IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received February 20, 2019, accepted March 2, 2019, date of publication April 1, 2019, date of current version April 16, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2908653

Hybrid Multiverse Optimization Algorithm With Gravitational Search Algorithm for Multithreshold Color Image Segmentation

HEMING JIA, (Member, IEEE), XIAOXU PENG, WENLONG SONG, CHUNBO LANG, ZHIKAI XING, AND KANGJIAN SUN
College of Mechanical and Electrical Engineering, Northeast Forestry University, Harbin 150040, China

Corresponding author: Wenlong Song (swl@nefu.edu.cn)

This work was supported by the National Natural Science Foundation of China under Grant 31470714.

ABSTRACT Multithreshold segmentation of color images is an important part of modern image processing. Multithreshold segmentation techniques that use the ideas of optimization in mathematics to process the information contained in the three channels of RGB images have received much attention in recent years. In this paper, a hybrid algorithm of multiverse optimization algorithm with a gravitational search algorithm (GSMVO) is proposed. On the one hand, the gravitational mechanism among individuals in the universe enables particles to share information and move toward particles with large mass (the optimal solution), thus optimizing local optimization. On the other hand, the problem of a slow interpopulation update is improved. On the premise of improving the optimization precision, the convergence speed of the whole system is accelerated and the robustness can be improved. The time complexity of a proposed hybrid algorithm is analyzed. The quality of a segmented image is evaluated by using peak signal-to-noise ratio (PSNR), feature similarity index (FSIM), structural similarity index (SSIM), and Wilcoxon test. The experimental results illustrate that the proposed method has obvious advantages in objective function value, image quality measurement, convergence performance, and robustness.

INDEX TERMS Multiverse optimization algorithm, gravitational search algorithm, multithreshold segmentation, Kapur's entropy, Otsu method.

I. INTRODUCTION

Image analysis, target recognition, computer vision and many other fields have been in the state of hot development. Image segmentation is the preprocessing stage of these higher-level processing [1], [2]. Accurate segmentation technology determines the performance of higher level processing system, therefore it is particularly important to obtain a high quality result of image segmentation. Image segmentation refers to the process of dividing digital images into multiple image sub-regions with respect of unique characteristics. More precisely speaking, image segmentation is a process of tagging each pixel in the image, which makes pixels with the same label have some common visual characteristics [3].

The existing image segmentation methods are mainly divided into the following categories: threshold segmentation, region growth, region division and merging, watershed algorithm, edge detection, histogram method, cluster analysis, wavelet transform, etc. Threshold segmentation is one of the most widely used methods, which can be divided into bi-level threshold and multilevel threshold [4], [5]. Bi-level threshold is the simplest segmentation method, as long as one gray value can be determined to divide the image into two regions of interest(ROI) [6]. In actual image processing, color images contain more than two ROI, so only multilevel threshold methods can be adopted. The pixels are divided into groups, and within each group, the pixels have intensity values within a specific range.

FIGURE 1. **Back hole, white hole, and wormhole [39], [40].**

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

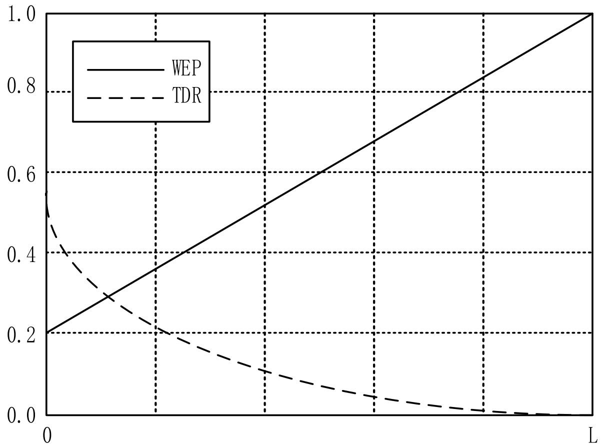


FIGURE 2. WEP versus TDP.

TABLE 1. The Pseudocode of the proposed local update method.

1	Calculate the Inflation_rates of each universe;
2	Update the best solution Best_universe ;
3	Update the inertial mass M_i^t ,universal gravitation $G(t)$, acceleration a_i^t , speed v_i^t , and position x_i^t by using (21), (25), (18), (16), (27) respectively ;
4	Continue global update with MVO;

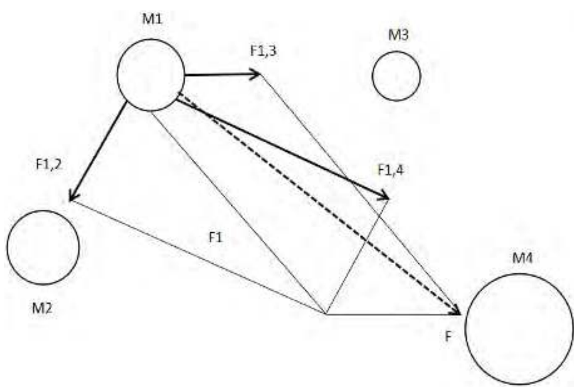


FIGURE 3. Universal gravitation [32].

Sezgin et al. divided the image threshold into six groups according to information. These methods include histogram based method, clustering based method, entropy based method, object attribute based method, spatial method and local method [7]. Histogram based method and entropy based method are simple and widely used. Otsu method is based on the representation of histogram method. Each rectangle in the histogram represents each pixel gray cluster, and the segmentation threshold is obtained by calculating the maximum between-cluster variance of each pixel gray cluster [8].The higher the between-class variance is, the better the segmentation effect will be. Kapur entropy is one of the more commonly used entropy based methods. In the image, the uniform region corresponds to the minimum entropy, while the non-uniform region defines the maximum entropy. Therefore, a better segmentation effect can be obtained by obtaining a larger entropy of the segmented image [9]. Different entropy based methods, such as fuzzy entropy [10], [11], cross entropy [12], [13], Tsallis entropy [14], Renyi entropy [15], Shannon entropy [16], were proposed successively.

TABLE 2. The Pseudo-code of the proposed multilevel threshold method.

1	Initialize parameters N (the number of population size), Level (the number of thresholds), Max_time (maximum number of iteration) ;
2	Randomly initialize the population Sorted Universes (SU) ;
3	Initialize the counter Time = 0, WEP and TDR by using (14) and (15) respectively , $G_0 = 100$, $a = 20$;
4	**While** Time < Max_time **do**
5	Time=Time+1;
6	**For** each universe indexed by ***i***
7	Check if any search agent goes beyond the search space and amend it ;
8	Calculate the objective function value of each universe (Inflation_rates of the universe) by using (3) or (9) (NI) ;
9	Update the best solution Best_universe ;
10	Update the inertial mass M_i^t ,universal gravitation $G(t)$, acceleration a_i^t , speed v_i^t , and position x_i^t by using (21), (25), (18), (16), (27) respectively ;
11	Update WEP and TDR
12	Black_hole_index = ***i*** ;
12	**For** each object indexed by ***j***
13	r1 = random ([0,1]) ;
14	**If** r1<NI(U_i) **Then**
15	White_hole_index=Roulette Wheel Selection (NI) ;
16	U(Black_hole_index, j) = SU(White_hole_index, j) ;
17	**End if**
18	r2 = random([0,1]) ;
19	**If** r2 < WEP **Then**
20	r3 = random ([0,1]) ;
21	L= Levy(λ) ;
22	**If** r3 < 0.3 **Then**
23	U(i,j) = Best_universe(1,j) + TDR*((ub-lb)*L+lb) ;
24	**Else if**
25	U(i,j) = Best_universe(1,j) - TDR*((ub-lb)*L+lb) ;
26	**End if**
27	**End if**
28	**End for**
29	**End for**
30	**End while**

As the No Free Lunch (NFL) theorem for optimization was proposed [17], people realized the openness of the research field of optimization algorithm. There is no ideal meta-heuristic algorithm for all optimization problems, and new or existing algorithms are superior to other algorithms for specific optimization problem sets. Algorithms could solve different domain problems through improvement and

TABLE 3. Parameters of the compared algorithms.

Reference	Algorithm	Parameters	Value
		WEP	[0.2,1]
[24] Mirjalili etc.	MVO	TDR	[0,1]
		Random parameters r_1, r_2, r_3	[0,1]
		Norm R	2
[33] Mittal etc.	GSA	Power R	2
		Gravitational constant G_0	100
		The variable parameter a	20
[44] Wang etc.	ALO	Switch possibility	0.5
		Inertial weight	[0.5,0.9]
		Seperation weight	
[45] Xu etc.	DA	Alignment weight	[0,0.2]
		Cohesion weight	
		Maximum velocity	25.5
		Food attraction weight	[0,2]
		Enemy distraction weight	[0,0.1]
		Switch possibility	0.4
[47] Shen etc.	FPA	Lévy constant λ	1.5
		Maximum inertia weight	0.9
[49] Maitra etc.		Minimum inertia weight	0.4
[50] Liu etc.	PSO	Learning factors c_1 and c_2	2
[51] Gao etc.		Maximum velocities	+120
		Minimum velocities	-120
		Mutation probability value P_a	0.25
[53] Bhandari etc.	CS	Scale factor β	1.5

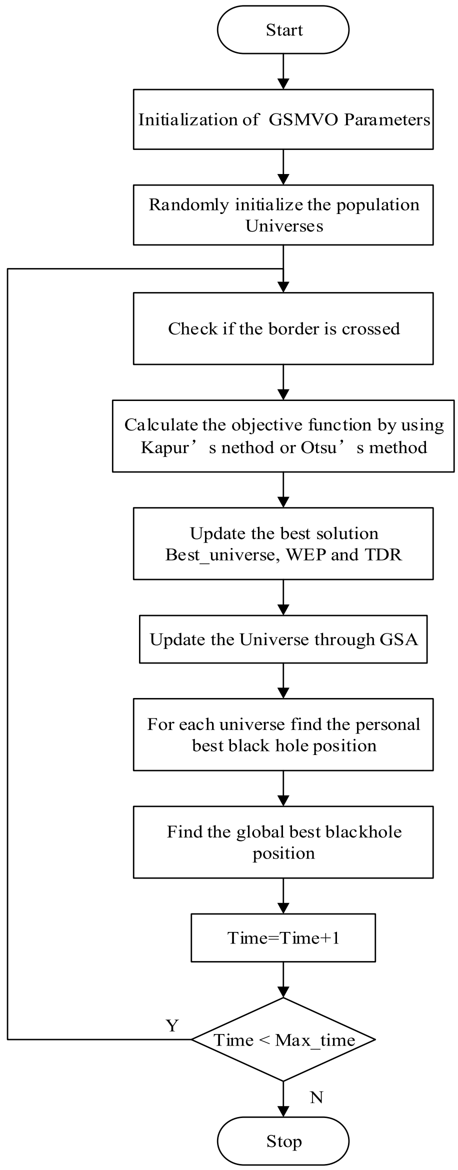


FIGURE 4. The GSMVO algorithm flow chart.

adjustment, so the combination of image segmentation and swarm intelligence algorithm became inevitable. There a multilevel threshold quantum behaved particle swarm optimization algorithm(PSO) based on Otsu function is proposed [18]. Using firefly algorithm (FA) to solve optimal multilevel image thresholding based on Otsu [19]. Multilevel image thresholding using Otsu and chaotic bat algorithm (BA) was promoted [20]. A hybridization of genetic algorithms and cross entropy methods (GACE) was proposed for solving continuous optimization [21]. An improved fuzzy entropy and Levy flying firefly algorithm (FA) method is used for color image threshold segmentation [22]. By maximizing Shannon entropy or fuzzy entropy the firefly algorithm (FA) is utilized to image segmentation [23].

MVO was proposed by S. Mirjalili in 2015 and also used to solve practical engineering optimization problems, such as pressure cylinder design, wheel design, and welding beam design [24]. It originated from the fact that the universe has an inflation rate. White holes have a higher inflation rate and black holes have a lower inflation rate. Regardless of the inflation rate of matter in the universe, all matter in the universe will move randomly through the wormhole to reach the optimal location of universe. In 2016, by utilizing multiverse optimization algorithm, S. Harminder et al. solved the problem of economic load scheduling (ELD) [25]. Hybrid MVO and PSO algorithm was presented to solved global numerical optimization and reactive optimization scheduling problem by Jangir et al. [26]. H. Faris et al. solved the feedforward neural network problem of training by using MVO algorithm [27]. In 2017, H. Faris et al. proposed a multiverse optimization method based on robust system architecture feature selection and optimization of SVM parameters [28]. Compared with the traditional algorithm, MVO algorithm has better performance, but there is no information exchange between its particles, so it has the problems

TABLE 4. The average CPU time of each algorithm under Kapur.

TEST IMAGES	K	GSMVO	MVO	GSA	ALO	DA	FPA	PSO	CS
Starfish	4	0.3500	0.3300	0.4060	1.6510	0.4360	0.4210	0.4670	**0.3290**
	6	0.4740	**0.3120**	0.4050	2.2770	0.6890	0.4270	0.4540	0.4000
	8	0.5170	**0.3280**	0.4410	2.8720	0.6880	0.4430	0.4720	0.4400
	10	0.5460	**0.3430**	0.4700	3.5540	0.7010	0.4610	0.4940	0.4499
	12	0.6110	**0.3900**	0.5230	4.2170	0.7220	0.4900	0.5210	0.4800
Corn	4	0.3700	**0.3700**	0.3750	1.7110	0.8220	0.6710	0.7400	0.6700
	6	0.4680	**0.3080**	0.4970	2.2410	0.7190	0.6630	0.6900	0.7100
	8	0.5010	**0.3300**	0.4370	2.8570	0.6730	0.8710	0.7950	0.8100
	10	0.5460	**0.3430**	0.4850	3.5980	0.6870	0.9420	0.8560	0.8319
	12	0.5930	**0.3750**	0.5200	4.1890	0.7210	0.9510	1.0060	0.9539
Tiger	4	0.3850	**0.3450**	0.3790	1.6720	0.7990	0.4060	0.4210	0.3909
	6	0.4600	**0.2970**	0.3950	2.2100	0.6600	0.4990	0.4410	0.4400
	8	0.5150	**0.3280**	0.4390	2.8730	0.6800	0.5080	0.4870	0.4860
	10	0.5460	**0.3450**	0.5930	3.8070	0.6860	0.5370	0.4990	0.5400
	12	0.5950	**0.3910**	0.5150	4.1780	0.7920	0.5520	0.5840	0.5600
Flower	4	0.4430	0.3910	**0.3750**	1.6400	0.7980	0.3900	0.4210	0.3820
	6	0.4530	**0.3120**	0.4240	2.4220	0.6570	0.4130	0.4660	0.4100
	8	0.5140	**0.3670**	0.4680	3.0080	0.7120	0.4260	0.4370	0.4300
	10	0.5610	**0.3590**	0.4680	3.4560	0.6860	0.5610	0.4980	0.5400
	12	0.5770	**0.3740**	0.4990	4.1190	0.7170	0.5730	0.5450	0.5500
Tower	4	0.4630	0.4880	0.4440	1.9390	0.7650	0.3920	0.4390	**0.3799**
	6	0.4680	**0.2960**	0.4550	2.4430	0.7560	0.4070	0.4220	0.4300
	8	0.4990	**0.3120**	0.4370	2.8540	0.6710	0.4570	0.4680	0.4700
	10	0.5460	**0.3590**	0.4840	3.5100	0.7090	0.4820	0.5010	0.5400
	12	0.5930	**0.3740**	0.5300	4.1650	0.7490	0.4930	0.4850	0.5519
Horse	4	0.3960	**0.3430**	0.3810	1.6830	0.7490	0.3900	0.4230	0.4000
	6	0.4680	**0.3120**	0.4370	2.2780	0.6400	0.5150	0.5470	0.4999
	8	0.5260	**0.3280**	0.4370	2.8200	0.6870	0.5320	0.5940	0.5500
	10	0.5610	**0.3590**	0.4880	3.9830	0.7560	0.5610	0.6100	0.6099
	12	0.6590	**0.3900**	0.5000	4.1500	0.7330	0.5620	0.6860	0.6599
River	4	0.4030	**0.3130**	0.4350	1.7380	0.7860	0.3920	0.4390	0.3520
	6	0.4680	**0.3290**	0.4050	2.8150	0.7660	0.5160	0.4060	0.4100
	8	0.5360	**0.3530**	0.4370	3.3310	0.7440	0.4520	0.4370	0.5100
	10	0.5960	**0.3830**	0.4690	3.5100	0.7020	0.4770	0.4780	0.5200
	12	1.0310	**0.4030**	0.5200	4.1970	0.7180	0.4822	0.4880	0.5600
Penguin	4	0.4620	0.4430	0.4060	1.7520	0.9150	**0.3900**	0.4210	**0.3900**
	6	0.5000	**0.3100**	0.4350	2.3240	0.7760	0.4990	0.4450	0.4499
	8	0.6200	**0.3740**	0.4600	3.3250	0.7310	0.4220	0.4380	0.5200
	10	0.6660	**0.4070**	0.5030	3.8080	0.7670	0.4380	0.4880	0.5500
	12	0.7510	**0.4330**	0.5560	4.2430	0.7590	0.4520	0.4990	0.5500
Dog	4	0.5050	0.4950	0.4830	2.0300	0.9320	0.3900	0.4530	**0.3800**
	6	0.5340	**0.3360**	0.4060	2.2780	0.6970	0.3900	0.4080	0.4199
	8	0.5620	**0.3590**	0.4960	2.9480	0.7180	0.4160	0.4530	0.4300
	10	0.7150	**0.4170**	0.4840	3.9760	0.7020	0.4370	0.4860	0.4399
	12	0.6390	**0.3910**	0.5460	4.4400	0.8370	0.4520	0.4840	0.4700
Wolf	4	0.4950	0.4590	0.4770	1.6840	0.8730	0.3900	0.4360	**0.3899**
	6	0.5340	**0.3150**	0.4290	2.3610	0.7020	0.4069	0.4830	0.4200
	8	0.5410	**0.3590**	0.4680	2.9400	0.6940	0.4219	0.4530	0.4049
	10	0.5710	**0.3580**	0.4990	3.6190	0.7150	0.4369	0.4840	0.4300
	12	0.6040	**0.3740**	0.5600	4.4080	0.7920	0.4539	0.5309	0.4600

of slow convergence, low accuracy, unstable system and easy to be trapped in the local optimal problem.

With more and more new algorithms proposed, it can be seen that some pure algorithms will have certain ranges of application [29]. Therefore, hybrid meta-heuristic algorithm is proposed. In order to balance the two conflicting standards of diversification of search space and enhancement of the optimal solution found, a mixed model algorithm - memetic algorithm was generated [29]. In 2002, E.G. Talbi published an article about a taxonomy of hybrid metaheuristics [30]. E.A. Baniani proposed a hybrid particle swarm optimization (PSO) algorithm and a genetic algorithm (GA) in 2013 for multi-level maximum entropy criterion threshold selection [31]. A hybrid whale algorithm and simulated annealing optimization algorithm have been applied to feature selection in 2017 [29]. In this study, we aim to construct a new hybrid algorithm using the recently proposed MVO and gravitational search algorithm (GSA) to solve the problem of multithreshold image segmentation.

The GSA is a meta-algorithm based on the concept of Newtonian gravity. Compared with traditional meta-heuristic algorithms, it performs better in searching for solutions of nonlinear functions in multidimensional space [32], [6]. The local part of GSA updates through Newton's law of gravity, and gradually develops the search space through the law of motion. Optimizes it under the condition of ensuring information exchange between particles. GSA converges fast but is easily in the local optimal, and the accuracy of the solution is poor [33], [34]. In this regard, the partial idea of GSA can be introduced into MVO, then the combination of the roulette mechanism in MVO is a global optimization, and the fusion of the algorithm is finally implemented.

Hybrid with GSA is helpful to obtain a better balance between the exploration and exploitation of algorithms, and has advantage in speeding up local update, improving the accuracy of local optimization. Therefore, this paper proposes an improved version of MVO based on GSA, called GSMVO, which aims to improve the convergence speed and accuracy of traditional MVO. On this basis, combining the Otsu and Kapur's entropy, the multithreshold segmentation of color images is realized. By analyzing and evaluating the image quality after segmentation, the performance of improved algorithm is evaluated. The rest of this paper is organized as follows: section II briefly introduces the basic principles of multithreshold segmentation Kapur's entropy and Otsu. Traditional MVO and GSA are described in section III and IV. In section V, the proposed hybrid GSMVO algorithm is introduced. In order to prove the superior performance of GSMVO, in section VI, various experimental results, discussion and analysis of image segmentation are given. Section VII, the last section, gives the final conclusion.

TABLE 5. The average CPU time of each algorithm under Otsu.

TEST IMAGES	K	GSMVO	MVO	GSA	ALO	DA	FPA	PSO	CS
Starfish	4	0.3770	0.4900	0.5610	1.8410	0.7890	0.4600	0.5170	**0.3380**
	6	0.3920	**0.3800**	0.5630	2.5680	0.7968	0.4060	0.4870	0.4100
	8	0.4800	**0.3900**	0.5930	3.3520	0.8260	0.4680	0.4590	0.5360
	10	0.4500	**0.3900**	0.6550	4.2130	0.8890	0.5360	0.5170	0.5500
	12	0.4800	**0.4200**	0.6710	4.5920	0.9030	0.4690	0.5230	0.5970
Corn	4	0.6650	0.4300	0.6090	1.8900	0.9520	**0.4220**	0.4720	0.9890
	6	0.7050	**0.3699**	0.5930	2.4930	0.8890	0.4000	0.4400	0.8200
	8	0.7250	**0.3799**	0.6700	3.1680	0.8320	0.5210	0.5460	0.8810
	10	0.8320	**0.3999**	0.6550	4.1560	0.8430	0.4520	0.4990	0.8630
	12	0.9050	**0.4100**	0.6550	4.5290	0.8730	0.6710	0.4990	0.9080
Tiger	4	**0.3900**	0.4200	0.5620	1.9830	0.9120	0.4370	0.4680	0.5310
	6	0.4650	**0.3500**	0.5460	2.4960	0.8270	0.4220	0.4220	0.4840
	8	0.4810	**0.3799**	0.5770	3.0580	0.8430	0.4520	0.4680	0.5150
	10	0.4970	**0.3900**	0.6550	3.7280	0.8430	0.4520	0.4990	0.5170
	12	0.5120	**0.4099**	0.6550	4.3830	0.8730	0.5000	0.4840	0.5460
Flower	4	**0.3820**	0.4000	0.7300	1.8720	1.0430	0.4520	0.4370	0.5000
	6	0.4070	**0.3599**	0.5620	2.3560	0.7950	0.4210	0.4210	0.4690
	8	0.4130	**0.3700**	0.6800	3.0580	0.8510	0.4680	0.4530	0.5000
	10	0.4960	**0.4000**	0.7400	3.9600	0.8740	0.4680	0.5160	0.5570
	12	0.5360	**0.4099**	0.6860	4.7290	0.9830	0.4930	0.5400	0.5470
Tower	4	**0.4280**	0.8100	0.6280	1.9690	1.1130	0.4520	0.4370	0.5310
	6	0.4500	**0.3819**	0.5790	2.4650	0.9460	0.4520	0.5040	0.4680
	8	0.4520	**0.4330**	0.6240	3.3150	0.9680	0.4520	0.4530	0.5070
	10	0.4810	**0.3999**	0.6550	3.7290	0.8890	0.4370	0.4680	0.5410
	12	0.4820	**0.4300**	0.6890	4.8260	0.8900	0.4680	0.4840	0.5460
Horse	4	**0.3960**	0.4099	0.5940	2.0570	1.1510	0.4370	0.5140	0.4990
	6	0.4660	**0.3880**	0.5620	2.4480	0.8780	0.4800	0.4210	0.6400
	8	0.5240	**0.3800**	0.5920	3.0580	0.8260	0.4360	0.4580	0.6500
	10	0.5500	**0.3899**	0.7180	3.9950	0.8420	0.4530	0.4780	0.7080
	12	0.5550	**0.4100**	0.7210	4.6620	0.8750	0.4680	0.4680	0.5510
River	4	0.3850	**0.3099**	0.5620	1.9450	0.9820	0.4370	0.4520	0.4990
	6	0.3900	**0.3480**	0.6020	2.6850	0.8580	0.4680	0.4360	0.4680
	8	0.4610	**0.3800**	0.6090	3.1070	0.8270	0.4370	0.4530	0.4990
	10	0.4760	**0.3899**	0.6400	3.8660	0.9680	0.5420	0.5320	0.5310
	12	0.4820	**0.4100**	0.6710	4.5030	0.9110	0.5140	0.5970	0.5460
Penguin	4	0.3820	**0.3790**	0.5300	1.8710	0.9550	0.4210	0.4530	0.4840
	6	0.4650	**0.3902**	0.5610	2.4030	0.8580	0.4210	0.4210	0.4840
	8	**0.3910**	0.3998	0.5980	3.0420	0.8430	0.4520	0.4370	0.5150
	10	**0.3960**	0.4164	0.6560	3.7440	0.8420	0.4370	0.4840	0.5120
	12	**0.4130**	0.4302	0.6700	4.4460	0.8580	0.4680	0.4990	0.5630
Dog	4	**0.3820**	0.4529	0.5300	1.8710	0.9550	0.4210	0.4530	0.4850
	6	0.3910	**0.3500**	0.5610	2.4030	0.8580	0.4210	0.4210	0.4530
	8	0.4110	**0.3710**	0.5980	3.0420	0.8430	0.4520	0.4370	0.4990
	10	0.4303	**0.3900**	0.6560	3.7440	0.8420	0.4370	0.4840	0.5160
	12	0.4520	**0.4100**	0.6700	4.4460	0.8580	0.4680	0.4990	0.5080
Wolf	4	**0.3829**	0.5000	0.5460	1.8410	0.9050	0.4060	0.4210	0.4840
	6	0.3950	**0.3829**	0.5460	2.4020	0.7960	0.4520	0.4210	0.4840
	8	**0.3999**	**0.3999**	0.5770	3.0420	0.8110	0.4520	0.4520	0.4990
	10	0.4160	**0.4100**	0.6390	3.6820	0.8270	0.4520	0.4360	0.5299
	12	0.4529	**0.4139**	0.6400	4.3680	0.8890	0.4680	0.4680	0.5459

II. MULTILEVEL THRESHOLDING

This section reviews two traditional thresholding techniques as follows.

Kapur et al. (1985) assumed that there are two different sources in the given images, one is target the other is background. If the image can be represented by its grey histogram, thresholds subdivide it into different classes. Then by maximizing the entropy of classes, the optimal thresholds will be found which are going to correspond to the each maximum objective function.

Otsu. (1979) supposed there are a series of pixel gray clusters within the grey histogram. Calculating the between-cluster variance of the each pixel gray cluster, the maximum result is the maximum objective function value.

Assuming that an image has L gray value $[0, 1, \cdots, L-1]$, let n_i represent the number of pixels with gray value of i in the image. Correspondingly, the total number of pixels is $N = n_0 + n_1 + \ldots + n_{L-1}$, and the distribution probability p_i of the *ith* gray value is indicated as:

$$p_i = n_i/N \tag{1}$$

Which $p_i \geq 0$, and $\sum_{0}^{L-1} p_i = 1$. Suppose there are K thresholds, then $t_1, t_2, \cdots, t_K$ can divide the gray levels of the given image into K+1 classes. Define different classes as: $[0, t_1 - 1] \in M_0, [t_1, t_2 - 1] \in M_1, \ldots, [t_K, L-1] \in M_K$, where $t_1 < t_2 < \ldots < t_K$. Then $t_0 = 0$ and $t_{K+1} = L$.

A. KAPUR'S ENTROPY METHOD

The Kapur's entropy criterion method proposes the concept of entropy, and the entropy of the image can be represented as:

$$H_n = -\sum_{i=0}^{L-1} p_i \ln p_i \tag{2}$$

For image segmentation of K thresholds, the objective function is defined as:

$$\begin{aligned}\psi(t_1, t_2, \cdots, t_K) = &-\sum_{i=0}^{t_1-1} \frac{p_i}{P_0} \ln \frac{p_i}{P_0} \\ &-\sum_{i=t_1}^{t_2-1} \frac{p_i}{P_1} \ln \frac{p_i}{P_1} - \cdots - \sum_{i=t_K}^{L-1} \frac{p_i}{P_K} \ln \frac{p_i}{P_K}\end{aligned} \tag{3}$$

where

$$P_j = \sum_{i \in M_j} p_i, \quad j = 0, 1, \cdots, K \tag{4}$$

Represent the probabilities of each class in the segmented classes $M_1, M_2, \ldots, M_K$, respectively. And the optimal threshold $\{t_1^*, t_2^*, \cdots, t_K^*\}$ is defined as follows:

TABLE 6. The optimal fitness value of each algorithm under Kapur.

TEST IMAGES	K	GSMVO	MVO	GSA	ALO	DA	FPA	PSO	CS
	4	**18.6487**	18.6485	17.8917	**18.6487**	18.6484	18.3501	**18.6487**	18.5886
	6	**23.9568**	23.8859	22.8937	23.9567	23.9563	23.0721	23.9567	23.5935
Starfish	8	**28.7074**	28.2657	26.1773	28.6943	28.7063	27.5602	28.7073	28.0042
	10	**33.0287**	32.4963	31.0656	32.9946	33.0112	31.3403	32.3147	31.8093
	12	**37.0170**	36.3465	33.1393	37.0153	36.8930	34.6132	36.8724	35.6828
	4	**18.6487**	18.6484	18.5830	**18.6487**	**18.6487**	18.4174	**18.6487**	18.5830
	6	**23.9567**	23.9282	23.8272	23.9566	23.9566	23.2916	23.9566	23.8272
Corn	8	**28.7072**	28.3950	28.0510	**28.7072**	28.6815	27.4610	28.7067	28.0510
	10	**33.0294**	32.3542	31.8782	**33.0294**	32.9772	31.4954	33.0291	31.8782
	12	**37.0178**	36.0237	35.4604	37.0067	36.9606	34.4249	37.0067	35.4604
	4	**18.7457**	18.7446	16.8203	**18.7457**	18.7454	18.5836	**18.7457**	18.7183
	6	**24.0157**	23.9078	22.3389	24.0139	24.0140	23.5795	24.0130	23.6780
Tiger	8	**28.7737**	28.5268	25.0839	28.7641	28.7103	27.6302	28.7731	28.1269
	10	**33.1110**	32.3747	29.9632	33.0999	33.0613	31.2407	33.1108	31.9593
	12	**37.1149**	35.9476	32.0982	37.1074	36.9899	34.7610	37.0135	35.6222
	4	**23.9078**	22.3389	**23.9078**	22.3389	**23.9078**	22.3389	**23.9078**	22.3389
	6	**25.5268**	25.0839	**25.5268**	25.0839	23.9268	25.0839	24.5226	25.0839
Flower	8	**27.9078**	22.3389	27.5056	22.3389	23.9978	22.3389	25.9078	25.3389
	10	**28.5268**	25.0839	28.0068	25.0839	25.5268	25.0839	25.8265	25.9764
	12	**29.9078**	26.3389	28.9078	26.3389	27.9078	26.3389	28.9078	22.3389
	4	**17.7408**	17.7013	16.7375	17.7232	17.723	17.4051	17.7407	17.6264
	6	**23.1306**	23.0635	19.6365	23.1130	23.1274	22.4431	23.1141	22.5809
Tower	8	**27.9828**	27.6141	24.2381	27.9297	27.8347	26.2793	27.9401	27.0202
	10	32.3357	31.6485	29.7799	32.3865	32.1599	30.565	**32.4376**	31.2620
	12	**36.4895**	35.5306	30.5024	36.4483	36.2947	34.3697	36.1192	34.6203
	4	**18.2922**	18.2863	16.3340	**18.2922**	**18.2922**	17.8418	**18.2922**	18.2159
	6	23.4658	23.4001	19.8465	23.4370	23.4937	23.0050	**23.4939**	23.1449
Horse	8	**28.1641**	27.8693	22.8181	28.1582	28.1358	26.9022	28.1419	27.6354
	10	**32.4642**	31.7308	30.6632	32.4279	32.3328	30.1471	32.3426	31.5491
	12	36.3198	35.0085	30.9706	**36.3706**	36.267	33.7251	35.7835	34.6119
	4	**17.7485**	17.7481	15.7614	**17.7485**	**17.7485**	17.5556	**17.7485**	17.6802
	6	**22.7148**	22.7005	19.7721	22.7144	22.7146	21.9675	22.7147	22.4987
River	8	**27.2343**	26.6765	25.0595	27.2103	27.2019	25.9951	27.2340	26.6621
	10	31.3253	30.9596	26.6303	**31.3908**	31.3292	29.4426	31.3686	30.2725
	12	**35.2951**	34.5001	30.5639	35.2949	34.8878	31.9370	35.2131	34.0655
	4	**18.7236**	18.7235	17.1694	**18.7236**	18.7233	18.5785	**18.7236**	18.6606
	6	**24.0251**	24.0159	17.7367	24.0162	24.0250	23.4457	24.0163	23.7478
Penguin	8	28.7675	28.4945	25.7443	**28.7716**	28.7681	27.861	28.7712	27.9145
	10	32.9884	32.4836	26.6705	33.0030	32.8651	30.8507	**33.0046**	31.7606
	12	**36.8537**	35.4513	33.0370	36.6857	36.8228	34.4137	36.2420	34.9899
	4	**18.5261**	18.5247	17.3078	**18.5261**	18.5258	18.2229	**18.5261**	18.4150
	6	23.6809	23.6778	21.6486	23.6845	**23.7411**	23.2099	23.7316	23.2927
Dog	8	**28.4227**	28.0775	23.1606	28.4174	28.4093	26.8028	28.4082	27.4945
	10	**32.6963**	32.0706	30.4658	32.6947	32.6562	31.2007	32.6477	31.5823
	12	**36.6155**	35.8466	33.0160	36.5290	36.5394	34.0841	36.0076	34.8832
	4	**18.3501**	18.3490	15.8957	**18.3501**	18.3489	18.1872	18.3500	18.2992
	6	23.5150	23.4431	20.5844	23.5154	23.5155	22.8723	**23.5156**	23.2843
Wolf	8	**28.1461**	27.5990	24.3579	28.1456	28.1317	27.0374	28.1299	27.3173
	10	32.2682	31.6558	28.5047	**32.3488**	32.3024	30.5153	32.3450	31.1612
	12	**36.2521**	35.3278	32.8685	35.6301	36.0286	33.1081	36.1317	34.4601

$$\left\{t_1^*, t_2^*, \cdots, t_K^*\right\} = \mathop{\arg\max}_{0<t_1<t_2\cdots<t_K<L-1} \left(\psi\left(t_1, t_2, \cdots, t_K\right)\right) \tag{5}$$

B. Otsu METHOD

In the past decades, the Otsu's between class variance method has been known for its unsupervised automatic threshold selection technology.

Respectively provides that the class probabilities ω_k, the class mean levels μ_k and the total mean level μ_T are expressed by (6), (7) and (8).

$$\omega_k = \sum_{j=t_k}^{t_{k+1}-1} P_j \tag{6}$$

$$\mu_k = \sum_{j=t_k}^{t_{k+1}-1} \frac{jP_j}{\omega_j} \tag{7}$$

$$\mu_T = \sum_{j=0}^{L-1} jP_j \tag{8}$$

Then the objective function can be mathematical described by:

$$Fit = \sum_{k=0}^{K} \omega_k \left(\mu_k - \mu_T\right)^2 \tag{9}$$

Which the definition of the optimal threshold of Otsu is as follows:

$$\left(t_1^*, t_2^*, \ldots, t_K^*\right) = \mathop{\arg\max}_{0<t_1<t_2\cdots<t_K<L-1} \left(Fit\left(t_1, t_2, \ldots, t_K\right)\right) \tag{10}$$

III. MULTIVERSE OPTIMIZATION

A. BASIC CONCEPTS

MVO is inspired by the big bang and quantum mechanics [35], [36]. In the standard MVO algorithm we treat each universe as a possible solution vector, and an object in the universe as a variable in the corresponding solution vector. Just as the concept of universe, every possible solution has its own inflation rate which refers to the fitness function of solution. There are four rules during the entire optimization:

1. Black holes [37]: In a universe with a lower inflation rate, the black holes tend to receive more objects through in.

2. White holes [38]: Behave completely in contrast to black wholes, the white holes tend to send objects throughout in a universe with a higher inflation rate.

3. Wormholes [39]: Also known as "grey lanes", which connect black holes to white holes. Randomly, it can send objects from the universe to the best without being limited by the inflation rate.

TABLE 7. The optimal fitness value of each algorithm under Otsu.

TEST IMAGES	K	GSMVO	MVO	GSA	ALO	DA	FPA	PSO	CS
	4	**3071.0220**	**3071.0220**	2537.1640	**3071.0220**	3071.0032	3039.2425	**3071.0220**	3060.7670
	6	**3140.9117**	3140.9001	3108.8202	**3140.9117**	3140.8689	3109.0333	3140.8985	3125.7660
Starfish	8	**3170.0265**	3168.0077	3077.5937	3170.0098	3169.9256	3135.3826	3169.9953	3159.4584
	10	**3184.3301**	3183.2131	3137.4838	3184.3268	3182.5098	3162.6371	3184.2842	3166.4263
	12	**3192.4060**	3189.7146	3162.6578	3191.9911	3189.0695	3169.2211	3191.9683	3181.8771
	4	**3738.0119**	**3738.0119**	3700.5353	**3738.0119**	**3738.0119**	3696.7425	**3738.0119**	3724.8631
	6	**3812.9643**	3812.9546	3745.4551	3812.9629	3812.6615	3776.6618	**3812.9643**	3794.5566
Corn	8	**3843.9380**	3843.7309	3774.8099	3843.9021	3843.7346	3826.6457	3843.9367	3832.2334
	10	3860.2602	3857.8584	3828.1338	**3860.3635**	3858.2986	3834.8897	3855.9065	3836.4589
	12	**3869.7609**	3868.6701	3834.6984	3868.3956	3868.6147	3839.1028	3865.7612	3853.6750
	4	**1526.6241**	**1526.6241**	1442.0585	**1526.6241**	1526.6238	1489.8723	1513.6832	1518.4929
	6	**1580.5490**	1580.5145	1497.8294	1580.5460	1580.5435	1559.2558	1580.5484	1566.7080
Tiger	8	**1603.0900**	1602.9956	1564.9892	1603.0875	1602.4652	1575.7799	1602.9836	1584.1056
	10	**1614.4553**	1614.3936	1586.8919	1614.4193	1612.6998	1585.1113	1608.4164	1601.4807
	12	**1620.7134**	1620.2080	1593.7020	1620.6364	1617.4007	1605.2895	1619.8679	1609.6516
	4	**2319.2628**	2319.2426	2281.4346	**2319.2628**	**2319.2628**	2305.1710	**2319.2628**	2309.3899
	6	**2374.0065**	2371.7573	2334.7560	**2374.0065**	2373.8011	2345.7656	**2374.0065**	2360.0897
Flower	8	**2398.7273**	2397.3091	2360.1324	2398.5875	2398.6086	2382.2012	2398.7027	2386.2317
	10	2410.6860	2409.7282	2384.8546	**2410.7376**	2410.0159	2392.2554	2399.0096	2394.5237
	12	2417.2394	**2417.4012**	2407.5498	2416.8590	2415.4150	2400.6074	2415.6071	2404.2547
	4	**1922.6304**	**1922.6304**	1823.0203	**1922.6304**	**1922.6304**	1898.7170	**1922.6304**	1915.5464
	6	**1969.6134**	1969.5986	1877.4597	1969.6068	1966.9062	1953.9635	1969.6098	1953.5761
Tower	8	**1989.0757**	1987.7523	1928.9302	1989.0648	1988.6071	1971.8455	1987.7888	1978.9366
	10	**1999.7221**	1999.3380	1954.1717	1999.5350	1998.2540	1974.4560	1996.0304	1986.1842
	12	**2004.9033**	2004.8888	1981.1256	2004.0389	2002.1234	1984.5875	2004.5132	1991.1834
	4	**1960.2297**	**1960.2297**	1782.8356	1960.2297	1960.2083	1915.2116	1960.2297	1953.9886
	6	**2016.4234**	2016.4149	1956.9189	2016.4080	2016.4171	1992.6753	2016.4212	2005.8019
Horse	8	**2038.6265**	2034.9849	1690.7832	2038.6208	2034.9847	2020.5910	2033.0526	2025.1378
	10	**2049.9723**	2047.7570	1987.9079	2048.1435	2049.7119	2021.2855	2044.5825	2035.7076
	12	**2056.1554**	2054.7641	2030.9139	2055.4503	2053.0422	2036.4605	2052.1811	2044.4973
	4	**3602.5126**	**3602.5126**	3282.9351	**3602.5126**	**3602.5126**	3568.9238	**3602.5126**	3593.8210
	6	**3676.1680**	3676.1483	3492.3286	**3676.1680**	3675.6902	3640.2587	3666.4701	3656.0719
River	8	**3705.3651**	3698.2279	3654.5629	3705.3353	3705.3003	3662.2698	3705.3444	3691.6482
	10	**3719.6464**	3715.8296	3654.5192	3719.4672	3718.4862	3691.1300	3717.6187	3703.6129
	12	**3727.4533**	3723.2935	3684.9529	3726.6097	3724.2316	3702.5741	3725.2104	3714.3571
	4	**2279.7675**	**2279.7675**	1812.5033	**2279.7675**	2279.7623	2249.3131	**2279.7675**	2272.0301
	6	**2338.1507**	2338.1290	2190.3621	2338.1403	2338.1181	2301.0181	**2338.1507**	2328.4121
Penguin	8	2364.1014	2356.8159	2335.6804	2364.1090	2364.0702	2327.3393	**2364.1140**	2350.9065
	10	**2377.6691**	2375.5599	2300.6606	2377.6487	2373.6796	2347.7418	2377.5479	2361.1150
	12	2384.9886	2380.9525	2362.7541	**2385.3470**	2384.3190	2367.5597	2383.2787	2375.0166
	4	2249.9848	**2249.9855**	2125.3918	**2249.9855**	**2249.9855**	2220.7397	**2249.9855**	2243.8614
	6	**2299.7411**	2299.6760	2246.2174	2299.7334	2296.9837	2262.1426	2293.2366	2284.2330
Dog	8	**2320.5300**	2317.4685	2267.8396	2317.6120	2319.4526	2295.7322	2320.5116	2304.4725
	10	**2331.1519**	2328.4166	2302.1501	2328.1101	2328.7200	2307.7168	2330.9643	2315.2940
	12	**2337.4718**	2335.0826	2321.1324	2337.3562	2335.5096	2314.5081	2333.5868	2327.7875
	4	**1194.7186**	1194.7033	1085.0241	**1194.7186**	**1194.7186**	1166.7150	**1194.7186**	1186.0943
	6	1234.5610	1234.5368	1171.7776	**1234.5701**	1234.5433	1213.3320	1234.5690	1223.0778
Wolf	8	**1253.1184**	1250.6969	1203.3949	1249.8918	1252.0728	1222.9033	1250.8688	1243.6199
	10	**1262.0460**	1261.5346	1236.1052	1261.9544	1261.1986	1243.5822	1257.1796	1254.0373
	12	**1266.8299**	1265.5588	1241.9932	1265.8203	1266.2885	1248.1427	1266.6028	1260.5045

4. The rate of expansion is the inflation rate of universe, which is various among universes [40]. In each iteration, the universe is sorted according to the value of inflation rate and the white hole realizes through the roulette wheel mechanism.

Therefore, by following these rules, objects can move from a universe with high inflation to one with low inflation, thus ensuring an increase in average inflation and achieve a stable state. Fig 1. enumerates these three main components of the multiverse theory, respectively.

B. MATHEMATICAL MODEL

Assuming that

$$U = \begin{bmatrix} x_1^1 & x_1^2 & \cdots & x_1^d \\ x_2^1 & x_2^2 & \cdots & x_2^d \\ \cdots & \cdots & \cdots & \cdots \\ x_n^1 & x_n^2 & \cdots & x_n^d \end{bmatrix} \tag{11}$$

where d is the number of parameters and n is the number of solutions which is correspond to the universe.

$$x_i^j = \begin{bmatrix} x_k^j & r_1 < NI\left(U_i\right) \\ x_i^j & r_1 \geq NI\left(U_i\right) \end{bmatrix} \tag{12}$$

where x_i^j represents the jth parameter of ith universe, U_i represents the ith universe, $NI\left(U_i\right)$ refers to the standard inflation rate of the ith universe and x_k^j indicates the jth parameter of kth universe selected by a roulette wheel selection mechanism.

The position updates in the optimal universe can be calculated as follows:

$$x_{i+1}^j = \begin{cases} \begin{cases} x_i^j + TDR \times \left(\left(ub_j - lb_j\right) \times r_4 + lb_j\right) & r_3 < H \\ x_i^j - TDR \times \left(\left(ub_j - lb_j\right) \times r_4 + lb_j\right) & r_3 \geq H \end{cases} & r_2 < WEP \\ x_i^j & r_2 \geq WEP \end{cases} \tag{13}$$

where, TDR (Travelling distance rate) is a dynamic parameter, and H is the threshold. Here, we take the empirical value of $H = 0.5$, r_1, r_2, r_3, r_4 are random numbers in the interval [0,1],x_i^j which represents the current position of optimal cosmic black hole. During iteration, the optimal position is updated, constantly. When a new optimal black hole location x_{i+1}^j is generated, the fitness value of x_i^j is compared with that of x_{i+1}^j. If the fitness value of x_{i+1}^j is better than x_i^j, then, it will replace the last generation of new black hole individuals. Otherwise, x_i^j will be saved to the next generation. The expression

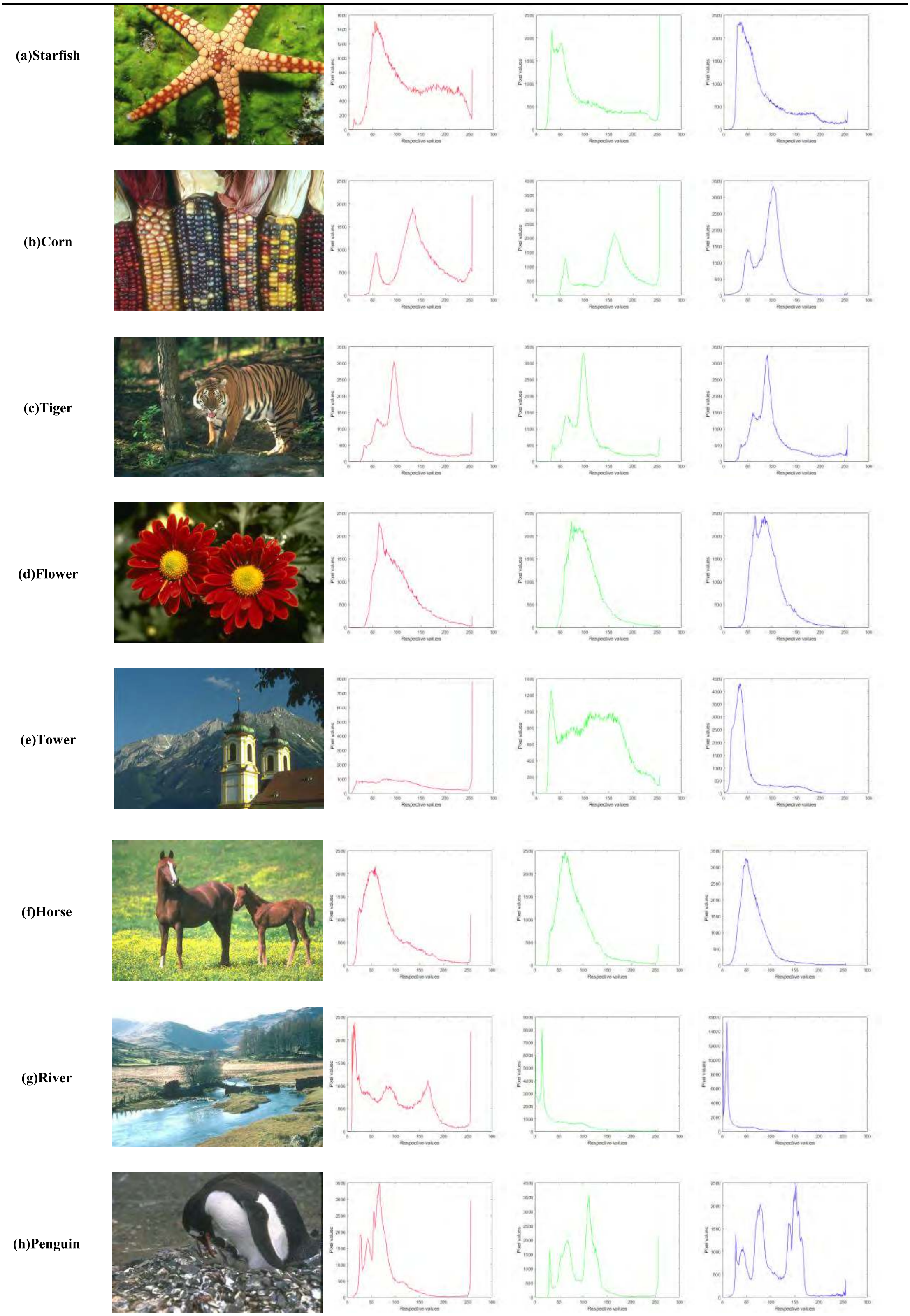


FIGURE 5. Original images and RGB three channels.

of WEP is as follows:

$$WEP = \min + l \times \left(\frac{\max - \min}{L} \right) \tag{14}$$

where, min and max take the empirical value, min $=$ 0.2, max $=$ 1, l represents the current iteration times, and L represents the maximum iteration times. The expression of (i)Dog

(j)Wolf

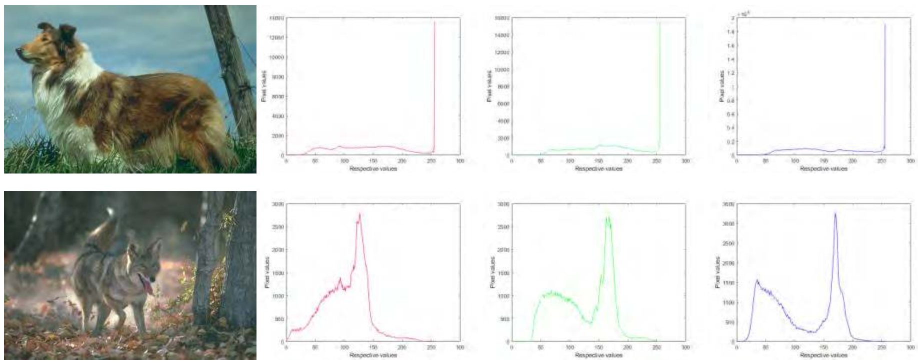


FIGURE 5. ***(Continued.)*** **Original images and RGB three channels.**

travel distance rate TDR is as follows:

$$TDR = 1 - \frac{l^{1/p}}{L^{1/p}} \tag{15}$$

Both TDR and WEP are coefficient and the relationship between them is shown in Fig 2. Where, p denotes the accuracy of mining capability. The iteration process of (13) is applied to the update of optimal particles in universe, while (12) corresponds to the iterative update of optimal universe (optimal solution). These two update mechanisms respectively correspond to local and global optimization and are triggered by their unique probability factors. $NI\ (U)$ is the inflation rate of universe, then it is the optimal fitness function value in the multithreshold segmentation problem.

IV. THE GRAVITATIONAL SEARCH ALGORITHM

A. BASIC CONCEPTS

GSA is a heuristic optimization algorithm proposed by Rashedi et al., in 2009. It originates from the swarm intelligence optimization algorithm generated by simulating the gravity in physics. The principle is that the search particles are regarded as a group of objects running in space, and the objects are attracted to each other by the universal gravitation. The movement of objects follows the laws of dynamics. The larger the fitness function value is, the larger the inertial mass will be. Therefore, gravity will urge the objects to move towards the object with the largest mass, so as to gradually find the optimal solution to the optimization problem [41], [42].

B. MATHEMATICAL MODEL

During each iteration, the object updates its speed and position according to the following formula [32]:

$$v_i^{t+1} = rand_i \times v_i^t + a_i^t \tag{16}$$

$$x_i^{t+1} = x_i^t + v_i^{t+1} \tag{17}$$

In the ith agent at iteration t.

Where, v_i^t is velocity which determined by the gravitational forces exerted by its neighbors according to the law of gravity, x_i^t represents the current position, and a_i^t denotes the acceleration. $rand_i$ is a random number in the interval [0,1].

$$a_i^t = \frac{F_i^t}{M_{ii}^t} \tag{18}$$

where F_i^t indicates the total force which is the sum of all other agents acting on it, M_{ii}^t is the inertial mass which is updated based on fitness values:

$$F_i^t = \sum_{j=1, j\neq i}^{N} rand_j \times F_{ij}^t, \tag{19}$$

$$m_i(t) = \frac{fitness_i(t) - worst(t)}{best(t) - worst(t)} \tag{20}$$

$$M_i(t) = \frac{m_i(t)}{\sum_{j=1}^{n} m_j(t)} \tag{21}$$

For a maximization problem, $worst(t)$ and $best(t)$ are defined as follows:

$$worst(t) = \min_{j\in\{1,\ldots,N\}} fit_j(t) \tag{22}$$

$$best(t) = \max_{j\in\{1,\ldots,N\}} fit_j(t) \tag{23}$$

where F_{ij}^t represents the force acting on agent i from agent j at iteration t and $rand_j$ is a random number in the interval [0,1]. F_{ij}^t is shown as follows:

$$F_{ij}^t = G(t)\frac{M_{pi}^t \times M_{aj}^t}{R_{ij}^t + \varepsilon}\left(x_j^t - x_i^t\right) \tag{24}$$

Respectively, M_{pi} is the passive gravitational masses and M_{aj} is the active gravitational masses. ε is a small constant. $G\,(t)$ stands for universal gravitation changing with time. R_{ij}^t is the Euclidean distance between two agents x_i and x_j:

$$G(t) = G_0 \times e^{-at/T} \tag{25}$$

$$R_{ij}(t) = \left\|X_i(t), X_j(t)\right\|_2 \tag{26}$$

where G_0 is a constant and the empirical value is 100, a = 20, t is the current generation, and T is the maximum number of iterations.

Gravitational search algorithm has a great advantage, through its gravitational interaction between the object to realize the optimization of information sharing. No environmental factors also can perceive the global information, guide the group to the optimal solution mining area, rather

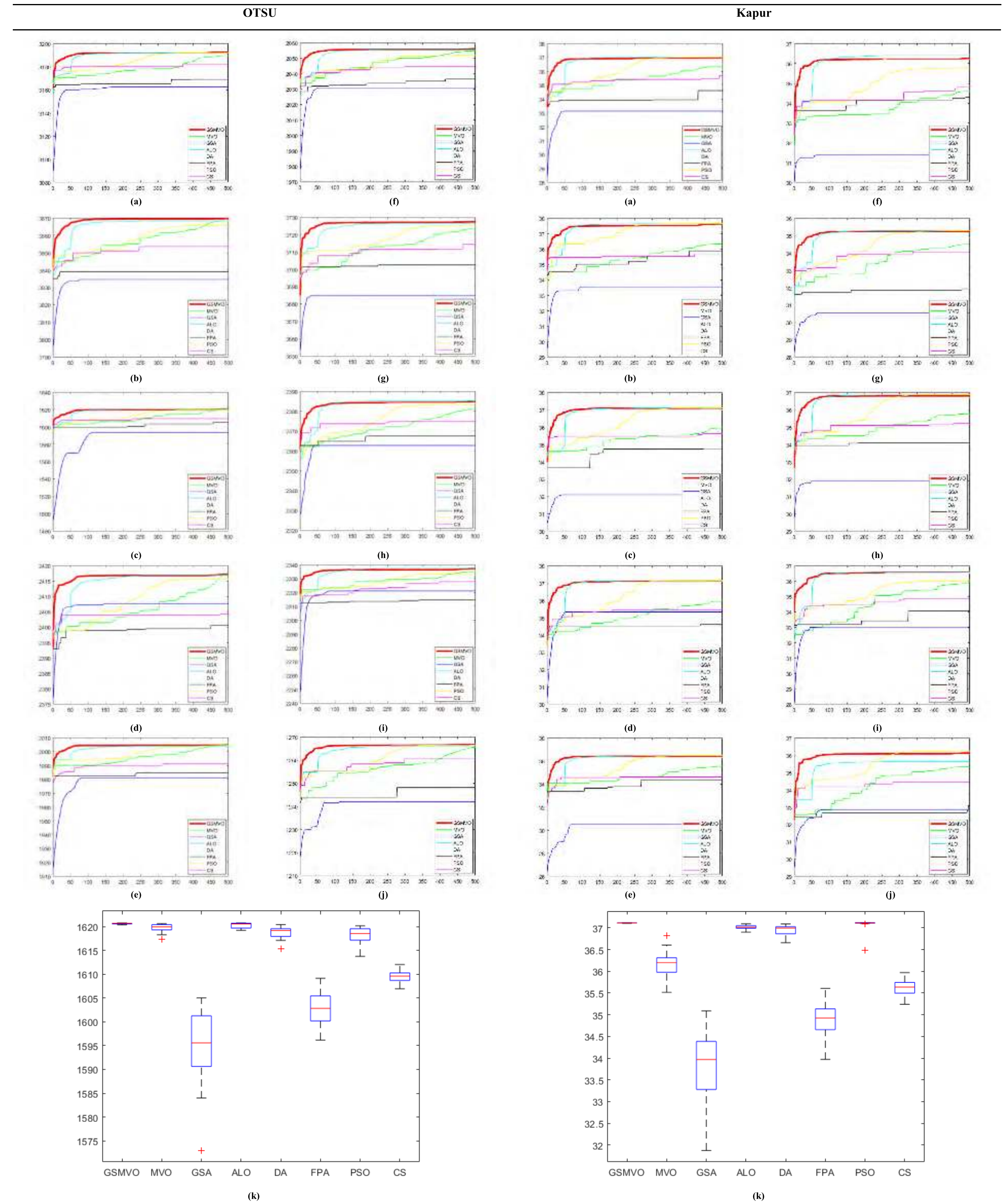


FIGURE 6. The fitness function curves and box charts obtained by GSMVO method.

than the other swarm algorithm (such as ant colony algorithm) by external environment factors are needed to perceive optimization. The gravity effect between objects is shown in Fig 3.

V. THE GSMVO ALGORITHM

A. BASIC CONCEPTS

Both MVO and GSA belong to the optimization algorithm based on physics. In the basic concept, objects in the universe

TABLE 8. **The PSNR of each algorithm under Kapur.**

TEST IMAGES	K	GSMVO	MVO	GSA	ALO	DA	FPA	PSO	CS
Starfish	4	18.2670	18.2358	18.8869	18.2358	18.2346	**19.0478**	18.2358	18.0264
	6	**20.5110**	20.2827	20.2471	20.1914	20.2156	20.2370	20.1946	19.7078
	8	**24.2179**	23.4211	20.7425	22.0779	23.3209	20.4658	23.2085	20.9865
	10	**26.3321**	25.0481	22.5085	24.7806	24.6296	23.9327	24.5652	23.3168
	12	**26.8813**	26.4828	24.1607	26.7306	25.7703	24.4965	26.7755	25.0777
Corn	4	**18.2670**	18.2357	17.7302	18.2357	18.2357	17.1761	18.2357	17.7302
	6	**20.4183**	20.1843	19.7705	20.1843	20.1933	20.2767	20.1978	19.7705
	8	**23.9204**	23.3037	23.1401	23.3037	21.8837	21.8864	21.9837	23.1401
	10	**26.0422**	24.8925	24.1125	24.8925	24.5467	22.7085	24.9165	24.1125
	12	**27.2936**	26.6496	23.8971	26.6496	26.7404	22.1828	26.7097	23.8971
Tiger	4	16.2736	16.2737	**16.9521**	16.2737	16.1272	14.3968	16.2737	15.2960
	6	**21.0837**	19.6450	20.9658	19.3973	19.6478	19.2279	19.3973	16.2353
	8	**24.4083**	23.2013	21.6459	22.5946	21.7234	22.1109	22.4047	20.9859
	10	**27.0711**	25.0945	23.2576	24.8732	23.8266	23.3113	24.3459	22.1040
	12	**28.1152**	26.9604	23.3142	26.7401	25.1206	25.2716	26.625	24.1944
Flower	4	21.4066	21.4046	20.8207	21.4046	21.4262	20.9907	21.4046	**21.5833**
	6	**24.1657**	24.0717	22.5897	24.0082	24.0218	23.3139	23.9927	23.0275
	8	**26.2691**	25.8739	24.6119	25.3663	25.2631	24.7726	25.3786	24.9535
	10	**27.7051**	27.1283	25.1330	27.0268	26.6337	24.3966	27.0001	25.6754
	12	**29.1949**	28.4692	28.4047	27.9570	27.7013	25.0970	27.9838	25.9418
Tower	4	17.7270	14.6605	**18.2650**	15.7299	15.7851	16.8267	14.6053	14.4052
	6	**21.2854**	19.7583	20.3589	19.1613	19.1411	17.8023	19.7510	20.8011
	8	**23.4873**	22.3460	18.6727	22.2113	19.9948	21.5577	22.2580	20.3627
	10	**25.6729**	25.5665	22.6363	24.5105	23.6406	20.5044	24.8870	24.7053
	12	**27.5286**	26.529	24.4018	25.9651	24.7830	23.0770	26.5728	22.8889
Horse	4	18.5501	18.2656	16.9496	18.2656	18.2656	18.2721	18.2656	**18.5858**
	6	**21.1532**	20.2070	18.0505	21.0793	20.2862	20.5480	20.2142	20.8933
	8	22.7045	22.2865	20.2196	22.2203	21.9634	20.7200	**23.1585**	21.3136
	10	25.1588	24.5801	24.7764	**25.9745**	24.2018	24.6881	24.4731	23.7671
	12	**27.4850**	25.8010	22.6250	25.7900	25.1383	24.2354	25.8908	22.9738
River	4	**17.1680**	16.4956	14.9493	16.4972	16.4972	**17.1680**	16.4972	16.5300
	6	**19.5782**	19.4958	16.7807	19.5291	19.2959	18.7488	19.5456	18.9096
	8	18.9487	21.7866	19.9403	21.8941	**22.0349**	19.8551	21.8084	21.2651
	10	**24.8426**	23.8564	19.6909	23.8958	21.5805	23.1399	24.1884	21.4485
	12	**26.5540**	25.4462	19.7167	26.5456	25.2791	24.1482	22.7387	25.1542
Penguin	4	17.2275	17.2228	**17.5651**	17.2228	17.1853	16.9090	17.2228	16.8054
	6	**22.0514**	21.9447	17.0441	21.1680	21.9393	18.9333	21.1718	20.9567
	8	**24.2630**	23.1296	20.9998	23.0639	23.1069	21.8359	23.0653	21.2843
	10	**25.8115**	25.2638	20.8513	25.0646	23.5876	21.6849	25.1730	24.6145
	12	**27.0422**	27.0265	25.0858	26.2059	25.6188	23.8631	25.7348	22.5593
Dog	4	18.3119	18.2605	17.9687	18.2605	**18.3708**	17.2014	18.2605	17.8688
	6	**21.8033**	20.6201	20.5754	21.7993	20.7421	19.2313	21.2494	19.5075
	8	**24.2840**	22.8203	21.0187	22.8940	22.3929	20.8185	22.6189	23.3172
	10	**26.3671**	25.0767	23.8880	25.3739	24.9025	23.8890	25.3617	25.1570
	12	**27.4536**	27.3486	25.2089	26.9742	26.3678	24.3897	26.7545	21.6394
Wolf	4	14.6796	14.3137	**17.4792**	14.3915	14.1280	14.3461	14.2481	14.1135
	6	19.5229	17.3456	**21.4221**	17.1925	17.2660	17.6070	17.2786	16.4433
	8	**25.6553**	22.1784	20.3816	21.8277	21.1046	22.6447	21.5599	17.4667
	10	**27.2347**	26.7578	24.0309	25.0611	24.4266	23.8742	25.8775	22.6413
	12	**28.7625**	28.1649	25.3589	27.4002	25.6077	21.4687	27.2897	23.0239

will be affected by the gravitational force. Therefore, the idea of gravity is added to the process of universe renewal (local optimization). Let the particles in the universe move towards the target with higher mass, that is, update to the optimal solution. The speed of local convergence is accelerated and the precision of optimization can be improved. In traditional MVO algorithm, the update of each object in the universe is just a simple position update without a proper screening mechanism, and the slow update of the optimal universe affects the local optimization of the algorithm. For these reasons, the universe is reformed through GSA, the algorithm will optimize each universe and improve the ability of local update. In each iteration, we ranked them according to the inflation rate of the universe, and selected one of them with a white hole through a roulette wheel. The discovery means that GSA is helpful for achieving a better balance of exploration and exploitation in MVO. In the two stages of algorithm: exploration versus exploitation, the white hole and the black hole explore the search space which is optimized by GSA, and the wormhole helps MVO to develop the search space under the action of roulette wheel.

In this case, the proposed local update method is:

$$x_i^{t+1} = x_{best} + v_i^{t+1} \tag{27}$$

where x_{best} represents the newly update global optimum. All the parameters except these few steps are same as the traditional algorithm. In theory, it is able to find a better solution than the traditional algorithm.

B. THE PROPOSED GSMVO-BASED MULTILEVEL THRESHOLDING METHOD

In this paper, the multilevel threshold method based on GSMVO algorithm is proposed, which takes the Otsu or Kapur's entropy as the objective function to find the optimal threshold of image segmentation. The universe represents search agents, and the threshold denotes the location of objects. Therefore, according to the number of thresholds, the matter in the universe moves through a wormhole to change the position vector in a one-dimensional, two-dimensional, or hyperdimensional space. At the beginning, randomly initialize the position of the universe. The fitness of the universe is determined by (3) or (9), match the GSA and black/white hole with (24) to update the position of the universe. Then the better position can be obtained by constraints such as roulette wheel selection and impulse presence rate. Iterate through the loop until the maximum number of iterations is completed. The optimal location of universe provides the desired threshold. In table 2, the pseudo-code of proposed multilevel threshold method is given.

C. THE FLOWCHART OF GSMVO (SEE FIGURE 4.)

VI. EXPERIMENTS

In this section, the performance of algorithm is evaluated through experiment. First of all, the experimental

TABLE 9. The PSNR of each algorithm under Otsu.

TEST IMAGES	K	GSMVO	MVO	GSA	ALO	DA	FPA	PSO	CS
Starfish	4	19.9176	19.9176	16.6212	19.9176	19.9186	19.1212	19.9176	**20.2145**
	6	**22.5633**	22.5577	21.8891	**22.5633**	22.5076	21.5765	**22.5633**	21.7567
	8	**24.7206**	24.3462	21.5436	24.6909	24.5779	21.7242	24.6907	24.0150
	10	**26.2557**	26.0127	23.7089	26.1622	25.6934	24.8769	26.2495	24.2545
	12	**27.5530**	27.2882	25.5847	27.2506	26.6119	25.3421	27.3900	26.8842
Corn	4	**17.5617**	**17.5617**	17.5138	**17.5617**	**17.5617**	16.3081	**17.5617**	17.2852
	6	**20.4797**	20.3557	19.6406	20.3580	20.3029	20.3555	20.3555	19.1508
	8	22.7422	**22.8139**	21.5856	22.5291	22.3722	21.9144	22.5242	21.6205
	10	**24.6699**	24.4278	23.5959	24.3252	23.7361	22.4022	24.0706	23.2929
	12	**26.5133**	26.3307	24.4351	26.0636	25.4260	23.2794	25.3172	23.0573
Tiger	4	**18.8825**	18.7987	18.1799	18.7987	18.7978	16.2364	18.3436	18.7987
	6	21.9612	21.9544	19.8708	21.9608	21.8363	**22.8249**	21.9577	19.5081
	8	**24.3247**	24.3151	23.0025	24.1038	23.8822	23.7236	24.1005	21.3769
	10	**26.2940**	26.1524	24.8950	25.8510	25.2804	23.7203	25.1491	24.9286
	12	27.6346	**27.6854**	26.5507	27.4465	26.0716	26.9492	27.4800	25.7327
Flower	4	22.6832	**22.6943**	21.6021	22.6832	22.6832	22.2830	22.6832	22.3485
	6	**25.5648**	24.9329	23.3238	**25.5648**	25.2606	23.7091	**25.5648**	24.2990
	8	**27.7220**	27.3733	24.9150	27.6614	27.7203	25.7995	27.7041	26.3383
	10	29.2816	28.9960	26.9412	**29.3984**	29.0890	26.9276	27.7840	27.4046
	12	**30.7465**	30.6518	28.9982	30.3783	29.8406	27.8348	30.1326	28.6507
Tower	4	**19.7307**	**19.7307**	18.1062	**19.7307**	**19.7307**	19.1713	**19.7307**	19.3720
	6	**22.5270**	22.4959	19.8820	22.4493	22.2424	21.1679	22.5039	21.1700
	8	**24.9762**	24.8428	21.2591	24.9087	24.9044	23.6393	24.2520	23.6844
	10	**26.3014**	26.1024	23.4197	26.1923	25.9674	24.7590	25.8467	24.2832
	12	**27.4944**	27.4192	25.1052	26.7937	26.5942	25.4262	26.8721	24.6714
Horse	4	**18.3751**	**18.3751**	17.5019	**18.3751**	18.3133	17.9032	**18.3751**	18.2281
	6	20.3162	20.3145	20.2899	20.2796	20.2675	20.2020	20.3060	**20.4316**
	8	**23.2658**	23.1861	18.4494	22.8184	22.9899	22.3329	22.0857	22.3351
	10	24.5428	24.4737	22.9947	**25.4132**	24.5244	21.8601	24.6638	23.3104
	12	**26.5363**	25.9240	25.2307	26.0970	26.2870	25.0074	25.1040	24.7340
River	4	**16.9661**	16.5803	14.6000	16.5803	16.5803	15.0768	16.5803	**16.9661**
	6	19.2104	**19.2578**	17.6594	19.2104	18.9086	18.2628	18.8517	17.9147
	8	**21.9603**	21.6770	21.3089	21.1463	21.2536	20.9558	21.1917	20.2744
	10	**23.3403**	22.9256	21.5167	22.5031	22.1457	22.0058	22.2687	22.7373
	12	24.6529	24.5072	23.3549	**25.0997**	24.6075	23.6928	23.6012	22.8758
Penguin	4	17.3192	17.3192	16.0988	17.3192	17.3144	16.4130	17.3192	**17.3461**
	6	**21.3508**	21.3459	18.9579	21.3494	21.2841	18.8336	**21.3508**	20.2347
	8	**23.7534**	23.1889	23.0551	23.5561	23.6033	21.1475	23.6090	22.0659
	10	25.3471	**25.3707**	21.9719	25.2851	24.2947	23.9748	25.2351	22.1646
	12	**26.8174**	26.3575	25.8671	26.369	25.5738	24.8789	26.6652	25.2647
Dog	4	**19.9346**	**19.9346**	17.8827	**19.9346**	**19.9346**	19.1474	**19.9346**	19.5027
	6	22.6622	**22.7193**	21.3925	22.6099	22.3531	21.3441	22.1130	21.4339
	8	**24.4071**	24.2646	22.3894	24.3341	24.0582	23.6097	24.3992	23.9708
	10	**26.3928**	26.0498	24.9187	26.2847	25.3812	25.1234	26.0019	25.4317
	12	**27.9728**	27.8542	26.5759	27.8590	26.6262	24.8294	26.5853	25.9879
Wolf	4	16.7945	16.8438	**18.8286**	16.7945	16.7926	16.2486	16.7945	16.7126
	6	**20.2130**	19.9155	**20.2130**	19.7916	19.5523	19.0401	19.792	19.9528
	8	**24.4954**	21.8300	22.6288	22.5402	21.4569	20.2768	21.5752	**24.4954**
	10	**25.5014**	23.9653	25.3470	23.0074	22.6863	23.5706	22.4766	22.3352
	12	**26.9579**	24.8074	26.3784	24.1120	23.2300	26.1674	24.0246	25.4100

environment is explained in section ***A*** and the compared algorithm is briefly introduced. The test images and the parameters of relevant algorithm will be respectively selected and set in section ***B***. In the section ***C***, six kinds of image segmentation quality measures are introduced which used to verify the segmentation performance. Finally, the data of experimental results are sorted and analyzed in section ***D***, meanwhile, statistic analysis and convergence performance are also given.

A. EXPERIMENTAL SETUP

All the algorithms in the experiment were run in Matlab R2017b, and the computer was configured as Intel (R) Pentium (R) CPU G4560 @3.50 GHz, Microsoft Windows 7 system. Due to the difference in the search strategies and mathematical formulation, all algorithms have various optimization performance. In order to prove the superiority of improved algorithm in image segmentation, a total of seven meta-heuristics algorithms are selected for comparison experiments. Each of them contains different characteristics. Including the traditional MVO algorithm; The traditional GSA algorithm; An interesting bionic algorithm named ant lion algorithm (ALO) which can always finds the maximum in the latest meta-heuristic algorithm [43]; A new complex swarm intelligent optimization technology, dragonfly algorithm (DA) [44]; FPA, inspired by the process of flower pollination of flowering plants in nature which is simple and requires fewer parameters to be adjusted [45], [46]; An earlier proposed evolutionary algorithm, PSO [47]–[50]; CS which is based on the brood parasitism of some cuckoo species, along with Levy flights random walks [51], [52]. The eight algorithms correspond to four relatively novel algorithms and four relatively basic algorithms respectively. These comparison algorithms are representative algorithms of multilevel thresholding, and the algorithm parameters involved are directly chosen from these references. These references are all the applications of corresponding algorithms in the field of image segmentation.

B. PARAMETER SETTING

According to the references of MVO, GSA, ALO, DA, FPA, PSO, CS the main parameters of all algorithms are set as shown in table 3. For comparison experiments, control variable method is adopted. Maximum number of iterations of all algorithms is 500 and the number of population size is 25. A total of 10 color test images were carefully selected from the Berkeley Segmentation Data Set and Benchmarks 500 (BSDS500) [53] [54], Starfish, Corn, Tiger, Flower, Tower, Horse, River, Penguin, Dog and Wolf.

The size of each image is 481 × 321, and both the original images and three channels of RGB are shown in Fig 5. The RGB image indicates the approximate number of thresholds and complexity of the selected image.

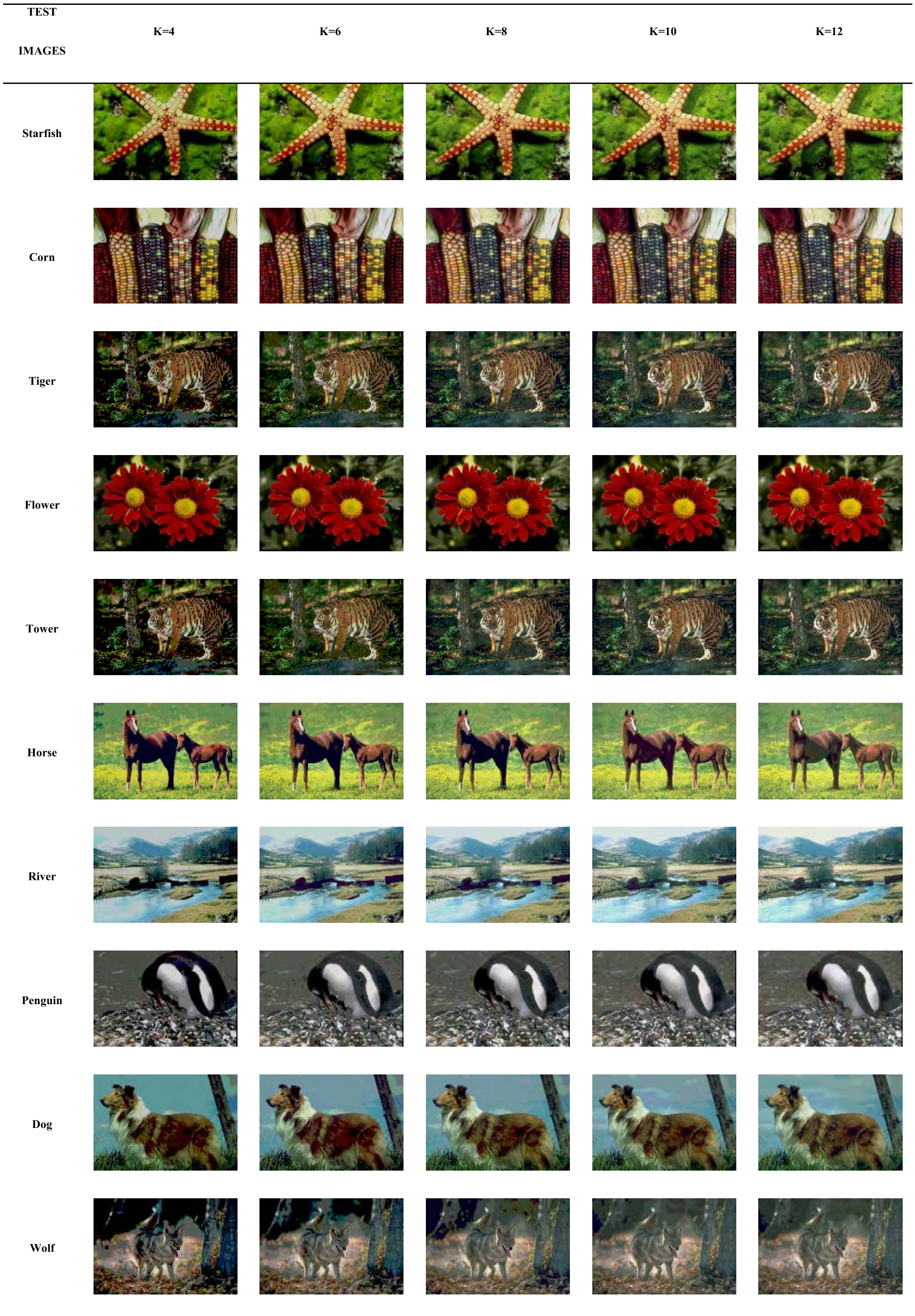


FIGURE 7. **The segmentation images obtained by GSMVO-Kapur method.**

The optimal threshold search of complex images is closer to the real search space, which is beneficial to benchmark the exploration and exploitation of the algorithm at the same time. In order to have a fair comparison, each algorithm runs each image 30 times independently. The threshold number of K includes: K = 4, 6, 8, 10, 12.

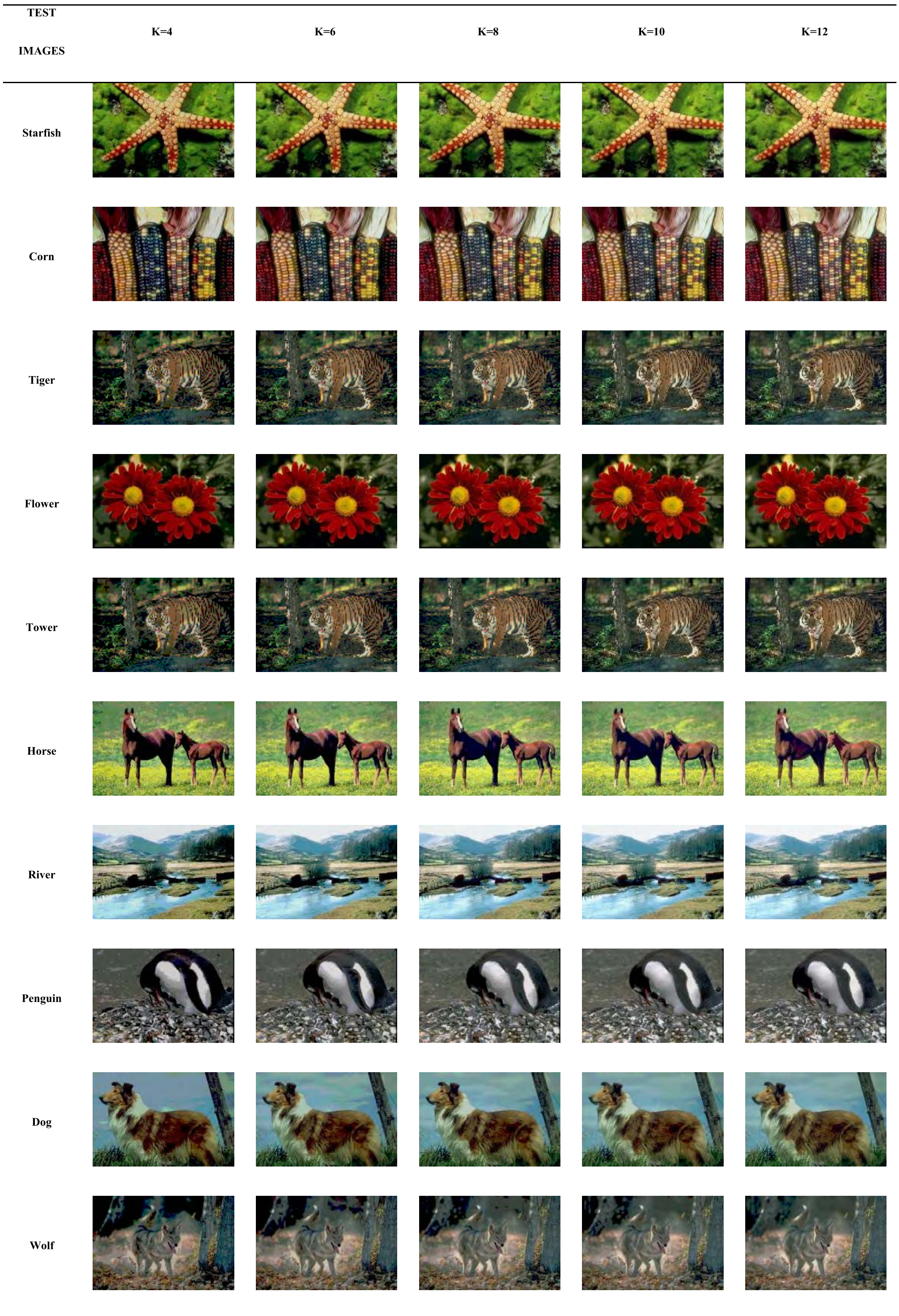


FIGURE 8. The segmentation images obtained by GSMVO-Otsu method.

C. THE TIME COMPLEXITY OF GSMVO ALGORITHM

For an excellent swarm intelligence optimization algorithm, on the one hand, its performance is required to be better, on the other hand, its time complexity is required to be low. The running time is mainly used to evaluate the fitness value of the population individuals, that is, the number of iterations needed by the algorithm to find the optimal solution or approach the optimal solution of

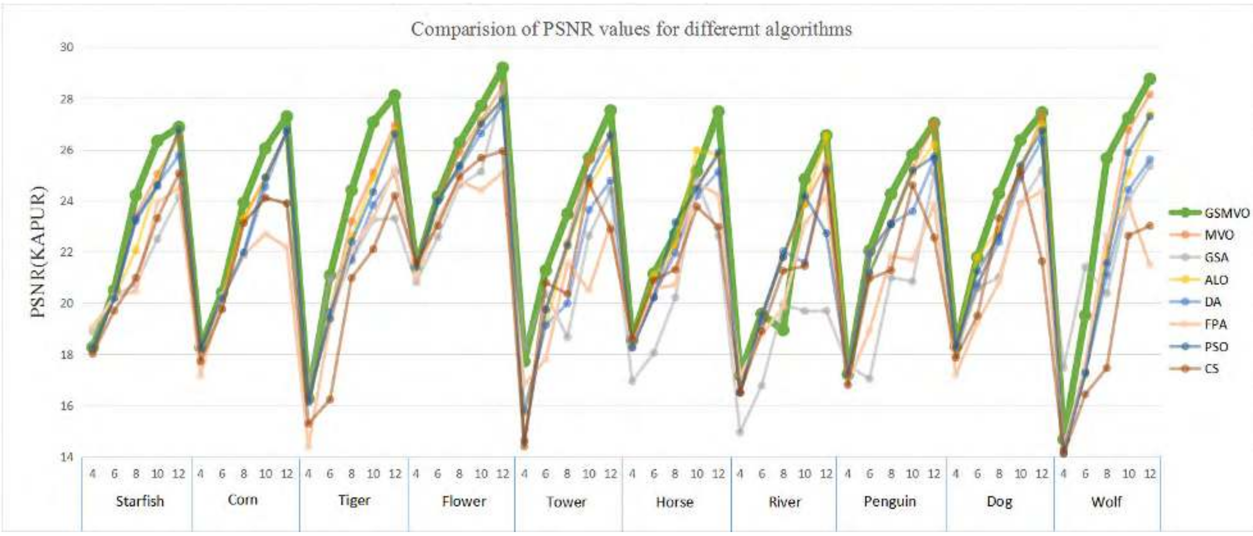


FIGURE 9. The PSNR obtained by GSMVO-Kapur method.

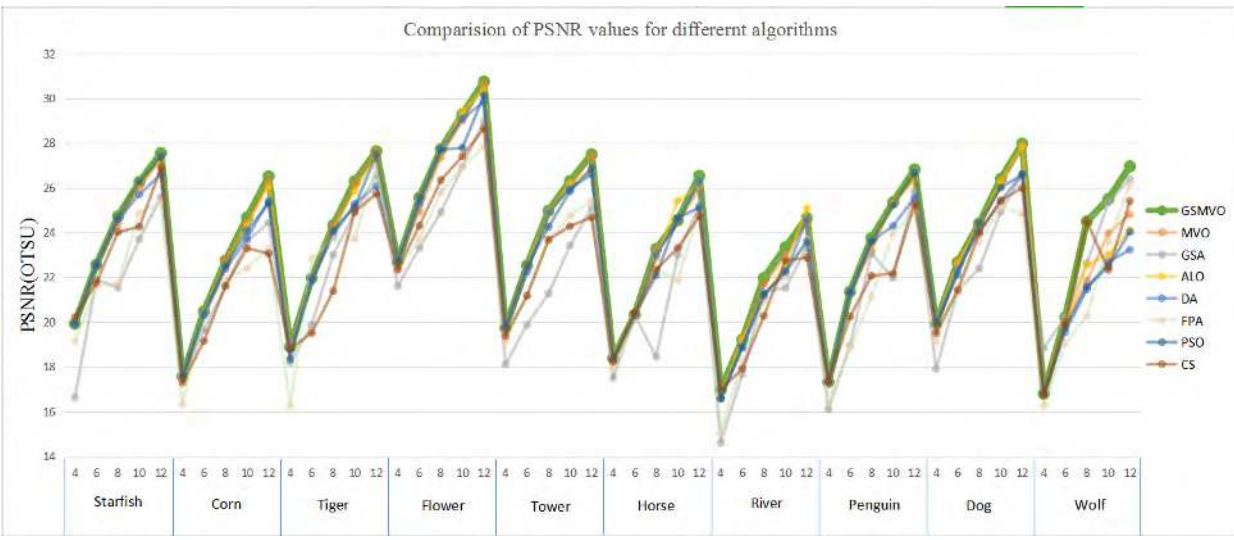


FIGURE 10. The PSNR obtained by GSMVO-Otsu method.

the problem. Assuming that the optimization problem function is f(x) and the dimension of the solution space is d, then according to the description of MVO algorithm and the operation rule of time complexity symbol O, the time complexity of MVO is T (MVO) = O(d + f (d)). GSMVO includes operations to calculate the coefficient of gravity, inertial mass, the sum of gravity, accelerated velocity, and basic operations of MVO. According to the detailed description of these operations above and the operation rule of the time complexity symbol O, the time complexity T of each operation can be derived. T(calculating universal gravitational coefficient operation)=O(1); T(calculating inertial mass)=O(d+f(d)); T(calculating the sum of the gravitational forces) =O(d+f(d)); T(calculating acceleration) =O(d+f(d)). Therefore, T(GSMVO)=T(MVO) + (calculating universal gravitational coefficient operation)+ T(calculating inertial mass) +T(calculating the sum of the gravitational forces) +T(calculating acceleration). After simplification, the time complexity of the GSMVO algorithm is also O(d + f (d)), which is of the same order of magnitude as the time complexity of the MVO algorithm. Indicating that the time complexity of the GSMVO algorithm is not greatly improved.

D. IMAGE SEGMENTATION QUALITY RESULTS AND ANALYSIS

(1) In the last part, we roughly deduced that the time complexity of the improved algorithm is the same as that of the original algorithm through mathematical formula calculation. Next, we will make statistics on the average CPU time (in seconds) of all algorithms through experiments.

Algorithm execution time, the average execution time of each algorithm running 30 times independently can also reflect its computational complexity directly. The shorter the

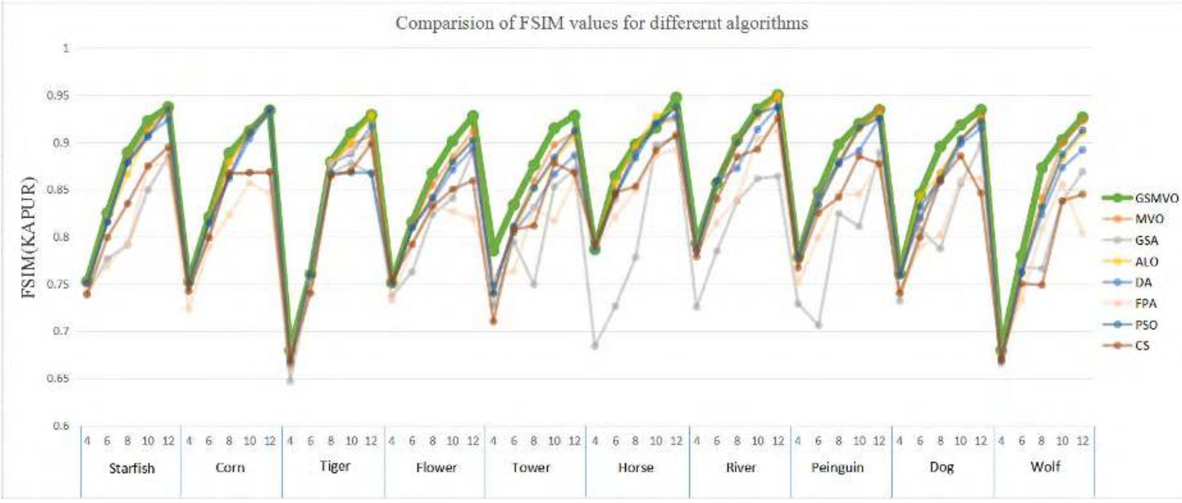


FIGURE 11. The FSIM obtained by GSMVO-Kapur method.

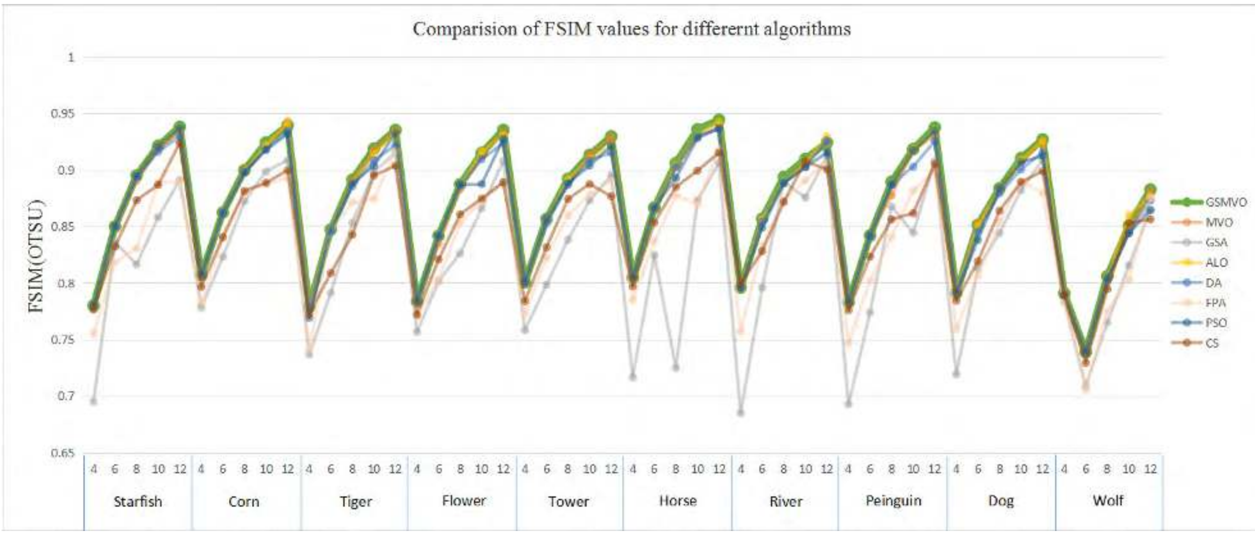


FIGURE 12. The FSIM obtained by GSMVO-Otsu method.

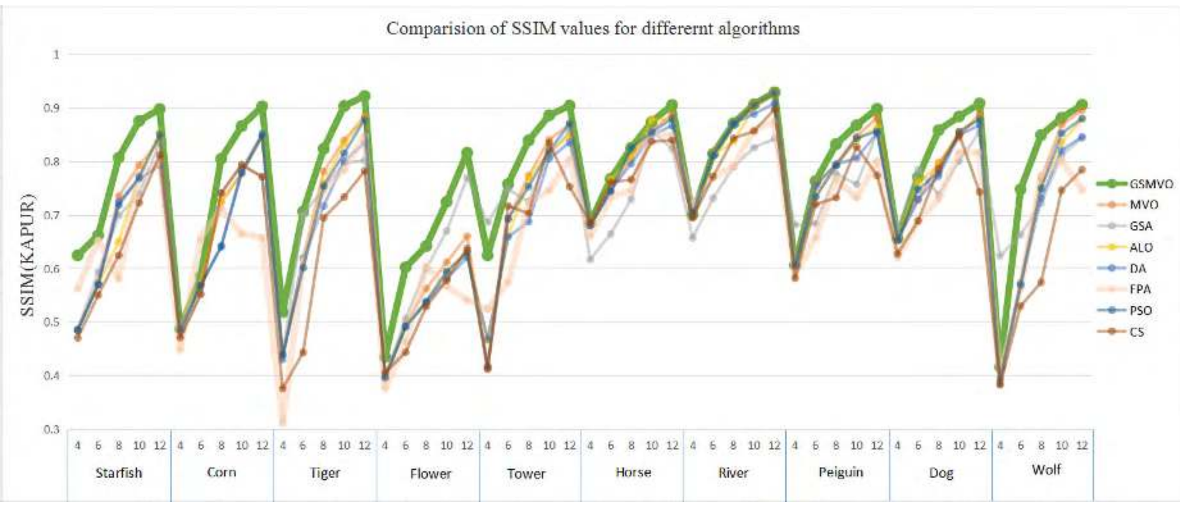


FIGURE 13. The SSIM obtained by GSMVO-Kapur method.

time, the faster the algorithm. Average CPU time of 10 images is shown in table 4 and 5 respectively, the shortest time is marked in bold. With the increase of threshold levels, the CPU time increases due to the computational complexity of algorithms. However, the update structure of MVO is more efficient, so the computational amount is less and

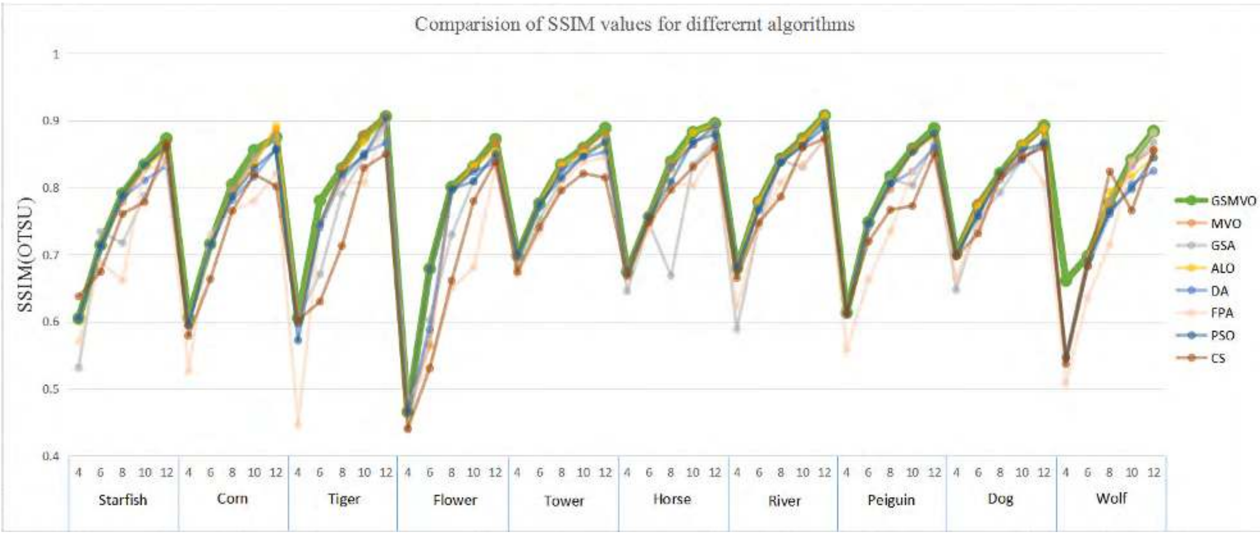


FIGURE 14. The SSIM obtained by GSMVO-Otsu method.

TABLE 10. The SSIM of each algorithm under Kapur.

TEST IMAGES	K	GSMVO	MVO	GSA	ALO	DA	FPA	PSO	CS
Starfish	4	**0.6249**	0.4849	0.4874	0.4849	0.4849	0.5624	0.4849	0.4705
	6	**0.6624**	0.5740	0.5931	0.5710	0.5705	0.6526	0.5699	0.5507
	8	**0.8069**	0.7352	0.6990	0.6505	0.7243	0.5818	0.7184	0.6246
	10	**0.8758**	0.7933	0.7390	0.7743	0.7675	0.7742	0.7707	0.7231
	12	**0.8980**	0.8388	0.8362	0.8522	0.7933	0.7987	0.8507	0.8123
Corn	4	**0.4874**	0.4849	0.4714	0.4849	0.4849	0.4483	0.4849	0.4714
	6	0.5865	0.5706	0.5524	0.5706	0.5688	**0.6570**	0.5692	0.5524
	8	**0.8050**	0.7242	0.7411	0.7242	0.6404	0.7116	0.6422	0.7411
	10	**0.8661**	0.7783	0.7936	0.7783	0.7841	0.6652	0.7786	0.7936
	12	**0.9030**	0.8472	0.7716	0.8472	0.8518	0.6583	0.8486	0.7716
Tiger	4	**0.5187**	0.4392	0.4393	0.4392	0.4304	0.3118	0.4392	0.3763
	6	**0.7066**	0.6180	0.7002	0.6019	0.6201	0.6138	0.6019	0.4437
	8	**0.8237**	0.7815	0.7495	0.7599	0.7169	0.7459	0.7542	0.6949
	10	**0.9036**	0.8395	0.7968	0.8325	0.8004	0.7849	0.8155	0.7338
	12	**0.9222**	0.8844	0.8022	0.8815	0.8355	0.8483	0.8784	0.7823
Flower	4	**0.4352**	0.3976	0.3992	0.3976	0.3948	0.3769	0.3976	0.4057
	6	**0.6025**	0.4961	0.5072	0.4923	0.4916	0.4630	0.4917	0.4438
	8	**0.6420**	0.5625	0.5995	0.5372	0.5360	0.6049	0.5387	0.5290
	10	**0.7247**	0.6116	0.6702	0.5934	0.5813	0.5672	0.5946	0.5781
	12	**0.8164**	0.6608	0.7700	0.6286	0.6214	0.5418	0.6310	0.6379
Tower	4	0.6249	0.4190	**0.6864**	0.4653	0.4685	0.5240	0.4157	0.4124
	6	**0.7579**	0.6924	0.7484	0.6599	0.6593	0.5753	0.6932	0.7170
	8	**0.8392**	0.7733	0.7255	0.7689	0.6876	0.7225	0.7532	0.7031
	10	**0.8862**	0.8407	0.8137	0.8177	0.8053	0.7459	0.8193	0.8350
	12	**0.9045**	0.8692	0.8556	0.8473	0.8356	0.8050	0.8713	0.7530
Horse	4	0.6850	0.6805	0.6167	0.6805	0.6805	0.6605	0.6805	**0.6855**
	6	**0.7676**	0.7438	0.6649	0.7663	0.7462	0.7333	0.7442	0.7613
	8	**0.8255**	0.8073	0.7296	0.8055	0.7958	0.7458	**0.8255**	0.7660
	10	0.8739	0.8570	0.8576	**0.8783**	0.8487	0.8540	0.8544	0.8371
	12	**0.9057**	0.8894	0.8239	0.8789	0.8670	0.8481	0.8801	0.8395
River	4	0.7009	0.7015	0.6575	0.7015	0.7015	**0.7197**	0.7015	0.6949
	6	0.8145	**0.8159**	0.7315	0.8119	0.8117	0.7701	0.8115	0.7722
	8	**0.8717**	0.8707	0.7901	0.8377	0.8708	0.7920	0.8705	0.8433
	10	**0.9070**	0.9066	0.8261	0.9017	0.8889	0.8596	0.9046	0.8567
	12	**0.9292**	0.9259	0.8428	**0.9292**	0.9087	0.8740	0.9287	0.8977
Penguin	4	0.6062	0.6059	**0.6817**	0.6059	0.6037	0.5901	0.6059	0.5822
	6	**0.7633**	0.7565	0.6839	0.7338	0.7549	0.6577	0.7338	0.7200
	8	**0.8325**	0.7957	0.7770	0.7921	0.7943	0.7677	0.7922	0.7323
	10	**0.8679**	0.8470	0.7573	0.8404	0.8065	0.7325	0.8436	0.8272
	12	**0.8986**	0.8817	0.8535	0.8639	0.8529	0.8009	0.8565	0.7740
Dog	4	0.6565	0.6534	**0.6752**	0.6534	0.6534	0.6213	0.6534	0.6277
	6	0.7704	0.7279	**0.7844**	0.7621	0.7299	0.6881	0.7479	0.6893
	8	**0.8583**	0.7981	0.7387	0.7922	0.7726	0.7298	0.7795	0.7872
	10	**0.8840**	0.8443	0.8010	0.8561	0.8485	0.8168	0.8556	0.8509
	12	**0.9083**	0.8900	0.8552	0.8808	0.8684	0.8171	0.8792	0.7429
Wolf	4	0.4164	0.3956	**0.6237**	0.4006	0.3848	0.3950	0.3918	0.3831
	6	**0.7482**	0.5732	0.6631	0.5647	0.5686	0.5639	0.5700	0.5299
	8	**0.8493**	0.7678	0.7217	0.7528	0.7314	0.7741	0.7498	0.5743
	10	**0.8815**	0.8685	0.8117	0.8358	0.8213	0.8022	0.8526	0.7463
	12	**0.9059**	0.8959	0.8439	0.8791	0.8460	0.7470	0.8805	0.7846

the computational time is shorter. GSMVO inherits all the advantages of MVO. Adding GSA mechanism in the local search process can improve the search efficiency, but relatively increase the running time of the algorithm. In terms of computing time, put all the statistics together and sort them, the order of the algorithm is MVO< GSMVO< CS< FPA< PSO< GSA< DA< ALO. In addition, compared with the CPU time shown in table 4 and table 5, the Kapur's entropy based method is relatively faster than the Otsu based method. The experimental results show that GSMVO can segment effectively in a short time under the condition of maintaining high segmentation accuracy.

(2) Fitness function value of the experiment. The amount of information contained in the segmented image and the good or bad segmentation effect depend on the size of the target function value.

TABLE 11. The SSIM of each algorithm under Otsu.

TEST IMAGES	K	GSMVO	MVO	GSA	ALO	DA	FPA	PSO	CS
	4	0.6049	0.6049	0.5313	0.6049	0.6049	0.5694	0.6049	**0.6378**
	6	0.7145	0.7147	**0.7341**	0.7142	0.7112	0.6884	0.7142	0.6746
Starfish	**8**	**0.7915**	0.7786	0.7173	0.7906	0.7847	0.6616	0.7906	0.7610
	10	**0.8344**	0.8290	0.7876	0.8281	0.8109	0.8275	0.8331	0.7786
	12	**0.8731**	0.8645	0.8418	0.8554	0.8325	0.8340	0.8599	0.8663
	4	**0.6063**	0.5950	0.5950	0.5950	0.5950	0.5262	0.5950	0.5792
	6	0.7161	0.7161	0.7125	0.7162	0.7132	**0.7302**	0.7161	0.6636
Corn	**8**	**0.8046**	0.7965	0.7935	0.7875	0.7816	0.7661	0.7877	0.7652
	10	**0.8556**	0.839	0.8456	0.8342	0.8165	0.7801	0.8293	0.8203
	12	0.8755	0.8875	0.8740	**0.8919**	0.8582	0.8217	0.8566	0.8020
	4	**0.6056**	0.5980	**0.6056**	0.5980	0.5978	0.4461	0.5723	0.6020
	6	**0.7807**	0.7464	0.6714	0.7464	0.7410	0.7465	0.7462	0.6300
Tiger	**8**	**0.8290**	0.8289	0.7899	0.8214	0.8113	0.8085	0.8214	0.7131
	10	0.8766	**0.8797**	0.8461	0.8677	0.8509	0.8069	0.8477	0.8294
	12	**0.9061**	0.9059	0.8928	0.9019	0.8669	0.897	0.9041	0.8502
	4	0.4655	0.4659	0.4384	0.4655	0.4655	**0.4838**	0.4655	0.4408
	6	**0.6791**	0.5650	0.6027	**0.6791**	0.5880	0.5318	**0.6791**	0.5304
Flower	**8**	**0.8010**	0.7970	0.7296	0.7954	0.7956	0.6496	0.7983	0.6611
	10	0.8314	0.8305	0.8100	**0.8329**	0.8237	0.6807	0.8092	0.7801
	12	**0.8723**	0.8681	0.8519	0.8542	0.8370	0.8302	0.8499	0.8390
	4	**0.6987**	**0.6987**	0.6826	**0.6987**	**0.6987**	0.6712	**0.6987**	0.6747
	6	**0.7763**	0.7761	0.7523	0.7731	0.7708	0.7330	0.7762	0.7411
Tower	**8**	**0.8344**	0.8323	0.8117	**0.8344**	0.8264	0.8075	0.8148	0.7958
	10	**0.8591**	0.8590	0.8513	0.8516	0.8455	0.8382	0.8465	0.8210
	12	**0.8888**	0.8801	0.8714	0.8645	0.8531	0.8454	0.8681	0.8151
	4	0.6748	0.6748	0.6453	0.6748	0.6727	**0.6754**	0.6748	0.6681
	6	**0.7563**	0.7561	0.7469	**0.7563**	0.7551	0.7372	0.7562	0.7485
Horse	**8**	**0.8390**	0.8357	0.6687	0.8265	0.8301	0.8073	0.8089	0.7967
	10	**0.8828**	0.8642	0.8344	**0.8828**	0.8647	0.8029	0.8693	0.8304
	12	**0.8958**	0.8896	0.8685	0.8943	0.8932	0.8576	0.8792	0.8602
	4	**0.6797**	**0.6797**	0.5886	**0.6797**	**0.6797**	0.6210	**0.6797**	0.6652
	6	0.7784	**0.7796**	0.7435	0.7784	0.7700	0.7410	0.7662	0.7478
River	**8**	**0.8431**	0.8427	0.8400	0.8362	0.8378	0.8083	0.8373	0.7862
	10	**0.8735**	0.8730	0.8310	0.8693	0.8624	0.8349	0.8627	0.8594
	12	**0.9076**	0.9061	0.8732	**0.9076**	0.8988	0.8691	0.8896	0.8732
	4	**0.6135**	0.6118	0.6100	0.6118	0.6117	0.5585	0.6118	**0.6135**
	6	**0.7484**	0.7481	0.7402	0.7483	0.7453	0.6636	0.7481	0.7195
Penguin	**8**	**0.8166**	0.7973	0.8123	0.8066	0.8053	0.7350	0.8065	0.7670
	10	**0.8580**	0.8579	0.8034	0.8530	0.8240	0.8235	0.8533	0.7727
	12	**0.8886**	0.8782	0.8659	0.8768	0.8598	0.8419	0.8805	0.8507
	4	**0.7016**	0.7003	0.6474	0.7003	0.7003	0.6652	0.7003	0.6973
	6	0.7735	**0.7782**	0.7539	0.7720	0.7626	0.7416	0.7577	0.7315
Dog	**8**	**0.8215**	0.8166	0.7928	0.8170	0.8114	0.8136	0.8212	0.8161
	10	0.8624	0.8592	0.8425	**0.8647**	0.8415	0.8552	0.8558	0.8456
	12	**0.8927**	0.8895	0.8711	0.8894	0.8659	0.8073	0.8663	0.8599
	4	**0.6614**	0.5482	0.5459	0.5459	0.5457	0.5081	0.5459	0.5370
	6	**0.6977**	0.6970	0.6915	0.6913	0.6827	0.6365	0.6915	0.6834
Wolf	**8**	0.7772	0.7743	0.7818	0.7940	0.7603	0.7149	0.7653	**0.8243**
	10	**0.8418**	0.8351	0.8306	0.8174	0.8052	0.8400	0.7979	0.7659
	12	**0.8843**	0.8570	0.8683	0.8446	0.8250	0.8821	0.8451	0.8555

TABLE 12. The FSIM of each algorithm under Kapur.

TEST IMAGES	K	GSMVO	MVO	GSA	ALO	DA	FPA	PSO	CS
	4	**0.7524**	0.7506	0.7398	0.7506	0.7505	0.7444	0.7506	0.7387
	6	**0.825**	0.8170	0.7766	0.8157	0.8154	0.7693	0.8155	0.7990
Starfish	**8**	**0.8892**	0.8816	0.7913	0.8663	0.8792	0.7923	0.8785	0.8350
	10	**0.9228**	0.9137	0.8498	0.9099	0.9077	0.8737	0.9055	0.8750
	12	**0.9375**	0.9345	0.8847	0.9352	0.9239	0.8788	0.9355	0.8948
	4	**0.7524**	0.7505	0.7425	0.7505	0.7505	0.7232	0.7505	0.7425
	6	**0.8206**	0.8147	0.7990	0.8147	0.8142	0.7890	0.8153	0.7990
Corn	**8**	**0.8885**	0.8791	0.8672	0.8791	0.8614	0.8235	0.8643	0.8672
	10	**0.9122**	0.9101	0.8680	0.9101	0.9039	0.8574	0.9103	0.8680
	12	0.9340	0.9342	0.8686	0.9342	0.9336	0.8455	**0.9351**	0.8686
	4	**0.6791**	0.6627	0.6467	**0.6791**	0.6742	0.6691	0.6672	0.6692
	6	**0.7601**	0.7583	0.7518	0.7521	0.7580	0.7510	0.7600	0.7401
Tiger	**8**	**0.8791**	0.8773	0.8672	**0.8791**	0.8772	0.8781	0.8672	0.8641
	10	**0.9101**	0.8981	0.8780	0.9001	0.8880	0.8901	0.8680	0.8701
	12	**0.9291**	0.9072	0.8672	0.9290	0.9172	0.8991	0.8672	0.8991
	4	0.7511	0.7505	0.7370	0.7505	0.7495	0.7325	0.7505	**0.7535**
	6	**0.8152**	0.8119	0.7626	0.8098	0.8101	0.7898	0.8092	0.7921
Flower	**8**	**0.8672**	0.8545	0.8234	0.8412	0.8372	0.8333	0.8414	0.8317
	10	**0.9013**	0.8845	0.8406	0.8805	0.8710	0.8269	0.8793	0.8503
	12	**0.9279**	0.9124	0.8903	0.9029	0.8942	0.8190	0.9024	0.8595
	4	**0.7851**	0.7399	0.7271	0.7493	0.7493	0.7560	0.7399	0.7101
	6	**0.8337**	0.8108	0.7949	0.8046	0.8044	0.7634	0.8110	0.8075
Tower	**8**	**0.8758**	0.8592	0.7491	0.8507	0.8310	0.8310	0.8516	0.8115
	10	**0.9149**	0.8973	0.8534	0.8810	0.8662	0.8167	0.8838	0.8777
	12	**0.9284**	0.9109	0.8713	0.9054	0.8864	0.8602	0.9123	0.8675
	4	0.7865	0.7876	0.6839	0.7876	0.7876	0.7882	0.7876	**0.7923**
	6	**0.8644**	0.8395	0.7262	0.8573	0.8461	0.8211	0.8445	0.8469
Horse	**8**	**0.8979**	0.8943	0.7780	0.8952	0.8887	0.8494	0.8835	0.8532
	10	0.9155	0.9193	0.8972	**0.9280**	0.9210	0.8861	0.9194	0.8914
	12	**0.9475**	0.9248	0.9055	0.9361	0.9279	0.8925	0.9372	0.9081
	4	**0.7926**	0.7853	0.7253	0.7854	0.7854	0.7838	0.7854	0.7788
	6	0.8565	**0.8594**	0.7848	0.8586	0.8592	0.8149	0.8590	0.8400
River	**8**	**0.9032**	0.8990	0.8377	0.9010	0.8725	0.8398	0.9002	0.8847
	10	**0.9353**	0.9276	0.8615	0.9284	0.9136	0.9041	0.9310	0.8925
	12	**0.9502**	0.9479	0.8641	0.9500	0.9376	0.9122	0.9376	0.9260
	4	**0.7783**	0.7781	0.7289	0.7781	0.7775	0.7507	0.7781	0.7670
	6	**0.8474**	0.8437	0.7058	0.8336	0.8428	0.7991	0.8338	0.8252
Penguin	**8**	**0.8974**	0.8801	0.8247	0.8772	0.8783	0.8450	0.8772	0.8418
	10	**0.9198**	0.9163	0.8108	0.9133	0.8911	0.8451	0.9155	0.8853
	12	0.9343	**0.9363**	0.8894	0.9297	0.9249	0.8766	0.9257	0.8773
	4	**0.7610**	0.7594	0.7319	0.7594	0.7595	0.7393	0.7594	0.7400
	6	**0.8452**	0.8181	0.8081	0.8432	0.8206	0.7894	0.8317	0.7994
Dog	**8**	**0.8954**	0.8680	0.7873	0.8659	0.8582	0.8018	0.8610	0.8595
	10	**0.9183**	0.9003	0.8560	0.9041	0.8981	0.8601	0.9036	0.8857
	12	**0.9345**	0.9285	0.8975	0.9245	0.9152	0.8620	0.9227	0.8462
	4	**0.6796**	0.6735	0.6654	0.6754	0.6691	0.6731	0.6718	0.6686
	6	**0.7799**	0.7633	0.7680	0.7602	0.7615	0.7332	0.7626	0.7502
Wolf	**8**	**0.8729**	0.8411	0.7659	0.8340	0.8236	0.8082	0.8315	0.7488
	10	**0.9024**	0.8988	0.8375	0.8826	0.8732	0.8562	0.8872	0.8379
	12	**0.9264**	0.9225	0.8689	0.9102	0.8920	0.8039	0.9130	0.8449

The objective function value can be used as an index to evaluate the performance of algorithm. Each algorithm is tested on 10 color images for 30 times. Table 6 and table 7 show the average value of 30 fitness function values obtained by GSMVO-Kapur method and the GSMVO-Otsu method, where the maximum fitness function value is

TABLE 13. The FSIM of each algorithm under Otsu.

TEST IMAGES	K	GSMVO	MVO	GSA	ALO	DA	FPA	PSO	CS
Starfish	4	**0.7801**	0.7800	0.6950	0.7800	0.7800	0.7548	0.7800	0.7765
	6	0.8499	**0.8503**	0.8359	0.8501	0.8499	0.8186	0.8501	0.8322
	8	**0.8951**	0.8903	0.8165	**0.8951**	0.8936	0.8305	0.8950	0.8732
	10	**0.9213**	0.9187	0.8582	0.9206	0.9162	0.8885	0.9208	0.8868
	12	**0.9380**	0.9334	0.8910	0.9354	0.9296	0.8900	0.9366	0.9237
Corn	4	**0.8070**	**0.8070**	0.7783	**0.8070**	**0.8070**	0.7824	**0.8070**	0.7967
	6	**0.8624**	0.8622	0.8232	0.8623	0.8615	0.8406	0.8623	0.8407
	8	**0.8998**	0.8995	0.8721	0.8975	0.8974	0.8797	0.8978	0.8815
	10	**0.9242**	0.9205	0.8987	0.9217	0.9182	0.8873	0.9178	0.8886
	12	0.9392	**0.9428**	0.9085	0.9417	0.9346	0.8932	0.9318	0.8997
Tiger	4	**0.7787**	**0.7787**	0.7367	**0.7787**	0.7784	0.7410	0.7692	0.7727
	6	**0.8473**	0.8463	0.7913	0.8462	0.8449	0.8460	0.8463	0.8088
	8	**0.8912**	0.8908	0.8533	0.8882	0.8855	0.8714	0.8882	0.8427
	10	**0.9188**	0.9174	0.8981	0.9141	0.9085	0.8743	0.9034	0.895
	12	**0.9353**	0.9348	0.9151	0.9328	0.9218	0.9166	0.9327	0.9037
Flower	4	**0.7838**	0.7835	0.7568	**0.7838**	**0.7838**	0.7698	**0.7838**	0.7720
	6	**0.8418**	0.8346	0.8018	0.8413	**0.8418**	0.8027	0.8413	0.8209
	8	**0.8873**	0.8830	0.8262	0.8848	0.8858	0.8532	0.8871	0.8607
	10	0.9150	0.9123	0.8660	**0.9163**	0.9094	0.8727	0.8875	0.8747
	12	**0.9354**	0.9328	0.9080	0.9308	0.9238	0.8891	0.9275	0.8890
Tower	4	**0.8006**	**0.8006**	0.7583	**0.8006**	**0.8006**	0.7728	**0.8006**	0.7839
	6	**0.8565**	0.8563	0.7985	0.8551	0.8529	0.8227	0.8563	0.8316
	8	**0.8922**	0.8898	0.8383	0.8915	0.8868	0.8599	0.8884	0.8743
	10	**0.9133**	**0.9133**	0.8730	0.9105	0.9087	0.8785	0.9039	0.8875
	12	**0.9296**	0.9282	0.8951	0.9210	0.9152	0.8901	0.9210	0.8767
Horse	4	**0.8047**	**0.8047**	0.7167	**0.8047**	0.8042	0.7852	**0.8047**	0.7970
	6	**0.8666**	0.8665	0.8248	0.8664	0.8662	0.8367	**0.8666**	0.8536
	8	**0.9060**	0.9022	0.7249	0.9039	0.9029	0.8773	0.8929	0.885
	10	**0.9359**	0.9285	0.8732	0.9312	0.9306	0.8698	0.9282	0.8993
	12	**0.9443**	0.9387	0.9062	0.9425	0.9358	0.9188	0.9367	0.9153
River	4	0.7960	0.7960	0.6851	0.7960	0.7960	0.7571	0.7960	**0.7995**
	6	**0.8568**	0.8561	0.7960	0.8561	0.8542	0.8334	0.8490	0.8282
	8	**0.8937**	0.8890	0.8896	0.8880	0.8887	0.8732	0.8881	0.8716
	10	**0.9096**	0.9060	0.8756	0.9054	0.9027	0.8907	0.9023	0.9075
	12	0.9241	0.9254	0.9115	**0.9295**	0.9260	0.9077	0.9154	0.9005
Penguin	4	**0.7827**	**0.7827**	0.6927	**0.7827**	**0.7827**	0.7465	**0.7827**	0.7759
	6	0.8420	0.8420	0.7740	**0.8421**	0.8410	0.8018	0.8420	0.8234
	8	**0.8894**	0.8774	0.8676	0.8859	0.8869	0.8410	0.8867	0.8563
	10	**0.9184**	0.9176	0.8444	0.9181	0.9029	0.8818	0.9173	0.8617
	12	**0.9375**	0.9298	0.9079	0.9337	0.9254	0.8995	0.9341	0.9053
Dog	4	**0.7913**	0.7911	0.7193	0.7911	0.7911	0.7594	0.7911	0.7843
	6	0.8514	**0.8531**	0.8130	0.8509	0.8454	0.8068	0.8380	0.8192
	8	**0.8837**	0.8807	0.8442	0.8818	0.8793	0.8540	0.8835	0.8638
	10	**0.9104**	0.9066	0.8821	0.9086	0.9004	0.8898	0.9068	0.8892
	12	**0.9269**	0.9247	0.9087	0.9257	0.9155	0.8798	0.9124	0.8989
Wolf	4	**0.7909**	0.7893	0.7839	0.7893	0.7893	0.7829	0.7893	0.7893
	6	**0.7389**	0.7387	0.7090	0.7387	0.7385	0.7055	0.7387	0.7294
	8	**0.8060**	0.8045	0.7650	0.8042	0.8021	0.7754	0.8044	0.7945
	10	0.8507	0.8471	0.8154	**0.8597**	0.8443	0.8033	0.8439	0.8528
	12	**0.8826**	0.8803	0.8645	0.8770	0.8728	0.8754	0.8646	0.8563

TABLE 14. Average p value of Wilcoxon test after 30 times of operation under Kapur (the data of p>0.05 has been bolded, "+" indicates significant difference).

TEST IMAGES	GSMVO vs. MVO		GSMVO vs. GSA		GSMVO vs. ALO		GSMVO vs. DA		GSMVO vs. FPA		GSMVO vs. PSO		GSMVO vs. CS	
	p	h	p	h	p	h	p	h	p	h	p	h	p	h
Starfish	<0.05+	1	<0.05+	1	<0.05+	1	<0.05+	1	<0.05+	1	<0.05	1	<0.05+	1
Corn	<0.05+	1	<0.05+	1	<0.05+	1	<0.05+	1	<0.05+	1	**0.0677**	**0**	<0.05+	1
Tiger	<0.05+	1	<0.05+	1	<0.05	1	<0.05	1	<0.05+	1	<0.05	1	<0.05+	1
Flower	<0.05+	1	<0.05+	1	<0.05	1	<0.05+	1	<0.05+	1	<0.05	1	<0.05+	1
Tower	<0.05+	1	<0.05+	1	<0.05	1	<0.05+	1	<0.05+	1	<0.05	1	<0.05+	1
Horse	<0.05+	1	<0.05+	1	<0.05+	1	**0.0730**	**0**	<0.05+	1	<0.05+	1	<0.05+	1
River	<0.05+	1	<0.05+	1	<0.05	1	<0.05+	1	<0.05+	1	<0.05	1	<0.05+	1
Penguin	<0.05+	1	<0.05+	1	<0.05+	1	**0.0765**	**0**	<0.05+	1	<0.05+	1	<0.05+	1
Dog	<0.05+	1	<0.05+	1	<0.05+	1	<0.05	1	<0.05+	1	<0.05+	1	<0.05+	1
Wolf	<0.05+	1	<0.05+	1	<0.05	1	<0.05	1	<0.05+	1	**0.0520**	**0**	<0.05+	1

TABLE 15. Average p value of Wilcoxon test after 30 times of operation under Otsu (the data of p>0.05 has been bolded, "+" indicates significant difference).

TEST IMAGES	GSMVO vs. MVO		GSMVO vs. GSA		GSMVO vs. ALO		GSMVO vs. DA		GSMVO vs. FPA		GSMVO vs. PSO		GSMVO vs. CS	
	p	h	p	h	p	h	p	h	p	h	p	h	p	h
Starfish	<0.05+	1	<0.05	1	**0.0736**	**0**	<0.05+	1	<0.05+	1	<0.05+	1	<0.05+	1
Corn	<0.05+	1	<0.05+	1	<0.05+	1	<0.05+	1	<0.05+	1	<0.05+	1	<0.05+	1
Tiger	<0.05	1	<0.05+	1	<0.05+	1	<0.05	1	<0.05+	1	<0.05	1	<0.05+	1
Flower	<0.05+	1	<0.05+	1	**0.0699**	**0**	<0.05+	1	<0.05+	1	<0.05+	1	<0.05+	1
Tower	<0.05+	1	<0.05+	1	<0.05+	1	<0.05+	1	<0.05+	1	<0.05+	1	<0.05+	1
Horse	<0.05+	1	<0.05+	1	<0.05+	1	<0.05	1	<0.05+	1	<0.05+	1	<0.05+	1
River	<0.05+	1	<0.05+	1	<0.05+	1	<0.05+	1	<0.05+	1	**0.072**	**0**	<0.05+	1
Penguin	<0.05+	1	<0.05+	1	<0.05+	1	<0.05+	1	<0.05+	1	<0.05+	1	<0.05+	1
Dog	<0.05+	1	<0.05+	1	<0.05+	1	<0.05	1	<0.05+	1	<0.05+	1	<0.05+	1
Wolf	<0.05+	1	<0.05+	1	<0.05	1	<0.05	1	<0.05+	1	<0.05	1	<0.05+	1

indicated by bold. As can be seen from these two tables, the number of running time that the GSMVO-based segmentation method find the largest objective function value is higher than other algorithms. Therefore, the proposed multilevel threshold segmentation method based on GSMVO is more accurate than other algorithms. For smaller thresholds (such as K = 4, 6), the value of object function obtained by all algorithms is almost the same. For larger thresholds (such as K = 8, 10, 12), higher results are obtained. Fig. 7 is the segmentation image obtained by GSMVO-Kapur method.

TABLE 16. Optimal solution for GSMVO under Kapur.

TEST IMAGES	K	R	G	B
	4	54 100 145 190	68 113 159 205	55 99 142 186
	6	42 76 110 145 178 211	57 92 126 159 192 223	51 81 113 145 178 210
Starfish	8	36 64 91 118 145 173 199 224	48 74 101 127 153 178 203 229	30 54 80 107 134 160 186 210
	10	32 57 81 104 126 148 170 191 213 235	45 68 91 112 134 156 177 196 217 237	10 31 52 75 99 122 145 167 187 210
	12	28 48 68 89 110 130 149 168 187 205 223 241	21 43 63 84 104 124 144 163 183 202 220 238	10 28 47 64 82 101 120 140 160 182 200 211
	4	49 90 125 166	56 96 133 180	54 92 134 186
	6	39 65 91 125 160 194	41 68 97 129 163 202	53 83 112 142 173 207
Corn	8	16 48 90 123 152 174 199 220	44 70 97 128 148 173 197 223	32 57 82 108 134 160 188 213
	10	16 40 65 91 120 138 158 179 199 220	20 41 59 78 98 128 152 180 203 229	32 57 81 104 124 144 163 186 203 220
	12	16 40 65 91 119 135 152 167 185 203 221 241	20 48 73 96 113 128 145 162 180 199 218 238	31 57 81 99 117 134 151 167 186 199 212 225
	4	61 100 143 187	77 119 158 200	69 109 151 197
	6	48 81 114 148 182 212	54 85 117 147 178 211	63 98 128 159 191 222
Tiger	8	41 66 90 115 140 165 190 217	51 79 107 133 157 181 205 227	16 54 82 109 137 164 191 222
	10	39 62 84 107 130 151 171 190 210 231	48 70 92 114 136 154 174 193 212 232	16 39 62 84 106 130 156 185 209 232
	12	12 40 64 86 107 128 147 166 184 202 220 237	45 65 85 104 122 140 157 175 192 209 226 242	16 32 52 73 93 114 137 159 181 200 218 236
	4	53 99 143 187	49 106 153 202	27 75 118 175
	6	42 75 108 143 177 206	32 69 106 140 176 212	20 51 82 113 142 175
Flower	8	38 68 97 124 151 177 202 227	25 54 83 112 140 169 197 224	19 44 70 94 118 145 175 197
	10	31 55 79 103 128 152 175 195 216 236	23 49 74 100 124 146 169 192 214 236	19 44 68 90 113 135 155 175 191 211
	12	29 52 74 95 115 135 154 172 188 203 220 237	22 44 66 88 108 128 149 168 187 205 221 239	17 35 53 71 88 105 123 140 158 175 197 256
	4	85 129 173 222	87 138 177 218	62 95 132 172
	6	53 91 130 168 200 232	73 103 138 172 205 236	61 89 117 144 173 215
Tower	8	53 85 114 142 168 193 218 245	51 77 102 136 163 185 213 241	45 66 93 122 146 173 199 227
	10	22 52 78 100 124 147 170 197 222 246	51 77 101 120 140 163 182 203 225 245	12 23 46 66 93 122 145 173 201 227
	12	22 48 69 88 107 127 147 168 188 207 226 247	49 65 82 101 119 137 157 176 194 212 230 247	12 23 46 65 85 101 117 135 173 194 216 236
	4	37 84 132 180	51 105 157 210	66 120 168 211
	6	36 75 113 149 186 226	46 85 118 151 180 210	45 77 109 143 179 213
Horse	8	33 64 93 120 148 175 200 226	29 55 81 106 131 156 181 210	37 60 84 108 134 159 185 213
	10	27 49 71 94 118 141 163 187 210 226	25 47 68 91 113 136 157 178 199 214	37 60 83 106 132 153 174 194 214 241
	12	23 42 63 83 102 122 142 160 179 198 218 234	26 49 72 91 111 131 151 168 184 199 210 224	27 47 66 85 105 124 142 160 179 196 213 233
	4	67 105 143 181	90 125 160 195	84 121 158 196
	6	54 83 111 139 167 195	81 108 135 162 188 214	53 85 115 146 177 208
River	8	38 63 87 112 136 160 184 208	77 99 121 143 165 187 208 229	53 79 103 127 151 175 199 223
	10	28 54 78 101 124 146 168 190 211 233	71 90 108 127 146 165 183 201 219 236	37 54 78 102 125 147 169 190 210 230
	12	27 48 68 88 109 129 150 170 189 208 227 245	63 79 95 112 129 146 162 179 195 211 227 243	43 57 78 99 120 141 158 175 193 210 226 242
	4	55 94 133 173	68 124 179 218	40 78 117 155
	6	51 86 120 154 188 222	45 78 112 145 180 218	28 56 84 118 155 203
Penguin	8	37 63 91 119 148 175 202 231	38 67 95 122 150 179 204 229	26 52 79 105 128 155 182 204
	10	34 58 81 104 127 150 173 196 218 240	29 52 75 97 119 141 163 186 210 233	21 43 64 85 106 126 146 164 183 204
	12	26 44 62 82 102 123 144 164 184 203 222 242	28 49 71 94 115 136 155 174 190 207 223 239	19 38 56 75 93 111 127 145 163 182 195 211
	4	46 96 146 187	74 112 149 192	62 102 149 200
	6	38 73 110 145 174 206	63 91 120 150 187 219	55 89 123 157 195 217
Dog	8	31 58 86 115 144 167 190 214	63 91 120 149 177 194 213 233	45 69 93 117 141 162 194 215
	10	25 48 71 94 118 143 161 179 198 217	55 76 97 119 140 159 182 198 215 233	22 46 70 93 115 137 158 178 197 217
	12	18 35 52 70 88 105 122 143 163 183 202 220	51 68 85 104 122 140 159 178 192 207 222 237	18 39 57 77 97 118 138 158 176 195 212 229
	4	52 96 142 190	55 96 137 178	44 88 135 183
	6	28 65 102 139 174 210	43 77 111 145 178 208	36 72 108 143 178 210
Wolf	8	16 47 78 108 139 168 196 223	35 63 92 120 149 178 204 229	32 61 89 118 146 175 200 226
	10	14 41 75 89 109 134 175 198	20 46 71 94 118 141 164 186 210 232	25 47 70 93 116 140 164 185 209 232
	12	14 37 60 82 102 123 142 163 181 201 219 235	16 38 59 79 100 122 143 163 181 201 219 239	23 44 65 86 107 127 147 167 187 205 222 239

Fig. 8 is the segmentation image obtained by GSMVO-Otsu method. Optimal solution for GSMVO with different thresholds are given in table 16 and 17.

From the convergence curves, the proposed algorithm is superior to MVO, GSA, ALO, DA, FPA, PSO, and CS algorithms. Fig. 6 shows the convergence curve of all the algorithms utilizing Kapur and Otsu methods, respectively. The value of the y-coordinate represents the value of the fitness function and the convergence curves at the maximum threshold (K=12) are selected for comparison. The improved algorithm is represented by an obvious bright right curve. According to the curves, the overall performance of improved algorithm is the best and the approximate optimal solution can be obtained. ALO is the second best, from (h), (i) of OTSU and (a), (b), (f), (h) of Kapur in Fig. 6, it can be seen that ALO algorithm is able to find larger objective function values than other algorithms. From (g), (h), (j) of Kapur in Fig. 6 PSO algorithm can also find some larger objective function values. GSA is prone to fall into local optimality and thus the optimal fitness function value cannot be found. For the remaining graphs, it is clear that GSMVO is superior to all other algorithms in the optimization process. Generally, the convergence graph shows the superiority and high performance of GSMVO. GSA, FPA, PSO, CS and other algorithms have encountered premature convergence in the later stage. Other algorithms, such as the traditional MVO, the update of population is slow and mutation occurs occasionally. The shape of convergence curve is ladderlike rather than a rising smooth curve. But GSMVO can completely overcome these problems, it is shown in the figure that the convergence curve is almost perfect, which also proves that the application of GSA operator for global search strategy is successful.

It can be seen from the curve of convergence in Fig. 6 that GSMVO converges completely after about 50∼100 iterations in each figure. By contrast, the convergence rate is faster than ALO, DA, FPA, PSO, and CS.

The (k) in Fig. 6 is 8 kinds of algorithm independent running 50 times the best fitness value variance analysis contrast figure. Known from the Fig. 6, in the two kinds of image segmentation function, the stability of GSMVO optimization accuracy is superior to the rest of algorithms. In 50 separate runs, GSMVO can find the optimal fitness value of the theoretical optimal value or the value close to the theoretical optimal value every time, 50 times are very stable. ALO algorithm always is better than other algorithm in addition to the hybrid algorithm. The overall fluctuation ratio of GSA and FPA algorithms is large and CS algorithm is small, but the optimal value found by them deviates from the optimal value greatly and falls into the local optimal value.

TABLE 17. Optimal solution for GSMVO under Otsu.

TEST IMAGES	K	R	G	B
	4	56 101 149 209	63 105 147 193	31 55 92 137
	6	43 75 106 140 179 225	53 85 115 143 172 207	27 41 62 91 123 156
Starfish	8	35 60 85 109 135 164 198 234	47 71 95 118 140 162 185 215	25 37 52 75 102 131 161 252
	10	32 55 76 96 117 137 160 185 213 241	44 64 84 103 122 140 159 178 199 224	24 36 49 69 94 119 143 168 212 244
	12	29 48 67 84 102 119 137 157 178 200 221 243	42 61 79 97 113 128 144 160 176 194 213 235	22 32 41 54 73 94 115 135 155 177 233 255
	4	47 83 107 154	49 84 110 155	37 61 84 132
	6	46 81 104 122 157 212	39 65 93 111 135 192	31 53 69 89 137 251
Corn	8	35 57 82 100 109 128 166 228	38 64 89 106 115 140 188 238	25 42 57 68 78 95 124 174
	10	29 46 66 86 101 110 127 161 209 248	37 59 81 98 109 117 140 179 214 244	27 46 61 70 83 104 137 175 209 233
	12	29 46 65 84 99 107 116 133 170 215 237 253	32 48 65 84 99 108 116 131 163 206 232 251	25 40 54 65 72 84 101 128 153 177 213 252
	4	53 88 132 191	60 87 122 179	51 74 107 163
	6	42 63 87 117 154 205	52 72 93 119 155 205	43 59 77 100 137 192
Tiger	8	38 55 72 93 118 146 179 221	48 64 80 98 119 145 180 222	40 54 68 84 107 138 179 222
	10	36 52 67 84 105 128 151 175 201 233	47 62 76 91 107 126 151 182 211 235	37 49 60 72 85 102 125 155 193 231
	12	34 48 62 76 92 111 132 152 173 200 222 245	44 57 70 83 97 112 129 152 179 203 224 240	37 49 60 72 85 100 120 146 171 201 234 253
	4	51 98 143 201	31 69 111 168	19 44 75 119
	6	34 66 99 132 164 209	22 47 77 109 150 202	15 33 55 81 118 186
Flower	8	29 54 78 103 130 156 183 222	11 28 51 78 103 132 170 216	7 18 35 55 77 103 144 240
	10	26 45 66 86 107 130 152 171 195 230	10 24 43 63 86 109 138 171 204 239	7 17 33 51 70 92 125 185 240 249
	12	25 43 64 83 102 121 140 157 173 190 212 237	10 22 37 54 72 90 109 132 158 186 220 239	6 15 28 43 58 74 92 116 149 186 207 254
	4	50 76 111 190	58 91 124 190	58 94 131 192
	6	40 58 76 101 136 202	48 69 94 118 144 204	57 88 117 141 157 200
Tower	8	39 57 73 93 118 149 199 240	45 63 80 100 119 140 180 230	39 59 78 97 122 143 157 200
	10	35 50 64 77 94 116 144 185 221 245	44 62 78 97 112 124 144 177 217 244	39 59 78 96 121 142 156 179 213 241
	12	34 48 60 70 81 96 113 132 157 192 223 238	43 58 70 83 99 112 123 138 158 190 217 243	39 58 75 91 108 126 143 156 179 208 228 241
	4	40 71 106 152	43 77 109 148	72 120 156 187
	6	32 53 74 99 127 165	34 60 80 102 128 160	54 94 131 157 181 208
Horse	8	29 48 65 83 103 124 149 179	31 55 73 89 107 127 147 173	44 71 101 130 152 170 189 213
	10	28 47 63 79 96 115 135 160 186 218	31 54 71 85 100 117 136 157 182 250	43 69 96 122 140 155 170 184 199 219
	12	26 42 55 68 83 98 113 129 145 165 185 202	24 43 60 75 89 104 121 138 158 180 240 254	37 57 76 98 119 136 150 162 174 187 201 219
	4	79 124 168 219	100 141 180 224	99 142 184 227
	6	70 103 133 162 192 230	87 116 144 171 199 233	83 110 137 167 200 234
River	8	58 82 106 130 154 177 202 234	81 104 127 148 168 189 212 239	77 99 121 143 166 191 215 240
	10	55 77 98 117 136 155 174 193 214 240	76 93 111 129 146 163 180 198 218 241	73 92 110 127 145 165 184 202 222 243
	12	54 74 93 111 129 146 162 177 193 208 225 244	6 15 79 101 122 141 158 173 189 206 224 245	19 69 85 101 117 132 148 167 186 204 223 243
	4	46 92 138 203	40 80 127 192	23 52 99 149
	6	33 62 92 126 162 214	31 60 86 114 150 202	21 48 84 118 146 167
Penguin	8	29 54 78 102 127 153 185 227	28 52 75 97 120 149 184 221	14 29 46 65 93 123 145 166
	10	27 50 70 88 108 130 152 176 205 238	25 46 65 84 102 122 146 169 199 230	13 27 43 61 86 115 137 155 171 235
	12	27 50 69 87 106 125 143 162 182 207 233 246	23 40 57 74 89 105 123 143 164 187 213 236	7 19 34 50 68 94 120 140 157 173 227 255
	4	50 84 115 157	73 107 145 188	52 83 123 161
	6	39 67 91 113 134 169	62 85 109 137 162 194	46 68 94 127 157 178
Dog	8	37 64 86 107 126 148 183 251	57 75 93 113 136 157 172 200	42 60 79 103 132 158 175 194
	10	29 51 70 86 102 117 130 146 167 196	55 72 89 106 126 144 158 170 184 210	39 54 70 87 109 135 159 176 195 256
	12	29 50 67 82 96 110 122 132 145 163 185 208	55 70 86 102 119 138 154 165 174 187 209 225	38 52 66 82 102 126 149 164 178 196 234 252
	4	30 73 114 178	35 85 134 197	37 84 133 192
	6	24 56 85 113 149 204	26 60 95 129 163 213	24 50 82 118 152 203
Wolf	8	20 46 70 93 115 142 177 220	21 46 71 98 125 151 182 225	20 41 65 93 121 147 176 218
	10	19 41 62 82 100 118 140 167 200 234	20 42 63 84 105 125 145 166 193 230	17 34 52 73 97 120 141 162 189 226
	12	15 33 50 66 82 97 112 128 149 177 209 239	19 40 60 81 101 121 139 157 176 200 226 247	17 33 50 68 89 109 129 147 166 188 215 240

MVO and DA algorithm with less fluctuation, but both of them have some outliers.

To sum up, GSMVO has higher optimization accuracy, better robustness, and stability than the comparison algorithm. Which proves the effectiveness and superiority

of this algorithm and achieves the purpose of improving the basic MVO algorithm.

(3) PSNR [55]. Namely, the peak signal to noise ratio is an objective standard for image evaluation. Represents the ratio between the maximum possible power of a signal and the power of corrupting noise. The larger the PSNR value, the better the segmentation effect. However, it has certain limitations. Because the visual acuity of human eyes is not absolute, it is possible that those with higher PSNR may look worse than those with lower PSNR. It defined as follows:

$$PSNR = 20\log(\frac{255}{RMSE})(dB) \tag{28}$$

where RMSE is the mean square error between the original image and the processed image

$$RMSE = \sqrt{\frac{\sum_{i=1}^{M}\sum_{j=1}^{N}(I(i,j)-\widehat{I}(i,j))^2}{MN}} \tag{29}$$

where $I(i,j)$ and $\widehat{I}(i,j)$ represent the original image and the segmented image with size $M \times N$ respectively.

SSIM [56]. Finds the similarity between segmented image and uncompressed or distortion-free image. An index to measure the structural similarity of two images, if the value is close to 1, then the image segmentation effect is better. The mathematical model of the SSIM is defined as follows:

$$SSIM(I,\hat{I}) = \frac{(2\mu_I\mu_{\hat{I}}+c_1)(2\sigma_I\sigma_{\hat{I}}+c_2)}{(\mu_I^2+\mu_{\hat{I}}^2+c_1)(\sigma_I^2+\sigma_{\hat{I}}^2+c_2)} \tag{30}$$

where μ_I is the estimation of the original image based on average grayscale brightness measurement. $\mu_{\hat{I}}$ is the estimation of brightness measurement based on average grayscale of the segmented images. σ_I and $\sigma_{\hat{I}}$ represent the standard deviations of image I and image $\hat{I}$, respectively. Both c_1 and c_2 are constants.

FSIM [57]. Feature similarity index, which is another method to evaluate feature similarity after SSIM. Defines the quality score which reflects the significance of a local structure. The basic concept is to measure image quality by evaluating feature similarity between original image and segmented image. It is given by:

$$\mathrm{FSIM} = \frac{\sum_{x\in\Omega} S_L(x) \times PC_m(x)}{\sum_{x\in\Omega} PC_m(x)} \tag{31}$$

TABLE 18. The main symbols involved in the paper.

Provenance	Symbols	Paraphrase
Color images	L	The gray value
	n_i	The number of pixels with gray value of i
	N	The total number of pixels
	p	The distribution probability of gray value
	K	The total number of threshold
	t	The threshold
	M	The class
	RGB	The three channels of color images
Kapur's entropy method	H	The entropy of the image
	P	The probabilities of each class in the segmented classes
	ψ	The objective function of Kapur
Otsu method	ω	The class probability
	μ_k	The class mean levels
	μ_T	The total mean level
	Fit	The objective function of Otsu
MVO	U	The universe
	d	The number of parameters
	n	The number of solutions
	x	The parameter of universe
	$NI(U)$	The standard inflation rate universe
	TDR	The travelling distance rate
	WEP	The wormhole existence rate
	l	The current iteration times
	L	The maximum iteration times
	p	The accuracy of mining capability
	ub	The upper bound of parameter
	lb	The lower bound of parameter
GSA	v	The velocity of parameter
	a	The acceleration of parameter
	F	The gravity force
	M	The inertial mass
	R	The Euclidean distance between two parameters
	G	The universal gravitation
	$worst(t)$	The worst objective function
	$best(t)$	The best objective function
	t	The current iteration times
	T	The maximum iteration times
Time complexity	O	The complexity notation

where Ω is the pixel field of the entire image, $S_L(x)$ represents the similarity value and $PC_m(x)$ denotes the phase consistency measure.

$$PC_m(x) = \max(PC_1(x), PC_2(x)) \tag{32}$$

The $PC_1(x)$ and the $PC_2(x)$ are the phase consistency of two regions.

$$S_L(x) = [S_{PC}(x)]^{\alpha} \cdot [S_G(x)]^{\beta} \tag{33}$$

where

$$S_{PC}(x) = \frac{2PC_1(x) \times PC_2(x) + T_1}{PC_1^2(x) \times PC_2^2(x) + T_1} \tag{34}$$

$$S_G(x) = \frac{2G_1(x) \times G_2(x) + T_2}{G_1^2(x) \times G_2^2(x) + T_2} \tag{35}$$

Then, $S_{PC}(x)$ is the similarity measure of phase consistency, $S_G(x)$ represents the gradient magnitude of two regions $G_1(x)$ and $G_2(x)$, and α, β, T_1 and T_2 are all constants.

Table 8 and 9 show PSNR values, table 10 and 11 show SSIM values, and tables 12 and 13 show FSIM values. It can be seen from table 8 and 9 that PSNR has a higher value at high threshold, which indicates good segmentation quality of the output image. Due to the accurate search capability of GSMVO, the obtained PSNR values are generally better than other algorithms, especially at the high threshold level.

In order to observe the superiority of GSMVO more clearly, the PSNR values of each test picture are integrated and made into a line graph, as shown in Fig. 9 and 10. In order to facilitate observation, the data based on GSMVO algorithm is selected with green and bold. So as to the FSIM and SSIM, Fig. 11–14 are also obviously show that the evaluation value corresponding to GSMVO is above other algorithms. The results show that the algorithm has good image segmentation performance.

(4) Wilcoxon's rank-sum test [58], [59]. In order to further verify the performance of the improved algorithm, Wilcoxon's rank sum test was adopted. Statistical analysis was conducted under the condition of 5% significance level. The fitness function value of GSMVO algorithm with K=12 is compared with other seven algorithms including traditional MVO algorithm. All algorithms run the same 30 times. We generally believe that if $p<0.05$ (or $h=1$), this can be considered sufficient evidence against the null hypothesis. If the performance of GSMVO is better than that of other algorithms, and there is a significant difference between algorithms, then "+" is used to mark after 0.05. Less significant difference is not marked. Similar to other algorithms, the actual value is given.

The Wilcoxon test results evaluated by p value are shown in table 14 and 15, and p value is statistically significant in all cases. With Kapur entropy as the test function, it can be seen that four out of the seventy cases (10 images and 7 algorithms) show poor results. Under the condition of Otsu as the test function, the overall test results are satisfied. Only three out of the seventy cases are poor. These results suggest that there are significant differences between GSMVO algorithm and other seventy algorithms. In particular, compared with traditional MVO algorithm, there is no flawed result in both test functions, which indicates that there is a great improvement.

To sum up, GSMVO has made significant improvement on the basis of MVO. Compared with the same other meta-heuristic algorithms, GSMVO has better performance, accuracy and convergence in multithreshold color image segmentation. It can serve as a useful alternative in multi-level image thresholding segmentation.

VII. CONCLUSION

Color image segmentation problem based on multilevel threshold is studied in this paper. With the increase of the threshold value number, the computational complexity of multilevel threshold value multiplied and search space complexity increased exponentially. Therefore, in order to solve above problem, an improved GSMVO algorithm was adopted. Two image segmentation functions, Kapur's entropy and Otsu, are selected to evaluate the performance of improved algorithm. The multithreshold performance of algorithm is tested by 10 images of BSD500. The segmentation results are analyzed in terms of the algorithm execution time, the average value of fitness function values, PSNR, SSIM, FSIM indices, and Wilcoxon test. Experimental results show that GSMVO has universality and better robustness compared with the other seven algorithms. The multilevel threshold segmentation method based on GSMVO has a good application prospect. In the future work, a simpler and more efficient MVO algorithm will be found, for example, by combining with other algorithms and applying it to practical engineering optimization problems such as pattern recognition and computer vision.

REFERENCES

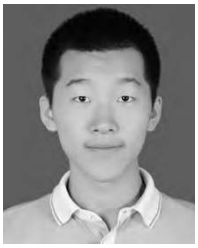
[1] R. M. Haralick and L G. Shapiro, "Survey: Image segmentation techniques," *Comput. Vis., Graph., Image Process.*, vol. 29, no. 1, pp. 100–132, 1985.
[2] H. Kaut and R. Singh, "A review on image segmentation techniques for future research study," *Pattern Recognit.*, vol. 26, no. 9, pp. 1277–1294, 1993.
[3] H. V. H. Ayala, F. M. dos Santos, V. C. Mariani, and L. dos Santos Coelho, "Image thresholding segmentation based on a novel beta differential evolution approach," *Expert Syst. Appl.*, vol. 42, no. 4, pp. 2136–2142, 2015.
[4] 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.
[5] M. Ali, C. W. Ahn, and M. Pant, "Multi-level image thresholding by synergetic differential evolution," *Appl. Soft Comput.*, vol. 17, pp. 1–11, Apr. 2014.
[6] H. Mittal and M. Saraswat, "An optimum multi-level image thresholding segmentation using non-local means 2D histogram and exponential Kbest gravitational search algorithm," *Eng. Appl. Artif. Intell.*, vol. 71, pp. 226–235, May 2018.
[7] M. Sezgin and B. Sankur, "Survey over image thresholding techniques and quantitative performance evaluation," *Proc. SPIE*, vol. 13, no. 1, pp. 146–168, Jan. 2004.
[8] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Trans. Syst. Man, Cybern.*, vol. 9, no. 1, pp. 62–66, Jan. 1979.
[9] 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, Jul. 1985.
[10] S. K. Pal, R. A. King, and A. A. Hashim, "Automatic grey level thresholding through index of fuzziness and entropy," *Pattern Recognit. Lett.*, vol. 1, no. 3, pp. 141–146, Mar. 1983.
[11] S. Patra, R. Gautam, and A. Singla, "A novel context sensitive multilevel thresholding for image segmentation," *Appl. Soft Comput.*, vol. 23, pp. 122–127, Oct. 2014.
[12] C. H. Li and C. K. Lee, "Minimum cross entropy thresholding," *Pattern Recognit.*, vol. 26, no. 4, pp. 617–625, Apr. 1993.
[13] S. Pare, A. Kumar, V. Bajaj, and G. K. Singh, "An efficient method for multilevel color image thresholding using cuckoo search algorithm based on minimum cross entropy," *Appl. Soft Comput.*, vol. 61, pp. 570–592, Dec. 2017.
[14] M. P. de Albuquerque, I. A. Esquef, A. R. G. Mello, and M. P. de Albuquerque, "Image thresholding using Tsallis entropy," *Pattern Recognit. Lett.*, vol. 25, no. 9, pp. 1059–1065, Jul. 2004.
[15] S. Sarkar, N. Sen, A. Kundu, S. Das, and S. S. Chaudhuri, "A differential evolutionary multilevel segmentation of near infra-red images using Renyi's entropy," in *Proc. Int. Conf. Frontiers Intell. Comput., Theory Appl. (FICTA)*. Berlin, Germany: Springer, 2014, pp. 699–706.
[16] C. E. Shannon, "A mathematical theory of communication," *ACM SIGMOBILE Mobile Comput. Commun. Rev.*, vol. 5, no. 1, pp. 3–55, Jan. 2001.
[17] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 67–82, Apr. 1997.
[18] H. Gao, W. Xu, Y. Tang, and J. Sun, "Multilevel thresholding for image segmentation through an improved quantum-behaved particle swarm algorithm," *IEEE Trans. Instrum. Meas.*, vol. 59, no. 4, pp. 934–946, Apr. 2010.
[19] N. S. M. Raja, V. Rajinikanth, and K. Latha, "Otsu based optimal multilevel image thresholding using firefly algorithm," *Modell. Simul. Eng.*, vol. 2014, no. 37, Jan. 2014, Art. no. 37.
[20] S. C. Satapathy, N. S. M. 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.

[21] P. Lopez-Garcia, E. Onieva, E. Osaba, A. D. Masegosa, and A. Perallos, "GACE: A meta-heuristic based in the hybridization of genetic algorithms and cross entropy methods for continuous optimization," *Expert Syst. Appl.*, vol. 55, pp. 508–519, Feb. 2016.
[22] 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. Elect. Eng.*, vol. 70, pp. 476–495, Aug. 2017.
[23] M. S. R. Naidu, P. R. Kumar, and K. Chiranjeevi, "Shannon and fuzzy entropy based evolutionary image thresholding for image segmentation," *Alexandria Eng. J.*, vol. 57, pp. 1643–1655, May 2017.
[24] S. Mirjalili, S. M. Mirjalili, and A. Hatamlou, "Multi-verse optimizer: A nature-inspired algorithm for global optimization," *Neural Comput. Appl.*, vol. 27, no. 2, pp. 495–513, Feb. 2016.
[25] H. Singh, S. Mehta, and S. Prashar, "Economic load dispatch using multi verse optimization," *Int. J. Eng. Res. Sci.*, vol. 6, no. 2, pp. 2395–6992, 2016.
[26] P. Jangir, S. A. Parmar, I. N. Trivedi, and R. H. Bhesdadiya, "A novel hybrid particle swarm optimizer with multi verse optimizer for global numerical optimization and optimal reactive power dispatch problem," *Eng. Sci. Technol. Int. J.*, vol. 20, no. 2, pp. 570–586, Apr. 2017.
[27] H. Faris, I. Aljarah, and S. Mirjalili, "Training feedforward neural networks using multi-verse optimizer for binary classification problems," *Appl. Intell.*, vol. 45, no. 2, pp. 322–332, Sep. 2016.
[28] H. Faris, M. A. Hassonah, A. M. Al-Zoubi, S. Mirjalili, and I. Aljarah, "A multi-verse optimizer approach for feature selection and optimizing SVM parameters based on a robust system architecture," *Neural Comput. Appl.*, vol. 30, no. 8, pp. 2355–2369, Oct. 2017.
[29] M. M. Mafarja and S. Mirjalili, "Hybrid whale optimization algorithm with simulated annealing for feature selection," *Neurocomputing*, vol. 260, pp. 302–312, Oct. 2017.
[30] E.-G. Talbi, "A taxonomy of hybrid metaheuristics," *J. Heuristics*, vol. 8, no. 5, pp. 541–564, Jan. 2002.
[31] E. A. Baniani and A. Chalechale, "Hybrid PSO and genetic algorithm for multilevel maximum entropy criterion threshold selection," *Int. J. Hybrid Inf. Technol.*, vol. 6, no. 5, pp. 131–140, 2013.
[32] E. Rashedi, H. Nezamabadi-Pour, and S. Saryazdi, "GSA: A gravitational search algorithm," *J. Inf. Sci.*, vol. 179, no. 13, pp. 2232–2248, Mar. 2009.
[33] Y. Kumar and G. Sahoo, "A review on gravitational search algorithm and its applications to data clustering & classification," *Int. J. Intell. Syst. Appl.*, vol. 6, pp. 79–93, May 2014.
[34] H. Mittal, R. Pal, A. Kulhari, and M. Saraswat, "Chaotic Kbest gravitational search algorithm (CKGSA)," in *Proc. Int. Conf. Contemp. Comput.*, 2016, pp. 1–6.
[35] F. Zhao, F. Xue, Y. Zhang, W. Ma, C. Zhang, and H. Song, "A hybrid algorithm based on self-adaptive gravitational search algorithm and differential evolution," *Expert Syst. Appl.*, vol. 113, pp. 515–530, Jul. 2018.
[36] J. Khoury, B. A. Ovrut, N. Seiberg, P. J. Steinhardt, and A. N. Turok, "From big crunch to big bang," *Phys. Rev. D, Part. Fields*, vol. 65, no. 8, 2001, Art. no. 086007.
[37] P. C. W. Davies, "Thermodynamics of black holes," *Russ. Phys. J.*, vol. 49, no. 10, pp. 1149–1150, 1978.
[38] D. M. Eardley, "Death of white holes in the early Universe," *Phys. Rev. Lett.*, vol. 33, no. 7, pp. 442–444, Aug. 1974.
[39] M. S. Morris and K. S. Thorne, "Wormholes in spacetime and their use for interstellar travel: A tool for teaching general relativity," *Amer. J. Phys.*, vol. 56, no. 5, pp. 395–412, May 1988.
[40] A. H. Guth, "Eternal inflation and its implications," *J. Phys. A, Math. Theor.*, vol. 40, no. 25, p. 6811, Jun. 2007.
[41] V. Kumar, J. K. Chhabra, and D. Kumar, "Automatic cluster evolution using gravitational search algorithm and its application on image segmentation," *Eng. Appl. Artif. Intell.*, vol. 29, pp. 93–103, Nov. 2013.
[42] 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, Feb. 2016.
[43] M. Wang, C. Wu, L. Wang, D. Xiang, and X. Huang, "A feature selection approach for hyperspectral image based on modified ant lion optimizer," *Knowl.-Based Syst.*, vol. 168, no. 15, pp. 39–48, Mar. 2019.
[44] L. Xu, H. Jia, C. Lang, X. Peng, and K. Sun, "A novel method for multilevel color image segmentation based on dragonfly algorithm and differential evolution," *IEEE Access*, vol. 99, pp. 19502–19538, Feb. 2019.
[45] X.-S. Yang, "Flower pollination algorithm for global optimization," *Unconventional Computation and Natural Computation*, vol. 7445. Berlin, Germany: Springer, pp. 240–249, 2012.
[46] L. Shen, C. Fan, and X. Huang, "Multi-level image thresholding using modified flower pollination algorithm," *IEEE Access*, vol. 6, pp. 30508–30519, May 2018.
[47] J. Kennedy and J. Kennedy, "Particle swarm optimization," in *Proc. IEEE Int. Conf. Neural Netw.*, Perth, WA, Australia, vol. 4, 1995, pp. 1942–1948. doi: .10.1007/978-0-387-30164-8_630.
[48] M. Maitra and A. Chatterjee, "A hybrid cooperative–comprehensive learning based PSO algorithm for image segmentation using multilevel thresholding," *Expert Syst. Appl.*, vol. 34, no. 2, pp. 1341–1350, Feb., 2008.
[49] Y. Liu, C. Mu, W. Kou, and J. Liu, "Modified particle swarm optimization-based multilevel thresholding for image segmentation," *Soft Comput.*, vol. 19, no. 5, pp. 1311–1327, May 2015.
[50] H. Gao, C.-M. Pun, and S. Kwong, "An efficient image segmentation method based on a hybrid particle swarm algorithm with learning strategy," *Inf. Sci.*, vol. 369, pp. 500–521, Nov. 2016.
[51] A. S. Joshi, O. Kulkarni, G. M. Kakandikar, and V. M. Nandedkar, "Cuckoo search optimization—A review," *Mater. Today*, vol. 4, no. 8, pp. 7262–7269, 2017.
[52] A. K. Bhandari, V. K. Singh, A. Kumar, and G. K. Sing, "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.
[53] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 5, pp. 898–916, May 2011.
[54] D. Mújica-Vargas, F. J. Gallegos-Funes, A. J. Rosales-Silva, and J. D. J. Rubio, "Robust c-prototypes algorithms for color image segmentation," *EURASIP J. Image Video Process.*, vol. 1, p. 63, Dec. 2013.
[55] R. Roy and S. Laha, "Optimization of stego image retaining secret information using genetic algorithm with 8-connected PSNR," *Procedia Comput. Sci.*, no. 60, pp. 468–477, Jan. 2015.
[56] 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.
[57] L. Zhang, L. Zhang, X. Mou, and D. Zhang, "FSIM: A feature similarity index for image quality assessment," *IEEE Trans. Image Process.*, vol. 20, no. 8, pp. 2378–2386, Aug. 2011.
[58] X. S. Yang, "Multiobjective firefly algorithm for continuous optimization," *Eng Comput*, vol. 29, no. 2, pp. 175–184, Apr. 2013.
[59] F. Wilcoxon, "Individual comparisons by ranking methods," *Biometrics Bull.*, vol. 1, no. 6, pp. 80–83, 1945.

HEMING JIA received the Ph.D. degree in system engineering from Harbin Engineering University, China, in 2012. He is currently an Associate Professor with Northeast Forestry University. His research interests include nonlinear control theory and application, image segmentation, and swarm optimization algorithm.

XIAOXU PENG was born in Harbin, China, in 1995. He is currently pursuing the M.S. degree in control theory and control engineering with Northeast Forestry University, China. His research interests include image segmentation and swarm intelligence algorithm.

WENLONG SONG received the Ph.D. degree in mechanical design and theory from Northeast Forestry University, China, in 2008, where he is currently a Professor. His research interests include image segmentation and swarm intelligence algorithm.



CHUNBO LANG was born in Shenyang, China, in 1998. He is currently pursuing the B.S. degree in automation with Northeast Forestry University, China. His research interests include image segmentation and swarm intelligence algorithm.

ZHIKAI XING was born in Daqing, China, in 1993. He is currently pursuing the M.S. degree in control engineering with Northeast Forestry University, China. His research interests include image segmentation and swarm intelligence algorithm.

KANGJIAN SUN was born in Jinzhou, China, in 1996. He is currently pursuing the M.S. degree in control engineering with Northeast Forestry University, China. His research interests include image segmentation and swarm intelligence algorithm.