	2
	7	5.2279	8.7116	5.0695	6.3485	6.8640	12.286	5.8968	3
	8	13.941	23.430	14.450	24.212	16.299	26.291	12.955	1
Test 10	4	2.2076	3.5924	3.2580	5.3527	5.4262	3.3989	1.8474	1
	5	2.3589	3.8264	2.3513	3.7307	5.6812	4.1056	1.9604	1
	6	2.8166	4.5317	2.7621	6.1859	4.3601	5.1646	2.6748	1
	7	4.2634	7.0070	4.3252	9.1599	5.8822	6.9339	4.2847	2
	8	9.2835	14.811	9.2987	10.584	11.713	13.325	9.2468	1
Test 11	4	2.1758	3.7490	2.1665	3.6196	3.6506	3.3973	2.1096	1
	5	2.4512	3.9901	2.4233	3.7441	3.9009	4.1611	3.2589	3
	6	2.9751	5.0182	2.9796	4.4254	4.7271	5.3269	2.9704	1
	7	5.8383	8.7282	5.4820	6.6884	6.7284	8.2101	5.4696	1
	8	14.587	27.919	14.785	15.343	15.222	17.817	15.102	3

TABLE 6. *(Continued.)* The average execution time of each algorithm.

Test 12	4	2.3862	2.3940	3.5610	3.6237	3.8533	3.5090	2.1239	1
	5	2.3047	2.5445	3.6898	3.6936	4.1476	4.1650	2.3502	2
	6	2.5430	2.6360	3.8815	3.7717	4.4163	4.7628	2.3245	1
	7	2.9361	2.8506	4.2550	3.9892	4.6578	5.5420	2.6415	1
	8	3.3892	3.3271	4.8361	4.4560	5.0608	6.3479	3.1419	1
Test 13	4	2.2442	3.5445	2.9773	3.4699	3.5136	3.7083	1.8815	1
	5	2.3289	3.8736	3.2227	3.6443	5.6507	4.1277	2.0854	1
	6	2.7953	4.4660	2.6940	4.1316	5.7309	4.9628	2.5595	1
	7	4.1920	6.6801	5.7597	5.5452	9.4193	7.4494	4.1045	1
	8	9.4268	14.549	14.707	10.491	16.652	12.341	9.0069	1
Test 14	4	2.0729	2.5112	2.2739	3.4322	6.2262	3.3870	1.9324	1
	5	2.3549	3.6675	3.5547	3.5772	5.8460	4.0554	2.0722	1
	6	2.7006	4.3042	6.5669	3.9884	6.4919	4.7903	2.3951	1
	7	3.7075	5.7662	5.0214	4.9690	8.6358	6.5493	3.5848	1
	8	7.5366	8.2241	9.5286	8.2228	24.216	10.583	7.5239	1

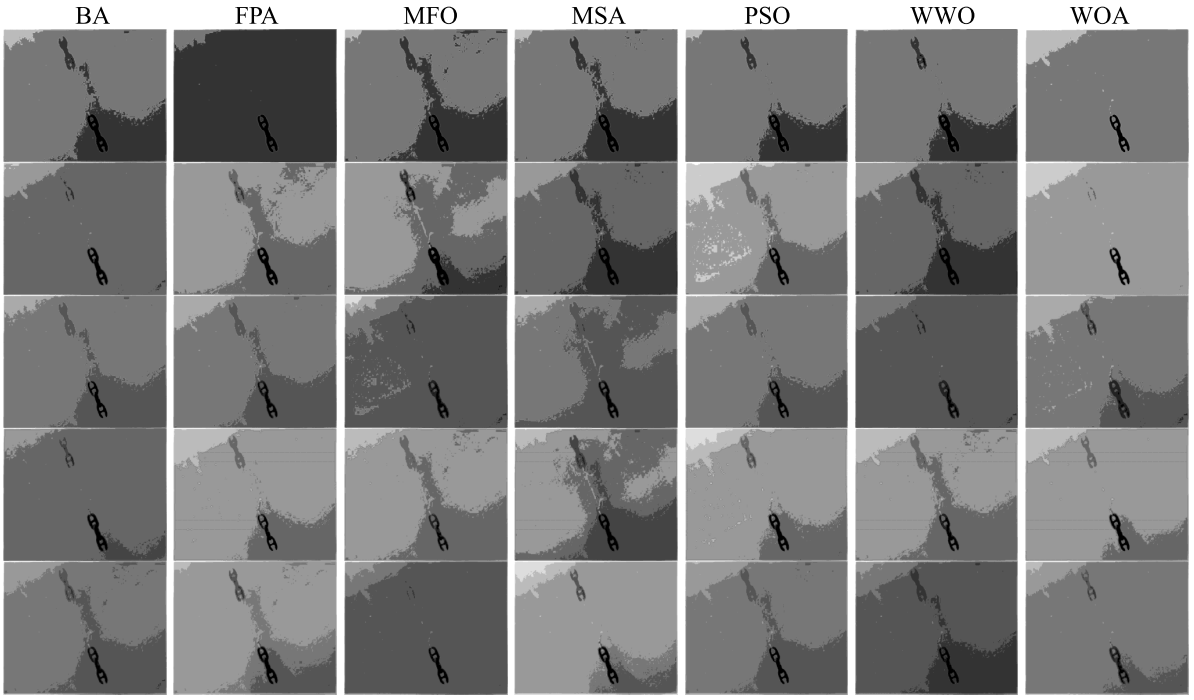

FIGURE 21. Segmented images of Test 13.

a faster convergence speed and requires less time because it has a higher search efficiency and thus can avoid premature convergence. The better fitness values, PSNR values and SSIM values show that the WOA has stronger segmentation performance. The average time of the algorithms over all images is given in Fig. 8. In terms of the computing time, the order of the algorithm is WOA < MFO < BA < MSA < PSO < FPA < WWO. The experimental results reveal that the WOA can complete the effective segmentation of images in a relatively shorter time and achieve a higher segmentation accuracy.

To verify the superiority and feasibility of the WOA, for all algorithms, the population size is 30, the maximum number of iterations is 100, and the number of independent operations is 30. In the data statistics, all experimental data based on Kapur's entropy are used for determining the *p-values* of the Wilcoxon rank-sum. A *p-value* less than 0.05 indicates a significant difference between the WOA and the other algorithms. If the *p-value* is greater than 0.05, then there is no significant difference between the WOA and the other algorithms (shown in bold). The experimental results reveal that there is a significant difference between the two groups

TABLE 7. The p-values of the Wilcoxon rank-sum.

Images	k	Wilcoxon test					
		WOA vs BA	WOA vs FPA	WOA vs MFO	WOA vs MSA	WOA vs PSO	WOA vs WWO
Test 1	4	4.0345E-11	1.5983E-10	7.0134E-08	2.6422E-02	1.3951E-03	2.9897E-11
	5	4.2635E-11	8.8672E-10	2.4712E-09	4.2136E-04	2.2691E-05	3.0104E-11
	6	5.5574E-10	9.9186E-11	2.5106E-10	1.3345E-02	1.5584E-07	3.1589E-10
	7	8.8910E-10	5.4941E-11	2.9924E-11	1.9963E-05	5.3211E-08	3.0199E-11
	8	1.9523E-10	3.0199E-11	2.2695E-11	2.1702E-03	1.4553E-07	3.0199E-11
Test 2	4	8.5590E-10	1.5227E-10	1.6859E-06	1.1211E-02	2.3570E-03	5.5270E-10
	5	1.4029E-09	1.0696E-09	1.6571E-10	2.8126E-02	5.3839E-06	3.0180E-11
	6	5.8342E-10	7.3891E-11	3.0084E-10	**5.7929E-01**	1.7507E-03	2.1544E-10
	7	2.0862E-10	4.5043E-11	2.5126E-11	3.5638E-04	6.2346E-06	3.0199E-11
	8	3.0123E-11	1.6132E-10	2.3905E-11	1.1882E-03	5.0270E-06	3.0199E-11
Test 3	4	8.4567E-09	3.9966E-10	4.2798E-05	1.2595E-02	3.6228E-03	5.5635E-10
	5	9.4432E-10	3.4645E-10	1.3330E-08	**9.3519E-01**	5.5423E-03	3.0104E-11
	6	3.0161E-11	2.1544E-10	2.1709E-11	6.7869E-03	9.4818E-04	3.0199E-11
	7	7.8830E-11	3.0199E-11	2.1150E-11	2.9205E-02	3.3629E-08	3.0199E-11
	8	2.9524E-11	3.0199E-11	2.2319E-11	1.9073E-03	2.3307E-08	3.0199E-11
Test 4	4	8.4560E-11	7.9975E-10	5.1094E-09	5.5119E-10	2.0790E-04	9.8013E-11
	5	3.7693E-10	5.4998E-10	5.5799E-10	9.9145E-03	1.1775E-03	2.9747E-11
	6	7.7384E-11	7.7384E-11	2.3496E-11	5.5727E-10	4.0639E-07	3.0199E-11
	7	4.4437E-11	9.8013E-11	2.0552E-11	5.5119E-10	2.7325E-06	2.9822E-11
	8	1.3250E-10	3.0179E-11	2.3316E-11	1.4635E-10	1.2791E-07	3.0180E-11
Test 5	4	1.5578E-10	4.0671E-11	1.1871E-01	1.7610E-02	**6.6261E-01**	3.0123E-11
	5	1.9762E-09	3.4742E-10	2.9409E-08	9.2113E-05	1.1668E-05	4.9752E-11
	6	1.6825E-09	6.6955E-11	3.6277E-11	**3.7904E-01**	3.8259E-05	5.5727E-10
	7	3.1235E-10	1.7769E-10	9.0094E-11	1.7613E-02	7.5457E-07	3.0199E-11
	8	4.0671E-11	3.0199E-11	2.6253E-11	**9.1171E-01**	1.3193E-04	3.0199E-11
Test 6	4	6.1443E-08	7.0146E-08	3.4537E-03	3.7715E-02	**6.2310E-02**	8.3848E-09
	5	4.4700E-10	5.5666E-10	5.8772E-08	5.2622E-04	5.8098E-04	3.3342E-11
	6	1.1857E-07	2.6099E-10	3.3255E-11	3.7904E-04	4.9156E-09	4.6159E-10
	7	8.8435E-10	3.0199E-11	2.8191E-10	4.9818E-04	7.5803E-06	3.0199E-11
	8	4.2767E-11	3.0199E-11	2.2842E-11	1.8090E-02	9.6072E-05	3.0199E-11
Test 7	4	9.6779E-10	8.0536E-10	1.8261E-02	9.7677E-05	1.8995E-03	2.9991E-11
	5	1.0071E-08	2.0328E-09	2.5694E-08	6.5668E-03	1.0515E-03	3.0180E-11
	6	3.0199E-11	3.3384E-11	2.6353E-11	**6.7350E-01**	3.3382E-03	3.0199E-11
	7	3.4723E-10	3.0199E-11	2.6087E-11	2.7726E-05	3.5906E-05	3.0199E-11
	8	3.0199E-11	3.0199E-11	2.7478E-11	6.7650E-05	2.8699E-06	3.0199E-11
Test 8	4	4.9266E-11	8.0753E-10	1.2531E-04	3.3631E-05	7.7565E-03	3.0085E-11
	5	2.5569E-10	1.6939E-09	2.8203E-09	1.3345E-03	1.3636E-03	2.6084E-10
	6	3.8249E-09	3.0199E-11	2.2507E-11	2.9047E-03	1.9984E-04	3.0199E-11
	7	3.0010E-11	3.0199E-11	1.9879E-11	2.0095E-02	3.8154E-09	3.0199E-11
	8	6.6793E-11	1.7769E-10	3.3778E-10	1.7649E-02	6.2568E-06	3.0199E-11
Test 9	4	5.9524E-11	7.3005E-11	4.6966E-09	1.0010E-08	5.0451E-04	2.9822E-11
	5	9.7084E-11	1.3289E-10	4.9722E-10	9.0307E-04	1.1791E-06	4.0772E-11
	6	3.0104E-11	3.0199E-11	2.4213E-11	**6.9522E-01**	1.7097E-06	3.0199E-11
	7	4.4739E-11	5.4941E-11	2.4524E-11	2.2823E-03	2.4273E-06	3.0199E-11
	8	1.0706E-10	8.1527E-11	2.7030E-11	1.0576E-03	2.7548E-05	4.6159E-10
Test 10	4	3.4401E-09	2.1384E-10	2.0694E-05	1.3255E-02	2.6669E-04	2.9953E-11
	5	3.7776E-10	5.4941E-11	2.8758E-10	4.0840E-05	1.8402E-06	3.0199E-11

TABLE 7. *(Continued.)* The p-values of the Wilcoxon rank-sum.

	6	1.6039E-10	3.0199E-11	1.6458E-10	**6.1001E-01**	5.4376E-08	3.0199E-11
	7	1.0679E-09	4.6159E-10	4.2005E-11	4.9818E-04	3.8243E-06	3.0199E-11
	8	2.1532E-10	6.0658E-11	2.7460E-11	7.9590E-03	3.4691E-07	3.0199E-11
Test 11	4	3.4586E-08	2.1482E-10	4.7256E-06	7.1971E-05	1.4384E-04	5.5696E-10
	5	3.7025E-10	8.8862E-10	1.7207E-09	4.0595E-02	5.3133E-07	3.0199E-11
	6	8.8720E-10	1.9568E-10	1.8206E-09	7.2884E-03	2.2715E-06	4.6159E-10
	7	6.8720E-11	4.5043E-11	3.7479E-11	4.1191E-03	1.2406E-06	3.6897E-11
	8	9.7142E-11	3.0199E-11	2.4260E-11	1.8577E-03	1.5977E-06	3.0199E-11
Test 12	4	3.6963E-10	2.8265E-08	4.6014E-04	1.2965E-03	2.0647E-02	7.1050E-09
	5	2.8190E-08	8.1014E-10	1.4999E-05	7.9590E-03	8.8346E-06	3.3384E-11
	6	6.1044E-10	1.2057E-10	2.3951E-11	6.4142E-04	1.1399E-07	3.0199E-11
	7	2.8393E-10	7.3891E-11	4.6764E-11	2.0283E-07	6.4901E-09	3.0199E-11
	8	1.2736E-09	3.0199E-11	1.9507E-11	2.3800E-03	2.6699E-06	3.0199E-11
Test 13	4	2.6101E-10	3.0123E-11	1.6153E-05	2.0514E-03	3.2332E-04	3.0142E-11
	5	8.0555E-11	5.9385E-09	9.0467E-11	5.9964E-05	9.4896E-04	3.6669E-11
	6	5.9924E-11	1.7769E-10	2.5206E-11	7.9782E-05	1.2974E-03	3.0199E-11
	7	1.3211E-10	7.3891E-11	2.5558E-11	4.8252E-02	2.1252E-04	3.0199E-11
	8	4.1188E-10	3.0199E-11	4.3571E-11	3.2553E-02	2.4296E-05	3.0199E-11
Test 14	4	3.7651E-10	8.1186E-11	5.2138E-05	2.4865E-06	2.2530E-03	3.0085E-11
	5	1.7413E-10	1.1023E-08	1.3299E-09	1.3250E-04	3.8174E-05	3.0199E-11
	6	4.0746E-11	4.0772E-11	2.0453E-10	1.0869E-02	2.3505E-07	3.0199E-11
	7	4.9630E-11	6.6955E-11	N/A	4.4592E-04	1.3422E-07	3.0199E-11
	8	1.7759E-10	4.9752E-11	3.7340E-11	6.9125E-04	3.3555E-05	3.0199E-11

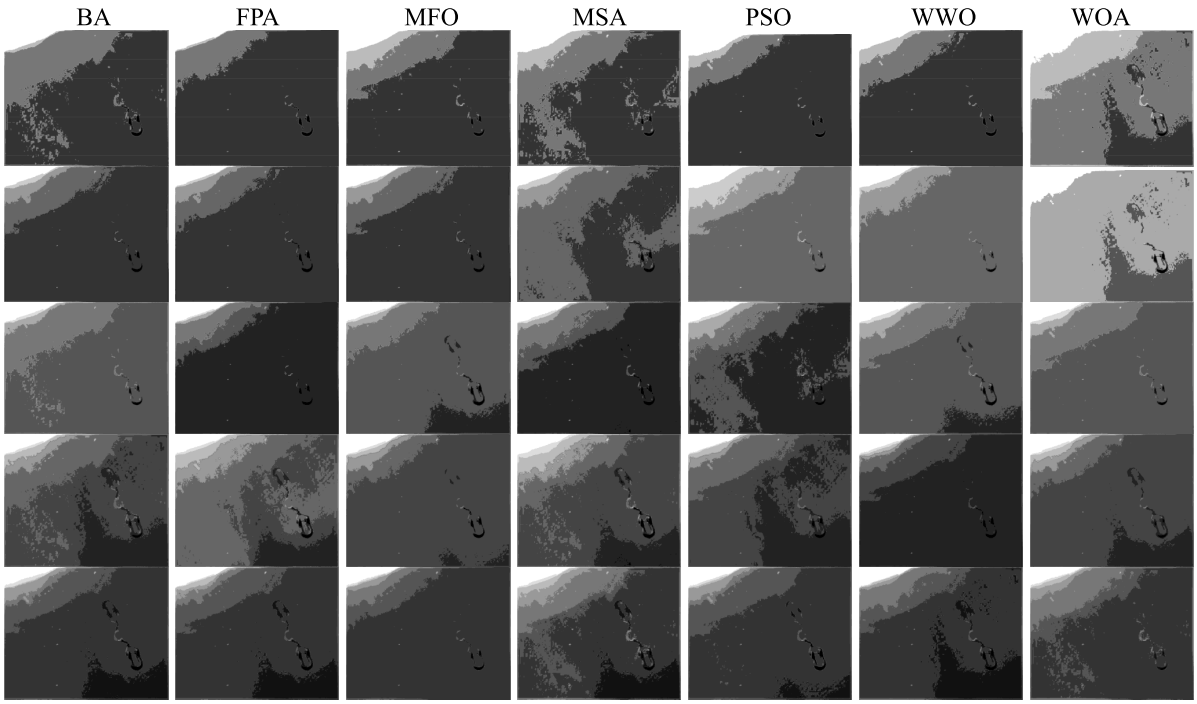

FIGURE 22. Segmented images of Test 14.

of data in most cases and that the data are not obtained by accident. The WOA is an effective and feasible method for achieving a better segmentation effect.

Figs. 9-22 show the results of image segmentation of the WOA and the other algorithms under different threshold levels. As the threshold level increases, the overall effect of

the image segmentation improves, which indicates that the segmented image contains more information. The segmentation effect of the WOA is obviously superior to that of the other algorithms, and a segmented image at a higher threshold level is closer to the original image. The WOA based on Kapur's entropy obtains the optimal fitness values and the best threshold values, which indicates that the WOA has a higher calculation accuracy and better segmentation effect. The PSNR and SSIM values of the WOA are better than those of the other algorithms, which indicates that the WOA has less image distortion and a higher structural similarity. The execution time of the WOA is less than those of the other algorithms by over 30 runs; thus, the WOA has a faster convergence speed. The p-values of the Wilcoxon rank-sum verify that there is a significant difference between the WOA and the other algorithms. In summary, the WOA has a better segmentation performance and stronger robustness in solving the image segmentation problem.

Statistically, the WOA simulates encircling of the prey, the bubble-net attacking method and the search for prey to obtain the global solution. The WOA can effectively solve the image segmentation problem for the following reasons. First, the WOA not only is simpler to operate, has few control parameters and is easy to implement but also has a wider search space and a stronger search ability; thus, it can avoid falling into a local optimum. Second, the WOA has a good position update mechanism. Once the optimal position of a humpback whale is determined, the remaining humpback whales will swim to the optimal position and constantly update their positions; hence, the WOA has a strong global search ability and robustness. Third, for control parameter $|\overrightarrow{A}|$, if $|\overrightarrow{A}| > 1$, the humpback whale randomly selects the positions of the whales to update its position. This search mechanism expands the optimization space. To summarize, the WOA can effectively switch between exploration and exploitation to enhance the overall optimization performance.

VI. CONCLUSIONS AND FUTURE RESEARCH

The purpose of image segmentation is to consume less time and obtain better segmentation results. This paper presents a WOA based on Kapur's entropy method to achieve multilevel thresholding image segmentation, and the objective function of the WOA is maximized to obtain the optimal threshold values. As the thresholds increase, the difference between the WOA and the other algorithms increases. Compared with the other algorithms, the WOA can switch between the global search ability and local search ability to obtain the optimal solution. The experimental results show that the WOA has a higher segmentation accuracy and faster convergence speed in terms of the fitness value, PSNR, SSIM, execution time and Wilcoxon's rank-sum test. Meanwhile, the WOA has good practicality and strong robustness and thus can effectively complete the image segmentation task.

In future research, the WOA will be used to address higher threshold values or solve the colour image segmentation problem. Underwater image segmentation methods will be applied to meet the needs of image segmentation, such as region-based method, edge-based method, clustering-based method and graph-based method. The thresholding-based method will be compared with other image segmentation methods to further verify the overall performance of the WOA in solving the image segmentation problem. In addition, advanced strategies or combined algorithms will be used to improve the convergence speed and calculation accuracy.

ACKNOWLEDGMENTS

The authors would like to thank everyone involved for their contribution to this article.

REFERENCES

[1] J. Xu, P. Bi, X. Du, and J. Li, "Robust PCANet on target recognition via the UUV optical vision system," *Optik*, vol. 181, pp. 588–597, Mar. 2019.
[2] Z. Yan, P. Gong, W. Zhang, Z. Li, and Y. Teng, "Autonomous underwater vehicle vision guided docking experiments based on L-shaped light array," *IEEE Access*, vol. 7, pp. 72567–72576, Jun. 2019.
[3] J. Xu, P. Bi, X. Du, J. Li, and D. Chen, "Generalized robust PCA: A new distance metric method for underwater target recognition," *IEEE Access*, vol. 7, pp. 51952–51964, Apr. 2019.
[4] N. Tang, F. Zhou, Z. Gu, H. Zheng, Z. Yu, and B. Zheng, "Unsupervised pixel-wise classification for chaetoceros image segmentation," *Neurocomputing*, vol. 318, pp. 261–270, Nov. 2018.
[5] Y. Li, X. Bai, L. Jiao, and Y. Xue, "Partitioned-cooperative quantum-behaved particle swarm optimization based on multilevel thresholding applied to medical image segmentation," *Appl. Soft Comput.*, vol. 56, pp. 345–356, Jul. 2017.
[6] W. Chen, H. Yue, J. Wang, and X. Wu, "An improved edge detection algorithm for depth map inpainting," *Opt. Lasers Eng.*, vol. 55, pp. 69–77, Apr. 2014.
[7] F. Breve, "Interactive image segmentation using label propagation through complex networks," *Expert Syst. Appl.*, vol. 123, pp. 18–33, Jun. 2019.
[8] M. P. van den Heuvel, S. C. de Lange, A. Zalesky, C. Seguin, B. T. T. Yeo, and R. Schmidt, "Proportional thresholding in resting-state fMRI functional connectivity networks and consequences for patient-control connectome studies: Issues and recommendations," *NeuroImage*, vol. 152, pp. 437–449, May 2017.
[9] H. V. H. Ayala, F. M. D. Santos, V. C. Mariani, and L. D. S. Coelho, "Image thresholding segmentation based on a novel beta differential evolution approach," *Expert Syst. Appl.*, vol. 42, no. 4, pp. 2136–2142, Mar. 2015.
[10] A. K. Bhandari, V. K. Singh, A. Kumar, and G. K. Singh, "Cuckoo search algorithm and wind driven optimization based study of satellite image segmentation for multilevel thresholding using Kapur's entropy," *Expert Syst. Appl.*, vol. 41, no. 7, pp. 3538–3560, Jun. 2014.
[11] L. He and S. Huang, "Modified firefly algorithm based multilevel thresholding for color image segmentation," *Neurocomputing*, vol. 240, pp. 152–174, May 2017.
[12] X. Yang and X. He, "Bat algorithm: Literature review and applications," *Int. J. Bio-Inspir. Com.*, vol. 5, pp. 141–149, Aug. 2013.
[13] X. S. Yang, "Flower pollination algorithm for global optimization," in *Proc. Int. Conf. Unconventional Comput. Natural Comput.*, Dec. 2012, vol. 7445, no. 2, pp. 240–249.
[14] S. Mirjalili, "Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm," *Knowl.-Based Syst.*, vol. 89, pp. 228–249, Nov. 2015.
[15] A.-A.-A. Mohamed, Y. S. Mohamed, A. A. M. El-Gaafary, and A. M. Hemeida, "Optimal power flow using moth swarm algorithm," *Electr. Power Syst. Res.*, vol. 142, pp. 190–206, Jan. 2017.
[16] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. IEEE Int. Conf. Neural Netw.*, vol. 4, Nov. 1995, pp. 1942–1948.
[17] Y.-J. Zheng, "Water wave optimization: A new nature-inspired Meta-heuristic," *Comput. Oper. Res.*, vol. 55, pp. 1–11, Mar. 2015.
[18] Y. Zhou, X. Yang, Y. Ling, and J. Zhang, "Meta-heuristic moth swarm algorithm for multilevel thresholding image segmentation," *Multimedia Tools Appl.*, vol. 77, no. 18, pp. 23699–23727, Feb. 2018.
[19] M. A. E. Aziz, A. A. Ewees, and A. E. Hassanien, "Whale optimization algorithm and moth-flame optimization for multilevel thresholding image segmentation," *Expert Syst. Appl.*, vol. 83, pp. 242–256, Oct. 2017.

[20] S. Ouadfel and A. Taleb-Ahmed, "Social spiders optimization and flower pollination algorithm for multilevel image thresholding: A performance study," *Expert Syst. Appl.*, vol. 55, pp. 566–584, Aug. 2016.
[21] M.-A. Díaz-Cortés, N. Ortega-Sánchez, S. Hinojosa, D. Oliva, E. Cuevas, R. Rojas, and A. Demin, "A multi-level thresholding method for breast thermograms analysis using dragonfly algorithm," *Infr. Phys. Technol.*, vol. 93, pp. 346–361, Sep. 2018.
[22] R. K. Sambandam and S. Jayaraman, "Self-adaptive dragonfly based optimal thresholding for multilevel segmentation of digital images," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 30, no. 4, pp. 449–461, Oct. 2018.
[23] G. Sun, A. Zhang, Y. Yao, and Z. Wang, "A novel hybrid algorithm of gravitational search algorithm with genetic algorithm for multi-level thresholding," *Appl. Soft Comput.*, vol. 46, pp. 703–730, Sep. 2016.
[24] L. Shen, C. Fan, and X. Huang, "Multi-level image thresholding using modified flower pollination algorithm," *IEEE Access*, vol. 6, pp. 30508–30519, 2018.
[25] H. Gao, Z. Fu, C.-M. Pun, H. Hu, and R. Lan, "A multi-level thresholding image segmentation based on an improved artificial bee colony algorithm," *Comput. Electr. Eng.*, vol. 70, pp. 931–938, Aug. 2018.
[26] S. Pare, A. K. Bhandari, A. Kumar, and G. K. Singh, "A new technique for multilevel color image thresholding based on modified fuzzy entropy and Lévy flight firefly algorithm," *Comput. Electr. Eng.*, vol. 70, pp. 476–495, Aug. 2018.
[27] S. C. Satapathy, N. Sri Madhava Raja, V. Rajinikanth, A. S. Ashour, and N. Dey, "Multi-level image thresholding using otsu and chaotic bat algorithm," *Neural Comput. Appl.*, vol. 29, no. 12, pp. 1285–1307, Jun. 2018.
[28] B. Akay, "A study on particle swarm optimization and artificial bee colony algorithms for multilevel thresholding," *Appl. Soft Comput.*, vol. 13, no. 6, pp. 3066–3091, Jun. 2013.
[29] X. Bao, H. Jia, and C. Lang, "A novel hybrid Harris Hawks optimization for color image multilevel thresholding segmentation," *IEEE Access*, vol. 7, pp. 76529–76546, 2019.
[30] H. Jia, J. Ma, and W. Song, "Multilevel thresholding segmentation for color image using modified moth-flame optimization," *IEEE Access*, vol. 7, pp. 44097–44134, 2019.
[31] V. K. Bohat and K. V. Arya, "A new heuristic for multilevel thresholding of images," *Expert Syst. Appl.*, vol. 117, pp. 176–203, Mar. 2019.
[32] Z. Lu, Y. Qiu, and T. Zhan, "Neutrosophic C-means clustering with local information and noise distance-based kernel metric image segmentation," *J. Vis. Commun. Image Represent.*, vol. 58, pp. 269–276, Jan. 2019.
[33] X. Li, J. Song, F. Zhang, X. Ouyang, and S. U. Khan, "MapReduce-based fast fuzzy c-means algorithm for large-scale underwater image segmentation," *Future Gener. Comput. Syst.*, vol. 65, pp. 90–101, Dec. 2016.
[34] M. A. Elaziz, D. Oliva, A. A. Ewees, and S. Xiong, "Multi-level thresholding-based grey scale image segmentation using multi-objective multi-verse optimizer," *Expert Syst. Appl.*, vol. 125, pp. 112–129, Jul. 2019.
[35] D. Oliva, M. S. R. Martins, V. Osuna-Enciso, and E. F. de Morais, "Combining information from thresholding techniques through an evolutionary Bayesian network algorithm," *Appl. Soft Comput.*, vol. 90, May 2020, Art. no. 106147.
[36] E. Rodríguez-Esparza, L. A. Zanella-Calzada, D. Oliva, A. A. Heidari, D. Zaldivar, M. Pérez-Cisneros, and L. K. Foong, "An efficient Harris Hawks-inspired image segmentation method," *Expert Syst. Appl.*, vol. 155, Oct. 2020, Art. no. 113428.
[37] H. S. N. Alwerfali, M. Abd Elaziz, M. A. A. Al-Qaness, A. A. Abbasi, S. Lu, F. Liu, and L. Li, "A multilevel image thresholding based on hybrid salp swarm algorithm and fuzzy entropy," *IEEE Access*, vol. 7, pp. 181405–181422, Dec. 2019.
[38] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Adv. Eng. Softw.*, vol. 95, pp. 51–67, May 2016.
[39] J. N. Kapur, P. K. Sahoo, and A. K. C. Wong, "A new method for gray-level picture thresholding using the entropy of the histogram," *Comput. Vis., Graph., Image Process.*, vol. 29, no. 3, pp. 273–285, Mar. 1985.
[40] A. Aldahdooh, E. Masala, G. Van Wallendael, and M. Barkowsky, "Framework for reproducible objective video quality research with case study on PSNR implementations," *Digit. Signal Process.*, vol. 77, pp. 195–206, Jun. 2018.
[41] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.
[42] F. Wilcoxon, S. Katti, and R. A. Wilcox, "Critical values and probability levels for the Wilcoxon rank sum test and the Wilcoxon signed rank test," *Select. Tables Math. Statist.*, vol. 1, pp. 171–259, Jun. 1970.

ZHEPING YAN was born in Hangzhou, China, in 1972. He received the B.S. degree in nuclear power plant, the M.S. degree in marine and marine engineering special auxiliary equipment and system, and the Ph.D. degree in control theory and control engineering from Harbin Engineering University, Heilongjiang, China, in 1994, 1997, and 2001, respectively.

He completed the Postdoctoral Research in mechanical and electrical engineering with the Harbin Institute of Technology, China, in 2004. He was a Visiting Scholar with the National University of Singapore, Singapore, in 2015, Australian National University, Australia, in 2016, and the University of Guelph, Canada, in 2017. He is currently a Professor with the College of Automation and the Director of the Marine Assembly and Automatic Technology Institute, Harbin Engineering University. His current research interests include navigation system and control system of unmanned underwater vehicles, cooperative control of unmanned underwater vehicles, and data fusion.

JINZHONG ZHANG was born in Fuyang, China, in 1989. He received the M.S. degree in computer science from Guangxi University for Nationalities, China, in 2018. He is currently pursuing the Ph.D. degree in control science and engineering with Harbin Engineering University, China. His research interests include computation intelligence, image processing, and coordination control of multiple unmanned underwater vehicles.

ZEWEN YANG was born in Harbin, China, in 1991. He received the M.S. degree in control engineering from Northeast Forest University, China, in 2017. He is currently pursuing the Ph.D. degree in control science and engineering with Harbin Engineering University, China. His research interests include control system of unmanned underwater vehicles and coordination control of multiple unmanned underwater vehicles.

JIALING TANG was born in Suining, China, in 1984. He is currently pursuing the Ph.D. degree in control science and engineering with Harbin Engineering University, China. His research interest includes coordination control of multiple unmanned underwater vehicles.