



Article

Otsu Image Segmentation Algorithm Based on Hybrid Fractional-Order Butterfly Optimization

Yu Ma, Ziqian Ding *, Jing Zhang and Zhiqiang Ma

Department of Electrical and Electronic Engineering, Ningxia University, Yinchuan 750021, China; mayu95@163.com (Y.M.); pepsiizhang@163.com (J.Z.); 12022130684@stu.nxu.edu.cn (Z.M.)

* Correspondence: ding_3477@163.com; Tel.: +86-177-5247-3477

Abstract: To solve the drawbacks of the Otsu image segmentation algorithm based on traditional butterfly optimization, such as slow convergence speed and poor segmentation accuracy, this paper proposes hybrid fractional-order butterfly optimization with the Otsu image segmentation algorithm. G-L-type fractional-order differentiation is combined with the algorithm's global search to improve the position-updating method, which enhances the algorithm's convergence speed and prevents it from falling into local optima. The sine-cosine algorithm is introduced in the local search step, and Caputo-type fractional-order differentiation is used to avoid the disadvantages of the sine-cosine algorithm and to improve the optimization accuracy of the algorithm. By dynamically converting the probability, the ratio of global search to local search is changed to attain high-efficiency and high-accuracy optimization. Based on the 2-D grayscale gradient distribution histogram, the trace of discrete matrices between classes is chosen as the fitness function, the best segmentation threshold is searched for, image segmentation is processed, and three categories of images are chosen to proceed with the segmentation test. The experimental results show that, compared with traditional butterfly optimization, the convergence rate of hybrid fractional-order butterfly optimization with the Otsu image segmentation algorithm is improved by about 73.38%; meanwhile, it has better segmentation accuracy than traditional butterfly optimization.



Citation: Ma, Y.; Ding, Z.; Zhang, J.; Ma, Z. Otsu Image Segmentation Algorithm Based on Hybrid Fractional-Order Butterfly Optimization. *Fractal Fract.* **2023**, *7*, 871. <https://doi.org/10.3390/fractfract7120871>

Academic Editor: Viorel-Puiu Paun

Received: 15 November 2023

Revised: 28 November 2023

Accepted: 28 November 2023

Published: 8 December 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Image segmentation is a key component of image processing aimed at separating important or relevant localized areas within an image region. Only by segmenting the target area accurately can the features of the target image be extracted, analyzed, and measured. Image segmentation has many applications, including license plate recognition, medical image segmentation, object defect detection, and so on. Image thresholding segmentation is a simple and efficient method, and it is one of the most commonly used methods. The Otsu image segmentation algorithm is widely used because it has low computational complexity, high stability, and good segmentation effects. The one-dimensional Otsu algorithm has less histogram information, and the image segmentation effect is poor, so researchers proposed the 2-D Otsu algorithm, which enhances the segmentation effects significantly, although the computation of the 2-D Otsu algorithm was intensive. Reddi et al. [1] combined the differential evolutionary algorithm with the 2-D Otsu algorithm, which reduced the computational complexity and improved the speed of the algorithm. Zhang et al. [2] improved the neighborhood mean algorithm by using 2-D gray and gradient histograms and introduced the separation information between classes to improve the threshold function, which was simplified to reduce the computational complexity; however, the threshold vector was calculated directly, requiring a lot of time.

The key to the 2-D Otsu image segmentation algorithm is to find the threshold vector that best maximizes the dispersion between classes, and group intelligence algorithms are

gradually applied to threshold selection. The group intelligence algorithm significantly improves the effect of threshold segmentation, making it better than other methods. Huang et al. [3] proposed the Otsu image segmentation algorithm based on the Fruit Fly Optimization Algorithm (FOA), which was a fast image segmentation algorithm with high real-time capability. Liu et al. [4] suggested a multi-threshold Otsu image segmentation method, using a cell membrane and adaptive step size to improve the Firefly Algorithm (FA), which solved the problem of the local optimum and achieved a higher convergence speed and better segmentation results. The researchers in [5,6] used some methods to enhance the whale optimization algorithm (WOA) and proposed a hybrid whale optimization algorithm for Otsu multi-threshold image segmentation, which improved the efficiency and accuracy of image segmentation. However, the convergence rate of the above algorithm was slow, reaching convergence only after many iterations. Therefore, the algorithm convergence rate needed to improve.

The Butterfly Optimization Algorithm (BOA) is an intelligent algorithm [7] that simulates the behavior of butterflies, such as their search for food and mate selection. It is used to solve global optimization problems, and it has a simple principle, few parameters, and good results for optimal searches. It is also widely used in path planning, particle filtering, photovoltaic power generation, and so on. However, the traditional BOA algorithm has some shortcomings, such as low optimization accuracy and slow convergence speed. Therefore, there is considerable scope for enhancing its optimization stability and accuracy. Wu et al. [8] proposed a revised BOA algorithm that combines refractive opposition learning and adaptive inertia weights and used it to achieve multi-threshold image segmentation. This method improved the image segmentation accuracy and segmentation efficiency, but the convergence speed was slow.

Fractional-order calculus is a part of high-level mathematics and can be used in a wide range of fields, including engineering, artificial intelligence, and medicine. Pu et al. [9] illustrated the physical significance and an algorithm for implementing the fractional order in signal processing, which set the stage for using the fractional order in image processing. Micael et al. [10] enhanced the Particle Swarm Optimization algorithm through fractional-order differentiation and suggested that the convergence speed of an algorithm is related to the fractional order. The adaptive order of fractional-order differentiation was proposed by Wei [11], and she used fractional-order differentiation to improve the Particle Swarm Optimization (PSO) algorithm. This method improved the convergence speed of the algorithm and prevented it from falling into the local optimum; further, it introduced a symmetric particle distribution to improve the algorithm's optimization accuracy. Fractional-order calculus can improve the convergence rate of group intelligence algorithms, but few people use the fractional order to solve the disadvantages of traditional algorithms. Only several researchers have used fractional-order and intelligence algorithms to process image segmentation [12,13].

Therefore, this paper introduces fractional-order differentiation and other methods to avoid the shortcomings of the traditional BOA algorithm, and it proposes the hybrid fractional-order butterfly optimization algorithm (HFBOA), which uses fractional-order differentiation and adaptive flavor perception strength to improve the global position search of the BOA algorithm. At the local stage, it combines fractional-order differentiation with the sine-cosine algorithm as the position from which the local search is updated, improving the convergence speed and optimal search accuracy. This paper introduces dynamic conversion probability so that the algorithm performs a more intensive global search in the early stage and an enhanced local search in the later stage, preventing the BOA algorithm from falling into a local optimum. Finally, this paper proposes a 2-D Otsu image segmentation algorithm based on the HFBOA and demonstrates the use of this algorithm to perform image segmentation, improving the efficiency of the process.

2. The 2-D Otsu Threshold Segmentation Algorithm

The 2-D Otsu image segmentation algorithm adds the distribution of the difference between the original pixel gray level and the average gray level of the domain pixel to the pixel gray level distribution and produces a 2-D distribution histogram. The segmentation principle of the 2-D Otsu image segmentation algorithm is as follows: for a digital image $f(x, y)$ whose size is $M \times N$, the total pixel gray level is L , so the average gray value in the $k \times k$ neighborhood centered on (x, y) is $g(x, y)$:

$$g(x, y) = \frac{1}{k \times k} \sum_{l=-k}^k \sum_{q=-k}^k f(x+l, y+q) \quad (1)$$

where k , l , and q are integer numbers, meaning that $f(x, y)$ and $g(x, y)$ will take integer values.

The neighborhood grayscale gradient of the pixel is $|f(x, y) - g(x, y)|$. When the pixel's gray value $f(x, y) = i$ and the neighborhood grayscale gradient of the pixel $|f(x, y) - g(x, y)| = j$, then i, j are taken as a binary group (i, j) . The appearance frequency of the pixel point (i, j) is n_{ij} , and the corresponding point probability density p_{ij} is defined as follows:

$$p_{ij} = \frac{n_{ij}}{M \times N} \quad (2)$$

Assuming the binary group threshold (s, t) of the segmented image, where s is the grayscale threshold and the gradient threshold is t , the image can be segmented into four parts: the planar 2-D histogram is shown in Figure 1, where I and III represent the background and the target, respectively, and the II and IV parts represent the edge and the noise, respectively.

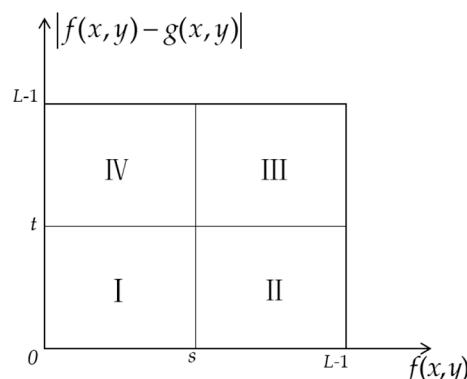


Figure 1. Planar 2-D distribution histogram. I and III represent the background and the target, and the II and IV parts represent the edge and the noise.

Because the target and the background cover a relatively large area of the whole image, the edge and the noise parts can be ignored when calculating the probability of every part. The probability of the background class is P_A , and the appearance probability of the target class is P_B . The calculations are as follows:

$$P_A(s, t) = \sum_{i=1}^s \sum_{j=1}^t p_{ij}, \quad P_B(s, t) = \sum_{i=s+1}^L \sum_{j=t+1}^L p_{ij} \quad (3)$$

The mean vectors of the background class μ_A and target class μ_B are

$$\mu_A(s, t) = (\mu_{Ai}, \mu_{Aj})^T = \left(\sum_{i=1}^s \sum_{j=1}^t \frac{ip_{ij}}{P_A}, \sum_{i=1}^s \sum_{j=1}^t \frac{jp_{ij}}{P_A} \right)^T \quad (4)$$

$$\mu_B(s, t) = (\mu_{Bi}, \mu_{Bj})^T = \left(\sum_{i=1}^s \sum_{j=1}^t \frac{ip_{ij}}{P_B}, \sum_{i=1}^s \sum_{j=1}^t \frac{jp_{ij}}{P_B} \right)^T \quad (5)$$

The total mean vector μ of the pixels is

$$\mu(s, t) = (\mu_i, \mu_j)^T = \left(\sum_{i=1}^L \sum_{j=1}^L ip_{ij}, \sum_{i=1}^L \sum_{j=1}^L jp_{ij} \right)^T \quad (6)$$

The inter-class dispersion matrix between the background class and the target class is

$$S(s, t) = P_A \times (\mu_A - \mu) \times (\mu_A - \mu)^T + P_B \times (\mu_B - \mu) \times (\mu_B - \mu)^T \quad (7)$$

The inter-class dispersion value means the difference between the background class and the target class. The larger the value, the larger the difference between the background and the target, and the better the image segmentation. To measure the dissimilarity, the trace of the dispersion matrix is used as the fitness function:

$$\text{tr}(S(s, t)) = P_A [(\mu_{Ai} - \mu_i)^2 + (\mu_{Aj} - \mu_j)^2] + P_B [(\mu_{Bi} - \mu_i)^2 + (\mu_{Bj} - \mu_j)^2] \quad (8)$$

When the trace of the dispersion matrix is the maximum, the best segmentation threshold (s^*, t^*) is obtained.

$$\text{tr}(S(s^*, t^*)) = \max(\text{tr}(S(s, t))) \quad (9)$$

3. Hybrid Fractional Butterfly Optimization Algorithm

3.1. Butterfly Optimization Algorithm

Butterflies use their senses of smell, sight, taste, touch, and hearing to find food, seek mates, and move from place to place. One of the most important senses is smell, which helps butterflies find food, and butterflies use it to locate the source of flavor; meanwhile, they can isolate different smells and perceive their intensity. In the BOA algorithm, when a butterfly senses the flavor of another butterfly, it will move toward it; when a butterfly is unable to sense the flavor of other butterflies, it will move randomly. Thus, the characteristics of the ideal butterfly population are as follows:

- All of the butterflies can emit fragrance so that they can attract each other;
- Each butterfly moves randomly or toward the best-flavored butterfly and emits more fragrance;
- The stimulus intensity of the butterflies is influenced by the fitness function.

Butterflies are affected by the fragrance intensity during movement, and the fragrance can be formulated as follows:

$$f = cI^\theta \quad (10)$$

where f is the perceived intensity of fragrance, c is the sensory modality, and I is the stimulus intensity. θ is the fragrance factor, which reflects the absorption degree of the fragrance; when $\theta = 1$, it is an ideal environment, and the fragrance of each butterfly can be perceived anywhere in the area, and when $\theta = 0$, no butterfly can perceive the fragrance of another butterfly. According to the literature [8], this paper sets the initial values of c to 0.01 and θ to 0.1.

The movement of the butterfly can be divided into a “global search” and a “local search”. When the butterfly perceives another butterfly that has the best fragrance, the butterfly will move to the best one: this behavior is called a global search. It can be calculated as follows:

$$x_i(t+1) = x_i(t) + [g^* \times r_1^2 - x_i(t)] \times f_i \quad (11)$$

where $x_i(t)$ represents the position of butterfly i in generation t , and g^* is the optimal position in the current iteration.

When the butterfly does not perceive the fragrance of the other butterflies, it will move randomly: this behavior is called a local search, and its formula is as follows:

$$x_i(t+1) = x_i(t) + [r_1^2 \times x_j(t) - x_k(t)] \times f_i \quad (12)$$

where $x_i(t)$, $x_j(t)$, and $x_k(t)$ are the positions of butterflies i , j , and k in generation t , respectively, and r_1 in Equations (11) and (12) is a random number within $[0, 1]$.

A butterfly that seeks food and mates can exhibit behavior in the global stage or the local stage. The BOA algorithm uses the conversion probability p to complete the switch between the global search and the local search; p usually takes a value of 0.8.

The BOA algorithm pseudocode is shown in Algorithm 1.

Algorithm 1. The BOA algorithm.

```

1 Initialize the population, set butterflies  $x_i$ , initialize the sensory modality  $c$ , fragrance factor  $\theta$ ,  

switch probability  $p$ , and number of iterations  $T$ ;  

2 Define fitness function  $f(x_i)$  and calculate the value of the fitness function for each butterfly;  

3 Calculate stimulus intensity  $I$  using the fitness function;  

4 while  $t < T$   

5   for  $i = 1 : n$   

6     Calculate perceived intensity of fragrance using Equation (10);  

7     if  $r_1 < p$   

8       Update position using Equation (11);  

9     else  

10      Update position using Equation (12);  

11    end if  

12    Update the optimal position  $g^*$ ;  

13  end for  

14  Update the value of sensory modality  $c$ ;  

15 end while  

16 Output the global optimal solution.

```

3.2. Fractional-Order Calculus

Fractional-order calculus is the differentiation and integration of fraction order, and unlike integer-order calculus, its definition takes different forms according to the perspective of the problem being studied [14]. The most commonly used definitions are the Grünwald–Letnikov-type (G-L) definition, Riemann–Liouville-type (R-L) definition, and Caputo-type definition. The G-L definition is obtained by finding the limit of the difference approximation formula in integer-order calculus, where the positive and negative values of the order represent the fractional differential and the fractional integral. The Caputo-type definition can be regarded as a generalization form of the G-L-type definition, which is a form of the realization integral, and it can solve some engineering problems effectively [15].

3.2.1. G-L-Type Definition

The G-L-type definition extends the calculus from the integer order to the fractional order. According to the literature [9], it is defined as follows: function $x(t) \in [a, t]$ ($a < t, a \in R, t \in R$) has an $m + 1$ ($m \in Z$)-order continuous derivative; when $v > 0$, the v -order derivative of the function $x(t)$ is

$${}_a^G D_t^v x(t) \stackrel{\Delta}{=} \lim_{h \rightarrow 0} x_h^{(v)}(t) \stackrel{\Delta}{=} \lim_{h \rightarrow 0} h^{-v} \sum_{r=0}^n \begin{bmatrix} -v \\ r \end{bmatrix} x(t - rh) \quad (13)$$

$nh = t - a$

where v is the order ($v \in R$), $\begin{bmatrix} -v \\ r \end{bmatrix} = \frac{(-v)(-v+1)\cdots(-v+r-1)}{r!}$. To make $x_h^{(-v)}(t)$ achieve the non-zero limit, when $h \rightarrow 0$, we take $n \rightarrow \infty$. When $h = \frac{t-a}{n}$, $n = \lceil \frac{t-a}{h} \rceil$, and by using the

mathematical induction method to arrange Formula (13), followed by integration by parts, Equation (13) becomes the following:

$${}_a^G D_t^v x(t) = \sum_{k=0}^m \frac{x^{(k)}(a)(t-a)^{-v+k}}{\Gamma(-v+k+1)} + \frac{1}{\Gamma(-v+m+1)} \int_a^t (t-\tau)^{-v+m} s^{(m+1)}(\tau) d\tau \quad (14)$$

where the Gamma function $\Gamma(\alpha) = \int_0^\infty e^{-x} x^{\alpha-1} dx = (\alpha-1)!$. If the continuous interval between a and t of the function $x(t)$ is divided into equal intervals, and the equal interval $\Delta h = 1$, then $n = [\frac{t-a}{\Delta h}] \stackrel{\Delta h=1}{=} [t-a]$, and the fractional-order differential discrete expression $x(t)$ is

$$\begin{aligned} \frac{d^v x(t)}{dt^v} &\approx x(t) + (-v)x(t-1) + \frac{(-v)(-v+1)}{2}x(t-2) + \frac{(-v)(-v+1)(-v+2)}{6}x(t-3) \\ &+ \cdots + \frac{\Gamma(-v+1)}{n! \Gamma(-v+n+1)} x(t-n) \end{aligned} \quad (15)$$

3.2.2. Caputo-Type Definition

Applying the Laplace transform to the G-L-type definition is simpler. The Caputo-type definition is obtained by improving the G-L-type definition. The Caputo-type differential form of function $x(t) \in [a, t]$ is [16]

$${}_a^C D_t^v x(t) = \frac{1}{\Gamma(m-v)} \int_a^t \frac{1}{(t-\xi)^{v-m+1}} x^{(m)}(\xi) d\xi \quad (16)$$

where v is the differential order, and m is an integer within $m-1 < v < m$.

When $0 < m-1 < v < m$, $\forall T > a$, the function $x(t) \in [a, t]$ has an $m+1$ ($m \in \mathbb{Z}$)-order continuous and bounded derivative, and $\lim {}_a^C D_t^v x(t) = x^{(m)}(t)$. When $m=1$, order $v \in (0,1)$, the Caputo-type differential is defined as follows:

$${}_a^C D_t^v x(t) = \frac{1}{\Gamma(1-v)} \int_0^t \frac{1}{(t-\xi)^v} f'(\xi) d\xi \quad (17)$$

Generally, $a=0$, and the value range of the function is between 0 and t , dividing the value range into N parts, where each part is $\Delta t = \frac{t}{N}$. According to the forward differential formula of the first order, the differential expression of the Caputo-type definition with respect to the function $x(t)$, which is shown in Equation (18), is obtained when $\Delta t = 1$ and $t_k = k \frac{t}{N} = k$.

$${}_a^C D_t^v x(t) \approx \frac{1}{\Gamma(2-v)} \sum_{k=0}^{N-1} [x(t-t_{k+1}) - x(t-t_k)] \times [(k+1)^{1-v} - k^{1-v}] \stackrel{def}{=} \sum_{k=0}^{N-1} c_k x(t-t_k) \quad (18)$$

where c_k values are as follows:

$$\left\{ \begin{array}{l} c_0 = -\frac{1}{\Gamma(2-v)} \\ c_1 = \frac{2-2^{1-v}}{\Gamma(2-v)} \\ \vdots \\ c_k = \frac{2(n-k)^{1-v} - (n-k+1)^{1-v} - (n-k-1)^{1-v}}{\Gamma(2-v)} \\ \vdots \\ c_N = \frac{N^{1-v} - (N-1)^{1-v}}{\Gamma(2-v)} \end{array} \right. \quad (19)$$

3.3. Hybrid Fractional Butterfly Optimization Algorithm

In order to solve the problems of the BOA algorithm, such as low accuracy, slow convergence speed, and the fact that it easily falls into local optima, this paper mainly improves the BOA algorithm from three aspects: firstly, fractional-order differentiation is combined with the global update method of the BOA algorithm, increasing the algorithm convergence rate; secondly, the local position update method is improved by using fractional-order differentiation and the sine-cosine algorithm, increasing the accuracy of optimization; thirdly, a dynamic conversion probability is introduced to prevent the BOA algorithm from falling into local optima. The improved algorithm is called the hybrid fractional-order butterfly optimization algorithm (HFBOA).

3.3.1. Fractional-Order Differentiation Improves the Global Search

The global search of the BOA algorithm does not consider the butterfly's previous positional state. Therefore, the search result and convergence rate are largely determined by the perceived intensity of fragrance f . The BOA algorithm has poor stability and slow convergence. Fractional-order differentiation has memory and heredity; when using it to improve the global position update method of the BOA algorithm, the butterfly's position update is affected by the historical state, and it will improve the convergence speed of the algorithm.

According to Equation (11), the global position update formula of the BOA algorithm can be described as follows:

$$x_i(t+1) - x_i(t) = [g^* \times r_1^2 - x_i(t)] \times f_i \quad (20)$$

The left side of Equation (20) can be regarded as the difference form of the first order in the G-L-type definition, for example, $D^v[x(t+1)]$; when $n = 4$ in Equation (15), $D^v[x(t+1)]$ becomes the following:

$$D^v[x(t+1)] \approx x(t+1) - vx(t) + \frac{1}{2}v(v-1)x(t-1) - \frac{1}{6}v(v-1)(v-2)x(t-2) + \frac{1}{24}v(v-1)(v-2)(v-3)x(t-3) \quad (21)$$

Combining Equations (20) and (21), the global position update method of the BOA algorithm is

$$x_i(t+1) = vx_i(t) - \frac{1}{2}v(v-1)x_i(t-1) + \frac{1}{6}v(v-1)(v-2)x_i(t-2) - \frac{1}{24}v(v-1)(v-2)(v-3)x_i(t-3) + [g^* \times r_1^2 - x_i(t)] \times f_i \quad (22)$$

According to the above equation, it can be seen that the butterfly position update is affected by the fractional order. To achieve better results, evolutionary factors f_y are introduced, and the fractional order is adjusted adaptively according to the position information of the butterflies [11].

The average distance between butterfly i and other butterflies is

$$d_{ix} = \frac{1}{N-1} \sum_{j=1, j \neq i}^N \sqrt{\sum_{k=1}^D (x_{ik} - x_{jk})^2} \quad (23)$$

where N is the number of butterflies, and D is the spatial dimension.

The evolutionary factor f_y can be described as follows:

$$f_y = \frac{d_g - d_{\min}}{d_{\max} - d_{\min}} \quad (24)$$

where d_g is the average distance between the global optimal butterfly position and other butterflies, and d_{\max} and d_{\min} are the maximum and minimum values among d_{ix} . When

$v \in [0.5, 0.8]$, the algorithm convergence is faster, and it can be calculated as $v(f_y) = \frac{1}{2}e^{-0.47f_y}$.

In the BOA algorithm, the initial value of the sensory modality c and the fragrance factor θ will affect the convergence speed and optimization accuracy. They are usually set based on manual experience and may introduce some ambiguity and randomness. If the initial parameter values are not suitable, the algorithm will fall into the local optimum and have a slow convergence rate. Therefore, this paper introduces the adaptive fragrance perception intensity [17], and in the initial search stage, the fragrance perception intensity is large. With the increase in iterations t , the fragrance perception intensity gradually decreases, which will ensure high search ability in the early stage and high accuracy in the later stage. The adaptive fragrance perception intensity can be calculated as follows:

$$f_i = \mu \times (1 - \frac{t}{T}) \quad (25)$$

where μ is a random number between 0 and 1 and follows a normal distribution, which ensures a balanced distribution of the fragrance perception intensity. t is the number of current iterations, and T is the maximum number of iterations.

3.3.2. Fractional-Order Sine-Cosine Algorithm Improves Local Search

In the local search stage of the BOA algorithm, butterflies have a small movement range and lack the ability to communicate information with each other. So, it is easy to fall into the local optimum, which leads to low optimization accuracy. While the sine-cosine algorithm can achieve a global search due to the oscillatory nature of the cosine function [18], this paper introduces the sine-cosine algorithm into the local search stage, meaning that every butterfly can play a full role in optimization and enhance the local optimization ability. The sine-cosine algorithm can be formulated as follows:

$$x_i(t+1) = \begin{cases} x_i(t) + r_2 \times \sin(r_3) \times |r_4 g^* - x_i(t)| & r_5 < 0.5 \\ x_i(t) + r_2 \times \cos(r_3) \times |r_4 g^* - x_i(t)| & r_5 \geq 0.5 \end{cases} \quad (26)$$

where $r_2 = 2 - 2t/T$, and r_3, r_4, r_5 are random numbers within $[0, 2\pi]$.

However, the sine-cosine algorithm still has slow convergence speed and low optimization accuracy when it deals with complex problems, so fractional-order differentiation is used to improve the sine-cosine algorithm. The Caputo-type definition can be regarded as an improvement of the G-L-type definition, and it has memory effects and nonlinear properties. Combining the Caputo-type fractional-order differentiation with the sine-cosine algorithm to achieve global optimization will improve the algorithm's optimization accuracy and increase its convergence speed. When $N = 3$, according to Equation (18), the difference form of Caputo-type fractional-order differentiation with respect to the function $x(t)$ is shown as follows. The difference form of $D^v[x(t+1)]$ is shown in Equation (28):

$$D^v[x(t)] = -c_0 x(t) + c_1 x(t-1) + c_2 x(t-2) \quad (27)$$

$$D^v[x(t+1)] = -c_0 x(t+1) + c_1 x(t) + c_2 x(t-1) \quad (28)$$

According to Equations (26) and (28), the position update formula combining fractional-order differentiation and the sine-cosine algorithm is obtained, and it is used as the local position update method of the BOA algorithm.

$$x_i(t+1) = \begin{cases} \frac{c_1 x_i(t) + c_2 x_i(t-1) - \sin(r_3) \times (r_4 g^* - x_i(t))}{c_0} & r_5 < 0.5 \\ \frac{c_1 x_i(t) + c_2 x_i(t-1) - \cos(r_3) \times (r_4 g^* - x_i(t))}{c_0} & r_5 \geq 0.5 \end{cases} \quad (29)$$

where $c_0 = -1/\Gamma(2-v)$, $c_1 = (2 - 2^{1-v})/\Gamma(2-v)$, and $c_2 = (2 \times 2^{1-v} - 3^{1-v} - 1)/\Gamma(2-v)$.

3.3.3. Dynamic Conversion Probability

In the BOA algorithm, the conversion probability p is a constant. The global search or the local search is chosen depending on the size relationship of p and r_1 , and the search process is random and easily falls into local optima. To achieve better optimization accuracy, a dynamic conversion probability p [19] that balances the ratio of the global stage to the local stage is introduced. So, in the early stage, the range of the global optimal butterfly is determined quickly, and with the increase in iterations t , the local stage capability is enhanced for high-precision optimization.

$$p_t = 0.8 - 0.3 \sin\left(\left(\frac{t}{T}\right)^2 \frac{\pi}{2}\right) \quad (30)$$

3.4. Improved Algorithm Evaluation of Benchmark Functions

To verify the improvement effects of the BOA algorithm, 12 benchmark functions were selected to test its optimization performance. The detailed descriptions of all benchmark functions are listed in Table 1. $F_1 \sim F_4$ are single-mode benchmark functions that have only one global optimal solution, and they can test the convergence rate of the algorithm. $F_5 \sim F_{12}$ are multi-mode benchmark functions that have some local optimal solutions, and they test the global optimization ability and optimization accuracy of the algorithm. In the experiment, the HFBOA algorithm was compared with four other algorithms, namely, the BOA algorithm, the Fractional Firefly Algorithm (FFA), the Fractional Particle Swarm Optimization (FPSO) algorithm, and the Particle Swarm Optimization (PSO) algorithm. For all algorithms, the population size is 30, and the total number of iterations is 500.

Table 1. Basic information of benchmark functions.

Function Expression	Dimension	Range	f_{\min}
$F_1 = \sum_{i=1}^n x_i^2$	30	[-100, 100]	0
$F_2 = \sum_{i=1}^n x_i + \prod_{i=1}^n x_i $	30	[-10, 10]	0
$F_3 = \sum_{i=1}^n \left(\sum_{j=1}^i x_j \right)^2$	30	[-100, 100]	0
$F_4 = \max\{ x_i , 1 \leq i \leq n\}$	30	[-100, 100]	0
$F_5 = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10]$	30	[-5.15, 5.12]	0
$F_6 = -20 \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right) - \exp\left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)\right) + 20 + e$	30	[-32, 32]	0
$F_7 = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	30	[-600, 600]	0
$F_8 = \sum_{i=1}^n x_i \sin(x_i) + 0.1 x_i $	30	[-10, 10]	0
$F_9 = \left(\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^2 (x_i - a_{ij})^6} \right)^{-1}$	2	[-65, 65]	1
$F_{10} = \sum_{i=1}^{11} \left[a_i - \frac{x_1(b_i^2 + b_1 x_2)}{b_i^2 + b_1 x_3 + x_4} \right]^2$	2	[-5, 5]	3.075×10^{-4}
$F_{11} = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1 x_2 - 4x_2^2 + 4x_3^4$	2	[-5, 5]	-1.0316
$F_{12} = [1 + (x_1 + x_2 + 1)^2 (19 - 14x_1 + x_1^2 - 14x_2 + 6x_1 x_2 + 3x_2^2)] \times [30 + (2x_1 - 3x_2)^2 \times (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1 x_2 + 27x_2^2)]$	2	[-2, 2]	3

To reduce errors during the experiment, each benchmark function was run 30 times independently. The optimal fitness value (BEST), the average fitness value (MEAN), and the standard deviation (STD) are shown in Table 2. In Table 2, the $F_1 \sim F_4$ results are shown in the first column to the fourth column. The BEST, the MEAN, and the STD of the HFBOA

algorithm are all the theoretical optima (0), which shows that the HFBOA algorithm has good stability and high optimization accuracy. The BEST value of the FPSO algorithm is the theoretical optimum of 0, but the MEAN and the STD are not 0, which means that the optimization results are unstable. Other algorithms do not obtain the theoretical optima (0). According to Figure 2, the convergence speed of the HFBOA algorithm is better than that of other algorithms, so the HFBOA algorithm has some advantages in the optimization problem of the single-mode function.

Table 2. Comparison of the benchmark function results.

Function	Index	BOA	FFA	FPSO	PSO	HFBOA
F_1	BEST	1.0888×10^{-11}	5.2955×10^{-2}	0	0.9744	0
	MEAN	1.2853×10^{-11}	6.8202×10^{-2}	18.0763	1.7449	0
	STD	7.9268×10^{-13}	6.2858×10^{-2}	18.5856	0.5707	0
F_2	BEST	1.3086×10^{-9}	10.9756	0	3.6652	0
	MEAN	4.4458×10^{-9}	1.1082	19.3553	6.8920	0
	STD	1.4377×10^{-9}	5.6678×10^{-2}	12.9518	1.2864	0
F_3	BEST	1.0335×10^{-11}	5.8945×10^{-2}	0	3.9691	0
	MEAN	1.2540×10^{-11}	8.2000×10^{-2}	50.7183	11.1283	0
	STD	9.3301×10^{-13}	1.0152×10^{-2}	61.1040	4.0186	0
F_4	BEST	5.4100×10^{-9}	8.1970×10^{-2}	0	0.4859	0
	MEAN	6.1565×10^{-9}	9.3128×10^{-2}	1.5025	0.7659	0
	STD	4.6739×10^{-10}	4.8686×10^{-3}	0.9839	0.1510	0
F_5	BEST	2.3357×10^{-10}	4.6991×10^{-2}	0	1.1007	0
	MEAN	8.2502×10^{-10}	6.3942×10^{-2}	10.1079	3.3182	0
	STD	7.7165×10^{-10}	5.9537×10^{-3}	6.6926	1.6393	0
F_6	BEST	0	8.9905	0	53.2521	0
	MEAN	13.5721	12.8963	34.1000	85.0095	0
	STD	51.6486	1.4408	52.6022	18.5541	0
F_7	BEST	4.9271×10^{-9}	0.2543	8.8818×10^{-16}	1.9942	8.8818×10^{-16}
	MEAN	6.0044×10^{-9}	0.3001	3.7122	2.5389	8.8818×10^{-16}
	STD	3.8142×10^{-10}	0.0191	1.9064	0.2774	0
F_8	BEST	1.1319×10^{-12}	2.0533×10^{-3}	2.5146×10^{-2}	4.7492×10^{-2}	0
	MEAN	4.3214×10^{-12}	2.9130×10^{-3}	0.6818	8.4649×10^{-2}	0
	STD	2.2965×10^{-12}	3.0735×10^{-4}	0.2817	3.0695×10^{-2}	0
F_9	BEST	0.9980	2.2026	12.6705	12.6705	0.9980
	MEAN	1.4449	11.2555	12.6735	12.6705	2.9673
	STD	0.6096	3.2662	8.2600×10^{-2}	1.4254×10^{-13}	3.2104
F_{10}	BEST	3.2212×10^{-4}	3.1048×10^{-4}	8.6673×10^{-4}	3.0749×10^{-4}	3.1388×10^{-4}
	MEAN	4.3588×10^{-4}	3.2493×10^{-3}	3.7876×10^{-2}	1.1178×10^{-3}	5.6483×10^{-4}
	STD	1.4855×10^{-4}	7.2934×10^{-6}	4.5871×10^{-2}	3.6490×10^{-3}	1.4503×10^{-3}
F_{11}	BEST	-1.0415	-0.9507	-1.0194	-1.0316	-1.0316
	MEAN	-1.0864	-0.8073	-0.4961	-1.0316	-1.0309
	STD	7.6611×10^{-2}	4.1317×10^{-2}	0.3563	6.4539×10^{-16}	3.8581×10^{-4}
F_{12}	BEST	3.0021	4.6238	3	3	3.0004
	MEAN	3.1861	59.0129	3	3.0027	3.043
	STD	0.3929	32.7770	1.5139×10^{-15}	4.1661×10^{-3}	0.11328

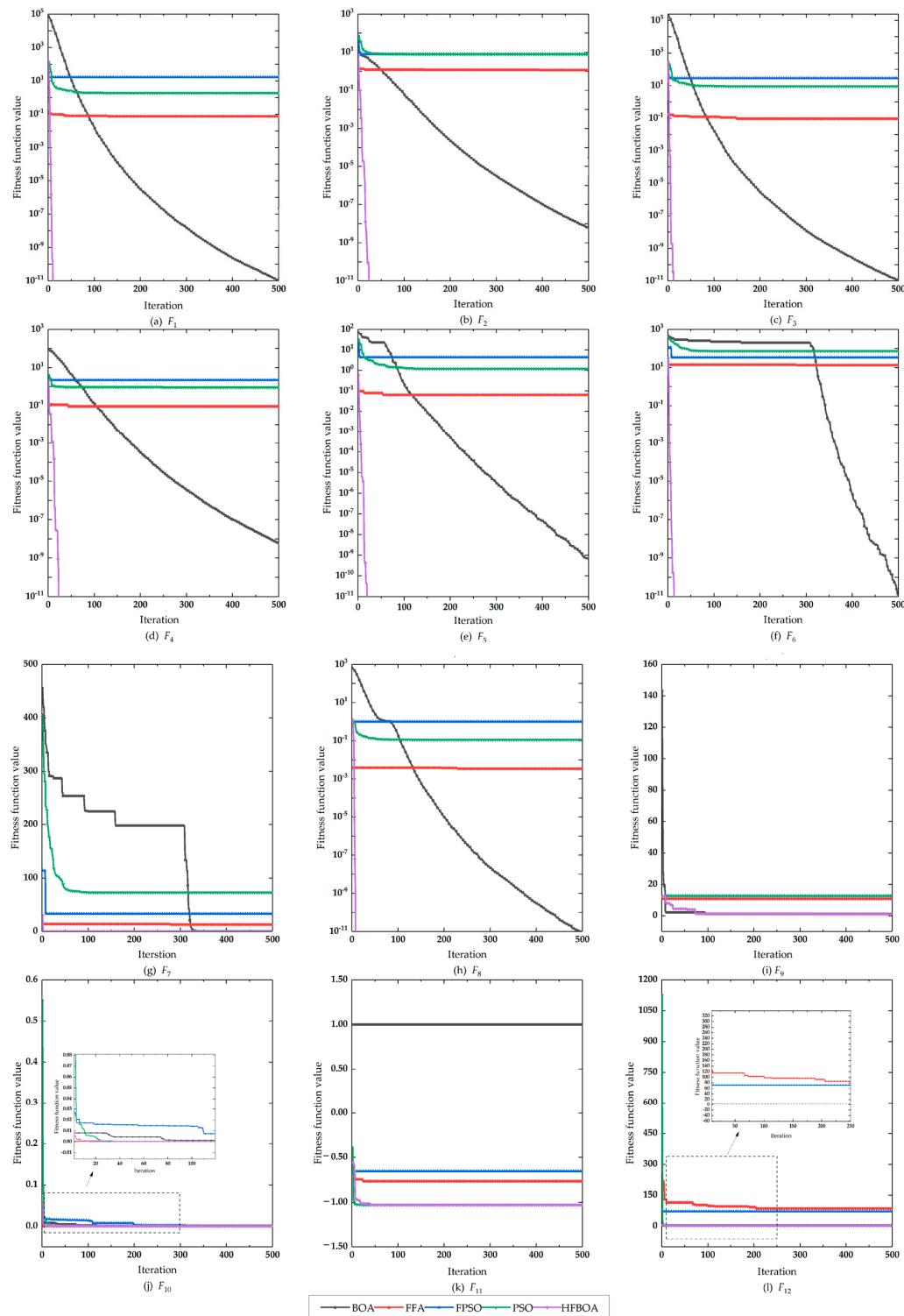


Figure 2. Convergence curves of 5 algorithms on 12 benchmark functions.

For the multi-mode functions F₅~F₈, the optimization ability of the HFBOA algorithm is better than that of the other algorithms. The BEST value, the MEAN value, and the STD value of functions F₅, F₆, and F₈ achieve the theoretical optimum of 0. Although F₇ does not reach the theoretical optimal value, the optimization accuracy is still higher than that of the other algorithms. The BOA algorithm has a probability of finding the theoretical optimal value of F₆, but the MEAN value and the STD value are not 0, indicating that the BOA algorithm will easily fall into local optima. For functions F₅, F₇, and F₈, the

BOA algorithm does not reach the theoretical optimum, which shows that the HFBOA algorithm effectively improves the optimization accuracy of the BOA algorithm and avoids falling into local optima. The FPSO algorithm finds the theoretical optimal value for F_5 and F_6 , indicating that it can solve some optimal problems well. For the fixed-dimensional multi-mode functions $F_9 \sim F_{12}$, five algorithms similarly achieve the theoretical optima, but when combining the MEAN value, the STD value, and the convergence rate, the HFBOA algorithm is still better than the other algorithms.

4. Hybrid Fractional Butterfly Optimization Algorithm for Otsu Image Segmentation

Because the BOA algorithm for 2-D Otsu image segmentation has slow convergence speed and low segmentation accuracy, this paper introduces the HFBOA algorithm into 2-D Otsu image segmentation and proposes the Hybrid Fractional Butterfly Optimization Algorithm—Otsu (HFBOA-Otsu). The specific steps are as follows:

- Step 1 Enter the image to be segmented, calculate the 2-D gray-gradient distribution histogram of the image, and choose the trace of the dispersion matrix as the fitness function.
- Step 2 Initialize the parameters and the butterfly population. Set the population size to 80 and the total number of iterations to 100.
- Step 3 Calculate the fitness function value of each butterfly according to Formula (9) and select the butterfly with the largest fitness value as the current global optimal value.
- Step 4 Update the fragrance perception intensity f_t and dynamic conversion probability p_t according to Equations (25) and (30).
- Step 5 When $rand < p_t$, choose the global search and update the butterfly position according to Equation (22). When $rand \geq p_t$, choose the local search and update the butterfly location according to Equation (29).
- Step 6 Determine whether the total number of iterations T is reached. If reached, output the global optimum; otherwise, return to Step 4.
- Step 7 Output the butterfly position that will maximize the fitness function value, use it (s^*, t^*) as the threshold to perform threshold segmentation on the image, and output the segmented image.

The flow chart of the HFBOA-Otsu algorithm is shown in Figure 3.

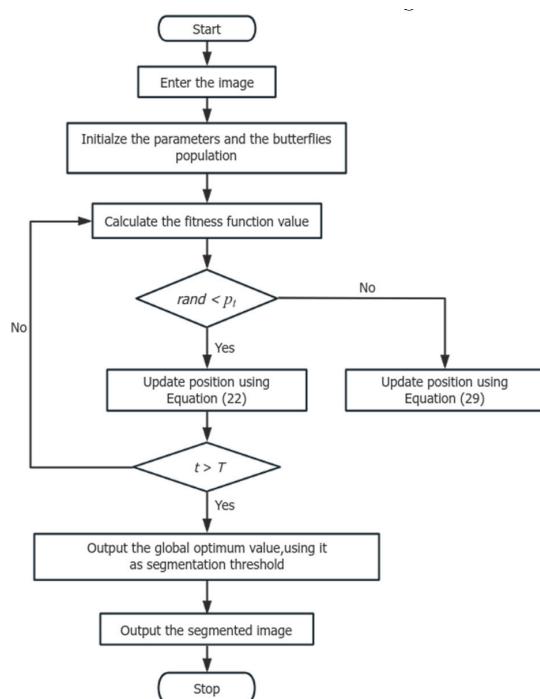


Figure 3. The flow chart of HFBOA-Otsu.

5. Experimental Results and Analysis

To verify the convergence speed and the segmentation accuracy of the HFBOA-Otsu algorithm, different types of images were chosen to perform threshold segmentation. The HFBOA-Otsu algorithm was compared with the 2-D Otsu image threshold segmentation based on the BOA algorithm (BOA-Otsu), the 2-D Otsu image threshold segmentation based on the PSO algorithm (PSO-Otsu), the 2-D Otsu image threshold segmentation based on Fractional PSO (Im-FpsOtsu) [11], and the 2-D Otsu image threshold segmentation based on the Fractional Firefly Algorithm (FFA-Otsu) [20]. The experimental hardware environment was Intel(R) Core(TM) i5-8265U CPU 1.80 GHz, with 8 GB of memory, and the software environment was MATLAB R2017a.

Three types of images were selected for the segmentation experiment: human images, scenery images, and medical images. The image segmentation effects were objectively evaluated on the basis of five indexes: the fitness function value, iterations, Peak-Signal-to-Noise Ratio (PSNR), mean square error (MSE), and Structural Similarity Index (SSIM). The larger the fitness function value, the greater the gap between the background class and the target class, and the better the segmentation effects. The number of iterations reflects the algorithm convergence rate, and the smaller the value, the faster the algorithm convergence rate. The PSNR value reflects the noise performance of the algorithm, and the greater the value, the higher the quality of the segmented image. The MSE value represents the mean square errors between the original image and the segmented image, and the smaller the value, the higher the segmentation accuracy. The SSIM value measures the similarity between the original image and the segmented image, and the larger the value, the lower the image distortion.

There are four human images: the Lena image, the Pirate image, the Woman-blonde image, and the Kodim image. The segmentation results of human images are shown in Figure 4. The first column of Figure 4 is the original images, and the second column to the sixth column are the segmentation results of the five image segmentation algorithms. From the subjective visual perspective, in the Lena image segmentation results, the HFBOA-Otsu algorithm generates more detailed segmentation results in the hat and window areas. In the Pirate image, the HFBOA-Otsu algorithm retains more details than other algorithms. In the Woman-blonde image segmentation results, the HFBOA-Otsu algorithm segments a clear and complete hand. For the Kodim image, the segmentation results of the HFBOA-Otsu algorithm largely retain the texture at the collar. Therefore, the HFBOA-Otsu algorithm has greater advantages in human image segmentation. The differences in image segmentation results between the five algorithms are highlighted by the red squares.

The objective evaluation indexes of the human segmentation images are shown in Table 3, and Figure 5 shows the fitness curves of the human images. For the Lena image, combining Table 3 and the fitness curve indicates that the HFBOA-Otsu algorithm finds the best segmentation threshold in 10 iterations, while the BOA-Otsu algorithm reaches convergence at 30 iterations, with the FFA-Otsu algorithm, the Im-FpsOtsu algorithm, the PSO-Otsu algorithm converging at 19, 48, and 88, respectively. This shows that the convergence rate of the HFBOA-Otsu algorithm is faster. Moreover, the fitness function value and the SSIM value of the HFBOA-Otsu algorithm are higher than those of the other algorithms, indicating that the HFBOA-Otsu algorithm not only improves the convergence speed but also has a better segmentation effect. The PSNR value of the HFBOA-Otsu algorithm is also higher than those of the other algorithms, and the MSE value is smaller, so the HFBOA-Otsu algorithm can guarantee a high segmentation accuracy.



Figure 4. Human image segmentation results of five algorithms. The first row is the Lena image, the second row is the Pirate image, the third row is the Woman-blonde image, and the fourth row is the Kodim image. The first column is the original images, and the second column to the sixth column are the segmentation results of the BOA-Otsu algorithm, the FFA-Otsu algorithm, the Im-FpsO Otsu algorithm, the PSO-Otsu algorithm, and the HFBOA-Otsu algorithm, respectively.

Table 3. Objective evaluation indexes of human image segmentation results.

Image	Segmentation Algorithm	Iterations	Fitness Value	PSNR	MSE	SSIM
Lena	BOA-Otsu	30	2990.2579	8.8206	8470.0803	0.3489
	FFA-Otsu	19	2982.9960	8.8244	8401.1874	0.3502
	Im-FpsO Otsu	48	2994.2793	8.6099	8667.3892	0.2176
	PSO-Otsu	88	2994.2378	8.8816	8412.4531	0.1316
	HFBOA-Otsu	10	2994.6627	8.8952	8386.1521	0.3676
Pirate	BOA-Otsu	33	3286.4006	8.8189	8534.6528	0.2631
	FFA-Otsu	18	3282.9542	8.8855	8448.7213	0.2809
	Im-FpsO Otsu	27	3286.6442	8.7928	8632.6822	0.2629
	PSO-Otsu	78	3286.8501	8.3563	9493.9568	0.1106
	HFBOA-Otsu	5	3287.4031	8.8857	8404.2361	0.2811
Woman-blonde	BOA-Otsu	28	2520.2425	8.8446	8484.2954	0.3604
	FFA-Otsu	15	2492.5715	8.8632	8448.1738	0.3561
	Im-FpsO Otsu	12	2520.5525	8.8616	8451.3193	0.3544
	PSO-Otsu	29	2519.9378	8.8607	8453.0329	0.3550
	HFBOA-Otsu	3	2521.4283	8.8767	8421.8425	0.3649
Kodim	BOA-Otsu	22	1626.2219	8.3051	9430.6818	0.1962
	FFA-Otsu	13	1624.8449	8.3093	9597.3716	0.1828
	Im-FpsO Otsu	15	1627.0425	8.3586	9488.9288	0.1986
	PSO-Otsu	29	1626.5371	8.3779	9446.9456	0.1968
	HFBOA-Otsu	4	1627.1348	8.3868	9427.6422	0.1993

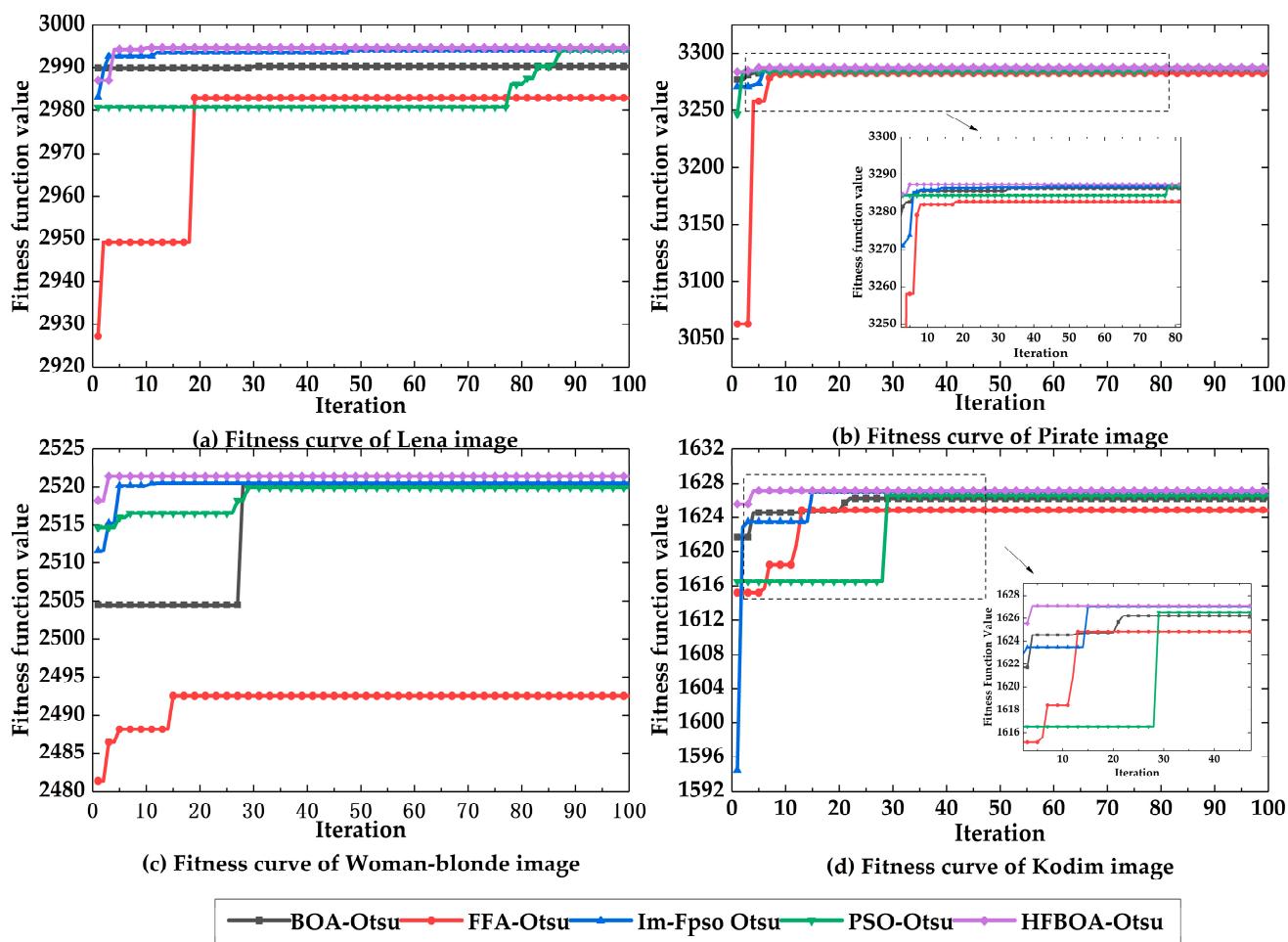


Figure 5. The fitness curves of human images. (a) The fitness curve of the Lena image; (b) the fitness curve of the Pirate image; (c) the fitness curve of the Woman-blonde image; (d) the fitness curve of the Kodim image.

For the Pirate image, the number of iterations with the HFBOA-Otsu is five. As can be seen in Figure 5b, the HFBOA-Otsu algorithm reaches convergence earlier, and it has a higher fitness function value and better segmentation effects. The PSNR value and the SSIM value of the HFBOA-Otsu algorithm are higher than those of other algorithms, and the MSE value is smaller, which shows that the segmentation accuracy and anti-noise performance of the HFBOA-Otsu algorithm are better than those of the other algorithms.

For the Woman-blonde image, the convergence of the HFBOA-Otsu algorithm, compared with the BOA-Otsu algorithm, the FFA-Otsu algorithm, the Im-FpsO Otsu algorithm, and the PSO-Otsu algorithm, is accelerated by 89.28%, 80%, 75%, and 89.65%, respectively. The fitness function value, the PSNR value, and the SSIM value are higher, indicating that the segmentation effects and anti-noise performance of the HFBOA-Otsu algorithm are improved. For the Kodim image, the convergence rate of the HFBOA-Otsu algorithm is faster, and the objective evaluation indexes are better than those of the other algorithms, which shows the HFBOA-Otsu algorithm has higher segmentation accuracy and better effects.

The scenery images include the Wall image, the Gorge image, the Butterfly image, and the Mandril image. The original images and the segmented images are shown in Figure 6. And the first column is the original images, the second column is the images segmented by the BOA-Otsu algorithm, and the third column to sixth column are the images segmented by the FFA-Otsu algorithm, the Im-FpsO Otsu algorithm, the PSO-Otsu algorithm, and the HFBOA-Otsu algorithm, respectively. By comparison, for the segmented image of the Wall, the HFBOA-Otsu algorithm provides more details of the bricks. In the Gorge image, the

HFBOA-Otsu algorithm can segment a clear line for the distant snow mountains, and the details of the segmentation effects are better than the other algorithms. In the segmentation of the Butterfly image, the HFBOA-Otsu algorithm can completely segment the petals at the edge of the image. For the Mandril image, the HFBOA-Otsu algorithm attains more details of the whiskers and hairs. Compared to the other algorithms, the HFBOA-Otsu algorithm can segment a clear scenery image and retain more details.

Table 4 show the objective evaluation indexes of the scenery segmentation images, and Figure 7 shows the fitness curves of the scenery images.

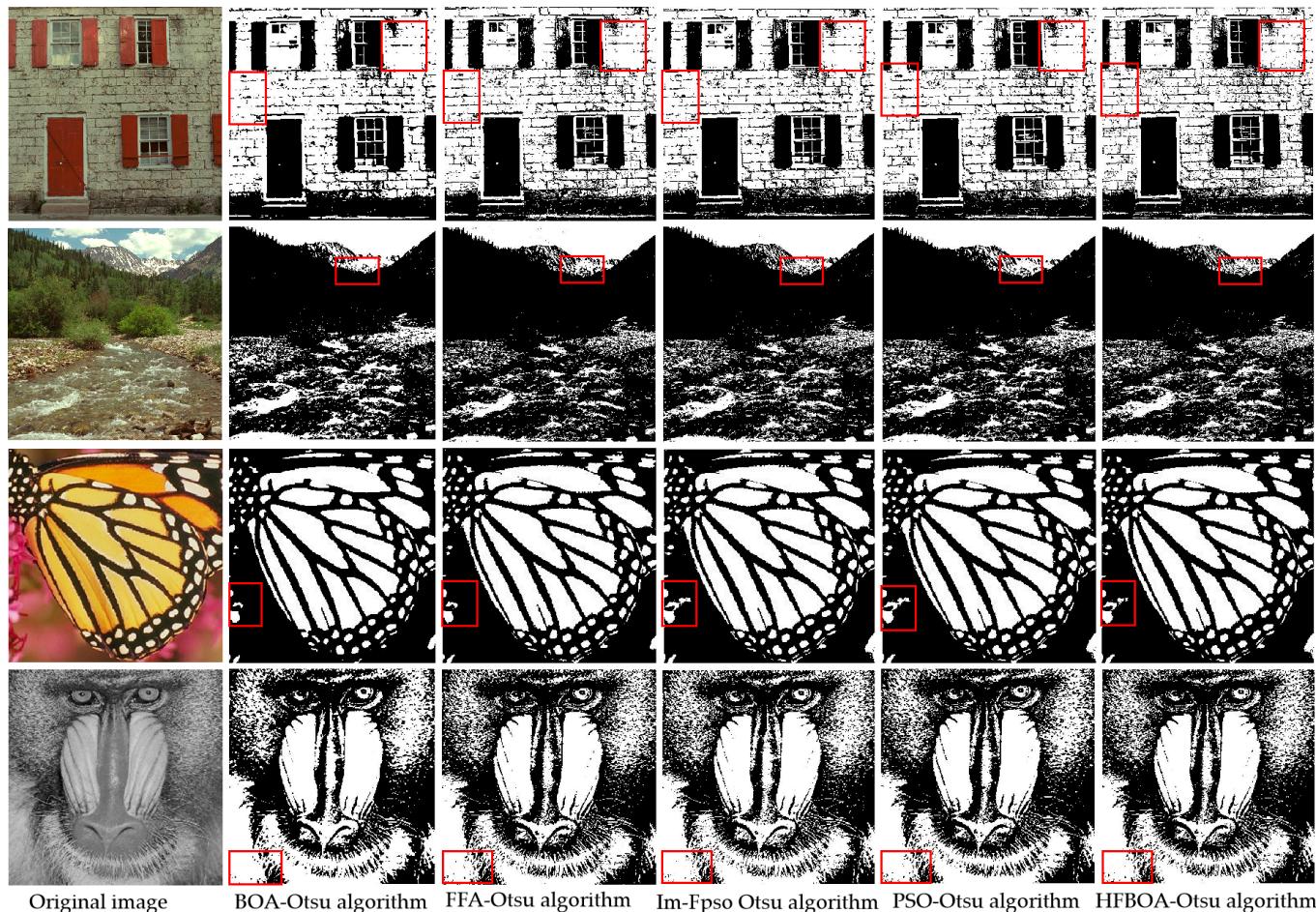


Figure 6. Scenery images and segmentation results of five algorithms. The first row is the Wall image, the second row is the Gorge image, the third row is the Butterfly image, and the fourth row is the Mandril image. The first column is the original images, and the second column to the sixth column are the segmentation results of the BOA-Otsu algorithm, the FFA-Otsu algorithm, the Im-FpsO Otsu algorithm, the PSO-Otsu algorithm, and the HFBOA-Otsu algorithm, respectively.

According to Table 4 and the fitness curve of the Wall image, the convergence rate of the HFBOA-Otsu algorithm is faster, and the fitness function value is slightly higher than that of the Im-FpsO Otsu algorithm; the fitness function value is about 12 times higher than the BOA-Otsu algorithm, and the HFBOA-Otsu algorithm greatly improves the segmentation effects of the Wall image. The PSNR value and the SSIM value are higher than those of the other algorithms, and the average PSNR value has risen by 0.1604, while the SSIM value has risen by 0.0276. Moreover, the MSE value is lower than those of the other algorithms, indicating that the segmentation accuracy and anti-noise resistance performance of the HFBOA-Otsu algorithm are improved.

Table 4. Objective evaluation indexes of scenery image segmentation results.

Image	Segmentation Algorithm	Iterations	Fitness Value	PSNR	MSE	SSIM
Wall	BOA-Otsu	55	2004.3901	8.1399	9979.0455	0.2183
	FFA-Otsu	6	2013.9126	8.3209	9571.8541	0.2639
	Im-FpsOtsu	24	2016.4836	8.3222	9568.5428	0.2528
	PSO-Otsu	74	2016.1782	8.3431	9522.9874	0.2103
	HFBOA-Otsu	5	2016.5050	8.4214	9344.1992	0.2726
Gorge	BOA-Otsu	18	3854.2277	9.4975	7300.0599	0.1940
	FFA-Otsu	30	3851.2286	9.4697	7513.3970	0.1938
	Im-FpsOtsu	42	3868.6217	9.5107	7277.9782	0.2088
	PSO-Otsu	67	3867.9821	9.4874	7317.1937	0.1511
	HFBOA-Otsu	14	3868.8271	9.5209	7259.6398	0.2103
Butterfly	BOA-Otsu	35	5936.3788	10.8547	5340.8421	0.4591
	FFA-Otsu	18	5947.0194	10.8538	5341.9318	0.4579
	Im-FpsOtsu	58	5949.0871	10.8538	5341.3258	0.4579
	PSO-Otsu	92	5948.8356	10.8532	5342.6722	0.3361
	HFBOA-Otsu	35	5949.2934	10.8575	5337.3479	0.4594
Mandril	BOA-Otsu	18	1930.0532	8.4976	9190.1544	0.2482
	FFA-Otsu	14	1937.3952	8.5931	8996.6544	0.2664
	Im-FpsOtsu	45	1941.4290	8.5900	8996.6544	0.2664
	PSO-Otsu	46	1941.3159	8.5813	9014.5957	0.1704
	HFBOA-Otsu	13	1941.6848	8.5945	8987.7847	0.2691

For the Gorge image, the HFBOA-Otsu algorithm reaches convergence after 14 iterations, and the other four algorithms reach convergence after 18 iterations, 30 iterations, 42 iterations, and 67 iterations. The fitness curve of the Gorge image shows that the HFBOA-Otsu algorithm has a higher fitness function value than the other algorithms. Compared with the other algorithms, the PSNR value and the SSIM value of the HFBOA-Otsu algorithm are higher, and the MSE value is smaller. Therefore, the segmentation accuracy and anti-noise performance of the HFBOA-Otsu algorithm are better than those of the other algorithms.

For the segmentation results of the Butterfly image, the PSNR value of the HFBOA-Otsu algorithm is higher, and the MSE value is lower compared with the other algorithms; the SSIM value and the fitness function value are also higher, so the segmentation accuracy of the HFBOA-Otsu algorithm is higher and the segmentation effect is better. According to the fitness curve of the Butterfly image, the convergence of the HFBOA-Otsu algorithm is faster. For the Mandril image, the HFBOA-Otsu algorithm reaches convergence at 13 iterations, which is slightly smaller than that obtained with the FFA-Otsu algorithm, but its fitness function value is higher than that of the FFA-Otsu algorithm, about 4.2896, which shows that the HFBOA-Otsu algorithm ensures good image segmentation effects and has a faster convergence rate. The PSNR value, MSE value, and SSIM value of the HFBOA-Otsu algorithm are better than those of the other algorithms; therefore, the segmentation accuracy of the HFBOA-Otsu algorithm is higher and better.

There are four medical images: the Lung1 image, the Lung2 image, the Thorax image, and the Brain image. Figure 8 shows the original images and the medical images' segmentation results. From the first row, it can be seen that the HFBOA-Otsu algorithm has better segmentation effects and clearly segments the lung tissue. In the segmentation results of the Lung2 image, the BOA-Otsu algorithm cannot fully segment the lung tissue. The FFA-Otsu algorithm and the Im-FpsOtsu algorithm are able to segment the lung tissue completely; however, there are some shortcomings in the details, whereas the segmentation results of the HFBOA-Otsu algorithm are clearer and have more details. For the Thorax image, the HFBOA-Otsu algorithm enabled the precise segmentation of organ tissue, and the lesion tissue is clear and complete. For the segmentation of the Brain image, the HFBOA-

Otsu algorithm ensures the organ tissues are clearer and more complete. Therefore, the HFBOA-Otsu algorithm provides better segmentation results of medical images.

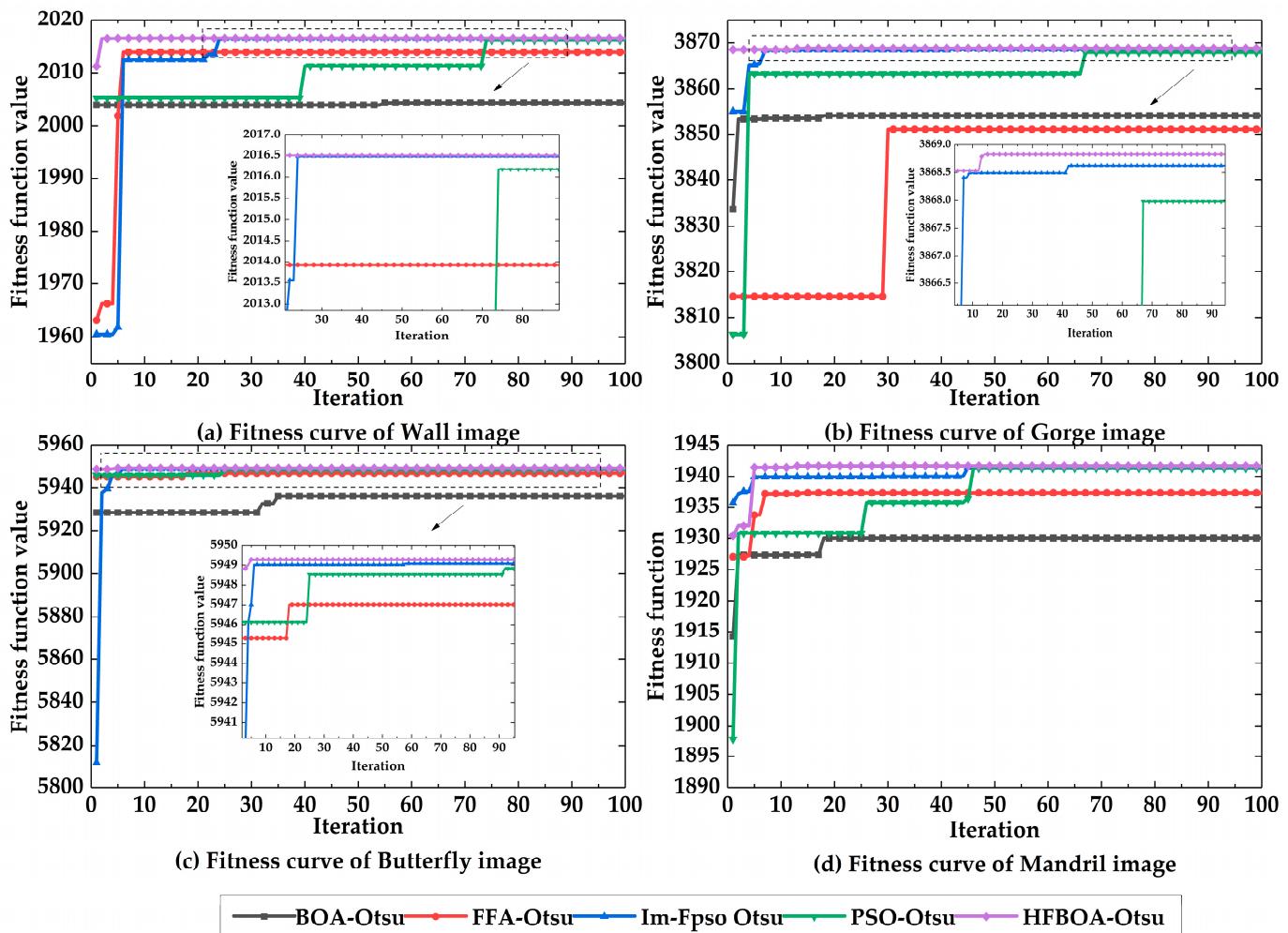


Figure 7. The fitness curves of scenery images. (a) The fitness curve of the Wall image; (b) the fitness curve of the Gorge image; (c) the fitness curve of the Butterfly image; (d) the fitness curve of the Mandril image.

The objective evaluation indexes of the medical segmentation images are shown in Table 5, which clearly shows the results of the medical segmentation images using the five algorithms, and it also shows the performance of each image threshold segmentation algorithm. The fitness curves of the medical images are shown in Figure 9, comparing the convergence speed of the five algorithms.

As shown in Table 5, for the Lung1 image, the number of iterations with the HFBOA-Otsu algorithm is four: compared with the BOA-Otsu algorithm, it is 92.72% faster. The FFA-Otsu algorithm, the Im-FpsO Otsu algorithm, and the PSO-Otsu algorithm find the optimal value after 13 iterations, 12 iterations, and 73 iterations, respectively. Therefore, the convergence rate of the HFBOA-Otsu algorithm is faster. The fitness function value, the SSIM value, the PSNR value, and the MSE value are better than those of the other algorithms. From the fitness curve of the Lung2 image, the HFBOA-Otsu algorithm has faster convergence, a higher fitness function value, and better segmentation effects. The MSE value and the SSIM value of the HFBOA-Otsu algorithm are better, indicating that the segmentation accuracy of the HFBOA-Otsu algorithm is higher. The PSNR value is also higher, and the segmentation results of the HFBOA-Otsu algorithm are least affected by noise.

According to the fitness curve of the Thorax image, the HFBOA-Otsu algorithm finds the optimal threshold after nine iterations. The other algorithms converged after 58 iterations, 14 iterations, 70 iterations, and 96 iterations, respectively. The fitness function value of the HFBOA-Otsu algorithm is larger, and the convergence speed and segmentation effects of the algorithm are improved. The PSNR value and SSIM value of the HFBOA-Otsu algorithm are larger than those of the other algorithms, and the MSE value is smaller, which means that the anti-noise performance of the HFBOA-Otsu algorithm is guaranteed, and the segmentation accuracy is improved. From the objective evaluation indexes of the Brain image segmentation results in Table 5, it can be seen that the indexes of the HFBOA-Otsu algorithm are better than those of the other algorithms, indicating that the segmentation accuracy and anti-noise performance of the HFBOA-Otsu algorithm are improved.

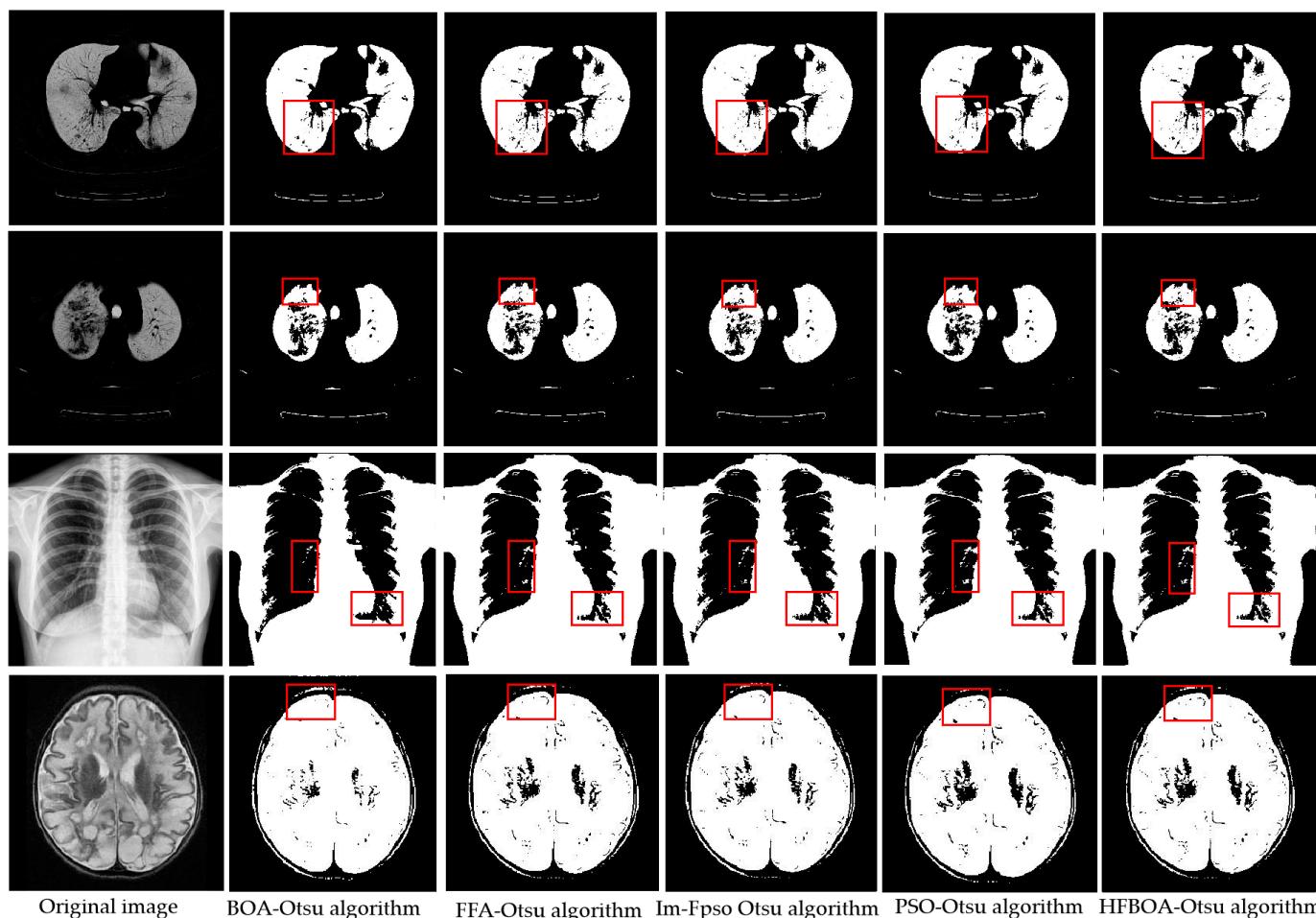
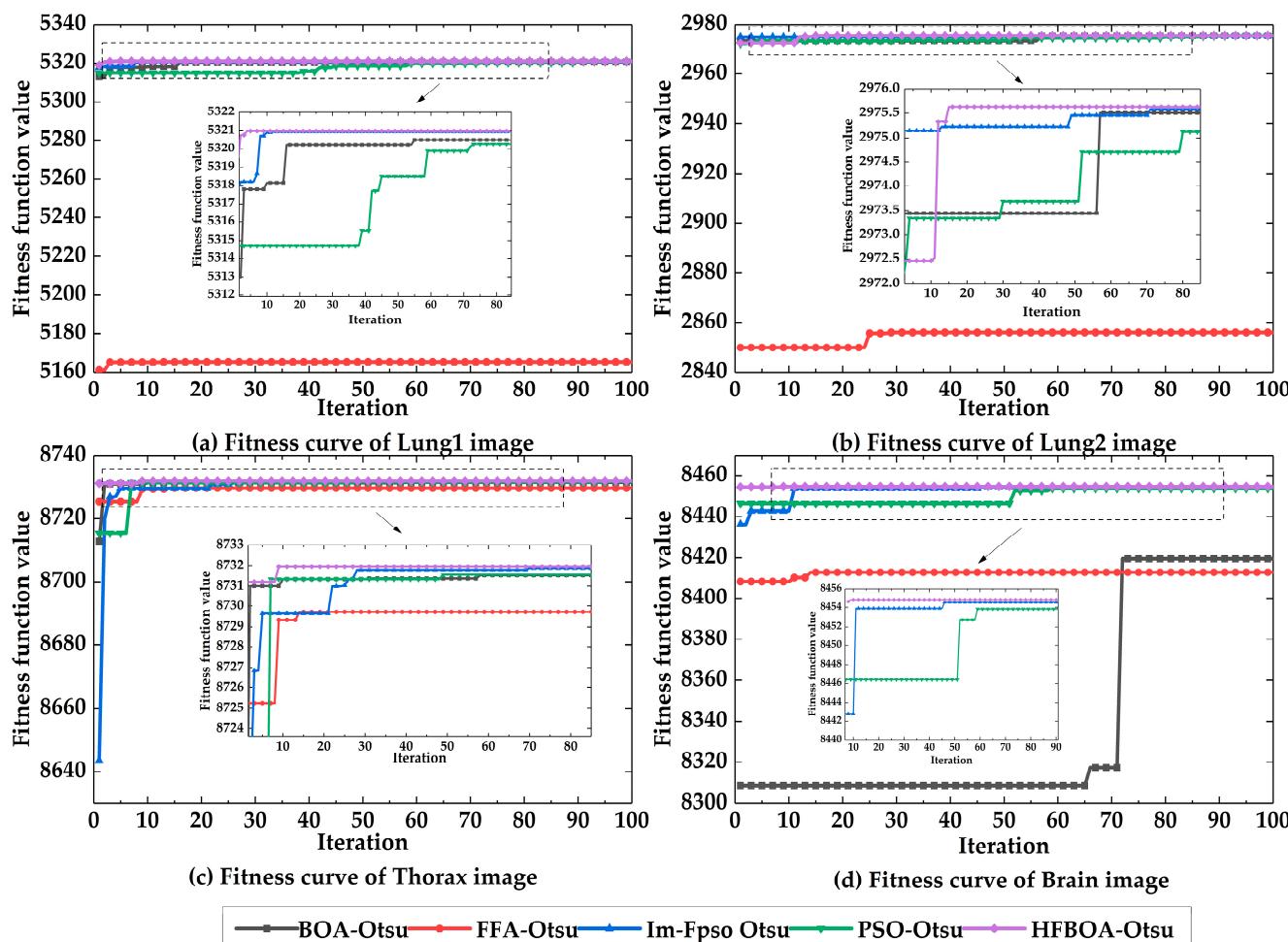


Figure 8. Medical images and segmentation results of five algorithms. The first row is the Lung1 image, the second row is the Lung2 image, the third row is the Thorax image, and the fourth row is the Brain image. The first column is the original images, and the second column to the sixth column are the segmentation results of the BOA-Otsu algorithm, the FFA-Otsu algorithm, the Im-FpsO Otsu algorithm, the PSO-Otsu algorithm, and the HFBOA-Otsu algorithm, respectively.

Table 5. Objective evaluation indexes of medical image segmentation results.

Image	Segmentation Algorithm	Iterations	Fitness Value	PSNR	MSE	SSIM
Lung1	BOA-Otsu	30	5320.5188	13.3348	3017.1746	0.6448
	FFA-Otsu	13	5165.1512	13.1634	3138.6246	0.6389
	Im-FpsoOtsu	12	5320.9561	13.0217	3242.7362	0.6433
	PSO-Otsu	73	5320.2684	13.3231	3025.3380	0.3078
	HFBOA-Otsu	4	5320.9911	13.3600	2999.7106	0.6507
Lung2	BOA-Otsu	57	2975.5037	15.0425	2036.2553	0.7309
	FFA-Otsu	29	2855.9622	15.0298	2042.1971	0.7324
	Im-FpsoOtsu	71	2975.5834	15.0674	2024.5906	0.7192
	PSO-Otsu	80	2975.1216	15.0318	2041.2665	0.3484
	HFBOA-Otsu	15	2975.6304	15.0794	2019.0304	0.7327
Thorax	BOA-Otsu	58	8731.5057	11.2223	4907.4865	0.4439
	FFA-Otsu	14	8729.7165	11.4004	4710.2191	0.4583
	Im-FpsoOtsu	70	8731.8715	11.3341	4782.6472	0.4509
	PSO-Otsu	96	8731.7093	11.3925	4718.7377	0.0558
	HFBOA-Otsu	9	8731.9556	11.4158	4693.4567	0.4584
Brain	BOA-Otsu	72	8419.6296	10.0260	6463.7597	0.3173
	FFA-Otsu	14	8412.6832	10.3471	6003.0255	0.3236
	Im-FpsoOtsu	46	8454.5139	10.2513	6136.9037	0.3213
	PSO-Otsu	59	8453.829406	10.3836	5952.8460	0.1204
	HFBOA-Otsu	9	8454.8034	10.3943	5938.1894	0.3239

**Figure 9.** The fitness curves of medical images. (a) The fitness curve of the Lung1 image; (b) the fitness curve of the Lung2 image; (c) the fitness curve of the Thorax image; (d) the fitness curve of the Brain image.

6. Conclusions

The BOA algorithm has some disadvantages, such as a slow convergence speed and low optimization accuracy, and it easily falls into local optima. This paper introduces the fractional-order differential, sine-cosine algorithm, and adaptive parameters and proposes the HFBOA algorithm to avoid the shortcomings of the BOA algorithm. The proposed algorithm was tested on 12 benchmark functions and compared with four other algorithms, such as the BOA algorithm and the PSO algorithm. The results show that the HFBOA algorithm converges faster and has better optimization accuracy. By combining the HFBOA algorithm with the 2-D Otsu algorithm, the HFBOA-Otsu algorithm is proposed to perform image segmentation. To verify the segmentation effects of the HFBOA-Otsu algorithm, four human images, four scenery images, and four medical images were used to perform the segmentation experiments. The experimental results show that, although there are small differences between indexes, the indexes of the HFBOA-Otsu are better than those of the other algorithms. Generally, the convergence speed and segmentation accuracy of the HFBOA-Otsu are better. Compared with the BOA-Otsu algorithm, the convergence rate of the HFBOA-Otsu algorithm is improved by about 73.38%, and the fitness function value and the SSIM value are increased by about 0.21% and 5.28%, respectively. The MSE value is decreased by approximately 2.32%, and the PSNR value is increased by about 1.18%. Therefore, the HFBOA-Otsu algorithm effectively improves the low segmentation accuracy and slow convergence rate of the BOA-Otsu algorithm.

Author Contributions: Conceptualization, Z.D. and Y.M.; methodology, Z.D. and Y.M.; software, Z.D.; validation, J.Z. and Z.M.; formal analysis, Z.D. and Y.M.; data curation, Z.D. and Y.M.; writing—original draft preparation, Z.D.; writing—review and editing, Z.D., Y.M., J.Z. and Z.M. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by Key Research and Development Projects of Ningxia Autonomous Region (no. 2023BDE03002); and National Natural Science Foundation of China (no. 42361056).

Data Availability Statement: The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Reddi, S.S.; Rudin, S.F.; Keshavan, H.R. An optimal multiple threshold scheme for image segmentation. *IEEE Trans. Syst. Man Cybern.* **1984**, *14*, 661–665. [[CrossRef](#)]
2. Zhang, J.B.; Yang, H.X.; Zhou, T.T.; Tan, Z.H.; Li, M. Improved segmentation method of 2-D Otsu infrared image. *Laser Technol.* **2014**, *38*, 713–717.
3. Huang, C.; Li, X.; Wen, Y. AN OTSU image segmentation based on fruitfly optimization algorithm. *Alex. Eng. J.* **2020**, *60*, 183–188. [[CrossRef](#)]
4. Liu, X.J.; Liu, Y.L.; Xu, X.X. Optimization of multi-thresholding Otsu image segmentation by glowworm swarm algorithm with cell membrane mechanism. *J. Chin. Comput. Syst.* **2020**, *41*, 410–415.
5. Ning, G.Y. Two-dimensional Otsu multi-threshold image segmentation based on hybrid whale optimization algorithm. *Multimed. Tools Appl.* **2022**, *82*, 15007–15026. [[CrossRef](#)]
6. Ma, G.Y.; Yue, X.F. An improved whale optimization algorithm based on multilevel threshold image segmentation using the Otsu method. *Eng. Appl. Artif. Intell.* **2022**, *113*, 104960. [[CrossRef](#)]
7. Arora, S.; Singh, S. Butterfly optimization algorithm: A novel approach for global optimization. *Soft Comput.* **2019**, *23*, 715–734. [[CrossRef](#)]
8. Wu, L.; Du, L.B.; Tang, X.F. Multi-level threshold image segmentation based on an improved butterfly optimization algorithm. *Chin. Sci. Technol. Pap.* **2021**, *16*, 1174–1180.
9. Pu, Y.F. Research on Application of Fractional Calculus to Latest Signal Analysis and Processing. Master’s Thesis, Sichuan University, Chengdu, China, 2006.
10. Micael, C.; Seenith, S. Novel fractional order particle swarm optimization. *Appl. Math. Comput.* **2016**, *283*, 36–54.
11. Wei, J.R.; Ma, Y.; Xia, R. Otsu image segmentation algorithm based on fractional particle swarm. *Comput. Eng. Des.* **2017**, *38*, 3284–3290.

12. Dalia, Y.; Mohamed, A.E.; Seyedali, M. Fractional-order calculus-based flower pollination algorithm with local search for global optimization and image segmentation. *Knowl.-Based Syst.* **2020**, *197*, 105889.
13. Chen, L.P.; Gao, J.H.; Lopes, A.M.; Zhang, Z.Q.; Chu, Z.B.; Wu, R.C. Adaptive fractional-order genetic-particle swarm optimization Otsu algorithm for image segmentation. *Appl. Intell.* **2023**, *53*, 26949–26966. [[CrossRef](#)]
14. Wang, X.M.; Zhang, W.Y.; Xing, J.Y. Higher-order Caputo fractional order differential operators and its application in image enhancement. *J. Comput. Res. Dev.* **2023**, *60*, 448–464.
15. Nguyen, P.T.; Nguyen, S.H.; Mai, T.V. Finite-time dissipative control design for one-sided Lipschitz nonlinear singular Caputo fractional order systems. *Int. J. Syst. Sci.* **2023**, *54*, 1694–1712.
16. Minh, V.T.; Wei, W. Caputo fractional differential variational–hemivariational inequalities involving history-dependent operators: Global error bounds and convergence. *Commun. Nonlinear Sci.* **2024**, *128*, 107654.
17. Fan, Y.Q.; Shao, J.P.; Sun, G.T. A Self-Adaption Butterfly Optimization Algorithm for Numerical Optimization Problems. *IEEE Access*. **2020**, *2020*, 88026–88041. [[CrossRef](#)]
18. Singh, N.; Singh, S. A novel hybrid GWO-SCA approach for optimization problems. *Eng. Sci. Technol.* **2017**, *20*, 1586–1601. [[CrossRef](#)]
19. Xia, Q.Y.; Ding, Y.M.; Zhang, R.; Liu, M.T.; Zhang, H.T. Blind Source Separation Based on Double-Mutant Butterfly Optimization Algorithm. *Sensors* **2022**, *22*, 3979. [[CrossRef](#)]
20. Kong, C.Y. Research on Otsu Image Segmentation Algorithm Based on Firefly Algorithm Optimized by Fractional-Order. Master's Thesis, Ningxia University, Yinchuan, China, 2019.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.