scientific reports

OPEN

A multi-strategy improved crow search algorithm for multi-level thresholding image segmentation

Xiaoping Zhang[1], Chengliang Huang[2]✉ & Weixia Gui[3,4]

The standard crow search algorithm suffers from low convergence accuracy, insufficient stability, and susceptibility to getting stuck in local optima. To tackle these formidable challenges, this paper proposes a novel multi-strategy improved crow search algorithm (MSICSA) specifically designed for multi-level image segmentation. The proposed approach incorporates three key enhancements: firstly, opposition-based learning (OBL) is utilized to improve the quality of initial solutions within MSICSA; secondly, an adaptive awareness probability mechanism is introduced to better balance the trade-off between exploration and exploitation; lastly, two differential mutation operators are developed to enhance global search capabilities, increase population diversity, and reduce the risk of converging on local optima. To validate the performance of the proposed algorithm, two sets of experiments are conducted. In the first set of experiments, CEC 2020 benchmark test functions are selected to compare the performance of MSICSA with other group intelligent optimization algorithms. In the second set of experiments, Otsu's method and fuzzy entropy are employed as objective functions for performing multilevel threshold segmentation on twelve grayscale images. The experimental results demonstrate that MSICSA outperforms seven comparison algorithms in terms of both convergence speed and segmentation quality.

Image segmentation technology is a fundamental step in image analysis and processing, and the quality of image segmentation directly impacts the accuracy and effectiveness of image analysis and processing, such as feature extraction, image recognition, and image measurement. Therefore, image segmentation finds applications in various fields, including facial recognition, medical image processing, agricultural plant disease and pest monitoring, and computer vision. For instance, in the domain of medical image processing, image segmentation technology plays a crucial role in accurately delineating regions of interest within medical images. This capability provides a vital foundation for subsequent disease diagnosis, treatment planning, and pre-evaluation. Not only does it enhance the accuracy and efficiency of medical image analysis, but it also offers physicians more objective and quantitative diagnostic information. Consequently, this advancement improves both the precision of disease diagnoses and the effectiveness of treatment interventions. The primary methods for image segmentation include threshold-based segmentation methods, edge-detection-based segmentation methods[1], clustering-based methods[2], fusion-based multi-feature processing methods[3] and deep learning-based methods[4]. Although deep learning-based methods can achieve good segmentation results, they are often accompanied by challenges such as prolonged training times, high algorithmic complexity, and intricate structures. On the contrary, the threshold segmentation methods are widely employed due to their simplicity, operational efficiency, segmentation effectiveness, and stability. However, the challenge with threshold segmentation methods lies in selecting the appropriate thresholds to achieve optimal image segmentation, especially in multi-level thresholding segmentation (MLTS). As the number of thresholds increases, the computational cost associated with threshold segmentation algorithms escalates, making it increasingly difficult to identify the optimal solution within a reasonable time. To address this issue of high time complexity inherent in MLTS when determining suitable thresholds, researchers have turned to meta-heuristic optimization algorithms[5–7] aimed at enhancing the speed of the segmentation process. Over the past decade, a variety of meta-heuristic algorithms have been employed in MLTS. These algorithms include Whale Optimization Algorithm (WOA)[8], the Gray Wolf Optimizer (GWO)[9], Efficient Krill Herd Algorithm (EKH)[10], Human Mental Search Algorithm (HMS)[11], Symbiotic Organisms Search Algorithm[12] and Gradient-Based Method (GM)[13]. Mousavirad et al. conducted a

[1]School of Computer, Electronics and Information, Guangxi University, Nanning 530004, China. [2]Information Technology Management Department, Toronto Metropolitan University, Toronto M5B 2K3, Canada. [3]School of Big Data and Artificial Intelligence, Guangxi University of Finance and Economics, Nanning 530004, China. [4]Guangxi Key Laboratory of Big Data in Finance and Economics, Guangxi University of Finance and Economics, Nanning 530004, China. ✉email: c28huang@torontomu.ca

study that focuses on the evaluation of population-based algorithms[14] specifically for multi-level variance-based image thresholding. The study conducted a comparative analysis of 23 population-based algorithms utilizing a standardized benchmark image set. The evaluation was based on various image quality metrics as well as statistical tests. The results of the experiment demonstrated that imperialist competitive algorithm and gorilla troops optimizer significantly outperformed all other compared approaches.

These algorithms have demonstrated encouraging outcomes in improving both segmentation quality and efficiency. To further enhance accuracy and convergence speed, a variety of improved algorithms have been implemented in MLTS. Wu et al. proposed an improved teaching and learning optimization algorithm (DI-TLBO)[15] using mutation and crossover mechanisms for MLTS. Comparative experiments showed that DI-TLBO outperformed other comparative algorithms in terms of solution accuracy and segmentation stability. Basu Dev et al. had developed a randomized spiral-based whale optimization algorithm for image segmentation[16]. This approach demonstrated impressive predictability in determining the most suitable threshold value. Table 1 lists some recent improved meta-heuristic algorithms for MLTS (Table 2).

In 2016, Askarzadeh proposed the Crow Search Algorithm (CSA)[30], a meta-heuristic algorithm inspired by the foraging behavior of crows. The CSA presents several advantages, including a straightforward structure, minimal parameter tuning requirements, ease of implementation, high efficiency, and robust global search capabilities. Upadhyay and Chhabra applied CSA to MLTS of grayscale images using Kapur method as the objective function[31]. The experimental results showed that CSA performed better overall than particle swarm optimization (PSO), GWO, differential evolution (DE), cuckoo search (CS), and moth flame optimization (MFO). CSA was applied to an adaptive neural fuzzy inference system for network intrusion detection[32].Analysis showed that its detection rate reached 95.8%, and the false alarm rate was only 3.45%, achieving good detection results. However, like other meta-heuristic algorithms, CSA also has problems such as slow convergence speed and being prone to falling into local optima. Therefore, to overcome the limits of CSA, many scholars have made improvements to it. Farid and Hamdi presented a modified CSA based on adaptive flight length adjustment and innovative individual selection to solve the economic load scheduling problem[33]. Surendar and Ponni introduced a hybrid deep neural network adaptive sine cosine CSA algorithm for the diagnosis of lung cancer[34]. CSA is widely used in various engineering problems, but there is relatively little research in the field of MLTS,

Refs.	Years	Description	Objective function	Results
17	2023	This paper proposed a boosted crow search algorithm(VMCSA) using variable neighborhood descent and information exchange mutation strategies	2D's Renyi entropy	The VMCSA demonstrated a notable advantage in segmentation outcomes and exhibits superior robustness compared to the nine other algorithms evaluated
18	2023	A crossover-enhanced optimizer that augments the search capabilities of the artificial electric field algorithm	Kapur	The proposed method demonstrated a superior capacity to skip local optima and exhibited exceptional overall performance in comparison to other comparative schemes
19	2023	A hybrid method combining quantum computing and optimal foraging algorithm is proposed	Kapur	The experimental results obtained from real abdominal CT liver images demonstrated that this scheme offered superior segmentation outcomes when compared to the standard Optimal Foraging Algorithm
20	2023	This work proposed an improved heap-based optimizer(IHBO) based on opposition-based learning	Otsu and Kapur	The proposed algorithm demonstrated superior performance compared to its counterparts, particularly in terms of fitness values. Additionally, it excelled in other performance indicators, such as the structural similarity index, feature similarity index, and peak signal-to-noise ratio
21	2023	This manuscript integrated opposition-based learning into the Runge Kutta optimization algorithm and proposes an improved algorithm(RUN_OBL)	Minimum cross-entropy	This algorithm had the capability to escape from suboptimal solutions, particularly in high-dimensional problems. Furthermore, it exhibited superior quality and consistency for segmentation purposes
22	2024	A hybrid algorithm was proposed which used the chaos theory and the modified cuckoo search	Otsu, Kapur, Tsallis and Cross entropy	The proposed scheme demonstrates a notable improvement over several state-of-the-art methods, both in terms of quantitative outcomes and segmented output
23	2024	An enhanced differential evolution algorithm was proposed, which used the Cauchy inverse cumulative distribution to adjust the scaling factor	Fuzzy entropy	This approach improved segmentation accuracy compared to the other compared algorithms
24	2024	An enhanced Snake Optimization algorithm (SO_OBL) was suggested by using opposition-based learning	Otsu	This algorithm had achieved impressive segmentation metric results, including FSIM = 0.947, SSIM = 0.941, PSNR = 24.876, MSE = 236.88, and an execution time of 0.281 s. These outcomes highlight the model's efficiency and its promising potential for accurate diagnosis in CAD systems
25	2024	A hybrid optimization approach combining the golden jackal algorithm with a sine cosine algorithm (SCGJO) has been proposed	Kapur	The SCGJO outperforms other algorithms, exhibiting a faster convergence rate, enhanced computational accuracy, superior segmentation quality, and greater stability
26	2024	This study improved Slime mould algorithm (SMA_MLS) using the mechanism of leadership and self-phagocytosis	Kapur	The proposed scheme demonstrated superiority in terms of convergence accuracy, convergence speed, and stability
27	2024	This work proposed a complex-valued encoding golden jackal optimization(CGJO)	Kapur	The CGJO effectively explored and exploited its capabilities to achieve superior convergence speed, enhanced numerical precision, and improved segmentation quality
28	2024	An enhanced Exponential Distribution Optimizer was introduced based on two opertors: phasor operator and adaptive p-best mutation strategy	Otsu and Kapur	The proposed algorithm surpassed the competing methods across a range of statistical measures
29	2024	An improved honey badger algorithm was proposed by using Enhanced Solution Quality	Otsu and Kapur	This scheme demonstrated an impressive overall capacity, particularly in terms of optimization accuracy and convergence speed

Table 1. Recent algorithms proposed for solving MLTS.

Type	No.	Description	f_{min}
Unimodal function	1	Shifted and rotated Bent Cigar function	100
Multimodal shifed and rotated functions	2	Shifted and rotated Schwefel's function	1100
	3	Shifted and rotated Lunacek bi-Rastrigin function	700
	4	Expanded Rosenbrock's plus Griewangk's funciton	1900
Hybrid functions	5	Hybrid function 1 (N = 3)	1700
	6	Hybrid function 2 (N = 4)	1600
	7	Hybrid function 3 (N = 5)	2100
Composition functions	8	Composition function 1 (N = 3)	2200
	9	Composition function 2 (N = 4)	2400
	10	Composition function 3 (N = 5)	2500

Table 2. Description of CEC 2020 benchmark functions.

it is necessary to further improve the application of CSA in the field of MLTS. Although the multi-threshold segmentation algorithms mentioned above have obtained many effective results, there are still problems such as insufficient solution accuracy, lack of stability, and being prone to falling into local optima. There is still some room for improvement. And according to No Free Run (NFL)[35]: no single algorithm can outperform other algorithms in all problems. This motivates us to improve CSA for MLTS of grayscale images, to achieve high accuracy and stability, and to avoid falling into local optima. The main contributions and innovations of this study can be summarized as follows:

(1) A novel multi-strategy improved crow search algorithm (MSICSA) is proposed. There are mainly three strategies to improve the performance of CSA: firstly, using OBL strategy to improve the quality of initial solutions; secondly, adaptive awareness probability is introduced to better balance the exploitation and exploration during iterative process; thirdly, the diversity of the search space is further expanded through two differential mutation operators, which facilitate jumping out of local optima.
(2) The CEC 2020 benchmark function sets have been selected to compare the performance of MSICSA with other algorithms, thereby validating the effectiveness of MSICSA.
(3) The performance of image segmentation using the MSICSA method is evaluated by employing 12 grayscale images, with Otsu's method and fuzzy entropy serving as objective functions.
(4) The quality of image segmentation is assessed using PSNR, SSIM, and FSIM metrics.

The remainder of this paper is structured as follows: "Related works" section discusses the theory of multi-level thresholding based on Otsu's method and fuzzy entropy, along with standard CSA. "The proposed MSICSA" section provides a detailed introduction to the proposed scheme. In the "Experiments on CEC 2020 benchmark functions" section, an experimental analysis of the proposed algorithm is presented utilizing the CEC 2020 test function set, accompanied by a comparative evaluation against other algorithms. The "Experiments for image segmentation" section applies the proposed algorithm to various images for multi-level threshold segmentation using Otsu's method and fuzzy entropy as objective functions, followed by a comprehensive analysis and discussion of the experimental results. Finally, in the "Conclusions and future works" section, this study concludes appropriately while also highlighting potential avenues for future research.

Related works
This section will review the Otsu's method, fuzzy entropy and the crow search algorithm in detail.

The Otsu's method
The Otsu's method is a non-parametric image segmentation algorithm proposed in 1979[36]. This algorithm uses the maximum variance between different classes as the threshold selection criterion, and it is entirely based on the histogram of the image for calculation.

Let image I has a total of L gray levels, and *NP* represents the total number of pixels in the image, h_i denotes the number of intensity in the image histogram. The probability distribution intensity Ph_i is calculated using Eq. (1):

$$Ph_i = \frac{h_i}{NP}, \sum_{i=1}^{NP} Ph_i = 1 \tag{1}$$

For bi-level thresholding segmentation, it is only necessary to find a suitable threshold *t* to segment the image into two classes C_0 and C_1. The purpose of the Otsu's method is to maximize Eq. (2):

$$f(t) = \sigma_0 + \sigma_1 \tag{2}$$

where σ_0 and σ_1 represent the variance of C_0 and C_1, and they can be computed using Eq. (3) to Eq. (7).

$$C_0=\frac{Ph_1}{\omega_0},\frac{Ph_2}{\omega_0},\ldots,\frac{Ph_t}{\omega_0},C_1=\frac{Ph_{t+1}}{\omega_1},\ldots,\frac{Ph_L}{\omega_1} \tag{3}$$

$$\omega_0=\sum\nolimits_{i=1}^{t}Ph_i,\omega_1=\sum\nolimits_{i=t+1}^{L}Ph_i \tag{4}$$

$$\mu_0=\sum\nolimits_{i=1}^{t}\frac{iPh_i}{\omega_0},\mu_1=\sum\nolimits_{i=t+1}^{L}\frac{iPh_i}{\omega_1} \tag{5}$$

$$\sigma_0=\omega_0(\mu_0-\mu_T)^2,\sigma_1=\omega_1(\mu_1-\mu_T)^2 \tag{6}$$

$$\mu_T=\omega_0\mu_0+\omega_1\mu_1 \tag{7}$$

When extending the problem to MLTS, m thresholds (t_1,t_2,…,t_m) are needed to divide the image into $m+1$ classesC_0,C_1,…,C_m , and its fitness function $f(t_1,t_2,\ldots,t_m)$ can be shown as Eq. (8).

$$f\left(t_1,t_2,\ldots,t_m\right)=\sigma_0+\sigma_1+\cdots+\sigma_m \tag{8}$$

where,

$$\sigma_0=\omega_0(\mu_0-\mu_T)^2,\omega_0=\sum\nolimits_{i=1}^{t_1-1}Ph_i,\mu_0=\sum\nolimits_{i=1}^{t_1-1}\frac{iPh_i}{\omega_0} \tag{9}$$

$$\sigma_1=\omega_1(\mu_1-\mu_T)^2,\omega_1=\sum\nolimits_{i=t_1}^{t_2-1}Ph_i,\mu_1=\sum\nolimits_{i=t_1}^{t_2-1}\frac{iPh_i}{\omega_1} \tag{10}$$

$$\sigma_m=\omega_m(\mu_m-\mu_T)^2,\omega_m=\sum\nolimits_{i=t_m+1}^{L}Ph_i,\mu_m=\sum\nolimits_{i=t_m}^{L}\frac{iPh_i}{\omega_m} \tag{11}$$

$$\mu_\mathrm{T}=\omega_0\mu_0+\omega_1\mu_1 \tag{12}$$

where ω_0, ω_1,…, and ω_m are the probability of class appearing in the image, and μ_0, μ_1, …, μ_m denote the average gray value of each class, μ_T is the average gray value of the original image. The goal of MLTS using the Otsu's method is to select m thresholds (t_1,t_2,…,t_m) to maximize function shown as Eq. (8).

Fuzzy entropy

The classical set V can be understood as a collection of elements that either fully belong to V or do not belong at all. In contrast, the fuzzy set[37] V serves as an extension of the classical concept, allowing for the possibility that an element may partially belong to the set V. The expression for V is given in Eq. (13).

$$\mathrm{V}=\{(\mathrm{x},\mu\mathrm{v}\left(\mathrm{x}\right))\mid\in\mathrm{X}\} \tag{13}$$

where, $\mu V(x)$ refers to the membership function, which takes on values ranging from 0 to 1. The value of $\mu V(x)$ indicates how closely x aligns with V. To simplify our approach, we will utilize a trapezoidal membership function in this project to estimate the number of segmented regions within the fuzzy membership (m), represented as (μ_1, μ_2, ⋯,μ_m). The number of unknown fuzzy parameters is given by $2\times(m-1)$. These parameters will be denoted as a_1, c_1,…, a_{m-1}, c_{m-1}. Besides, their values satisfy $0<a_1<c_1<a_2<c_2<\ldots<a_{m-1}<c_{m-1}\le L-1$, where L represents the grey level and is equal to 256. Figure 1 presents an example of a fuzzy membership function for 5 threshold levels (m = 5). The derivations shown in Fig. 1 are detailed in Eqs. (14), (15), and (16)[38].

$$\mu_1\left(k\right)=\begin{cases}1 & k\le a_1\\ \frac{k-a_1}{c_1-a_1} & a_1<k\le c_1\\ 0 & k>c_1\end{cases} \tag{14}$$

$$\mu_{m-1}\left(k\right)=\begin{cases}0 & k\le a_{m-2}\\ \frac{k-a_{m-2}}{c_{m-2}-a_{m-2}} & a_{m-2}<k\le c_{m-2}\\ 1 & c_{m-2}<k\le a_{m-1}\\ \frac{k-c_{m-1}}{a_{m-1}-c_{m-1}} & a_{m-1}<k\le c_{m-1}\\ 0 & k>c_{m-1}\end{cases} \tag{15}$$

$$\mu_m\left(k\right)=\begin{cases}0 & k\le a_{m-1}\\ \frac{k-a_{m-1}}{c_{m-1}-a_{m-1}} & a_{m-1}<k\le c_{m-1}\\ 1 & k>c_{m-1}\end{cases} \tag{16}$$

$$H_1=-\sum\nolimits_{i=0}^{L-1}\frac{Ph_i*\mu_1\left(i\right)}{P_1}*\ln\left(\frac{Ph_i*\mu_1\left(i\right)}{P_1}\right) \tag{17}$$

$$H_{m-1}=-\sum\nolimits_{i=0}^{L-1}\frac{Ph_i*\mu_{m-1}\left(i\right)}{P_{n-1}}*\ln\left(\frac{Ph_i*\mu_{m-1}\left(i\right)}{P_{m-1}}\right) \tag{18}$$

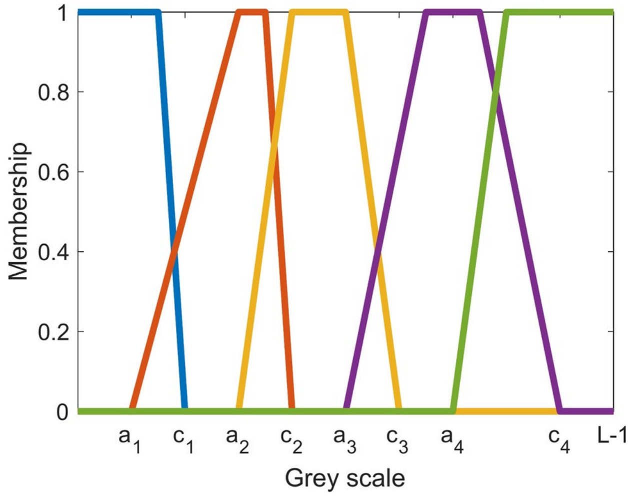


Fig. 1. Fuzzy membership function for 5 threshold level.

$$H_m = -\sum_{i=0}^{L-1} \frac{Ph_i * \mu_m(i)}{P_m} * \ln\left(\frac{Ph_i * \mu_m(i)}{P_m}\right) \tag{19}$$

where, Ph_i is the probability distribution intensity.

$$P_1 = \sum_{\mathrm{i=0}}^{\mathrm{L-1}} Ph_{\mathrm{i}} \times \mu_1(i) \tag{20}$$

$$P_{m-1} = \sum_{\mathrm{i=0}}^{\mathrm{L-1}} Ph_{\mathrm{i}} \times \mu_{\mathrm{m-1}}(i) \tag{21}$$

$$P_{\mathrm{m}} = \sum_{\mathrm{i=0}}^{\mathrm{L-1}} Ph_{\mathrm{i}} \times \mu_{\mathrm{m}}(i) \tag{22}$$

The optimal parameter values will be obtained by maximizing the total entropy. The global optimization method is shown in Eq. (23), where Argmax is an operation to determine the argument that obtains the maximum value from the target function. Moreover, the threshold values t_i can be easily obtained using the fuzzy parameters as shown in Eq. (24).

$$\varphi(\mathrm{a}_1, \mathrm{c}_1, \ldots, \mathrm{a}_{m-1}, \mathrm{c}_{\mathrm{m}-1}) = \mathrm{Argmax}([\mathrm{H}_1 + \mathrm{H}_2 + \cdots + \mathrm{H}_{\mathrm{m}}]) \tag{23}$$

$$t_i = \frac{a_i + c_i}{2}\, i = 1, 2, \ldots, m-1 \tag{24}$$

Crow search algorithm (CSA)

CSA is a novel meta-heuristic algorithm that simulates the behavior of crow populations. The crow is considered the smartest bird, with a brain to body ratio greater than other birds. Crows can hide food, and in order to obtain a better source of food, they will observe the location where other crows hide food and steal it after the owner leaves. Therefore, CSA follows four principles: (1) Crows live in groups; (2) Crows can remember where they hide their food; (3) Crows will track each other and steal their food; (4) Crows have a certain probability of protecting their food from theft.

Assuming the population size is N, D denotes dimension. The position of crow *i* in iteration g is represented by a vector $X_i^g = [x_1^g, x_2^g, \ldots, x_D^g]$, representing a feasible solution in the search space. A crow *i* randomly selects another crow, such as *j*, to track and discover the location where it hides its food. There are two possible cases that may occur:

Case 1 Crow *j* is not aware of crow *i* tracking. Crow *i* finds the location where crow *j* stores food, and crow *i* updates the location by Eq. (25):

$$X_i^{g+1} = X_i^g + r_i \times fl \times \left(M_j^g - X_i^g\right) \tag{25}$$

where fl represents the flight length, and r_i is a random number with intervals between [0, 1]. Using a D-dimensional memory vector $M_{\mathrm{j}}^{\mathrm{g}}$ to represent the location where crow *j* stores the food in its memory.

Case 2 Crow *j* discovers the tracking of crow *i* and randomly flies to a random location to mislead crow *i*.

The conversion between these two cases mainly depends on the awareness probability *AP*, which can be represented by Eq. (26).

$$X_i^{g+1} = \begin{cases} X_i^g + r_i \times fl \times \left(M_j^g - X_i^g\right) & if rj \geq AP \\ arandomposition & otherwise \end{cases} \tag{26}$$

where r_j is a random value with uniformly distributed between [0,1].

Supposed that *f(.)* is an objective function used to measure the fitness of candidate solutions, the update of memory positions is shown in Eq. (27).

$$M_i^{g+1} = \begin{cases} X_i^{g+1} & iff\left(X_i^{g+1}\right) isbetterthan f\left(M_i^g\right) \\ M_i^g & otherwise \end{cases} \tag{27}$$

The proposed MSICSA

In order to improve the optimization performance of CSA, it is necessary to find a good balance between exploitation and exploration. This paper proposes three improvement strategies: opposition-based learning, adaptive awareness probability, and two differential mutation operators to improve the CSA. The joint application of these three strategies can improve the accuracy of algorithm optimization and the ability to jump out of local optima, better balancing the relationship between global search and local search during the iteration process.

Opposition-based learning (OBL)

OBL is a technique proposed by Tizhoosh[39] to search for its corresponding opposition solution through the current feasible solution. The advantage of doing so is that it can expand the search scope and improve the optimization efficiency of the algorithm. According to the probability theorem, there is a 50% probability that the reverse solution is closer to the optimal solution of the problem to be solved than the current solution. OBL is commonly utilized to improve various metaheuristic algorithms, such as equilibrium optimizer[40], marine predators algorithm[41],emperor penguin optimization[42] and Sailfish optimizer[43]. It plays a significant role in improving these algorithms during different stages. In the initial phase, OBL aims to achieve a higher quality initialization population. During the optimization iteration stage, it helps expand the search range and enhances global search capabilities.

The calculation formula for the opposition solution is shown in Eq. (28).

$$\overline{\mathrm{X_i}} = \mathrm{Lb} + \mathrm{Ub} - \mathrm{X_i} \tag{28}$$

Lb and Ub are the lower and upper bounds of the search space respectively, and $\overline{X_i}$ is corresponding opposition value of X_i . In CSA, the initial solutions of the population are generated randomly. Although this method is simple and easy to use, it has blindness and sometimes cannot cover some good feasible solutions, which will directly reduce the optimization performance of the algorithm. To improve this issue, OBL is applied to the population initialization process. The initially generated random solutions and their reverse solutions are merged, and they are sorted according to their fitness values. Half of the candidate solutions with better fitness values are selected as the initial solutions. To a certain extent, this approach can avoid blindness, obtain high-quality initial solutions, and improve the search efficiency of the algorithm. The pseudo-code of initialization algorithm using OBL is shown in Algorithm 1.

Input: N: population size

Output: the set of initial individuals S^*

1: Randomly generate N random individuals and form a set $S \leftarrow \{X_1, X_2, \ldots, X_N\}$

2: Calculate the OBL individuals by Eq. (28) for each random individual and form a set $\overline{S} \leftarrow \{\overline{X_1}, \overline{X_2}, \ldots, \overline{X_N}\}$

3: $S \leftarrow S \cup \overline{S}$

4: Calculate the fitness value of each individual in set S

5: Individuals in S are sorted depending on their fitness;

6: N better individuals are selected from set S to form a set S^*

7: Return S^*

Adaptive awareness probability

Awareness probability *AP* is an important parameter in CSA, which balances the reduction and diversification of iteration processes. When the *AP* is relatively small, the algorithm focuses on local search. Otherwise, the algorithm focuses on global search. However, in the CSA, this parameter is usually set to a fixed value such as 0.1, which does not meet the requirements of the optimization process. Here, an adaptive awareness probability is proposed, where the AP value varies based on the number of iterations g. The improved AP is represented by Eq. (29).

$$AP = \begin{cases} AP_{max} - \frac{2\times(AP_{max}-AP_{min})\times g}{gmax} & if g < gmax/2 \\ AP_{min} + \frac{2\times(AP_{max}-AP_{min})\times g}{gmax} & otherwise \end{cases} \tag{29}$$

where AP_{min}, $AP_{max} \in [0,1]$ are two preset constants that represent the minimum and maximum values of perception probability respectively, and $gmax$ is the maximum number of iterations of the algorithm.

The improved awareness probability AP linearly decreases in the first half of the iterations, which is beneficial for the algorithm to transition from global search to local search. In the latter half of the algorithm, the candidate solutions have high repeatability. They are prone to rigidity and premature convergence. The AP increases linearly with the number of iterations, which is beneficial for improving the diversity of candidate solutions, increasing the probability of global search, and facilitating jumping out of the local optimal solution.

Two differential mutation operators

In the CSA, its local search mainly utilizes M_j^g as a guide, and the update method is relatively single. This update method does not fully utilize the global optimal solution currently found and prior knowledge of the best location for other crows to store food as a guide, making it easy to precocious and fall into local optima. In order to improve the diversity of the population and enhance global search ability, we consider using the global optimal solution obtained during the current iteration and the best location for other crows to store food as guidance. We propose two individual update methods, called differential mutation operators, as shown in Eq. (30) and Eq. (31).

$$X_i^{g+1} = X_i^g + GR \times (M_{r_1}^g - M_{r_2}^g) \tag{30}$$

$$X_i^{g+1} = X_i^g + GR \times (X_{best} - M_{r_1}^g) \tag{31}$$

where $GR = (gr_1, gr_2, \ldots, gr_D)$ is a D-dimensional vector, and gr_j ($j = 1,2,\ldots,D$) is the Gaussian random number. r_1 and r_2 are random numbers between 1 and N, and r_1 , r_2 , i are not equal to each other; X_{best} denotes the global optimal solution found at the current iteration; $M_{r_1}^g$ and $M_{r_2}^g$ are two randomly selected memory positions in the population. Gaussian random numbers are characterized by a normal distribution. The purpose of utilizing GR is to apply these random numbers to the original position vector, thereby generating a new position. Most of the newly generated locations tend to be situated close to the original location, which effectively resembles a neighborhood search within a limited range. Simultaneously, some new positions may be located further away from the current point, enhancing population diversity and facilitating exploration of potentially advantageous areas. This approach ultimately aims to improve search efficiency and accelerate the convergence trend of the optimization process.

Pseudo-code and flowchart of MSICSA

The modified algorithm is designed by using three strategies mentioned in above sections. The combined use of three strategies will help improve the algorithm's global search ability, convergence speed, and robustness. The pseudo-code and flowchart of MSICSA are presented in Algorithm 2 and Fig. 2.

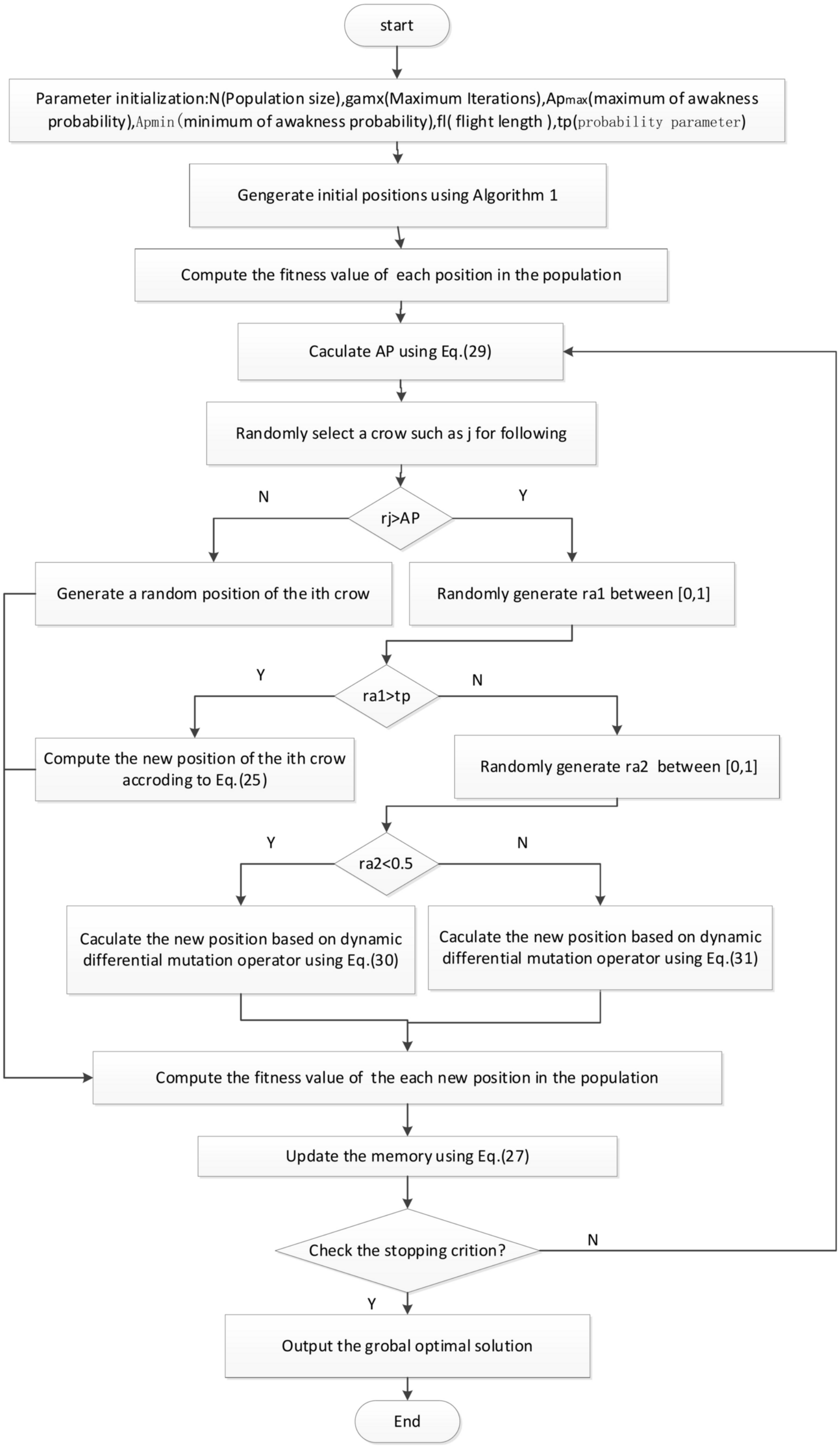


Fig. 2. Flowchart of MSICSA.

Input: N: population size; D: number of dimensions; $gmax$: max number of iterations; fl: flight length; AP_{max}: maximum value of awareness probability; AP_{min}: minimum value value of awareness probability; tp: probability parameter

Output: the optimal solution X_{best}

Generate initial solutions using Algorithm 1
Initialize the memories of N crows
for $g \leftarrow 1$ *to* $gmax$ **do**
    Update AP using Eq.(29)
    for $i \leftarrow 1$ *to* N **do**
        if $rj \geq AP$ **then**
            Randomly select $ra1$ from [0,1]
            if $ra1 < tp$ **then**
                Generate the position of each crow according to Eq.(25)
            else
                Randomly select $ra2$ from [0,1]
                if $ra2 < 0.5$ **then**
                    Update X_i^{g+1} according to Eq.(30)
                else
                    Update X_i^{g+1} according to Eq.(31)
                end if
            end if
        else
            Randomly select a crow for following
    end for
    Calculate the fitness value of new population
    Update the memory of crows according to Eq.(27)
    Update the optimal solution X_{best}
end for
Return X_{best}

Time complexity analysis of MSICSA

This section discusses the computational complexity of MSICSA. The sorting algorithm employed during population initialization via OBL exhibits a complexity of $O(N \times \log N)$. Consequently, the overall time complexity for this initialization process can be articulated as $O(N \times D + N \times \log N)$, where D and N represent the dimension of the problem and the size of the population, respectively. The time complexity of the iterative optimization process is $O(N \times D \times gmax)$, where gmax reprents the maximum number of iterations. The overall time complexity of MSICSA is $O(N \times D + N \times \log N + N \times D \times gmax) = O(N \times \log N + N \times D \times gmax)$.

Experiments on CEC 2020 benchmark functions

In this section, the CEC 2020 benchmark functions are used to evaluate the optimization performance of the proposed algorithm.

CEC 2020 benchmark functions description

CEC 2020 benchmak functions include 10 functions of four types. These four different types of functions will provide a comprehensive evaluation of the exploitation and exploration performance of the proposed algorithm. Table 2 provides the description of the types and details of the 10 functions in CEC 2020. f_{min} is the global optimal value. In order to facilitate understanding the differences and properties of each function, Fig. 3 shows

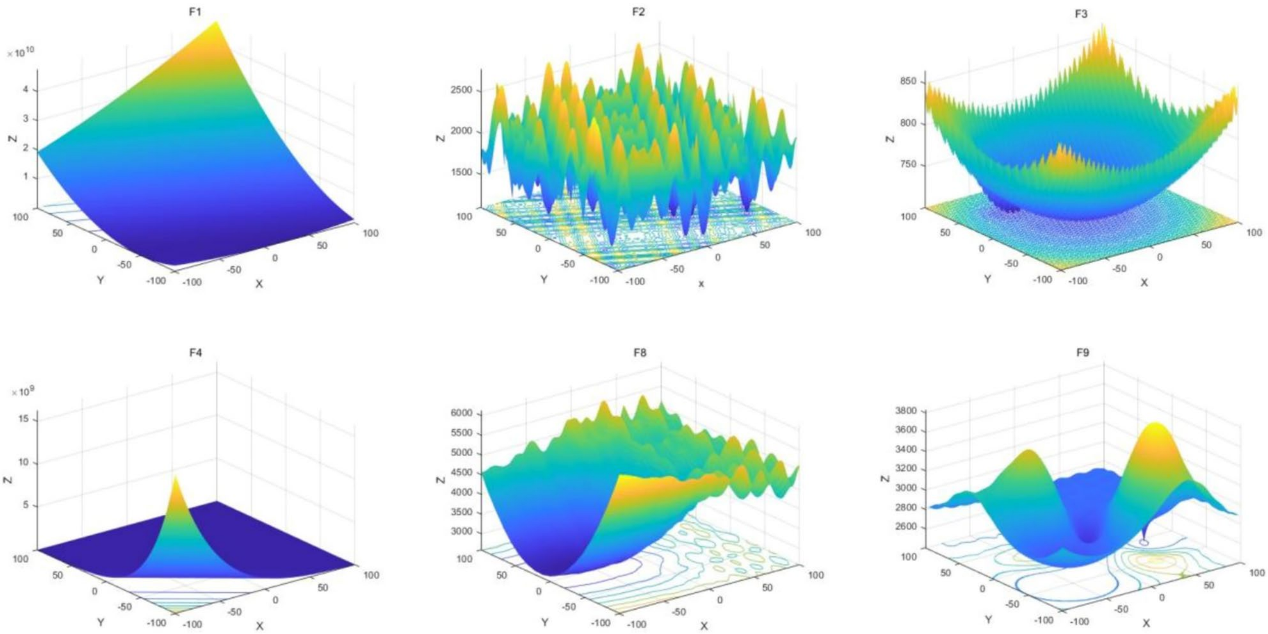


Fig. 3. 2D visualization of six CEC2020 benchmark functions.

No.	tp = 0.4	tp = 0.5	tp = 0.6	tp = 0.7	tp = 0.8	tp = 0.9
F1	1775.36	**1414.99**	1632.35	1760.28	2131.19	1954.94
F2	2988.74	3133.92	3010.37	3144.14	**2949.86**	3216.63
F3	**760.59**	762.86	764.83	776.30	771.72	778.21
F4	1904.95	1903.22	1903.54	1903.09	**1903.06**	1904.34
F5	65846.18	30990.06	16819.03	9278.47	7250.65	**6540.84**
F6	**1775.01**	1784.91	1862.67	1850.60	1846.24	1924.60
F7	12867.66	7136.48	6351.44	5726.74	5333.91	**5044.55**
F8	2300.61	2300.60	2300.37	2300.34	**2300.26**	2300.60
F9	2845.70	**2835.81**	2845.05	2847.19	2838.12	2868.03
F10	2961.16	2958.36	**2954.05**	2963.16	2954.22	2954.76
Friedman average rank	3.60	3.15	3.20	3.80	2.80	4.45
Final rank	4	2	3	5	1	6

Table 3. Table of test results for *tp*. Best values are in bold.

the 2D visualization of six CEC 2020 benchmark functions. For each function, the range of values for each dimension is between [– 100,100].

Parameter sensitivity testing

The proposed MSICSA introduces three additional parameters in comparison to standard CSA , namely AP_{min}, AP_{max} and *tp*. It is imperative that these parameters undergo rigorous testing to ascertain their optimal values. Table 3 presents the average fitness values of each function for *tp* ranging from 0.4 to 0.9, and each function has be tested 30 times. The Friedman ranking of each result is given in the last row. The test result with $tp = 0.8$ ranks first, therefore, $tp = 0.8$ is a more suitable value. The evaluation of other additional parameters follows a process akin to that of the *tp* and will not be reiterated here.

Parameter settings

To evaluate the effects of OBL, adaptive awareness probability, and differential mutation operators on the standard CSA, ablation studies will be conducted. Three variant CSA are generated using two strategies: DSICSA1, DSICSA2 and DSICSA3, and the specific details of which are presented in Table 4. In this table, 'Y' denotes that the strategy is employed, while 'N' indicates that it was not adopted. These variants were subsequently subjected to comparative testing. To assess the effectiveness of the proposed MSICSA, it is compared with seven other meta-heuristic algorithms. These algorithms include: CSA, DSICSA1, DSICSA2, DSICSA3, Particle Swarm Optimization(PSO)[44], Sine Cosine Algorithm(SCA)[45], Snake Optimization(SO)[46] and Golden Jackal Optimization(GJO)[47]. All algorithms are implemented in MATLAB R2022a on a personal computer running

Algorithm	OBL	Adaptive awareness probability	Differential mutation operators
DSICSA1	Y	N	Y
DSICSA2	N	Y	Y
DSICSA3	Y	Y	N

Table 4. Different variants using different strategies.

Algorithm	Parameters
Common parameters	Population size: 30, maximum evaluation: 1000, problem dimensions: 20, Number of independent runs : 30
MSICSA	$AP_{min}=0.01, AP_{max}=0.2, fl=2, tp=0.8$
CSA	$AP=0.1, fl=2$
DSICSA1	$AP=0.1, fl=2$, tp = 0.8
DSICSA2	$AP_{min}=0.01, AP_{max}=0.2, fl=2, tp=0.8$
DSICSA3	$AP_{min}=0.01, AP_{max}=0.2, fl=2,$
PSO	w = 1,c1 = 1.5,c_2 = 1.5
SCA	a = 2
SO	$\theta_1=0.25$, $\theta_2=0.6$,$c_1=0.5$, $c_2=0.5$, $c_3=2$
GJO	$c_1=1.5$,$\beta=1.5$

Table 5. Parameter setting for experiments on CEC 2020 benchmark functions.

windows 10 64bit system with 32 GB RAM and 2.5 GHz CPU. The parameters utilized in CSA, PSO, SCA, SO and GJO are consistent with those reported in the published papers. Table 5 shows the parameter settings for each algorithm.

Performance evaluation

The standard deviation(STD) and mean fitness of CEC2020 benchmark functions calculated by nine different algorithms when D = 20 are shown in Table 6. The final row displays the Friedman ranking. It can be observed that MSICSA obtains the minimum mean values in F1, F3, F4,F6,F7,F8 and F10. For F2, the mean value of MSICSA is higher than SO, but better than the other seven algorithms. Similar to the cases of F5 and F9, the mean value of MSICSA is only worse than CSA in F5 and higher than DSICSA2 in F9, which is smaller than the rest of the algorithms. Considering the STD values, MSICSA gets minimum values in F1, F3, F4, F6, F7 F8 and F10 compared with other methods, indicating that MSICSA has the strongest robustness in these functions.

From the perspective of the Friedman final ranking, the rankings of MSICSA, DSICSA1, DSICSA2, and DSICSA3 are superior to that of CSA. Notably, MSICSA occupies a higher position than DSICSA1, DSICSA2, and DSICSA3. This indicates that each improvement strategy positively contributes to overall performance. In general, DSCSA2 demonstrates superior performance compared to both DSCSA1 and DSCSA3, while the performances of DSCSA1 and DSCSA3 are comparable. This implies that the contribution of OBL in MSICSA is not as significant as that of adaptive awareness probability and Differential mutation operators. This may be attributed to the fact that OBL only operates during the initialization phase and does not participate in the optimization iteration process.

Boxplot analysis

Boxplot is an important tool for displaying the distribution characteristics of a set of data, which can intuitively show the maximum, minimum, median, and upper and lower quartiles of the data. The straight lines extending from the upper and lower ends of the box represent the maximum and minimum values of the data.The rectangle inside the box represents the interval where the median is located. Points outside the whisker line are considered outliers, which are values that are relatively special compared to the overall distribution. The length of the box displays the degree of data dispersion, and the longer the box length, the more dispersed the data. On the contrary, the narrower the box, the more concentrated the data distribution, and the smaller the volatility. From the boxplot shown in Fig. 4, it can be observed that MSICSA obtains the narrowest box length and the smallest boxplot value compared with other approaches in most cases. Especially in F3, F6, F7, F8 and F10, this situation is even more obvious.

Convergence curve analysis

Figure 5 presents the convergence plots of various algorithms applied to the CEC 2020 benchmark functions. FEs is the current number of evaluations. From Fig. 5, it can be found that the curve of MSICSA converges before 2000 evaluations in F1, F3, F7 and F10. The performance of the MSICSA curve demonstrates a notably faster convergence compared to other algorithms, particularly in F3, F7, F8, and F10. This suggests that MSICSA effectively balances the exploration and exploitation capabilities of the algorithm. In F4 and F6, MSICSA first converges to a local optimum, and then jumps out of the local optimum to converge to a near global optimum. Thus MSICSA exhibits the ability to escape from local optima.

No		MSICSA	CSA	DSICSA1	DSICSA2	DSICSA3	PSO	SCA	SO	GJO
F1	Mean	**1.336E+03**	2.261E+03	1.693E+03	1.571E+03	1.864E+03	1.717E+10	4.773E+09	2.432E+03	4.328E+09
	STD	1.048E+03	2.940E+03	1.848E+03	1.394E+03	2.166E+03	4.940E+09	8.505E+08	2.009E+03	1.992E+09
F2	Mean	3.041E+03	3.474E+03	3.333E+03	3.116E+03	3.305E+03	4.805E+03	4.597E+03	**2.099E+03**	3.417E+03
	STD	4.158E+02	7.097E+02	5.158E+02	4.311E+02	4.755E+02	3.169E+02	3.541E+02	2.438E+02	7.081E+02
F3	Mean	**7.682E+02**	8.302E+02	7.840E+02	7.811E+02	7.891E+02	1.153E+03	8.944E+02	7.942E+02	8.349E+02
	STD	1.583E+01	3.155E+01	2.083E+01	1.072E+01	2.220E+01	5.618E+01	1.604E+01	2.066E+01	2.069E+01
F4	Mean	**1.904E+03**	1.994E+03	1.970E+03	1.959E+03	1.984E+03	1.301E+05	2.107E+03	2.013E+03	2.356E+03
	STD	9.174E-01	5.244E+00	2.349E+00	2.289E+00	2.049E+00	1.472E+05	1.796E+02	5.425E+01	5.893E+02
F5	Mean	4.058E+03	**3.235E+03**	9.403E+03	7.315E+03	6.119E+03	1.464E+06	6.335E+05	1.247E+05	5.122E+05
	STD	1.176E+03	5.759E+02	5.766E+03	3.638E+03	3.258E+03	9.876E+05	3.141E+05	9.071E+04	6.971E+05
F6	Mean	**1.825E+03**	2.026E+03	1.863E+03	1.936E+03	1.955E+03	2.695E+03	2.167E+03	2.040E+03	2.084E+03
	STD	1.481E+02	2.178E+02	1.990E+02	1.878E+02	1.697E+02	1.502E+02	1.768E+02	2.256E+02	1.979E+02
F7	Mean	**3.068E+03**	5.247E+03	4.546E+03	4.988E+03	3.527E+03	3.937E+05	2.277E+05	6.982E+04	2.458E+05
	STD	4.916E+02	1.407E+03	1.792E+03	2.057E+03	3.538E+02	2.853E+05	1.964E+05	7.079E+04	3.310E+05
F8	Mean	**2.300E+03**	2.514E+03	2.447E+03	2.436E+03	2.428E+03	5.772E+03	3.710E+03	3.110E+03	3.072E+03
	STD	5.871 E−01	7.742E+01	5.892E+01	5.401E+01	2.798E+01	1.051E+03	1.654E+03	1.004E+03	5.651E+02
F9	Mean	2.855E+03	2.930E+03	2.861E+03	**2.849E+03**	2.884E+03	3.308E+03	2.963E+03	2.867E+03	2.910E+03
	STD	2.100E+01	6.095E+01	3.557E+01	2.143E+01	4.584E+01	1.124E+02	1.799E+01	2.448E+01	2.862E+01
F10	Mean	**2.915E+03**	2.963E+03	2.955E+03	2.950E+03	2.953E+03	4.063E+03	3.083E+03	2.942E+03	3.078E+03
	STD	1.004E+01	3.649E+01	3.302E+01	3.146E+01	3.789E+01	3.293E+02	4.235E+01	3.002E+01	9.553E+01
Friedman average rank		1.30	5.20	3.60	2.70	3.60	9.00	7.80	4.90	6.90
Final rank		1	6	3.5	2	3.5	9	8	5	7

Table 6. Mean and STD of the optimal fitness achieved by different algorithms on the CEC 2020 functions with D = 20 Best values are in bold.

Experiments for image segmentation

In this section, we will first introduce the test images used in the experiments, then list the relevant parameters set by all algorithms, finally provide a series of evaluation criteria.

Test images

This paper uses 12 test images that were randomly picked from two different databases. The first 8 images come from Berkeley publicly available image dataset (url: https://www2.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/resources.html) and are called Img1, Img2, Img3, Img4, Img5, Img6, Img7 and Img8 respectively; while the last 4 images are from X-ray images of COVID-19 (https://github.com/ieee8023/covid-chestxray-dataset) and are named CX1, CX2,CX3 and CX4. The test images and their corresponding histograms are shown in Table 7.

Parameter settings for different algorithms

The proposed algorithm will be compared with the CSA and six other recently and well-known meta-heuristic algorithms, namely: VMCSA[17], IHBO[20], RUN_OBL[21], SO_OBL[24], SCGJO[25] and SMA_MLS[26], all of which have achieved good threshold segmentation results in previous studies. The Otsu's method and fuzzy entropy are taken as the fitness function to select the optimal threshold. In order to test the segmentation ability of the algorithm under different threshold levels, five threshold levels(D) of 4, 8, 12, 16, 20 are used in this study. The purpose of selecting these threshold levels is to test the performance of each algorithm at both low and high threshold levels. The settings of other parameters for each algorithm are shown in Table 8.

Performance evaluation criteria

This study uses five criteria to evaluate the threshold segmentation performance of the proposed algorithm. The explanations of these criteria are as follows:

1. *The mean fitness value* It is the average of fitness values which are obtained by executing the optimization algorithm k times.

$$Mean = \frac{1}{k}\sum\nolimits_{i=1}^{k} fit_i \tag{32}$$

where k is the number of times the algorithm has been run (k = 30). fiti denotes the best fitness value obtained from the ith run of the algorithm, which uses Eq. (7) or Eq. (24) as the fitness value. Mean represents mean fitness value of k times.

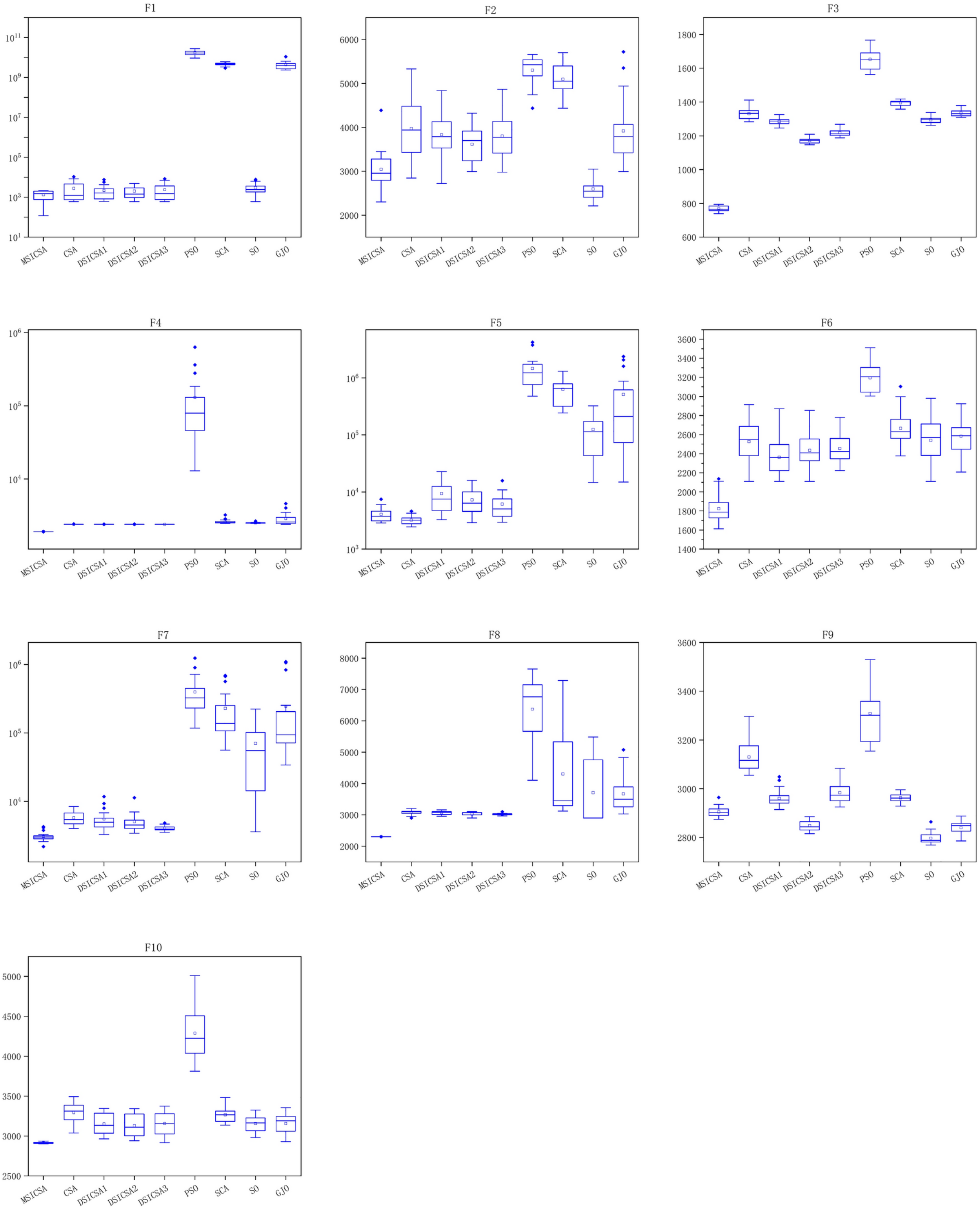


Fig. 4. The results of the boxplots achieved by different algorithms in the CEC 2020 benchmark functions with D = 20.

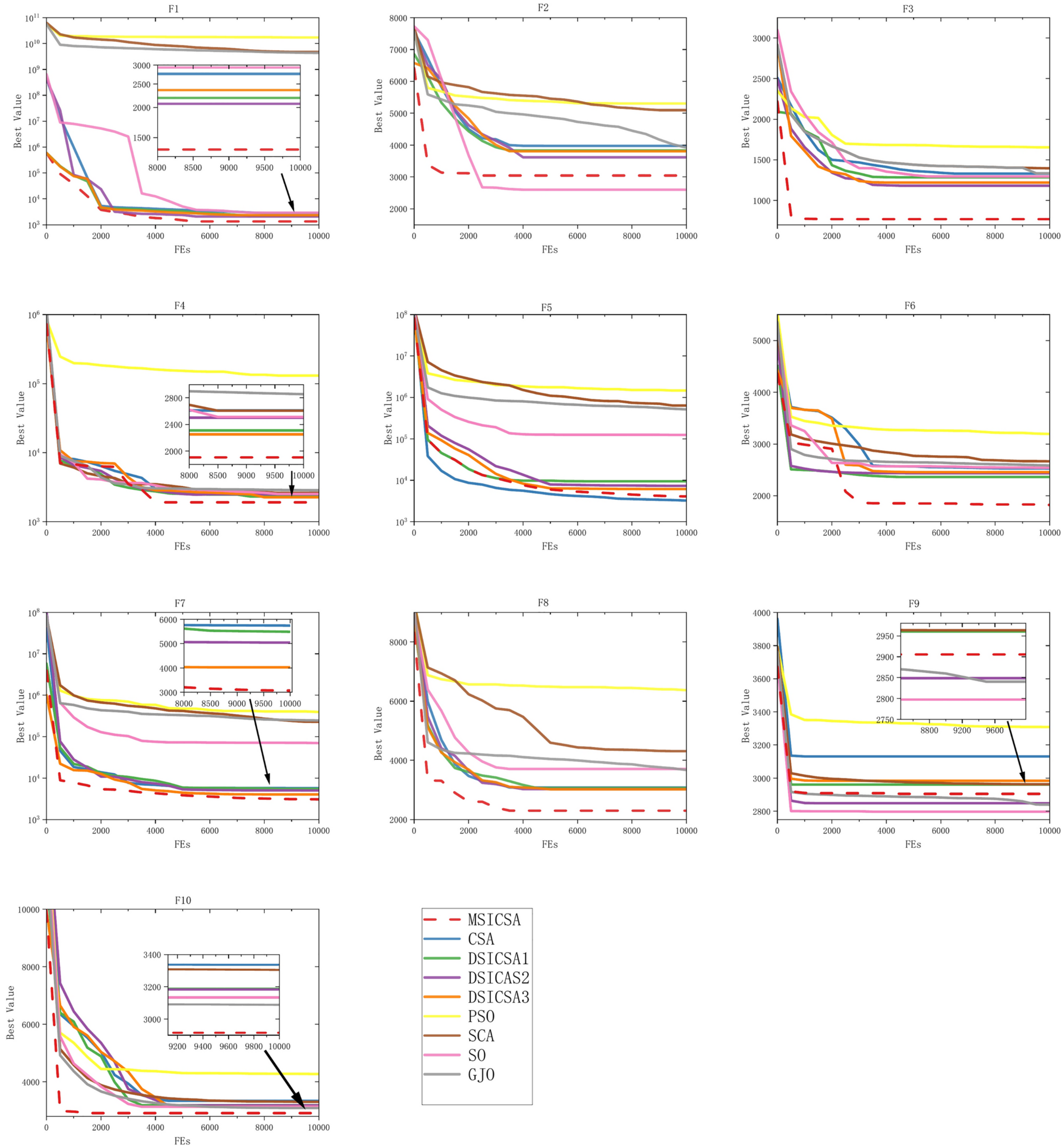


Fig. 5. The results of convergence curves achieved by different algorithms in the CEC 2020 benchmark functions with D = 20.

2. *The standard deviations (STD)* It is an indicator used to evaluate stability and robustness of algorithm. The closer it is to 0, the more stable the algorithm is. It can be defined as Eq. (33).

$$STD = \sqrt{\frac{1}{k}\sum\nolimits_{i=1}^{k}(fit_i - Mean)^2} \tag{33}$$

3. *Peak signal to noise ratio (PSNR)*[48] It is used to measure the error ratio between segmented images and the original image. The closer the PSNR value is to 1, the smaller the distortion of the segmented image is, and the better the image segmentation effect. It can be computed using Eq.(34):

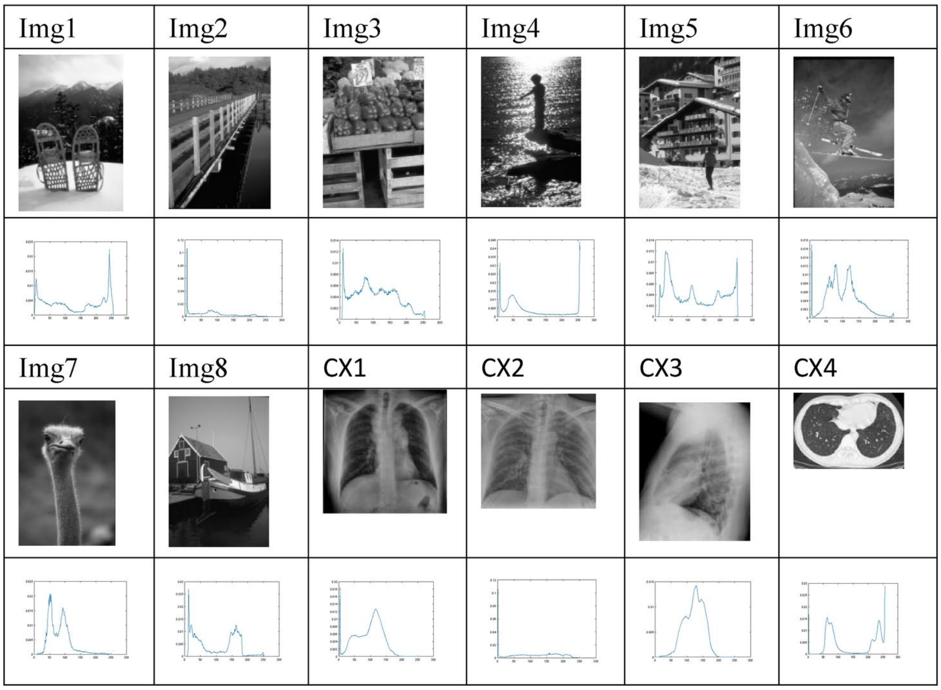


Table 7. Test images and their histogram.

Algorithm	Parameters
Common parameters	Population size: 30, maximum evaluation: 150, number of independent runs : 30
MSICSA	$AP_{min} = 0.01, AP_{max} = 0.2, fl = 2, tp = 0.8$
CSA	$AP = 0.1, fl = 2$
VMCSA	$AP = 0.1, fl = 2$
IHBO	$MutationRatio = 0.5$
RUN_OBL	w = 0.5
SO_OBL	$\theta_1 = 0.25$, $\theta_2 = 0.6$, $c_1 = 0.5$, $c_2 = 0.5$, $c_3 = 2$
SCGJO	$c_1 = 1.5$, $\beta = 1.5$, a = 2
SMA_MLS	$z = 0.03, rr = 0.3, p = 0.25$

Table 8. Parameter settings for experiments in image segmentation.

$$PSNR = 20log_{10}\left(\frac{255}{\sqrt{MSE}}\right) \tag{34}$$

where MSE represents the mean-squared error between two images, defined by Eq. (35):

$$MSE = \frac{1}{mn}\sum\nolimits_{i=1}^{m}\sum\nolimits_{j=1}^{n}\left(I\left(i,j\right) - Seg\left(i,j\right)\right)^2 \tag{35}$$

Here m and n are the height and width of the image, and $I(i, j)$ and $Seg(i, j)$ represent the pixel gray values of the original image and the segmented image.

4. *Structural similarity index (SSIM)*[49] It is mainly used to compare the similarity in brightness and structure between two images. Compared to PSNR, SSIM is more in line with the human visual system in terms of image quality. The closer the value of SSIM is to 1, the more similar the two images are. SSIM can be calculated using Eq. (36):

$$SSIM\left(I,Seg\right)=\frac{\left(2\mu_{I}\mu_{Seg}+c_{1}\right)\left(2\sigma_{I,Seg}+c_{2}\right)}{\left(\mu_{I}^{2}+\mu_{Seg}^{2}+c_{1}\right)\left(\sigma_{I}^{2}+\sigma_{Seg}^{2}+c_{2}\right)} \tag{36}$$

where μ_I and μ_{Seg} represent the average intensity of the original image and the segmented image respectively; σ_I and σ_{Seg} are the standard deviation of the original image and the segmented image; $\sigma_{I,Seg}$ is the covariance between the original image and the segmented image; c_1 and c_2 are two constants equal to 0.01 and 0.03 respectively.

5. *Feature similarity index(FSIM)*[50] It is a variant of SSIM and is an indicator used to compare the similarity of features between original and segmented images. It can be calculated using Eq. (37):

$$FSIM\left(I,Seg\right)=\frac{\sum_{X\in\Omega}S_{T}\left(X\right)PC_{m}\left(X\right)}{\sum_{X\in\Omega}PC_{m}\left(X\right)} \tag{37}$$

where Ω represents the entire image region, $ST(X)$ is a measure of similarity between the original image and the segmented image, phase consistency (PC) can effectively characterize the local structure of the image.

6. Friedman rank test[51]: It is a non-parametric statistical method, whose core idea is to use rank to represent the relative position of algorithm index values in each group, then calculate the average rank of each group, and identify which algorithm is better based on average rank of each group. The optimized performance of algorithm with ranking at the top is the best.

Analysis of Otsu's results

Concerning mean fitness values In Table 9, the mean fitness values of various algorithms at different threshold level are presented. The last row of Table 9 presents the Friedman average ranking.MSICSA outperforms all others, achieving superior results in 49 of 60 cases. Notably, it holds the highest rank among all algorithms with a Friedman average rank of 1.66. RUN_OBL follows closely, excelling in 8 of 60 cases and securing the second position with an average rank of 3.45. SMA_MLS ranks third with an average rank value of 4.02. The remaining ranking is SO_OBL,VMCSA,IHBO,CSA and SCGJO. Figure 6 presents the average fitness values of different algorithms at each threshold. According to Fig. 6, MSICSA exhibits the highest average mean fitness value at each threshold. Thus, it is evident that the performance of the proposed MSICSA method surpasses that of the other methods.

Concerning STD values The STD values of all algorithms are presented In Table 10. The MSICSA algorithm demonstrates superior performance, achieving better results in 45 out of 60 cases. Following closely behind is RUN_OBL, which excels in 8 out of 60 instances. SO_OBL ranks third, showing superiority in 4 cases. The last row of Table 10 showcases the Friedman average ranking. We are pleased to highlight that MSICSA achieves the highest rank among all algorithms in this comparison, with a Friedman average rank value of 1.66. In second place, RUN_OBL secures an average rank value of 3.22, while SO_OBL follows closely in third with an average rank value of 3.59. The algorithms ranked from 4 to 8th are CSA, IHBO, SMA_MLS, VMCSA,and SCGJO respectively. Figure 7 shows the average STD values at each threshold from various algorithms. It can be seen from Fig. 7 that the average STD value of MSICSA is relatively small at each threshold. It is evident that the robustness of the MSICSA algorithm surpasses that of the other comparative algorithms.

Concerning PSNR values Table 11 presents the PSNR values obtained by different algorithms. It is evident from the table that MSICSA consistently achieves higher PSNR values in 37 out of 60 cases compared to other algorithms. Following MSICSA, RUN_OBL obtains higher values in 9 cases. The average PSNR values at each threshold level are depicted in Fig. 8. Furthermore, Fig. 9 illustrates the average PSNR for all threshold levels, clearly indicating that MSICSA consistently attains the highest values. Considering the Friedman average ranks of PSNR under different thresholds, MSICSA still ranks first. Analysis reveals that MSICSA demonstrates the highest average PSNR values at each threshold level.

Concerning SSIM values According to the results presented in Table 12, MSICSA outperforms other methods in 41 cases, accounting for 68.3% of the total cases, in terms of SSIM. Additionally, RUN_OBL and CSA rank second and third, respectively, with 12 and 4 instances of better SSIM values. Considering the Friedman average ranks of SSIM values under different thresholds, MSICSA ranks first. Figure 10 illustrates the average SSIM at each threshold level. It is evident that MSICSA attains the largest value at all threshold levels except for threshold level 20. At threshold level 20, the average SSIM for RUN_OBL and MSICSA are 0.9364 and 0.9352, respectively. This suggests that MSICSA slightly underperforms compared to RUN_OBL at threshold level 20. Furthermore, in Fig. 11, the average SSIM for all threshold levels is presented, indicating that MSICSA achieves the maximum value.

Concerning FSIM values Table 13 presents the results of FSIM values for different algorithms. It is evident that MSICSA outperforms other comparison methods by achieving higher FSIM results in 49 cases. Following MSICSA, RUN_OBL achieves higher FSIM values in 4 cases. It can be seen from the last row of Table 7 that MSICSA method achieves the highest FSIM Friedman average rank, demonstrating excellent image segmentation performance and securing the top position in the rankings. The average FSIM at each threshold level is depicted in Fig. 12. It is observed that MSICSA attains maximum values at every threshold levels . Additionally, in Fig. 13,

Images	D	MSICSA	CSA	VMCSA	IHBO	RUN_OBL	SO_OBL	SCGJO	SMA_MLS
Img1	4	**7617.718**	7613.463	7615.433	7613.763	7616.001	7615.410	7586.153	7616.579
	8	**7738.365**	7733.638	7737.786	7734.764	7735.457	7733.930	7715.785	7734.574
	12	**7766.205**	7763.043	7764.039	7761.939	7764.570	7762.815	7741.491	7760.735
	16	**7776.574**	7773.313	7773.626	7770.991	7773.509	7773.945	7762.843	7767.859
	20	**7782.037**	7779.334	7780.921	7774.795	7774.425	7776.578	7768.361	7774.348
Img2	4	**3670.687**	3668.327	3669.619	3669.919	3669.500	3668.546	3636.188	3668.336
	8	**3776.121**	3771.188	3772.011	3772.099	3774.319	3773.054	3753.354	3772.215
	12	**3801.320**	3798.035	3796.432	3797.619	3799.377	3795.633	3781.070	3798.102
	16	**3810.759**	3807.681	3807.832	3805.918	3809.193	3806.675	3795.415	3809.592
	20	**3815.119**	3810.773	3811.745	3812.188	3813.402	3812.430	3803.794	3813.849
Img3	4	**3490.023**	3486.800	3489.573	3488.362	3488.207	3488.597	3464.208	3489.073
	8	**3615.298**	3611.659	3614.400	3613.255	3614.918	3614.504	3591.092	3614.230
	12	**3645.694**	3642.498	3644.794	3643.422	3645.177	3644.970	3624.787	3644.241
	16	**3656.683**	3654.284	3656.092	3655.098	3655.561	3655.966	3640.115	3654.419
	20	**3662.342**	3661.188	3660.738	3660.506	3660.676	3661.386	3647.261	3660.289
Img4	4	**6510.545**	6507.553	6510.275	6509.385	6508.629	6509.706	6481.636	6509.046
	8	**6643.592**	6640.885	6640.878	6641.106	6642.999	6640.544	6622.257	6643.295
	12	**6669.536**	6661.939	6663.575	6668.116	6662.873	6667.537	6652.446	6662.453
	16	**6679.150**	6677.309	6677.619	6676.965	6677.744	6678.525	6665.595	6678.032
	20	**6683.586**	6682.085	6683.157	6681.461	6682.033	6682.892	6671.808	6681.735
Img5	4	**5964.161**	5960.683	5963.411	5962.483	5963.972	5962.169	5946.549	5962.771
	8	**6083.085**	6079.617	6082.057	6081.768	6081.297	6081.188	6059.333	6080.655
	12	**6113.100**	6110.731	6112.881	6110.813	6110.116	6110.569	6091.807	6110.887
	16	**6124.227**	6120.379	6123.316	6121.757	6122.576	6122.497	6106.622	6121.164
	20	**6129.448**	6125.178	6126.592	6126.710	6127.762	6125.901	6111.951	6125.390
Img6	4	**1953.386**	1951.954	1952.165	1953.385	1953.017	1952.186	1929.930	1952.186
	8	2060.160	2059.280	2061.009	2060.566	**2061.512**	2061.418	2042.000	2061.497
	12	**2087.865**	2082.771	2084.611	2081.294	2082.130	2083.221	2069.373	2082.555
	16	**2098.067**	2091.214	2096.320	2094.881	2095.719	2093.228	2081.955	2095.776
	20	**2102.656**	2097.180	2097.004	2098.867	2098.108	2097.157	2090.931	2101.897
Img7	4	**1178.551**	1173.805	1174.998	1175.427	1177.274	1177.533	1157.028	1175.238
	8	**1235.667**	1231.045	1234.735	1234.172	1234.479	1232.554	1215.042	1232.716
	12	**1249.589**	1247.227	1246.694	1245.770	1248.843	1248.300	1232.499	1246.625
	16	**1255.524**	1253.862	1253.945	1251.291	1255.082	1251.689	1239.025	1253.661
	20	**1258.182**	1253.663	1255.127	1253.978	1254.349	1255.772	1241.857	1254.767
Img8	4	4138.207	4136.625	4140.301	**4140.356**	4140.356	4140.356	4118.751	4140.355
	8	4236.269	4232.573	4235.985	4236.651	**4236.724**	4235.958	4212.581	4235.777
	12	4252.952	4252.821	4254.299	4254.398	**4255.109**	4254.795	4238.127	4254.946
	16	4260.782	4261.408	4261.739	4262.318	**4262.968**	4262.636	4250.157	4262.528
	20	4266.648	4265.144	4265.810	4264.792	**4266.798**	4266.435	4254.967	4266.055
CX1	4	**1521.576**	1520.731	1519.893	1521.010	1518.587	1521.047	1488.435	1519.075
	8	**1591.005**	1590.337	1587.541	1545.329	1590.043	1588.103	1525.745	1590.662
	12	**1607.040**	1550.298	1594.376	1573.225	1604.919	1601.428	1563.161	1605.509
	16	**1612.632**	1556.476	1601.933	1584.547	1609.418	1602.480	1571.731	1611.226
	20	1612.404	1584.136	1603.595	1596.439	**1615.398**	1604.018	1578.956	1614.767
CX2	4	**860.162**	860.162	859.446	860.155	860.162	860.161	835.335	860.161
	8	903.332	903.179	900.970	**903.730**	903.358	903.649	882.161	903.287
	12	913.675	913.516	911.844	913.804	**913.943**	912.930	894.673	913.688
	16	917.584	917.828	916.553	916.998	**918.198**	916.303	902.390	917.947
	20	**920.111**	917.938	916.256	918.209	919.569	916.995	904.714	919.046
CX3	4	**3282.171**	3280.261	3280.652	3281.465	3281.922	3280.482	3249.146	3279.825
	8	**3369.271**	3367.707	3366.763	3366.805	3366.648	3368.094	3343.677	3368.225
	12	**3389.048**	3386.848	3386.029	3387.486	3385.907	3387.598	3369.221	3386.858
	16	**3396.273**	3394.484	3392.577	3393.432	3393.041	3393.130	3267.813	3394.984
	20	**3399.719**	3398.327	3396.748	3391.165	3397.041	3395.433	3273.715	3397.128

Continued

Images	D	MSICSA	CSA	VMCSA	IHBO	RUN_OBL	SO_OBL	SCGJO	SMA_MLS
CX4	4	**6401.453**	6399.155	6397.571	6400.402	6400.999	6400.280	6386.624	6400.701
	8	6493.795	6491.992	6490.787	6491.183	6492.366	**6493.830**	6474.905	6493.332
	12	**6511.078**	6510.424	6509.115	6509.715	6508.085	6508.192	6498.816	6510.795
	16	**6517.603**	6515.949	6514.451	6516.648	6513.833	6515.170	6507.362	6516.546
	20	**6520.557**	6518.422	6517.607	6519.220	6519.368	6518.935	6510.151	6519.430
Friedman average rank		1.62	5.60	4.52	4.67	3.45	4.17	7.97	4.02
Rank		1	7	5	6	2	4	8	3

Table 9. The mean fitness values of different algorithms using Otsu's method. The optimal values are highlighted in bold.

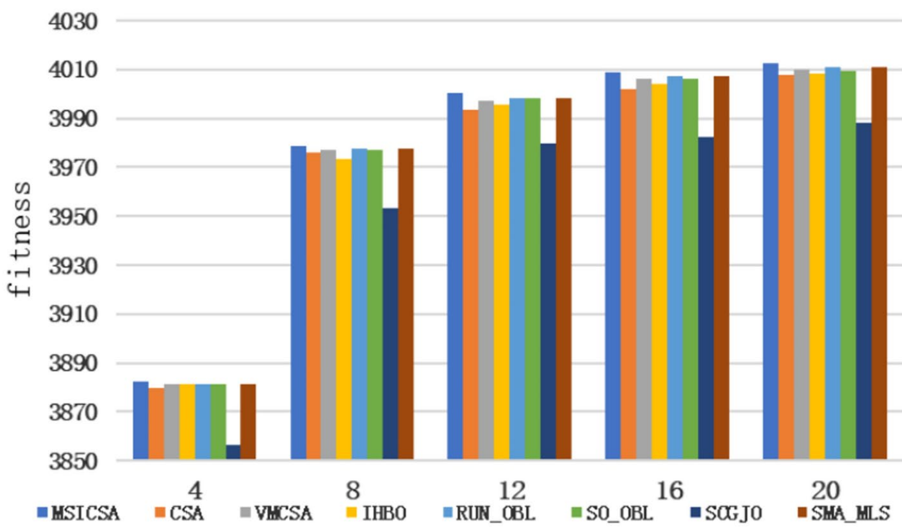


Fig. 6. Average fitness values at each threshold using Otsu's method.

the average FSIM for all threshold levels is presented, highlighting that MSICSA achieves the largest value. This demonstrates the superior performance of MSICSA in terms of FSIM.

Table 14 shows the optimal threshold values for test images from various algorithms. The convergence curves of various algorithms based on the Otsu's method for Img1, Img7, and CX4 are illustrated in Figs. 14, 15, and 16. Among these images, MSICSA demonstrates the highest fitness value along with a more rapid convergence rate in most instances. This indicates that MSICSA is capable of identifying the optimal threshold value with fewer evaluations compared to other algorithms.

The segmented images obtained through MSICSA with varying thresholds (D = 4, 8, 12, 16,20) for Img1, Img7, CX4 are presented in Tables 15. Additionally, the results illustrate the locations of the optimally determined thresholds within the histogram for Img1, Img7, CX4. In Table 15, the outcomes of segmentation are visualized using distinct colors, with each color representing a different segment.

The average CPU times spent by each algorithm at each threshold of Otsu are given in Fig. 17. It shows that the CPU time of MSICSA algorithm is satisfactory. At level 4 and 8, the CPU time of MSICSA is longer than CSA but shorter than other compared algorithms; at level 12, the CPU time of MSICSA is longer than CSA and SCGJO but shorter than other compared algorithms; at level 16 and 20, the CPU time of MSICSA is longer than CSA, SCGJO and SMA_MLS but shorter than other compared algorithms. It is worth noting that MSICSA runs only about 1 s longer than CSA at level 20. Considering its effectiveness, this running time is acceptable (Table 15).

Overall, MSICSA outperforms other comparative algorithms in terms of mean fitness values, STD, PSNR, SSIM and FSIM when employing Otsu's method. However, due to the incorporation of three improved strategies, its runtime is slightly longer than that of the CSA algorithm.

Analysis of results using fuzzy entropy

Concerning mean fitness values In Table 16, the mean fitness values of various algorithms at each threshold level are presented. MSICSA demonstrates superior performance, achieving the best results in 42 out of 60 cases. Closely following is SMA_MLS, which excels in 6 out of 60 instances. SO_OBL ranks third, showing superiority in 4 cases. The last row of Table 19 displays the Friedman average ranking. Notably, MSICSA attains

Images	D	MSICSA	CSA	VMCSA	IHBO	RUN_OBL	SO_OBL	SCGJO	SMA_MLS
Img1	4	**2.78 E−12**	1.34E−02	1.18E+00	2.61 E−02	1.32 E−02	3.86 E−12	8.59 E+00	4.13 E−12
	8	**1.93E−02**	7.74 E−02	1.16 E+00	4.61 E−01	4.38 E−01	7.35 E−02	9.48 E+00	8.73 E−01
	12	**9.38 E−02**	2.38 E−01	5.44 E−01	5.07 E−01	5.12 E−01	1.44 E−01	5.58 E+00	8.26 E−01
	16	**1.76 E−01**	2.45 E−01	3.37 E−01	6.47 E−01	2.93 E−01	2.05 E−01	3.62 E+00	6.32 E−01
	20	**7.63 E−02**	1.31 E−01	3.75 E−01	6.45 E−01	1.79 E−01	1.56 E−01	2.78 E+00	7.77 E−01
Img2	4	**2.31 E−12**	5.88 E−03	9.82 E−01	8.83 E−03	1.93 E−02	5.78 E−03	1.58 E+01	1.88 E−02
	8	**1.45 E−02**	5.88 E−01	1.03 E+00	1.81 E+00	3.33 E−02	6.52 E−01	1.22 E+01	1.22 E+00
	12	**1.03 E−01**	1.46 E−01	7.41 E−01	6.88 E−01	9.24 E−01	6.88 E−01	5.36 E+00	1.22 E+00
	16	**1.33 E−01**	3.22 E−01	5.00 E−01	7.13 E−01	2.97 E−01	5.41 E−01	4.61 E+00	9.14 E−01
	20	**1.76 E−01**	2.85 E−01	3.31 E−01	7.02 E−01	3.64 E−01	4.14 E−01	2.39 E+00	5.62 E−01
Img3	4	**4.63 E−13**	7.66 E−03	1.27 E+00	8.22 E−03	5.91 E−03	5.93 E−13	6.11 E+00	4.55 E−03
	8	**1.68 E−02**	4.84 E−02	1.10 E+00	1.76 E−01	4.23 E−02	4.77 E−02	1.14 E+01	1.59 E−01
	12	**9.08 E−02**	1.48 E−01	5.48 E−01	7.47 E−01	1.42 E−01	4.46 E−01	7.93 E+00	6.89 E−01
	16	**8.62 E−02**	1.50 E−01	2.79 E−01	7.54 E−01	2.05 E−01	3.38 E−01	5.36 E+00	5.37 E−01
	20	**7.10 E−02**	1.66 E−01	2.64 E−01	8.45 E−01	3.29 E−01	3.79 E−01	4.04 E+00	7.47 E−01
Img4	4	**1.85 E−12**	1.85 E−12	1.31 E+00	1.59 E−02	1.94 E−12	3.19 E−03	9.74 E+00	2.47 E−02
	8	2.86 E−02	5.19 E−02	9.40 E−01	1.20 E−01	**2.62 E−02**	6.69 E−02	8.40 E+00	1.21 E−01
	12	**7.37 E−02**	1.75 E−01	5.91 E−01	4.29 E−01	1.61 E−01	2.03 E−01	4.86 E+00	6.98 E−01
	16	**1.31 E−01**	2.06 E−01	4.52 E−01	4.51 E−01	2.40 E−01	2.79 E−01	4.39 E+00	5.56 E−01
	20	**1.04 E−01**	1.58 E−01	3.12 E−01	4.19 E−01	2.57 E−01	1.73 E−01	4.12 E+00	4.13 E−01
Img5	4	**2.78 E−12**	7.52 E−03	1.93 E+00	9.34 E−03	2.78 E−12	7.52 E−03	1.22 E+01	1.06 E−02
	8	**2.71 E−02**	4.15 E−02	9.92 E−01	9.91 E−02	5.53 E−02	3.17 E−02	1.03 E+01	5.96 E−02
	12	**1.20 E−01**	1.36 E−01	5.35 E−01	2.85 E−01	2.34 E−01	1.28 E−01	6.57 E+00	7.60 E−01
	16	**8.46 E−02**	1.37 E−01	4.10 E−01	6.19 E−01	5.82 E−01	1.67 E−01	5.39 E+00	8.42 E−01
	20	**1.24 E−01**	1.67 E−01	3.57 E−01	5.73 E−01	2.48 E−01	3.13 E−01	3.80 E+00	7.67 E−01
Img6	4	**0.00E+00**	8.18 E−04	7.52 E−01	1.37 E−03	3.04 E−03	1.16 E−03	3.67 E+00	2.45 E−03
	8	1.54 E−01	1.16 E−01	9.95 E−01	1.12 E−01	1.19 E−01	**1.08 E−01**	6.95 E+00	1.33 E−01
	12	**1.52 E−01**	3.11 E−01	4.66 E−01	5.43 E−01	2.14 E−01	2.05 E−01	6.70 E+00	4.52 E−01
	16	**1.77 E−01**	2.51 E−01	3.87 E−01	7.45 E−01	2.79 E−01	5.60 E−01	3.61 E+00	7.97 E−01
	20	**2.13 E−01**	4.34 E−01	3.18 E−01	7.76 E−01	3.42 E−01	4.55 E−01	3.32 E+00	8.45 E−01
Img7	4	**9.25 E−13**	5.79 E−04	4.94 E−01	1.02 E−02	4.25 E−12	1.29 E−03	1.68 E+01	1.55 E−03
	8	**1.38 E−01**	7.91 E−01	6.63 E−01	5.91 E−01	2.81 E−01	4.88 E−01	9.73 E+00	3.03 E−01
	12	4.54 E−01	5.25 E−01	6.48 E−01	6.57 E−01	**1.67 E−01**	7.05 E−01	4.73 E+00	6.79 E−01
	16	**3.85 E−01**	5.69 E−01	5.53 E−01	6.81 E−01	4.44 E−01	5.15 E−01	4.07 E+00	6.77 E−01
	20	**2.96 E−01**	5.91 E−01	6.66 E−01	7.16 E−01	4.36 E−01	7.51 E−01	4.92 E+00	5.88 E−01
Img8	4	**1.85 E−12**	3.50 E−04	9.92 E−01	6.62 E−04	3.50 E−04	7.26 E−04	1.74 E+01	8.61 E−04
	8	9.14 E−01	1.77 E+00	1.16 E+00	**3.90 E−01**	8.43 E−01	2.03 E+00	9.49 E+00	2.42 E+00
	12	**6.72 E−01**	1.13 E+00	9.70 E−01	1.01 E+00	8.91 E−01	1.06 E+00	5.39 E+00	9.85 E−01
	16	**4.36 E−01**	6.43 E−01	6.34 E−01	7.78 E−01	5.10 E−01	5.08 E−01	4.53 E+00	6.48 E−01
	20	3.57 E−01	2.88 E−01	4.12 E−01	6.01 E−01	**2.80 E−01**	4.19 E−01	3.59 E+00	5.63 E−01
CX1	4	**2.31 E−13**	2.31 E−13	5.77 E−01	4.89 E−02	2.40 E−02	1.27 E−02	4.40 E+00	2.31 E−02
	8	5.27 E−02	**4.07 E−02**	1.41 E+00	1.04 E+00	2.07 E−01	6.53 E−01	1.95 E+01	2.07 E−01
	12	**1.12 E−01**	1.33 E+01	1.74 E+00	1.23 E+01	9.47 E−01	6.13 E+00	5.66 E+00	6.77 E−01
	16	**3.94 E−01**	1.44 E+01	1.62 E+01	7.14 E+00	6.04 E−01	9.04 E+00	7.58 E+00	7.26 E−01
	20	8.58 E+00	1.63 E+01	1.82 E+01	1.93 E+01	**3.33 E−01**	6.22 E+00	1.51 E+01	7.26 E−01
CX2	4	**4.63 E−13**	1.27 E−05	6.25 E−01	2.06 E−02	1.27 E−05	3.14 E−05	8.97 E+00	3.36 E−04
	8	1.69 E+00	1.69 E+00	1.73 E+00	**1.90 E−01**	7.98 E−01	3.43 E−01	7.24 E+00	1.47 E+00
	12	1.33 E+00	1.34 E+00	1.28 E+00	8.11 E−01	**7.02 E−01**	8.05 E−01	7.54 E+00	1.27 E+00
	16	1.12 E+00	9.74 E−01	1.11 E+00	6.36 E−01	**5.32 E−01**	1.27 E+00	6.31 E+00	9.63 E−01
	20	6.53 E−01	6.16 E−01	6.09E−01	6.99 E−01	**3.22 E−01**	7.62 E−01	4.48 E+00	9.09 E−01
CX3	4	2.29 E−03	6.10 E−03	9.50 E−01	6.52 E−03	**0.00 E+00**	3.18 E−03	3.68 E+00	5.39 E−03
	8	**5.47 E−02**	7.50 E−02	7.03 E−01	1.65 E−01	2.74 E−01	1.12 E−01	1.27 E+01	4.29 E−01
	12	**1.11 E−01**	8.36 E−01	8.01 E−01	7.39 E−01	1.39 E+00	3.37 E−01	7.93 E+00	7.59 E−01
	16	**2.37 E−01**	4.06 E−01	1.51 E+00	1.29 E+00	1.27 E+00	5.82 E−01	1.76 E+01	1.01 E+00
	20	**2.38 E−01**	1.06 E+00	2.66 E+00	1.65 E+01	6.06 E−01	8.74 E+00	1.70 E+01	6.82 E−01

Continued

Images	D	MSICSA	CSA	VMCSA	IHBO	RUN_OBL	SO_OBL	SCGJO	SMA_MLS
CX4	4	**1.85 E−12**	5.69 E−04	8.95 E−01	2.98 E−12	6.83 E−12	3.25 E−12	1.02 E+01	7.15 E−12
	8	6.16 E−02	3.25 E−01	1.34 E+00	3.60 E−02	1.11 E−01	**1.86 E−02**	4.85 E+00	3.70 E−01
	12	**1.40 E−01**	5.46 E−01	6.91 E−01	3.60 E−01	8.30 E−01	2.19 E−01	3.84 E+00	4.80 E−01
	16	3.73 E−01	5.50 E−01	7.41 E−01	4.21 E−01	6.94 E−01	**3.09 E−01**	2.69 E+00	6.89 E−01
	20	3.65 E−01	6.18 E−01	5.11 E−01	4.27 E−01	4.54 E−01	**3.64 E−01**	2.99 E+00	5.73 E−01
Friedman average rank	1.66	3.67	5.60	5.16	3.22	3.59	7.85	5.26	
Rank	1	4	7	5	2	3	8	6	

Table 10. Result of STD for different algorithms using Otsu's method. The optimal values are highlighted in bold.

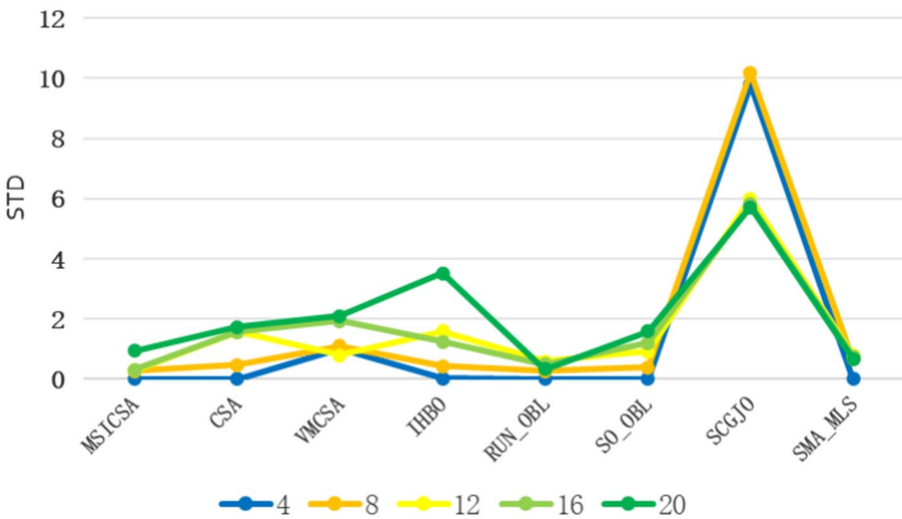


Fig. 7. Average STD at each threshold using Otsu's method.

the highest rank among all algorithms included in this comparison, with a Friedman average rank value of 2.08. Following closely behind is SMA_MLS, securing second place with an average rank value of 3.79. SO_OBL occupies third position with an average rank value of 3.87. The remaining rankings include RUN_OBL,IHBO, VMCAS, CSA and SCGJO. Figure 18 presents the mean fitness value at each threshold using fuzzy entropy as objective function. According to Fig. 18, MSICSA exhibits the highest average mean fitness value across all threshold levels. Therefore, it is evident that the performance of the proposed MSICSA method surpasses that of other methods evaluated.

Concerning STD values The STD values of all algorithms are presented In Table 17 . The last row of Table 17 showcases the Friedman average ranking. SO_OBL achieves the highest rank among all algorithms included in this comparison, with a Friedman average rank value of 2.51. Notably, SO_OBL consistently demonstrates lower STD values in 20 instances compared to other algorithms. In second place, RUN_OBL secures an average rank value of 2.84, while MSICSA closely follows in third position with an average rank value of 3.03; MSICSA consistently exhibits lower STD values in 13 cases relative to other algorithms. Figure 19 illustrates the average STD values of various algorithms at each threshold level. At thresholds of 8, 12, 16, and 20, the average STD value for MSICSA is only surpassed by those of SO_OBL and RUN_OBL; it outperforms CSA, VMCSA, IHBO, SCGJO, and SMA_MLS . At threshold level 2, the average STD value for MSICSA is comparable to that of CSA, VMCSA, IHBO, RUN_OBL, SO_OBL and SMA_MLS; however, it surpasses SCGJO in performance.

Concerning PSNR values Table 18 presents the PSNR values obtained by various algorithms. It is evident from the table that MSICSA consistently achieves higher PSNR values in 41 out of 60 cases when compared to other algorithms. Following MSICSA, SMA_MLS attains superior values in 5 instances. The average PSNR values at each threshold level are illustrated in Fig. 20. Furthermore, Fig. 21 depicts the average PSNR at all threshold levels, clearly indicating that MSICSA consistently reaches the highest values. Considering the Friedman average ranks of PSNR under different thresholds, MSICSA maintains its position at the top. Analysis reveals that MSICSA demonstrates the highest average PSNR values across all threshold levels.

Concerning SSIM values Table 19 displays the SSIM values derived from various algorithms. According to the results presented in Table 19, MSICSA outperforms other methods in 44 instances, representing 73.3% of the total cases, with respect to SSIM. Additionally, SMA_MLS ranks second , achieving better SSIM values in 7 instances. When considering the Friedman average ranks of SSIM values across different thresholds, MSICSA secures the top position. Figure 22 illustrates the average SSIM at each threshold level, clearly demonstrating that

Images	D	MSICSA	CSA	VMCSA	IHBO	RUN_OBL	SO_OBL	SCGJO	SMA_MLS
Img1	4	**18.909**	17.695	18.348	18.682	17.638	18.614	17.304	18.392
	8	**24.530**	22.199	22.577	22.766	24.412	23.463	21.625	24.028
	12	**27.862**	26.255	26.708	26.804	27.538	27.835	25.466	27.663
	16	**30.224**	28.746	30.175	28.972	30.013	28.553	27.459	29.298
	20	**32.062**	30.504	31.764	31.254	31.343	30.469	27.465	31.212
Img2	4	20.672	20.614	20.674	20.672	**20.678**	20.671	19.974	20.675
	8	**25.773**	25.154	24.031	23.901	23.969	24.922	22.276	25.543
	12	**28.628**	27.473	27.027	27.338	27.977	28.000	24.574	26.675
	16	**30.920**	28.664	29.581	29.681	30.209	28.872	26.448	28.738
	20	**32.533**	30.507	31.512	30.445	31.788	32.039	28.352	30.799
Img3	4	**19.419**	19.395	19.419	19.419	19.419	19.419	18.949	19.419
	8	24.383	24.243	**24.385**	24.383	24.385	24.380	23.030	24.380
	12	27.728	27.442	27.721	27.589	**27.730**	27.683	25.245	27.585
	16	**29.931**	28.808	29.054	28.469	28.979	28.204	25.710	28.630
	20	**31.743**	31.292	31.696	29.355	29.955	30.094	27.070	29.430
Img4	4	**19.192**	19.134	19.192	19.189	19.192	19.189	18.938	19.185
	8	24.609	24.315	24.605	**24.625**	24.615	24.616	23.929	24.611
	12	27.986	27.565	27.974	28.000	**28.031**	28.013	26.478	27.937
	16	30.376	29.805	30.268	30.365	30.409	**30.461**	28.097	30.235
	20	**32.219**	30.871	31.464	31.630	30.470	30.736	27.786	31.397
Img5	4	18.189	18.140	18.190	18.190	18.189	18.190	17.636	**18.191**
	8	23.209	23.032	23.312	**23.323**	23.214	23.210	21.364	23.273
	12	**27.292**	25.917	25.766	25.919	25.505	25.459	22.321	26.950
	16	**29.588**	25.379	27.791	28.229	27.686	26.472	24.587	27.166
	20	31.386	31.134	31.387	31.226	**31.556**	31.428	27.485	30.980
Img6	4	19.248	**19.286**	19.274	19.251	19.248	19.248	18.473	19.250
	8	25.134	24.964	25.103	24.897	25.170	24.963	23.626	**25.188**
	12	28.598	28.368	28.481	28.305	**28.642**	28.417	25.857	28.396
	16	31.053	30.751	**31.070**	30.714	31.007	30.897	27.605	30.803
	20	**32.926**	32.611	32.838	32.292	32.764	32.497	29.070	32.582
Img7	4	17.331	**17.339**	17.330	17.329	17.331	17.327	16.661	17.324
	8	**24.186**	22.196	23.083	21.629	22.672	22.459	19.827	23.151
	12	28.979	29.146	28.870	28.070	**30.159**	28.020	24.207	29.166
	16	32.491	31.926	32.375	31.749	**32.647**	31.814	26.874	32.066
	20	33.999	33.733	34.090	33.323	**34.351**	33.381	28.675	33.883
Img8	4	19.835	19.840	19.838	19.849	19.838	19.853	19.386	**19.863**
	8	25.171	24.930	25.129	25.194	**25.197**	25.099	23.217	25.075
	12	**28.263**	27.728	28.017	27.904	28.242	27.984	25.370	28.125
	16	**30.459**	28.469	29.602	29.734	28.281	30.205	25.752	27.536
	20	**31.814**	30.241	30.577	30.884	31.372	28.795	25.910	30.715
CX1	4	**22.445**	22.159	21.919	21.682	22.314	21.532	19.919	22.426
	8	**27.583**	27.547	27.219	27.146	27.397	26.908	24.399	27.334
	12	**30.961**	30.159	29.909	28.070	30.442	29.592	28.256	29.870
	16	**33.262**	32.446	28.833	26.905	32.578	29.904	19.434	32.005
	20	33.898	31.618	28.111	25.035	34.424	30.055	9.086	**34.433**
CX2	4	**20.453**	19.867	19.688	20.227	19.723	19.713	16.591	19.905
	8	**27.543**	24.968	25.186	24.863	26.292	25.688	23.107	24.968
	12	**31.011**	28.647	30.201	29.010	29.469	27.819	25.189	28.715
	16	**32.943**	30.348	31.775	31.216	31.303	31.179	26.052	32.198
	20	**34.818**	31.729	32.919	31.130	33.064	32.262	28.635	33.378
CX3	4	20.800	19.847	20.109	20.053	20.218	20.311	20.521	**20.812**
	8	**26.485**	24.649	24.435	25.625	25.386	24.710	23.525	25.492
	12	**29.957**	29.562	29.230	28.091	28.479	29.273	25.418	28.195
	16	**32.387**	31.526	31.354	31.189	30.403	31.366	27.124	30.875
	20	**34.236**	32.701	32.320	31.647	33.697	32.866	16.081	32.660

Continued

Images	D	MSICSA	CSA	VMCSA	IHBO	RUN_OBL	SO_OBL	SCGJO	SMA_MLS
CX4	4	**17.816**	17.572	16.695	16.942	16.118	16.125	15.839	16.317
	8	23.197	22.718	**23.216**	23.215	22.990	23.177	21.577	22.689
	12	**26.961**	25.402	26.188	26.788	26.800	26.934	23.139	25.919
	16	29.990	30.152	30.87	**31.049**	30.221	30.782	26.515	29.772
	20	**33.117**	30.137	31.233	30.993	30.823	30.347	27.527	29.891
Friedman average rank		2.11	5.24	3.84	4.68	3.33	4.61	7.90	4.29
Rank		1	7	3	6	2	5	8	4

Table 11. Result of PSNR for different algorithms using Otsu's method. The optimal values are highlighted in bold.

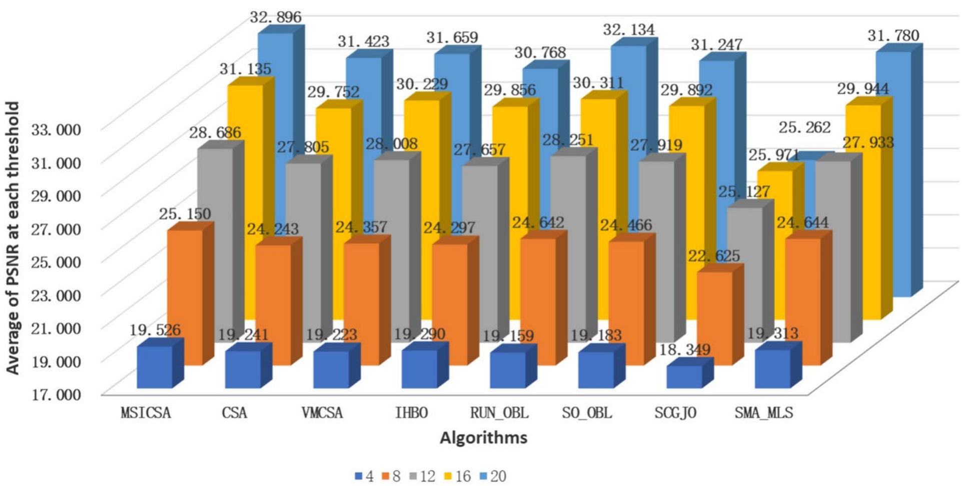


Fig. 8. The average PSNR at each threshold level using Otsu's method.

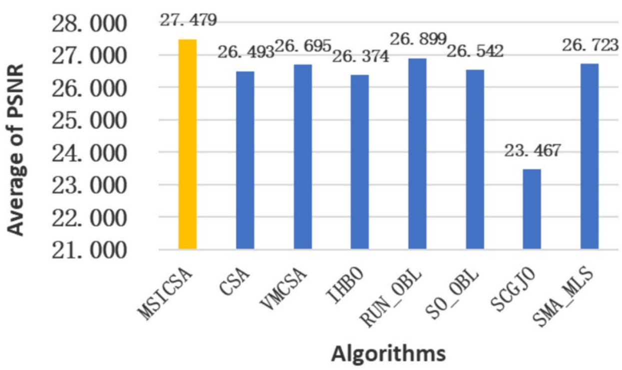


Fig. 9. The average PSNR at all threshold levels using Otsu's method.

MSICSA attains the highest value across all thresholds. Furthermore, Fig. 23 presents the average SSIM for all threshold levels, indicating that MSICSA consistently achieves maximum performance in terms of SSIM.

Concerning FSIM values Table 20 presents the results of FSIM values for various algorithms. It is clear that MSICSA outperforms other comparative methods, achieving higher FSIM results in 39 instances. Following MSICSA, SMA_MLS attains superior FSIM values in 7 cases. The last row of Table 20 indicates that the MSICSA method achieves the highest Friedman average rank for FSIM, demonstrating exceptional image segmentation performance and securing its position at the top of the rankings. The average FSIM at each threshold level is illustrated in Fig. 24, where it can be observed that MSICSA reaches maximum values across all threshold levels. Furthermore, Fig. 25 presents the average FSIM for all threshold levels, emphasizing that MSICSA consistently

Images	D	MSICSA	CSA	VMCSA	IHBO	RUN_OBL	SO_OBL	SCGJO	SMA_MLS
Img1	4	**0.7228**	0.7210	0.7165	0.7191	0.7137	0.7133	0.6543	0.7179
	8	0.8190	0.8143	0.8186	0.8190	0.8199	0.8195	0.7658	**0.8208**
	12	**0.8782**	0.8572	0.8728	0.8686	0.8600	0.8619	0.8101	0.8700
	16	**0.9031**	0.8658	0.8933	0.8880	0.8975	0.9012	0.8655	0.8890
	20	0.9258	0.9242	0.9264	0.9241	**0.9280**	0.9268	0.8894	0.9260
Img2	4	0.6341	0.6343	0.6354	0.6343	**0.6360**	0.6343	0.5759	0.6352
	8	**0.7602**	0.7410	0.7454	0.7515	0.7470	0.7561	0.6964	0.7447
	12	**0.8086**	0.7815	0.7987	0.7977	0.8059	0.7996	0.7509	0.7963
	16	0.8317	0.8140	0.8192	0.8242	**0.8334**	0.8131	0.7801	0.8203
	20	0.8481	0.8320	0.8462	0.8348	**0.8641**	0.8289	0.7998	0.8297
Img3	4	0.6222	0.6222	**0.6223**	0.6223	0.6222	0.6222	0.5954	0.6221
	8	**0.7791**	0.7656	0.7729	0.7599	0.7755	0.7618	0.7294	0.7628
	12	**0.8531**	0.8370	0.8480	0.8333	0.8387	0.8457	0.7729	0.8440
	16	**0.8876**	0.8717	0.8811	0.8769	0.8803	0.8759	0.8230	0.8859
	20	**0.9110**	0.8882	0.8989	0.9065	0.9106	0.8962	0.8615	0.9099
Img4	4	**0.7552**	0.7390	0.7471	0.7381	0.7354	0.7472	0.6719	0.7462
	8	**0.8411**	0.8248	0.8391	0.8323	0.8335	0.8298	0.7927	0.8385
	12	**0.8783**	0.8699	0.8717	0.8706	0.8598	0.8745	0.8455	0.8713
	16	**0.8974**	0.8825	0.8853	0.8927	0.8808	0.8915	0.8587	0.8829
	20	**0.9116**	0.8977	0.8894	0.8890	0.8997	0.8916	0.8732	0.8989
Img5	4	0.6509	**0.6511**	0.6479	0.6510	0.6509	0.6510	0.6177	0.6510
	8	0.8139	**0.8191**	0.8072	0.8124	0.8048	0.8010	0.7390	0.8161
	12	**0.9071**	0.8864	0.88678	0.8886	0.8925	0.9053	0.8001	0.8882
	16	**0.9319**	0.9203	0.9232	0.9144	0.9299	0.9283	0.8535	0.9206
	20	0.9462	0.9468	0.9462	0.9474	**0.9509**	0.9485	0.8977	0.9439
Img6	4	0.6992	0.7006	**0.7019**	0.6995	0.6992	0.6992	0.6541	0.6995
	8	**0.8592**	0.8484	0.8567	0.8412	0.8451	0.8520	0.8168	0.8438
	12	**0.9016**	0.8819	0.8926	0.8927	0.8845	0.8804	0.8561	0.8890
	16	**0.9299**	0.9120	0.9164	0.9239	0.9287	0.9171	0.8854	0.9234
	20	**0.9453**	0.9313	0.9390	0.9286	0.9400	0.9399	0.8958	0.9251
Img7	4	**0.5226**	0.5083	0.5102	0.5030	0.5184	0.5169	0.4809	0.5071
	8	**0.8083**	0.7937	0.7973	0.7793	0.7973	0.7852	0.6383	0.8083
	12	0.9052	0.9031	0.9093	0.8892	**0.9228**	0.8882	0.8058	0.9077
	16	0.9448	0.9435	0.9392	0.9358	**0.9454**	0.9364	0.8714	0.9400
	20	0.9549	0.9307	0.9530	0.9492	**0.9563**	0.9493	0.9067	0.9549
Img8	4	**0.6409**	0.6240	0.6254	0.6281	0.6378	0.6221	0.5471	0.6394
	8	**0.7387**	0.7230	0.7295	0.7206	0.7242	0.7290	0.6483	0.7184
	12	**0.8078**	0.7823	0.7886	0.7790	0.8010	0.7836	0.7196	0.7974
	16	**0.8527**	0.8322	0.8436	0.8341	0.8468	0.8309	0.7618	0.8333
	20	0.8713	0.8723	0.8756	0.8695	**0.8814**	0.8673	0.8340	0.8807
CX1	4	**0.8119**	0.7945	0.79459	0.8005	0.8104	0.8008	0.7128	0.7951
	8	**0.9495**	0.9336	0.9361	0.9269	0.9494	0.9309	0.8764	0.9485
	12	**0.9779**	0.9687	0.9720	0.9332	0.9652	0.9533	0.9437	0.9705
	16	**0.9863**	0.9500	0.9723	0.9072	0.9812	0.9533	0.9541	0.9799
	20	0.9822	0.9562	0.9686	0.8701	**0.9872**	0.9588	0.9753	0.9708
CX2	4	**0.8146**	0.87973	0.8043	0.8007	0.8024	0.7982	0.6952	0.7992
	8	**0.9658**	0.9474	0.9598	0.9602	0.9565	0.9601	0.8888	0.9653
	12	**0.9871**	0.9719	0.9783	0.9852	0.9866	0.9827	0.9276	0.9677
	16	**0.9922**	0.9732	0.9833	0.9729	0.9891	0.9715	0.9425	0.9792
	20	0.9954	0.9856	0.9935	0.9933	**0.9972**	0.9945	0.9610	0.9950
CX3	4	0.7275	**0.7276**	0.7263	0.7275	0.7275	0.7275	0.6948	0.7276
	8	**0.8085**	0.8026	0.8057	0.8079	0.7960	0.7980	0.7509	0.7908
	12	**0.8771**	0.8610	0.8683	0.8689	0.8523	0.8727	0.8013	0.8717
	16	**0.9136**	0.8828	0.8929	0.8928	0.9057	0.8997	0.8286	0.8987
	20	**0.9366**	0.9111	0.9261	0.8983	0.9294	0.9080	0.8535	0.9193

Continued

Images	D	MSICSA	CSA	VMCSA	IHBO	RUN_OBL	SO_OBL	SCGJO	SMA_MLS
CX4	4	**0.7747**	0.7631	0.7194	0.7548	0.7654	0.7589	0.7318	0.7611
	8	0.9011	**0.9082**	0.8994	0.9077	0.9028	0.9071	0.8614	0.8965
	12	**0.9489**	0.9307	0.9391	0.9458	0.9480	0.9476	0.8934	0.9375
	16	0.9749	0.9703	0.9765	**0.9851**	0.9786	0.9832	0.9394	0.9737
	20	**0.9943**	0.9822	0.9898	0.9915	0.9920	0.9930	0.9565	0.9854
Friedman average rank	1.99	5.25	4.06	4.92	3.20	4.53	7.83	4.22	
Rank	1	7	3	6	2	5	8	4	

Table 12. The result of SSIM for different algorithms using Otsu's method. The optimal values are highlighted in bold.

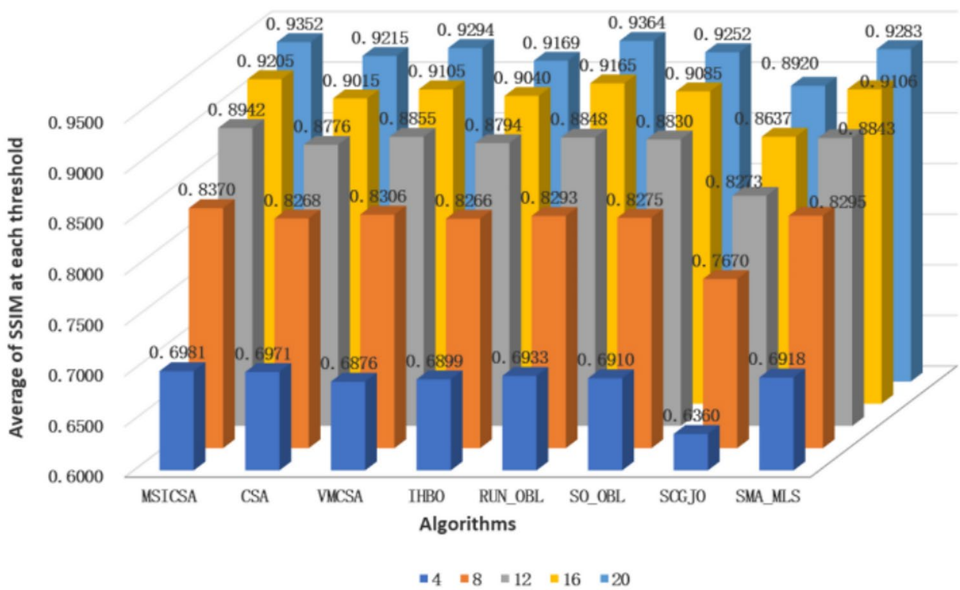


Fig. 10. The average SSIM at each threshold level using Otsu's method.

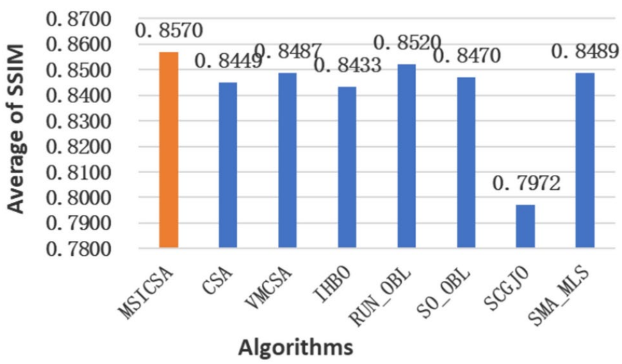


Fig. 11. The average SSIM at all threshold levels using Otsu's method.

achieves the largest value. This underscores the outstanding performance of MSICSA with respect to FSIM (Figs. 24 and 25).

Table 21 presents the optimal thresholds for various algorithms using fuzzy entropy. The convergence curves of various algorithms based on the fuzzy entropy for Img1, Img7, and CX4 are illustrated in Figs. 26, 27, and 28. With the exception of Img7 at threshold level 12 and CX4 at threshold level 16, MSICSA has demonstrated a notably faster convergence speed and achieved higher optimization accuracy.

The segmented images obtained through MSICSA with varying thresholds (D = 4, 8, 12, 16, 20) for Img1, Img7, CX4 are presented in Tables 22. Additionally, the results illustrate the locations of the optimally determined

Images	D	MSICSA	CSA	VMCSA	IHBO	RUN_OBL	SO_OBL	SCGJO	SMA_MLS
Img1	4	**0.8351**	0.8203	0.8220	0.8347	0.8299	0.8236	0.7944	0.8295
	8	**0.9053**	0.8949	0.9029	0.8942	0.8972	0.8885	0.8739	0.8861
	12	**0.9440**	0.9220	0.9236	0.9410	0.9343	0.9372	0.9018	0.9355
	16	**0.9578**	0.9386	0.9420	0.9562	0.9418	0.9422	0.9205	0.9481
	20	**0.9706**	0.9467	0.9585	0.9671	0.9618	0.9706	0.9444	0.9590
Img2	4	0.8215	0.8211	0.8216	0.8216	**0.8220**	0.8216	0.7985	0.8218
	8	**0.9144**	0.9020	0.9070	0.9095	0.9079	0.9019	0.8745	0.9089
	12	**0.9488**	0.9398	0.9431	0.9446	0.9386	0.9346	0.9106	0.9417
	16	**0.9659**	0.9469	0.9527	0.9599	0.9573	0.9513	0.9288	0.9552
	20	**0.9759**	0.9550	0.9659	0.9705	0.9703	0.9574	0.9509	0.9668
Img3	4	**0.7793**	0.7708	0.7743	0.7664	0.7753	0.7595	0.7495	0.7689
	8	**0.8812**	0.8699	0.8742	0.8613	0.8747	0.8759	0.8458	0.8729
	12	**0.9332**	0.9172	0.9236	0.9287	0.9243	0.9288	0.8762	0.9253
	16	**0.9562**	0.9393	0.9474	0.9456	0.9488	0.9535	0.9085	0.9465
	20	**0.9695**	0.9615	0.9648	0.9630	0.9636	0.9559	0.9192	0.9651
Img4	4	**0.8946**	0.8767	0.8797	0.8858	0.8804	0.8775	0.8803	0.8902
	8	**0.9601**	0.9463	0.9496	0.9413	0.9457	0.9498	0.9288	0.9488
	12	**0.9777**	0.9620	0.9736	0.9654	0.9649	0.9754	0.9481	0.9697
	16	**0.9866**	0.9649	0.9750	0.9703	0.9760	0.9770	0.9516	0.9720
	20	**0.9904**	0.9714	0.9792	0.9759	0.9758	0.9837	0.9613	0.9795
Img5	4	0.8294	0.8255	0.8284	0.8296	0.8294	0.8295	0.8216	**0.8297**
	8	**0.9072**	0.8923	0.8956	0.8922	0.9024	0.8975	0.8816	0.9048
	12	**0.9484**	0.9311	0.9407	0.9446	0.9373	0.9363	0.8984	0.9405
	16	**0.9634**	0.9512	0.9542	0.9457	0.9580	0.9544	0.9195	0.9464
	20	0.9729	0.9660	0.9717	0.9721	**0.9745**	0.9735	0.9440	0.9707
Img6	4	0.7918	0.7919	**0.7931**	0.7919	0.7918	0.7918	0.7850	0.7919
	8	0.8784	0.8787	0.8761	**0.8803**	0.8792	0.8799	0.8615	0.8781
	12	**0.9203**	0.9040	0.9120	0.8997	0.9178	0.9169	0.8882	0.9152
	16	**0.9470**	0.9345	0.9402	0.9388	0.9389	0.9397	0.9108	0.9391
	20	**0.9619**	0.9457	0.9576	0.9496	0.9577	0.9571	0.9061	0.9580
Img7	4	0.8196	0.8196	0.8195	0.8198	0.8197	0.8197	**0.8213**	0.8198
	8	**0.8456**	0.8314	0.8344	0.8247	0.8278	0.8274	0.8146	0.8334
	12	**0.8844**	0.8783	0.8794	0.8753	0.8764	0.8653	0.8320	0.8796
	16	**0.9090**	0.8911	0.8950	0.8828	0.8985	0.8850	0.8606	0.8917
	20	**0.9218**	0.9044	0.9140	0.8960	0.9157	0.9007	0.8755	0.9217
Img8	4	**0.8459**	0.8262	0.8358	0.8432	0.8445	0.8289	0.8030	0.8265
	8	**0.8958**	0.8727	0.8841	0.8863	0.8791	0.8780	0.8491	0.8898
	12	**0.9287**	0.9132	0.9170	0.9193	0.9171	0.9191	0.8756	0.9162
	16	**0.9490**	0.9340	0.9442	0.9272	0.9354	0.9435	0.9000	0.9470
	20	**0.9600**	0.9451	0.9557	0.9472	0.9453	0.9414	0.9206	0.9474
CX1	4	**0.8544**	0.8369	0.8492	0.8504	0.8413	0.8368	0.8069	0.8449
	8	**0.9584**	0.9371	0.9477	0.9382	0.9412	0.9437	0.9012	0.9521
	12	**0.9842**	0.9718	0.9741	0.9443	0.9799	0.9622	0.9413	0.9728
	16	**0.9917**	0.9722	0.9864	0.9163	0.9743	0.9608	0.8768	0.9851
	20	0.9866	0.9514	0.9602	0.8887	**0.9870**	0.9664	0.9511	0.9769
CX2	4	**0.8438**	0.8372	0.8361	0.8327	0.8417	0.8379	0.7895	0.8397
	8	**0.9614**	0.9444	0.9533	0.9468	0.9455	0.9421	0.8945	0.9496
	12	**0.9851**	0.9664	0.9706	0.9745	0.9799	0.9665	0.9229	0.9715
	16	**0.9913**	0.9731	0.9796	0.9836	0.9759	0.9903	0.9329	0.9889
	20	0.9948	0.9903	0.9940	0.9939	**0.9977**	0.9952	0.9619	0.9940
CX3	4	0.7493	**0.7494**	0.7488	0.7494	0.7493	0.7493	0.7334	0.7494
	8	**0.8193**	0.8034	0.8059	0.8070	0.8110	0.8162	0.7654	0.8041
	12	**0.8869**	0.8624	0.8757	0.8731	0.8655	0.8829	0.8190	0.8763
	16	**0.9272**	0.9071	0.9153	0.8951	0.9201	0.9191	0.8503	0.9156
	20	**0.9508**	0.9247	0.9383	0.9037	0.9327	0.9189	0.8524	0.9435

Continued

Images	D	MSICSA	CSA	VMCSA	IHBO	RUN_OBL	SO_OBL	SCGJO	SMA_MLS
CX4	4	**0.8651**	0.8558	0.8488	0.8530	0.8624	0.8634	0.8312	0.8538
	8	0.9407	**0.9448**	0.9395	0.9438	0.9415	0.9438	0.9254	0.9386
	12	**0.9601**	0.9563	0.9581	0.9583	0.9592	0.9589	0.9387	0.9540
	16	0.9783	0.9807	0.9811	**0.9869**	0.9821	0.9850	0.9603	0.9781
	20	**0.9942**	0.9835	0.9922	0.9925	0.9940	0.9932	0.9702	0.9920
Friedman average rank		1.69	5.93	4.28	4.48	3.70	4.24	7.82	3.87
Rank		1	7	5	6	2	4	8	3

Table 13. Result of FSIM for different algorithms using Otsu's method. The optimal values are highlighted in bold.

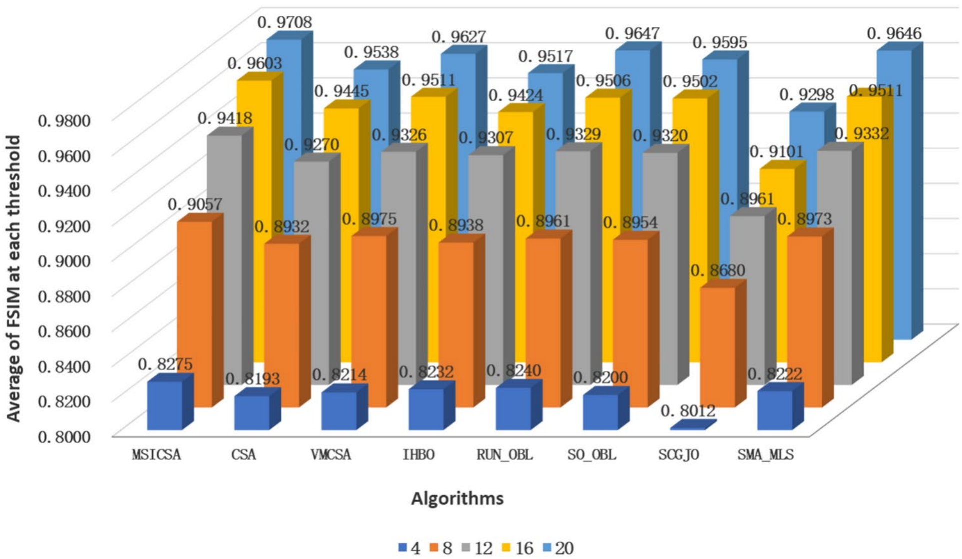


Fig. 12. Average FSIM at each threshold level using Otsu's method.

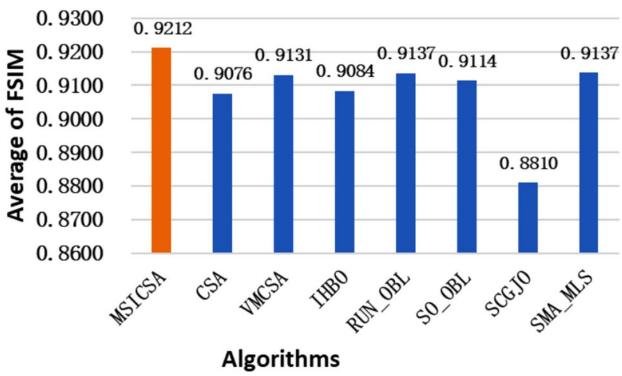


Fig. 13. Average FSIM at all threshold levels using Otsu's method.

thresholds within the histogram for Img1, Img7, CX4. In Table 22, the outcomes of segmentation are visualized using distinct colors, with each color representing a different segment.

The average CPU times spent by different algorithms at each threshold using fuzzy entropy are given in Fig. 29. It shows that the CPU time of MSICSA is acceptable. At all thresholds, the runtime of MSICSA is quite comparable to that of CSA, VMCSA, IHBO, SO_OBL, and SOGJO. However, it demonstrates a shorter runtime than RUN_OBL and SMA_MLS. It is worth noting that the largest difference in average running time between

Images	D	MSICSA	CSA	VMCSA	IHBO	RUN_OBL	SO_OBL	SCGJO	SMA_MLS
Img1	4	40 84 143 208	41 86 147 209	39 82 142 208	40 86 144 208	40 84 145 210	40 84 143 208	47 95 162 222	40 82 143 207
	8	24 52 81 111 148 182 208 232	24 54 81 111 146 181 208 232	26 53 84 116 155 184 211 235	26 52 81 111 145 182 208 234	24 51 81 111 147 182 207 232	24 52 81 111 148 182 208 232	28 67 91 115 158 182 214 238	23 52 81 111 146 182 210 233
	12	16 33 52 71 88 106 130 156 179 198 217 235	16 33 52 70 88 107 130 157 179 198 217 235	17 36 56 74 91 107 132 154 178 197 216 234	16 33 52 70 88 107 130 157 180 199 218 235	16 33 52 71 88 107 130 157 179 198 217 235	16 33 52 70 88 107 130 157 179 198 217 235	17 42 67 97 127 160 187 208 220 234 248 253	16 34 53 71 88 107 131 158 180 199 217 235
	16	13 25 39 54 68 82 97 113 134 156 175 190 206 220 233 244	13 26 40 55 70 84 99 116 136 157 174 188 204 219 233 244	14 25 39 56 70 84 94 112 131 150 171 186 205 217 232 243	13 26 39 54 69 84 99 115 135 158 176 191 206 220 233 244	13 26 39 53 67 81 95 111 131 153 171 186 202 217 232 244	13 26 39 54 69 84 99 116 136 158 176 191 207 221 234 245	7 29 53 74 89 103 113 143 162 185 200 208 225 231 243 253	13 26 39 53 68 82 97 113 134 157 176 192 208 222 235 245
	20	11 22 33 45 56 66 77 88 100 114 129 146 162 176 189 202 215 226 237 246	11 21 31 41 53 64 76 88 99 112 125 143 159 174 187 201 213 225 236 245	13 24 38 49 61 67 78 90 101 114 130 146 162 179 192 204 216 226 236 245	10 20 31 43 55 67 80 90 101 114 129 146 160 174 186 200 214 225 236 246	10 20 31 42 54 66 77 88 99 112 127 144 160 173 185 198 212 224 235 245	10 19 29 40 52 64 76 88 101 114 132 152 169 181 194 206 218 229 238 246	8 23 35 56 71 93 112 114 118 136 148 163 181 190 204 221 223 230 239 253	11 20 30 41 52 64 77 91 104 119 137 153 168 180 192 204 215 225 235 244
Img2	4	34 74 118 175	35 77 119 175	35 74 119 176	36 74 118 178	35 74 117 175	34 76 118 176	50 98 142 178	34 73 117 177
	8	20 44 66 87 110 137 168 200	20 44 66 85 111 137 167 200	23 48 68 89 108 136 170 200	22 44 67 87 110 135 167 200	20 44 66 87 113 138 165 200	20 46 66 87 112 135 168 202	32 64 93 108 131 153 186 238	22 43 68 87 110 137 168 200
	12	15 32 49 65 79 93 110 130 152 175 199 225	16 33 49 65 79 93 110 130 152 175 199 225	16 30 47 66 80 91 109 129 153 174 201 223	15 32 49 66 79 93 110 130 152 175 200 225	15 32 48 64 79 93 110 130 151 174 199 225	15 32 48 64 79 93 110 130 152 175 199 225	17 38 67 90 112 149 157 176 187 212 234 247	15 32 48 64 78 93 109 129 151 174 199 224
	16	14 28 41 53 66 77 87 98 112 128 144 159 177 195 210 228	13 26 38 51 64 76 86 97 110 126 143 159 176 194 209 231	16 32 46 57 68 78 86 96 109 125 139 153 171 190 207 229	12 24 36 50 63 75 86 98 113 130 147 164 180 197 212 230	13 26 39 52 65 76 86 97 111 127 144 161 177 194 209 228	13 25 40 53 66 77 88 99 113 130 147 164 181 197 211 229	20 45 58 73 91 99 105 122 133 140 152 172 186 203 209 242	12 25 38 50 64 76 86 97 111 127 143 159 176 194 209 226
	20	11 22 32 42 53 64 74 83 91 100 110 122 136 148 160 174 189 202 213 232	12 23 35 46 56 66 75 83 91 101 113 126 138 150 162 175 187 200 211 229	13 26 36 47 56 65 75 85 94 104 114 126 140 155 168 181 195 207 218 232	12 23 35 45 56 67 76 85 95 107 120 134 148 162 175 189 201 211 222 237	11 21 31 42 53 65 75 84 93 102 113 124 136 148 161 174 187 200 212 230	12 24 35 45 56 66 75 84 93 102 112 127 139 150 161 173 187 201 213 230	10 17 37 61 77 97 114 122 141 151 163 180 194 209 217 231 233 243 245 253	13 23 35 45 55 67 76 85 92 101 111 123 134 146 158 171 188 202 213 230
Img3	4	52 98 141 188	52 98 141 188	52 98 141 187	52 98 141 188	52 98 141 188	52 98 141 188	53 106 154 199	52 98 141 188
	8	31 59 84 108 134 159 187 220	31 57 85 109 135 159 186 220	32 61 83 111 136 161 186 217	30 61 82 108 134 159 187 221	30 59 84 106 134 159 189 223	31 56 84 108 132 159 187 220	35 69 93 116 146 163 198 234	31 59 84 108 134 159 187 220
	12	22 40 58 74 90 108 126 143 161 179 200 225	23 41 58 74 90 108 126 144 161 179 200 226	22 38 54 71 90 108 123 141 157 176 199 229	24 42 60 76 92 109 126 143 161 180 201 228	22 40 58 74 90 108 126 143 161 179 200 226	22 40 58 75 91 108 126 143 161 180 201 226	35 62 89 99 110 132 160 172 201 226 252 253	23 41 58 74 90 107 126 143 161 180 201 226
	16	20 35 49 63 76 88 102 115 128 141 155 168 181 197 213 231	21 36 51 65 77 90 103 116 129 142 154 167 181 197 214 234	18 34 45 59 73 87 100 113 127 140 151 163 179 194 209 228	18 31 44 57 70 83 97 111 125 139 153 166 181 197 213 232	19 32 45 58 70 82 94 107 121 134 147 160 174 190 207 229	20 36 50 64 78 91 104 118 131 143 155 167 182 198 214 233	24 37 54 73 76 98 122 135 147 176 194 216 221 242 245 253	17 30 43 56 69 81 94 109 123 137 150 163 176 190 206 229
	20	18 29 40 51 61 71 80 90 101 111 122 133 144 155 166 177 190 204 218 236	17 28 40 51 62 72 82 92 103 114 125 136 147 158 168 180 195 209 224 239	17 29 40 52 62 72 83 92 103 117 130 141 152 162 173 184 199 213 226 239	19 29 40 49 59 70 81 93 104 116 127 139 152 162 174 186 200 214 228 241	16 27 37 49 60 71 81 91 101 112 123 134 145 157 168 180 193 206 221 238	17 28 38 48 60 71 82 93 104 116 127 138 150 161 172 185 198 210 224 240	28 47 52 73 86 94 112 124 141 149 153 169 174 192 196 198 208 219 230 246	18 31 43 53 63 73 81 89 99 108 118 131 141 153 165 176 191 205 221 239
Img4	4	34 76 136 209	34 76 136 209	34 75 133 205	34 76 136 209	34 76 136 209	34 76 136 209	49 97 162 221	34 76 136 209
	8	24 46 66 91 123 159 197 234	22 45 66 94 123 157 197 234	22 45 67 92 128 165 204 236	24 46 68 91 123 157 197 232	24 46 66 91 123 159 197 234	24 46 65 92 123 160 197 234	33 67 102 130 173 202 236 253	24 46 66 90 124 158 198 234
	12	19 35 47 60 76 95 116 140 165 191 216 241	19 35 47 60 76 95 117 141 166 192 217 241	17 33 45 59 75 92 118 146 168 195 218 240	17 33 46 59 75 94 115 140 165 191 216 241	19 36 48 61 77 96 117 141 166 192 217 241	19 36 48 61 76 94 115 139 165 191 217 241	13 36 47 70 96 118 145 185 213 229 245 253	19 36 48 60 75 94 115 139 165 193 219 243
	16	16 29 40 49 60 71 83 97 112 130 148 168 188 208 227 245	15 29 40 49 59 70 82 97 112 129 149 169 189 209 227 245	16 30 40 49 59 68 81 98 111 131 146 169 189 208 230 243	16 29 40 49 58 69 82 98 114 129 146 164 184 205 226 245	16 29 39 48 58 69 81 95 111 128 147 167 187 207 227 245	17 30 41 51 61 72 84 98 114 131 147 169 189 209 227 245	24 39 50 61 67 89 119 144 168 184 188 212 221 234 249 253	17 31 43 51 61 73 86 99 114 129 146 166 185 207 226 244
	20	12 22 31 39 47 55 64 74 84 95 107 118 132 147 164 181 198 215 231 246	12 22 31 40 47 56 65 74 85 97 109 123 137 152 168 184 200 217 232 246	12 23 33 41 50 58 68 77 87 96 109 122 133 147 162 178 194 213 227 244	12 22 34 44 53 61 70 80 91 104 118 128 143 157 174 187 204 218 231 246	12 23 33 41 49 58 67 77 88 101 114 129 144 159 174 190 205 220 234 248	11 21 31 40 48 56 63 71 80 92 106 121 137 154 168 182 197 212 229 245	8 17 32 33 49 64 73 94 99 112 127 142 159 186 196 212 234 244 249 253	13 23 32 40 47 54 63 70 80 91 104 119 138 154 167 183 199 217 232 247
Continued									

Images	D	MSICSA	CSA	VMCSA	IHBO	RUN_OBL	SO_OBL	SCGJO	SMA_MLS
Img5	4	56 100 155 211	56 100 155 211	56 102 156 211	56 100 155 211	56 100 155 211	56 100 155 211	72 139 179 220	56 100 155 211
	8	33 53 78 105 135 171 203 230	31 53 76 102 135 174 203 230	38 62 91 116 143 175 205 233	35 53 77 103 135 171 203 230	33 53 81 104 135 171 203 232	36 57 78 105 134 168 203 230	34 59 94 118 154 188 213 239	33 54 78 107 135 171 203 232
	12	27 42 59 79 100 119 138 160 182 202 220 238	27 42 59 79 100 118 137 159 181 202 221 239	28 44 61 79 99 117 137 158 178 197 216 233	27 42 59 80 101 119 138 159 181 201 220 238	27 42 59 80 101 120 139 160 182 202 220 238	27 42 59 79 100 119 138 160 182 202 220 238	35 46 69 96 125 146 174 199 216 233 246 253	27 42 58 79 100 119 139 161 183 202 221 238
	16	24 36 47 60 75 90 106 119 135 152 169 185 200 215 229 242	23 36 47 60 75 91 105 119 134 151 168 185 200 214 228 242	24 36 49 62 78 93 108 125 145 158 171 188 203 217 231 242	24 35 46 59 75 90 104 119 134 150 167 184 199 212 226 241	24 35 46 60 74 89 104 118 133 150 167 184 199 214 228 242	24 35 46 59 74 89 104 119 135 152 169 185 200 214 228 241	33 49 67 84 99 125 151 164 184 195 214 225 236 246 252 253	25 37 48 62 77 92 108 123 139 158 174 189 201 214 228 241
	20	22 31 39 48 58 69 81 94 105 116 129 143 159 173 186 198 210 222 234 245	23 33 41 50 60 72 85 97 109 120 132 146 160 174 187 197 208 220 232 244	21 32 41 51 64 79 91 103 113 125 140 152 164 179 190 199 209 221 231 244	21 31 39 48 59 70 83 95 106 118 131 146 159 172 184 195 207 218 231 243	21 31 39 48 59 70 83 95 106 117 130 144 159 174 187 199 211 222 233 244	22 32 40 49 59 70 82 95 107 118 131 145 158 171 184 196 208 220 232 244	20 41 53 67 79 83 97 126 127 142 160 179 190 204 218 221 235 236 244 253	24 32 41 52 64 75 86 98 110 121 134 145 155 169 182 194 206 218 230 243
Img6	4	60 98 138 184	60 98 138 184	60 97 138 182	62 98 138 184	60 96 138 184	60 98 138 184	64 106 163 252	60 99 138 185
	8	28 55 73 96 118 140 168 206	25 59 73 96 118 144 165 206	25 57 73 95 117 140 168 208	28 55 73 96 119 142 170 208	29 55 75 96 118 142 168 208	31 53 73 98 118 140 168 206	42 69 98 111 122 152 177 220	29 55 73 96 118 140 168 206
	12	24 47 61 74 89 105 118 131 147 166 188 219	24 47 62 75 89 105 118 131 147 166 188 219	22 45 62 77 91 107 121 135 153 171 198 228	22 44 58 72 88 104 118 132 148 167 191 221	22 44 58 72 87 103 117 131 148 167 189 220	23 45 59 72 86 101 116 131 148 167 189 220	13 51 66 83 102 129 141 157 183 202 223 253	23 46 61 75 89 105 118 132 148 167 188 218
	16	19 38 52 63 74 84 96 108 119 130 142 155 169 185 207 231	17 38 52 63 73 84 96 108 119 130 142 155 170 186 207 233	21 35 49 62 73 83 95 107 118 130 141 152 166 187 209 231	18 39 53 63 74 85 97 110 121 132 144 157 171 186 206 232	18 39 53 64 75 86 99 111 121 132 144 157 171 187 206 232	18 39 53 64 74 85 97 109 119 130 142 156 170 187 208 233	15 40 66 92 112 119 121 138 149 169 198 201 208 221 246 253	17 37 52 63 73 84 96 108 117 128 140 152 166 181 202 229
	20	16 33 46 55 63 72 81 90 100 110 119 127 136 147 158 170 183 198 216 237	16 34 46 55 63 71 79 88 99 109 118 127 138 148 159 172 185 201 219 237	13 34 47 59 68 77 87 97 106 115 123 131 139 150 161 171 185 200 215 237	19 39 49 59 69 78 84 92 103 113 121 131 141 153 164 175 187 205 218 243	16 32 47 57 67 76 84 94 104 113 121 130 140 151 163 175 187 202 219 239	17 34 47 56 66 76 86 98 109 118 127 136 146 157 168 180 195 212 228 243	21 32 52 62 71 80 92 112 132 147 159 168 185 200 225 234 243 250 253 253	14 33 47 58 66 74 82 90 98 108 116 123 131 141 152 163 175 190 208 232
Img7	4	66 93 126 176	66 93 126 176	66 93 126 174	66 93 126 176	66 93 126 176	66 93 126 176	68 99 134 198	66 93 126 176
	8	46 60 76 91 105 126 156 195	46 60 76 93 104 126 153 196	44 59 75 90 105 131 163 195	45 60 76 91 103 126 156 195	46 60 78 91 105 126 156 195	46 60 76 91 105 126 156 195	48 71 86 103 124 159 205 234	45 59 75 90 104 125 155 194
	12	39 50 60 71 83 94 104 116 133 154 179 208	39 50 60 72 83 93 103 115 132 154 179 208	37 49 60 70 82 93 104 115 130 155 178 203	40 52 63 77 89 99 110 123 140 160 185 213	38 49 59 70 82 93 103 115 131 152 177 207	38 48 59 72 85 95 105 118 135 155 179 208	47 61 78 98 110 130 163 193 210 218 220 231	38 49 59 71 83 94 103 115 131 154 180 210
	16	27 42 50 58 66 77 86 94 102 111 123 139 156 174 193 218	28 42 50 58 67 77 86 94 102 112 125 140 156 174 195 217	30 42 51 60 68 75 85 93 103 113 126 140 154 173 184 209	30 43 51 59 69 79 89 98 108 121 138 156 174 192 209 226	29 41 50 57 67 77 87 95 104 114 126 141 158 175 194 217	29 42 50 59 69 80 90 99 108 119 132 147 163 183 204 223	45 60 68 71 87 98 104 119 150 165 195 201 213 225 237 242	29 42 49 57 65 75 85 94 102 110 120 135 153 172 194 214
	20	22 39 46 52 58 65 73 80 88 95 102 111 119 131 145 158 173 190 211 225	29 39 46 52 59 66 74 81 88 95 102 109 118 129 142 157 173 190 204 223	24 40 47 54 58 68 76 83 92 97 103 110 115 127 142 158 179 200 218 231	27 38 44 50 57 66 77 87 95 102 111 122 137 153 166 177 192 207 221 234	25 37 45 51 58 66 75 84 92 99 107 114 123 135 147 160 175 191 208 225	30 40 46 53 61 70 80 89 97 105 113 124 139 154 168 182 196 209 222 235	33 45 56 77 94 103 113 116 127 172 185 199 206 207 216 229 233 242 244 245	29 41 47 52 58 65 74 82 90 97 104 109 116 124 136 148 158 174 193 220
Img8	4	38 81 134 193	38 81 134 193	38 81 134 191	38 81 134 193	38 81 134 193	38 81 134 193	46 111 163 225	38 81 134 193
	8	26 44 68 98 129 152 168 203	26 44 67 98 129 151 168 205	28 48 69 98 128 153 168 207	26 44 68 97 130 152 168 203	25 45 68 98 129 152 168 203	26 44 68 98 128 151 168 203	36 70 96 131 161 190 210 234	27 42 68 98 129 152 168 203
	12	21 33 46 62 83 105 127 144 156 169 193 225	21 34 47 63 84 106 128 145 157 169 193 225	18 32 46 66 88 107 127 143 157 171 192 220	21 34 48 65 85 107 127 144 156 169 193 225	21 33 47 63 83 105 128 145 157 169 193 225	20 32 46 64 84 106 128 145 157 169 192 225	26 54 61 72 92 113 126 141 163 190 220 246	21 33 47 64 85 107 128 145 157 169 196 227
	16	17 27 37 47 59 72 88 104 119 135 147 156 165 174 195 224	17 27 37 45 56 71 87 105 124 139 151 161 171 188 211 233	17 25 36 47 61 74 87 107 124 137 146 155 166 176 193 222	18 27 38 49 62 79 97 113 130 143 153 163 173 190 213 233	18 27 37 47 57 70 85 101 117 133 146 156 165 174 195 226	18 28 38 50 65 82 99 115 131 144 154 164 173 189 213 234	23 41 67 96 130 143 154 159 167 179 202 210 213 228 246 249	19 28 37 47 58 71 85 99 113 130 144 154 164 174 195 226
	20	14 25 34 42 52 61 70 78 88 95 101 107 114 120 128 134 141 148 158 171	11 23 34 44 51 59 68 76 85 93 99 106 112 119 125 132 140 150 162 175	20 31 40 46 51 59 65 69 79 91 97 104 110 117 124 137 143 149 162 173	18 39 44 48 63 67 71 75 77 85 90 104 114 118 119 133 137 158 163 179	11 24 34 43 51 60 68 77 86 94 102 110 117 123 130 137 145 153 163 175	11 30 43 54 64 68 76 86 89 99 104 110 118 124 128 132 137 146 160 172	5 10 19 28 46 60 82 93 100 113 119 125 126 132 143 144 161 177 186 194	8 22 31 39 47 55 63 72 79 88 97 104 111 118 125 132 140 149 161 172
Continued									

Images	D	MSICSA	CSA	VMCSA	IHBO	RUN_OBL	SO_OBL	SCGJO	SMA_MLS
CX1	4	42 76 107 135	42 76 107 135	41 76 106 135	42 76 107 135	42 76 107 135	42 76 107 135	51 85 115 144	42 76 107 135
	8	21 45 65 85 103 119 135 154	20 47 65 85 104 117 135 157	20 46 65 85 103 116 131 152	21 46 65 88 103 117 135 155	23 45 65 85 103 121 137 157	21 45 65 87 105 117 135 154	25 56 86 105 127 150 173 186	21 45 65 84 103 118 133 156
	12	17 35 48 61 75 89 101 112 123 134 146 162	17 35 49 62 76 89 101 112 122 133 146 163	14 33 47 59 70 86 100 113 125 136 147 159	19 37 52 67 83 96 109 120 130 141 154 169	17 36 50 64 78 91 103 114 124 135 147 163	17 35 50 64 78 91 103 113 123 134 146 162	26 49 55 73 90 103 117 133 142 159 176 194	17 35 50 63 76 90 102 113 123 134 146 162
	16	15 31 41 51 60 71 82 91 100 109 117 125 134 143 154 169	14 30 41 52 63 74 84 94 102 110 117 125 134 144 155 168	11 23 30 42 52 65 75 88 96 105 115 124 133 140 150 166	20 38 47 56 68 81 96 105 114 123 132 139 140 151 163 169	13 26 36 46 56 66 76 87 97 106 115 123 132 141 152 167	16 32 44 56 68 80 92 102 112 120 128 135 143 151 160 173	8 11 24 30 42 51 72 96 108 123 135 144 167 179 187 194	13 29 39 49 59 69 79 90 98 105 113 121 129 138 148 162
	20	17 26 35 44 54 64 76 89 103 115 126 136 145 153 160 167 174 192 215 237	17 25 35 43 52 61 70 85 100 115 127 138 146 154 161 167 175 193 214 232	16 24 34 42 51 60 74 86 105 120 131 143 151 159 166 172 184 204 220 236	17 25 32 40 49 59 71 86 101 115 131 142 150 158 165 172 178 195 216 234	16 22 29 37 45 56 68 81 95 109 123 135 144 152 159 166 174 190 213 233	17 25 33 41 49 58 69 81 97 112 127 141 151 159 167 175 191 210 226 238	28 43 67 86 112 130 131 133 138 139 153 172 194 195 199 205 212 231 245 250	19 27 32 38 46 54 65 78 90 102 115 129 141 150 159 166 175 192 216 235
CX2	4	79 106 130 152	77 106 132 150	79 101 130 154	78 106 130 152	79 107 130 152	79 106 130 152	93 126 150 245	80 103 131 155
	8	57 80 95 110 124 137 151 166	57 82 95 110 127 135 151 168	61 83 99 114 125 138 150 167	55 80 93 110 123 137 151 165	57 81 95 110 125 136 151 167	56 80 95 110 125 137 150 165	65 94 118 141 159 175 205 235	57 82 97 112 127 138 151 168
	12	43 65 79 90 100 110 120 129 139 149 159 173	47 67 81 92 103 114 123 132 141 150 160 173	37 57 74 84 97 107 119 129 140 150 159 173	42 63 78 90 102 113 123 132 142 151 161 173	42 62 77 89 100 111 121 131 140 150 160 173	49 69 81 91 102 113 122 131 140 150 160 173	17 26 63 87 101 116 124 146 163 234 243 250	48 66 80 92 103 114 124 133 142 151 160 174
	16	42 60 71 80 88 96 104 111 120 127 135 143 151 158 166 177	39 60 73 82 90 97 105 112 119 126 134 142 150 158 167 178	47 62 71 79 88 97 105 114 123 129 137 144 152 160 174 185	28 56 66 77 84 93 99 108 119 129 140 147 154 161 170 183	42 52 66 76 86 95 104 113 121 128 135 143 151 159 167 178	33 46 65 77 84 95 106 117 125 132 140 148 155 162 168 181	39 57 93 103 113 117 129 146 158 163 182 228 231 239 241 249	40 56 70 78 87 94 103 111 118 123 130 137 144 152 162 174
	20	40 57 68 76 83 90 98 105 113 119 125 131 136 142 149 155 162 170 185 222	22 46 61 72 82 91 99 107 113 120 126 132 138 143 149 155 162 169 180 241	37 51 61 75 84 92 102 107 112 119 126 133 137 145 152 160 170 180 195 209	17 31 46 58 71 80 91 98 105 117 125 133 138 141 148 154 158 165 172 185	13 32 45 58 71 80 89 97 105 113 120 126 132 138 144 150 156 161 168 178	28 41 52 60 72 83 92 101 110 118 125 133 139 145 151 157 163 171 182 235	15 55 71 78 88 99 119 123 125 145 161 162 173 194 201 225 228 237 240 250	31 51 62 72 79 84 90 98 105 113 120 127 134 141 147 153 159 165 175 185
CX3	4	37 93 136 173	37 93 135 175	37 91 136 174	35 93 136 173	37 93 136 173	34 93 133 173	42 99 141 178	36 93 137 173
	8	23 59 89 115 136 154 174 193	23 61 89 115 137 154 178 193	23 60 87 114 134 154 177 194	26 59 89 115 139 157 173 193	23 59 90 115 134 154 174 193	25 59 89 117 133 154 172 193	16 62 99 129 156 179 195 208	21 59 89 115 137 156 177 195
	12	15 39 61 81 100 115 129 142 155 170 185 198	13 37 60 81 100 116 130 143 156 171 186 198	14 42 64 85 102 116 128 142 156 167 182 198	11 36 58 80 99 115 129 142 156 170 184 198	12 35 57 77 96 112 127 141 155 170 185 198	13 37 59 79 99 115 129 142 156 171 185 198	48 65 70 84 85 104 118 141 160 178 205 219	16 41 61 82 101 115 128 142 155 170 185 197
	16	11 28 45 64 80 95 108 120 130 141 152 162 172 183 193 202	10 28 46 61 77 93 106 117 128 140 150 161 172 183 193 202	14 29 44 57 75 92 107 119 128 139 149 160 171 181 191 202	9 26 43 59 76 91 104 117 130 141 151 161 172 183 193 202	6 25 44 60 76 91 105 117 129 141 152 163 174 185 194 202	10 29 48 65 82 96 108 119 129 140 152 162 172 183 192 201	13 25 41 70 80 91 115 126 140 155 167 173 183 204 218 219	13 38 57 74 91 103 115 127 138 146 155 163 172 183 192 202
	20	8 24 41 55 70 83 95 108 117 126 134 141 149 155 162 170 178 186 194 202	9 29 43 54 65 75 86 97 106 115 124 134 143 152 161 169 179 188 196 203	11 33 48 61 72 79 92 105 114 123 130 141 151 159 167 174 183 188 196 204	10 29 48 60 72 85 98 110 120 130 140 149 158 168 176 184 191 196 201 206	8 21 33 47 61 74 87 99 109 118 127 136 144 153 162 171 180 188 196 203	10 22 36 49 60 72 85 96 107 117 126 135 143 153 162 174 184 190 197 204	36 58 81 95 98 112 113 114 117 123 127 135 148 159 167 186 203 204 215 218	10 24 41 56 72 89 100 110 117 126 136 145 152 158 164 172 179 188 195 204
CX4	4	35 87 157 225	35 89 157 226	29 87 158 225	33 89 157 225	34 87 155 223	35 86 157 224	40 91 164 227	35 85 158 224
	8	31 68 84 107 146 191 223 242	30 70 84 107 146 193 223 244	24 70 85 107 145 192 225 242	31 69 84 109 146 190 223 242	33 67 84 107 146 193 223 240	31 68 84 108 144 191 225 242	36 81 108 163 201 219 235 252	31 65 84 109 146 190 223 245
	12	30 61 71 82 95 115 142 172 200 220 233 245	30 61 71 82 97 116 142 173 200 220 233 245	19 61 72 83 93 112 141 168 195 218 232 246	26 61 71 82 95 114 141 172 199 220 233 245	29 61 71 82 95 114 140 171 200 220 233 245	29 62 72 82 95 114 141 172 200 220 233 245	38 66 87 108 142 158 172 216 238 247 251 253	29 61 71 81 94 113 140 170 199 220 234 246
	16	24 57 66 75 84 95 110 130 152 176 196 212 223 232 239 248	34 58 66 74 83 94 108 128 149 172 196 213 223 232 240 248	28 60 69 75 84 96 112 127 150 177 200 213 220 229 236 247	27 58 66 75 83 93 107 125 148 173 196 212 223 232 239 248	14 53 62 70 79 89 102 118 139 163 187 205 217 228 237 247	27 57 65 73 81 91 104 120 139 161 184 203 216 227 237 247	6 49 68 87 103 120 135 138 146 173 199 216 228 236 250 253	27 55 64 72 81 90 103 118 137 161 185 203 217 228 237 247
	20	22 53 61 68 76 84 92 103 117 133 149 167 183 197 211 220 227 234 242 250	26 54 62 69 77 84 94 106 119 136 154 171 187 203 212 220 227 235 241 249	24 52 62 70 75 82 91 102 115 135 160 171 192 205 211 217 228 235 241 247	22 50 59 65 71 77 85 94 106 122 139 158 177 195 208 216 225 233 240 249	10 35 52 61 68 76 84 94 106 119 135 153 171 189 204 214 223 232 239 248	23 51 59 65 72 78 86 95 107 122 140 161 178 196 208 217 226 234 241 249	4 9 50 66 81 97 114 120 141 170 174 190 200 223 224 236 239 250 251 253	31 50 59 65 72 79 87 98 112 131 153 172 193 206 216 225 232 238 244 249

Table 14. Te best thresholds values for different algorithms using Otsu's method.

MSICSA and CSA occurs at threshold level 8, where MSICSA takes an additional 0.25 s compared to CSA, resulting in an increase of 5.39% in time.

In general, MSICSA demonstrates superior performance compared to other comparative algorithms when evaluated on mean fitness values, PSNR, SSIM, and FSIM using fuzzy entropy. The runtime of MSICSA is slightly longer than that of CSA across various thresholds. Furthermore, its robustness is inferior to that of RUN_OBL and SO_OBL, yet superior to that of CSA, VMCSA, IHBO, SCGJO and SMA_MLS.

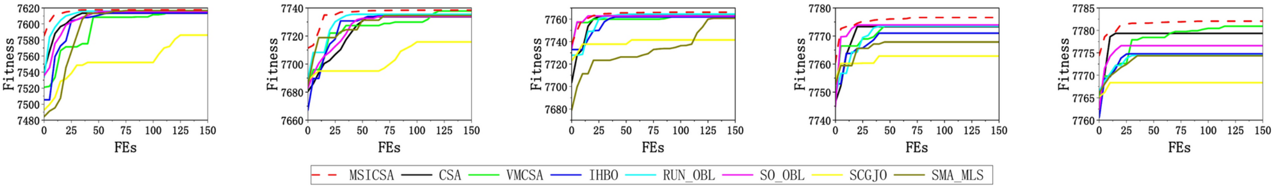


Fig. 14. Convergence curves of Img1, D = 4, 8, 12, 16 and 20 using Otsu's method.

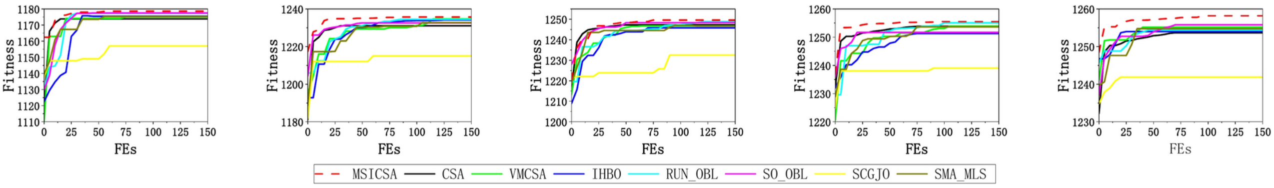


Fig. 15. The convergence curves of Img7, D = 4, 8, 12, 16 and 20 using Otsu's method.

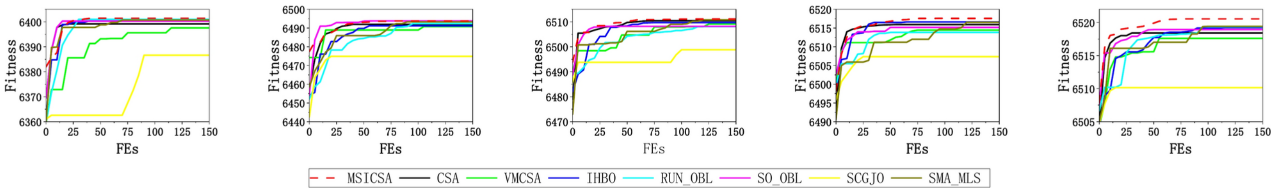


Fig. 16. The convergence curves of CX4, D = 4, 8, 12, 16 and 20 using Otsu's method.

Conclusions and future works

This paper introduces a novel approach called Multi-Strategy Improved Crow Search Algorithm (MSICSA) for solving the multi-threshold segmentation problem. The proposed algorithm incorporates three key enhancement strategies: Opposition-based Learning (OBL), adaptive awareness probability adjustment, and the utilization of two differential mutation operators. OBL enhances the quality of initial solutions, adaptive awareness probability balances the global and local search exploration, and two differential mutation operators strengthen global search capabilities and enhance population diversity. The CEC 2020 benchmark functions are utilized to assess and evaluate the performance of the proposed MSICSA. The results indicate that MSICSA outperforms the other algorithms tested. Additionally, multi-level threshold image segmentation is employed to examine the effectiveness of MSICSA. MAICSA evaluates grayscale benchmark images at various threshold levels based on Otsu's method and fuzzy entropy. According to the results from the image segmentation, MSICSA demonstrates a significant positive impact on image segmentation, showcasing excellent optimization capabilities. Experimental results further reveal that MSICSA performs admirably in both global optimization and image segmentation tasks.

Due to the effectiveness of the proposed MSICSA in this paper, in the future, we can try to use MSICSA for different application issues in the real world. Such applications include color image segmentation, satellite image segmentation, feature selection and clustering problems. We can also try combining OBL and differential mutation operators with other meta-heuristic optimization algorithms to see if they can improve the performance of the standard algorithms.

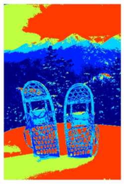
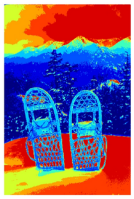
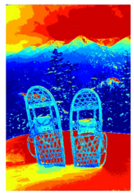
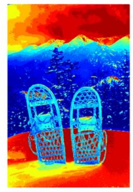
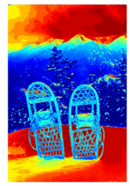
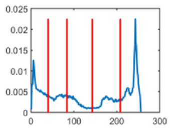
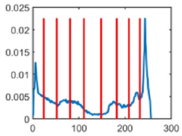
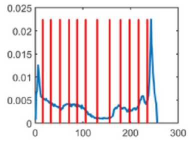
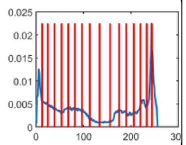
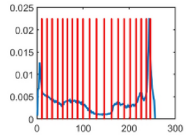
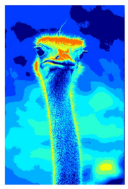
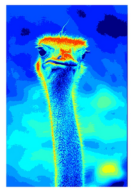
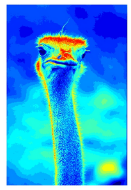
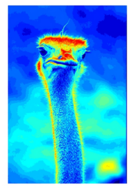
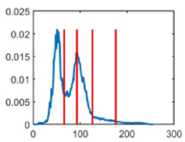
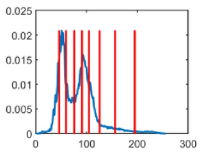
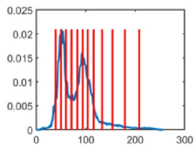
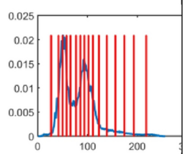
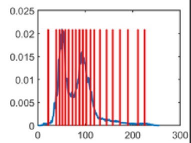
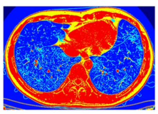
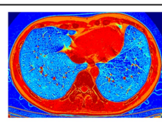
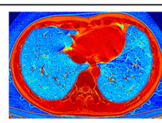
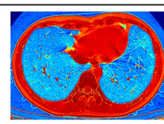
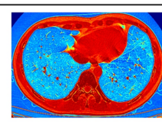
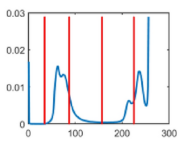
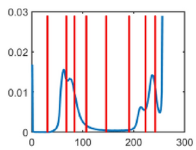
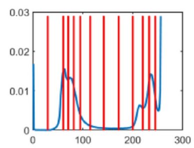
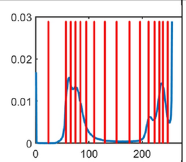
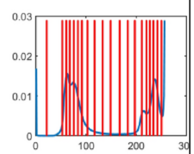
Images	D=4	D=8	D=12	D=16	D=20
Img1					
					
Img7					
					
CX4					
					

Table 15. Segmentation results of MSICSA using Otsu's method.

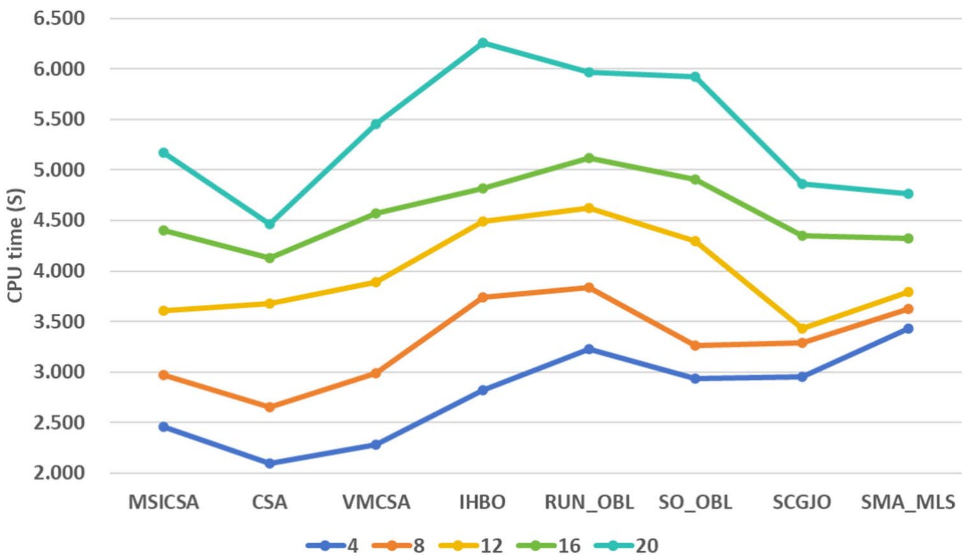


Fig. 17. Average CPU time of different algorithms at each threshold level using Otsu's method.

Images	D	MSICSA	CSA	VMCSA	IHBO	RUN_OBL	SO_OBL	SCGJO	SMA_MLS
Img1	4	**14.111**	12.869	13.366	13.435	13.301	12.822	13.727	14.077
	8	**21.545**	20.707	21.230	20.324	21.087	20.280	19.892	21.352
	12	**27.967**	26.232	26.673	26.746	27.020	27.425	26.296	26.840
	16	**33.882**	32.538	32.934	32.889	32.975	32.870	32.736	33.356
	20	**39.283**	38.674	38.893	37.087	37.225	37.413	36.104	38.217
Img2	4	13.813	13.814	13.811	13.812	13.814	13.813	13.742	**13.815**
	8	20.950	20.753	20.871	20.943	20.950	**20.955**	20.632	20.939
	12	27.006	26.040	27.030	27.002	**27.073**	27.066	26.597	27.002
	16	32.688	32.505	32.815	32.679	32.840	**32.902**	32.022	32.738
	20	37.815	37.207	37.684	37.686	38.031	38.025	37.020	**38.064**
Img3	4	**14.538**	13.270	12.985	14.499	13.872	13.587	13.185	13.596
	8	**21.770**	20.362	21.294	20.405	20.221	19.858	20.208	20.989
	12	**28.109**	27.493	27.401	27.443	27.283	26.435	26.972	27.260
	16	**33.796**	33.354	32.114	32.658	32.144	32.823	31.804	32.490
	20	**39.091**	38.529	37.652	37.011	37.151	38.669	37.486	37.591
Img4	4	**13.046**	12.138	12.877	11.138	12.081	11.461	11.579	12.292
	8	**20.339**	18.387	18.648	18.666	19.890	18.477	19.155	18.706
	12	**26.770**	25.074	25.864	25.188	25.494	25.914	24.540	25.973
	16	**32.516**	31.188	31.754	32.100	30.689	32.462	31.432	32.177
	20	**37.729**	36.649	35.954	37.212	37.164	36.122	34.743	35.839
Img5	4	**14.170**	12.627	12.282	13.280	13.916	13.511	13.475	13.696
	8	22.056	21.700	21.789	21.717	22.187	22.038	**22.195**	21.933
	12	**27.971**	26.430	27.449	26.456	27.811	27.895	25.749	27.788
	16	**33.698**	31.920	32.548	31.866	33.170	33.609	31.662	33.440
	20	**39.029**	37.620	38.050	38.761	38.509	38.413	36.725	37.084
Img6	4	**14.222**	13.082	12.269	12.239	13.113	13.532	13.780	14.219
	8	**21.340**	19.907	20.619	19.862	20.390	19.858	19.089	21.039
	12	**27.504**	25.479	26.884	26.050	26.003	26.281	26.408	26.553
	16	**33.055**	31.062	32.315	32.486	31.087	32.325	31.772	32.004
	20	**38.015**	37.705	36.012	36.767	36.667	36.799	35.948	37.587
Img7	4	**13.776**	13.220	13.257	13.523	12.842	13.631	12.558	13.620
	8	**21.144**	19.908	20.928	20.327	20.246	20.755	20.381	20.700
	12	27.717	27.367	27.590	28.538	28.644	28.381	28.780	**28.828**
	16	**33.244**	32.771	32.593	32.525	31.626	31.863	30.544	32.912
	20	**38.478**	37.800	36.079	37.989	38.233	38.032	36.403	37.218
Img8	4	13.354	13.363	13.355	13.364	13.353	13.345	13.173	**13.368**
	8	20.739	20.758	20.671	**20.795**	20.761	20.743	20.430	20.777
	12	27.560	27.439	27.516	27.608	27.609	**27.628**	26.742	27.600
	16	33.235	33.164	32.689	33.222	33.365	33.365	32.185	**33.424**
	20	38.275	38.226	37.529	38.346	**38.583**	38.577	37.007	38.316
CX1	4	**13.432**	13.120	12.832	13.032	13.393	13.105	12.966	13.072
	8	**20.124**	19.457	19.827	19.844	19.689	19.807	18.982	20.054
	12	**25.786**	24.845	25.105	25.773	24.809	25.755	24.660	25.431
	16	**30.746**	29.994	29.463	30.415	30.123	30.263	28.233	29.861
	20	**35.051**	34.270	34.299	34.294	34.121	34.617	32.375	34.968
CX2	4	**13.342**	13.340	13.339	13.340	13.340	13.340	13.133	13.340
	8	19.981	19.998	19.906	**20.016**	20.001	20.011	19.607	19.973
	12	26.236	26.260	25.946	26.285	26.277	**26.311**	25.393	26.206
	16	31.697	31.764	31.201	31.748	31.606	31.870	30.667	**31.917**
	20	36.706	36.813	36.021	36.958	**37.134**	37.128	35.517	36.863
CX3	4	**12.981**	12.707	12.156	12.157	12.023	12.539	12.107	12.494
	8	**19.351**	18.552	19.089	18.567	19.060	19.021	18.529	19.213
	12	**25.132**	24.635	24.928	24.458	25.066	24.176	23.698	24.169
	16	30.577	**30.602**	29.519	30.036	30.431	29.749	28.909	29.950
	20	**35.484**	34.896	34.305	35.065	34.959	35.192	33.339	34.846
Continued									

Images	D	MSICSA	CSA	VMCSA	IHBO	RUN_OBL	SO_OBL	SCGJO	SMA_MLS
CX4	4	**12.878**	12.621	12.580	12.097	12.675	12.737	12.480	12.066
	8	**20.872**	20.252	20.798	20.452	20.518	20.842	20.229	20.407
	12	**26.597**	25.401	26.340	26.535	26.147	26.479	25.100	26.118
	16	31.970	31.692	31.972	32.609	**32.858**	32.591	31.454	32.117
	20	**36.498**	36.124	36.237	35.074	35.783	35.662	34.842	36.295
Friedman average rank		2.08	5.41	5.18	4.49	4.13	3.87	7.05	3.79
Rank		1	7	6	5	4	3	8	2

Table 16. The mean fitness values of different algorithms using fuzzy entropy. The optimal values highlighted in bold.

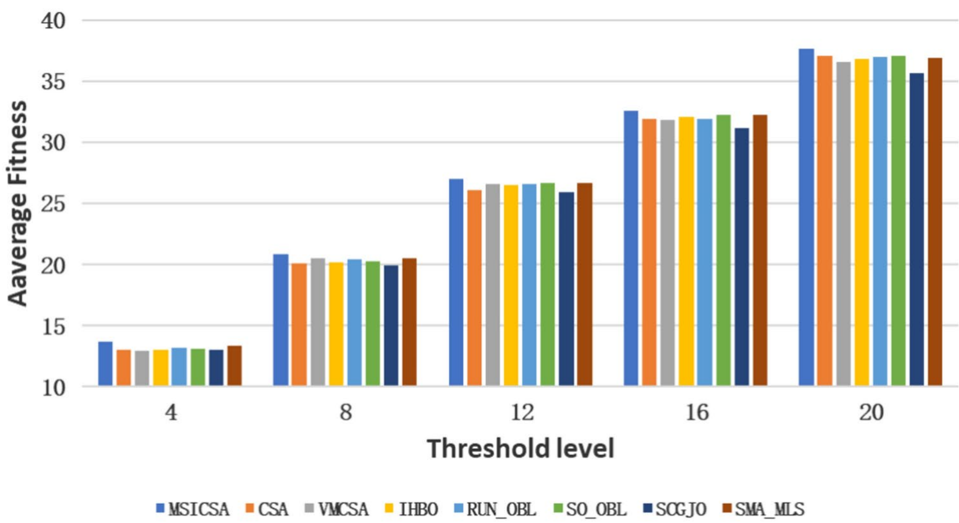


Fig. 18. Mean fitness value at each threshold using fuzzy entropy.

Images	D	MSICSA	CSA	VMCSA	IHBO	RUN_OBL	SO_OBL	SCGJO	SMA_MLS
Img1	4	**5.42 E−15**	5.52 E−15	6.89 E−04	5.48 E−15	4.62 E−04	5.72 E−15	6.03 E−02	4.76 E−04
	8	9.70 E−03	8.95 E−03	3.37 E−02	9.24 E−03	1.03 E−02	**8.09 E−03**	6.43 E−02	1.39 E−02
	12	4.14 E−02	5.48 E−02	7.66 E−02	4.23 E−02	**2.69 E−02**	3.06 E−02	9.11 E−02	7.70 E−02
	16	7.48 E−02	1.37 E−01	1.35 E−01	6.99 E−02	5.27 E−02	**4.98 E−02**	1.37 E−01	9.49 E−02
	20	1.17 E−01	1.33 E−01	1.31 E−01	1.40 E−01	**5.47 E−02**	9.00 E−02	2.09 E−01	1.45 E−01
Img2	4	8.53 E−03	6.14 E−03	3.88 E−03	1.06 E−03	**9.03 E−15**	9.03 E−15	4.07 E−02	8.48 E−05
	8	1.33 E−02	1.83 E−02	5.11 E−02	1.07 E−02	1.36 E−02	**3.84 E−03**	6.52 E−02	1.90 E−02
	12	5.96 E−02	5.05 E−02	7.95 E−02	6.04 E−02	**2.85 E−02**	2.89 E−02	1.13 E−01	6.01 E−02
	16	9.23 E−02	9.78 E−02	1.18 E−01	1.23 E−01	6.74 E−02	**6.12 E−02**	1.44 E−01	1.52 E−01
	20	1.23 E−01	1.21 E−01	1.40 E−01	1.64 E−01	**1.01 E−01**	1.10 E−01	1.72 E−01	1.84 E−01
Img3	4	**1.08 E−14**	1.13 E−14	1.13 E−03	1.18 E−04	2.63 E−04	1.08 E−14	7.24 E−02	1.25 E−14
	8	5.52 E−03	9.14 E−03	3.96 E−02	1.13 E−02	8.80 E−03	**3.55 E−03**	9.84 E−02	2.09 E−02
	12	4.03 E−02	8.30 E−02	1.00 E−01	5.14 E−02	2.76 E−02	**2.46 E−02**	9.16 E−02	5.12 E−02
	16	8.93 E−02	1.18 E−01	1.23 E−01	9.31 E−02	6.54 E−02	**4.62 E−02**	1.27 E−01	8.43 E−02
	20	1.03 E−01	1.21 E−01	1.94 E−01	1.90 E−01	**7.09 E−02**	7.90 E−02	1.65 E−01	1.45 E−01
Img4	4	**3.61 E−15**	3.65 E−15	6.04 E−04	3.67 E−15	3.72 E−15	3.63 E−15	4.82 E−02	2.71 E−04
	8	1.60 E−02	9.59 E−03	3.03 E−02	1.32 E−02	**5.40 E−03**	8.41 E−03	8.19 E−02	2.32 E−02
	12	4.11 E−02	5.96 E−02	7.35 E−02	3.27 E−02	**2.24 E−02**	2.78 E−02	1.28 E−01	4.67 E−02
	16	6.26 E−02	9.11 E−02	1.08 E−01	1.21 E−01	**5.78 E−02**	6.46 E−02	1.37 E−01	1.04 E−01
	20	1.07 E−01	1.16 E−01	1.58 E−01	1.66 E−01	**9.46 E−02**	1.14 E−01	1.48 E−01	1.45 E−01
Img5	4	**9.03 E−15**	9.05 E−15	9.67 E−04	9.03 E−15	9.08 E−15	9.09 E−15	7.39 E−02	4.47 E−04
	8	2.96 E−03	5.55 E−03	2.78 E−02	7.27 E−03	6.14 E−03	**2.62 E−03**	1.12 E−01	1.07 E−02
	12	2.26 E−02	8.35 E−02	8.02 E−02	4.24 E−02	2.11 E−02	**1.45 E−02**	1.22 E−01	4.57 E−02
	16	9.03 E−02	1.08 E−01	1.43 E−01	7.65 E−02	4.56 E−02	**3.59 E−02**	1.30 E−01	8.24 E−02
	20	1.21 E−01	1.13 E−01	1.33 E−01	1.38 E−01	6.63 E−02	**6.04 E−02**	1.74 E−01	1.24 E−01
Img6	4	**3.61 E−15**	3.67 E−15	1.53 E−03	3.64 E−15	4.62 E−05	3.69 E−15	7.90 E−02	4.09 E−05
	8	1.14 E−02	4.18 E−03	3.95 E−02	7.11 E−03	3.87 E−03	**7.23 E−04**	9.83 E−02	1.61 E−02
	12	4.05 E−02	5.46 E−02	9.40 E−02	4.48 E−02	**2.94 E−02**	3.24 E−02	1.52 E−01	5.05 E−02
	16	**7.58 E−02**	7.87 E−02	1.18 E−01	1.14 E−01	9.24 E−02	8.85 E−02	1.64 E−01	1.01 E−01
	20	1.30 E−01	**1.17 E−01**	1.74 E−01	2.95 E−01	1.41 E−01	1.24 E−01	2.00 E−01	1.86 E−01
Img7	4	**5.42 E−15**	5.47 E−15	9.72 E−04	5.42 E−15	5.45 E−15	5.53 E−15	6.98 E−02	9.48 E−05
	8	1.17 E−02	4.46 E−02	4.57 E−02	2.30 E−02	**5.20 E−03**	1.00 E−02	1.07 E−01	1.64 E−02
	12	6.89 E−02	8.24 E−02	8.44 E−02	6.63 E−02	**4.65 E−02**	6.51 E−02	9.33 E−02	7.53 E−02
	16	7.94 E−02	6.94 E−02	1.09 E−01	7.51 E−02	3.96 E−02	**2.83 E−02**	1.39 E−01	1.30 E−01
	20	1.18 E−01	8.63 E−02	1.54 E−01	2.04 E−01	**6.91 E−02**	1.16 E−01	1.87 E−01	1.98 E−01
Img8	4	1.66 E−02	2.22 E−02	1.45 E−02	1.86 E−02	1.15 E−02	**1.51 E−03**	6.26 E−02	2.05 E−02
	8	**2.79 E−02**	5.43 E−02	2.92 E−02	5.30 E−02	5.87 E−02	3.80 E−02	7.99 E−02	7.08 E−02
	12	4.01 E−02	1.06 E−01	1.27 E−01	2.06 E−02	3.64 E−02	**1.40 E−02**	1.45 E−01	3.11 E−02
	16	1.37 E−01	2.02 E−01	1.62 E−01	**1.06 E−01**	1.19 E−01	1.22 E−01	1.72 E−01	1.10 E−01
	20	1.68 E−01	1.66 E−01	2.05 E−01	1.89 E−01	**9.01 E−02**	9.06 E−02	2.37 E−01	2.10 E−01
CX1	4	**5.42 E−15**	5.48 E−15	2.25 E−03	5.45 E−15	1.38 E−03	5.48 E−15	9.41 E−02	2.17 E−05
	8	1.51 E−02	**4.19 E−03**	4.37 E−02	1.36 E−02	1.13 E−02	1.77 E−02	2.18 E−01	2.66 E−02
	12	4.93 E−02	4.88 E−02	9.71 E−02	4.96 E−02	1.03 E−01	**4.08 E−02**	2.76 E−01	5.95 E−02
	16	9.23 E−02	**8.96 E−02**	1.87 E−01	1.53 E−01	2.97 E−01	3.29 E−01	5.06 E−01	1.60 E−01
	20	**1.30 E−01**	1.63 E−01	1.92 E−01	4.19 E−01	1.60 E−01	5.12 E−01	6.39 E−01	2.29 E−01
CX2	4	**1.81 E−15**	1.84 E−15	1.15 E−03	1.89 E−03	1.88 E−15	1.87 E−15	7.48 E−02	1.84 E−15
	8	1.45 E−02	**1.27 E−02**	4.02 E−02	2.36 E−02	1.43 E−02	2.30 E−02	1.04 E−01	2.68 E−02
	12	6.04 E−02	1.07 E−01	9.22 E−02	4.35 E−02	6.36 E−02	**3.66 E−02**	1.54 E−01	1.19 E−01
	16	1.15 E−01	9.42 E−02	1.57 E−01	1.48 E−01	8.13 E−02	**4.83 E−02**	2.27 E−01	1.36 E−01
	20	1.68 E−01	1.61 E−01	2.11 E−01	1.53 E−01	1.37 E−01	**6.76 E−02**	2.41 E−01	1.92 E−01
CX3	4	**0.00 E+00**	4.49 E−04	2.45 E−03	1.69 E−05	6.65 E−07	6.52 E−10	5.96 E−02	5.23 E−05
	8	1.07 E−02	**3.78 E−03**	2.98 E−02	1.78 E−02	2.57 E−02	1.42 E−02	1.08 E−01	2.66 E−02
	12	3.47 E−02	**3.27 E−02**	6.83 E−02	4.50 E−02	5.42 E−02	3.76 E−02	1.76 E−01	5.79 E−02
	16	6.04 E−02	**5.77 E−02**	1.20 E−01	1.61 E−01	7.15 E−02	7.47 E−02	2.23 E−01	9.73 E−02
	20	9.13 E−02	**8.20 E−02**	1.38 E−01	2.00 E−01	9.84 E−02	1.55 E−01	2.42 E−01	1.69 E−01

Continued

Images	D	MSICSA	CSA	VMCSA	IHBO	RUN_OBL	SO_OBL	SCGJO	SMA_MLS
CX4	4	**7.23 E−15**	5.98 E−03	2.46 E−03	2.80 E−04	7.63 E−15	8.30 E−03	3.30 E−02	1.79 E−04
	8	7.89 E−03	1.12 E−02	2.86 E−02	**4.73 E−03**	1.94 E−02	5.18 E−03	5.63 E−02	1.54 E−02
	12	2.78 E−02	3.65 E−02	8.17 E−02	3.34 E−02	2.88 E−02	**1.56 E−02**	1.55 E−01	3.70 E−02
	16	8.15 E−02	1.00 E−01	1.26 E−01	4.12 E−01	**7.70 E−02**	3.98 E−01	6.16 E−01	8.66 E−02
	20	3.78 E−01	4.83 E−01	4.15 E−01	**1.31 E−01**	5.11 E−01	1.61 E−01	5.22 E−01	5.02 E−01
Friedman average rank		3.03	3.92	6.22	4.38	2.84	2.51	7.78	5.33
Rank		3	4	7	5	2	1	8	6

Table 17. STD values of different algorithms using fuzzy entropy. The optimal values highlighted in bold.

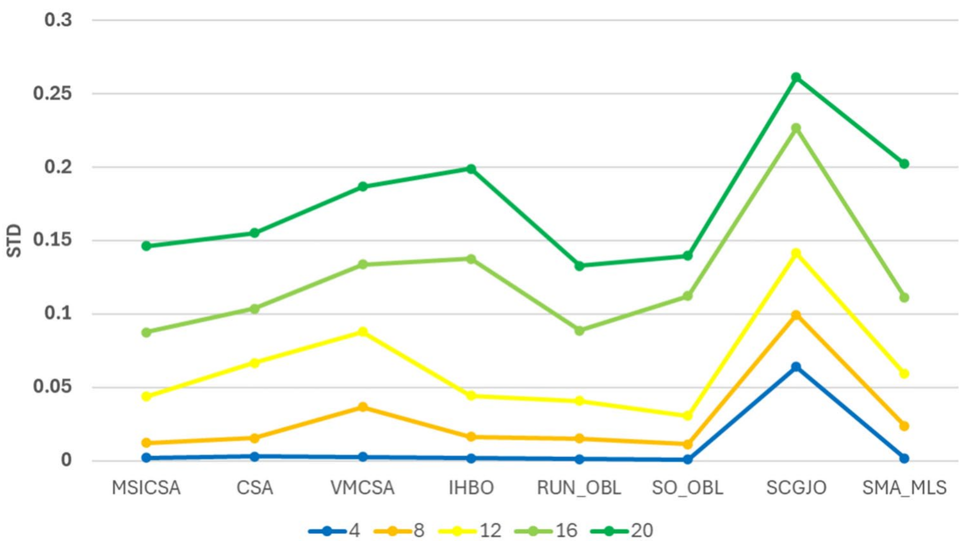


Fig. 19. Average STD value at each threshold using fuzzy entropy.

Images	D	MSICSA	CSA	VMCSA	IHBO	RUN_OBL	SO_OBL	SCGJO	SMA_MLS
Img1	4	**12.681**	10.701	10.698	10.701	10.703	10.701	10.701	11.705
	8	**16.884**	14.365	14.247	14.705	14.795	14.629	14.509	14.436
	12	**20.779**	18.178	18.577	18.700	18.451	18.430	17.972	18.127
	16	**23.482**	20.301	21.351	21.143	21.506	21.934	20.277	21.387
	20	**25.617**	23.760	24.720	23.864	24.172	23.838	23.457	24.291
Img2	4	**16.225**	15.113	15.732	15.874	15.356	15.472	15.855	15.483
	8	**17.391**	16.275	16.402	16.437	16.429	16.429	16.121	16.632
	12	20.559	19.480	20.573	20.756	20.262	**20.804**	19.976	19.949
	16	**23.912**	23.053	23.541	22.545	23.171	22.582	21.897	23.271
	20	**25.955**	25.131	25.441	25.094	25.355	24.545	25.104	25.447
Img3	4	**13.044**	11.860	11.861	11.865	11.841	11.960	11.830	11.878
	8	**15.959**	14.537	14.832	14.740	14.580	14.534	14.450	14.444
	12	**20.510**	19.163	19.376	19.562	19.533	19.536	19.188	18.964
	16	**23.016**	22.263	22.602	22.356	22.265	22.831	22.509	22.314
	20	**25.789**	24.610	24.925	24.575	24.802	24.606	24.028	24.517
Img4	4	**14.737**	13.958	13.831	13.982	13.958	13.981	13.973	13.979
	8	**16.701**	15.246	15.576	15.272	15.212	15.292	15.564	15.431
	12	**20.921**	18.095	18.999	18.542	18.321	18.397	18.293	18.317
	16	**22.735**	21.449	20.411	19.986	22.021	21.823	21.535	21.718
	20	24.212	23.454	23.750	24.243	23.537	24.212	23.578	**24.543**
Img5	4	**11.721**	10.909	10.912	10.973	10.924	10.956	10.964	11.313
	8	**15.496**	14.676	14.785	14.832	14.792	14.819	14.754	14.817
	12	**19.692**	18.308	18.217	18.109	18.260	18.893	18.653	18.906
	16	**21.579**	20.421	20.811	20.557	20.497	20.693	20.177	20.155
	20	**24.512**	23.358	23.356	23.306	22.580	22.738	22.676	22.356
Img6	4	**11.733**	11.293	11.280	11.295	11.289	11.297	11.299	11.364
	8	**16.536**	14.237	14.871	14.243	14.201	14.235	14.425	14.489
	12	**19.382**	17.939	18.098	18.585	18.168	18.817	18.738	18.332
	16	22.665	21.961	22.088	22.994	21.449	22.554	21.675	**23.069**
	20	24.624	24.309	25.676	**25.946**	24.328	25.101	24.101	24.736
Img7	4	**13.217**	12.604	12.614	12.604	12.604	12.604	12.603	12.610
	8	**15.137**	14.650	14.135	14.858	14.778	14.709	14.882	14.811
	12	**20.124**	18.697	18.590	17.484	18.915	17.669	18.001	19.101
	16	23.532	23.641	**24.077**	23.449	23.423	23.304	23.576	23.853
	20	**26.482**	24.920	26.440	25.144	25.341	24.449	25.086	25.317
Img8	4	11.638	10.531	11.217	10.634	11.373	13.993	13.499	**13.856**
	8	16.195	15.072	16.909	15.991	15.569	15.751	18.277	**18.344**
	12	18.161	18.105	18.532	17.880	18.064	17.907	**21.243**	20.899
	16	20.486	21.380	21.837	20.520	21.210	20.493	**23.306**	22.516
	20	23.499	23.714	24.428	23.144	23.625	22.919	**24.895**	23.755
CX1	4	**13.959**	11.542	11.504	11.542	11.466	11.547	11.545	12.437
	8	**19.213**	16.870	17.729	17.191	17.107	17.184	16.993	17.576
	12	**22.782**	21.529	21.690	20.968	22.394	21.387	21.790	21.338
	16	**25.811**	25.251	24.951	24.921	25.247	24.733	25.314	24.245
	20	27.208	27.464	**28.368**	27.842	27.657	27.175	28.310	26.406
CX2	4	12.944	12.944	12.935	12.958	12.944	12.947	**14.544**	13.926
	8	**20.838**	19.656	19.922	18.278	20.350	19.525	18.496	19.379
	12	21.510	21.923	21.830	21.434	21.877	21.316	21.696	**21.973**
	16	24.610	23.374	24.654	23.634	23.828	23.726	**25.057**	23.655
	20	26.204	26.026	27.024	25.388	26.264	25.598	26.925	**27.110**
CX3	4	**15.450**	15.160	15.279	15.159	15.160	15.160	15.160	15.163
	8	**19.967**	19.255	18.489	19.060	19.116	19.202	19.135	19.565
	12	**23.427**	20.552	20.983	21.268	20.014	21.148	21.003	22.575
	16	**26.121**	23.289	23.284	24.348	22.336	23.831	23.105	24.587
	20	**28.115**	25.158	26.074	27.281	24.672	26.918	25.588	26.117

Continued

Images	D	MSICSA	CSA	VMCSA	IHBO	RUN_OBL	SO_OBL	SCGJO	SMA_MLS
CX4	4	**14.530**	13.661	13.625	13.622	13.606	13.716	13.606	13.994
	8	17.692	17.195	16.579	16.952	16.070	17.145	17.909	**17.926**
	12	**19.862**	18.391	17.108	19.614	17.375	17.977	18.721	19.404
	16	**21.573**	20.090	19.287	20.329	18.139	20.780	19.219	20.257
	20	**23.389**	20.952	22.971	22.348	22.264	22.234	22.335	22.608
Friedman average rank		1.99	5.91	4.37	4.76	5.41	4.79	5.01	3.77
Rank		1	8	3	4	7	5	6	2

Table 18. PSNR values of different algorithms using fuzzy entropy. The optimal values highlighted in bold.

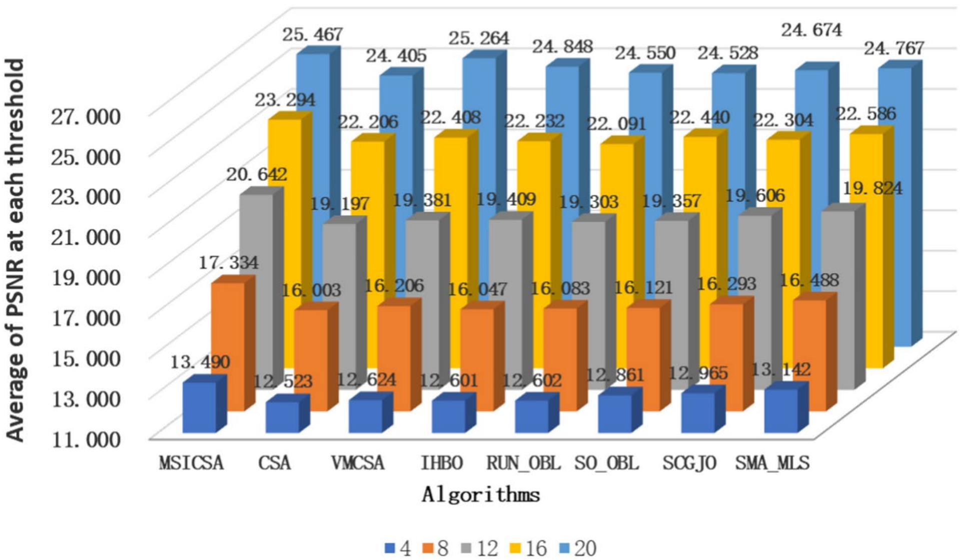


Fig. 20. Average PSNR at each threshold level using fuzzy entropy.

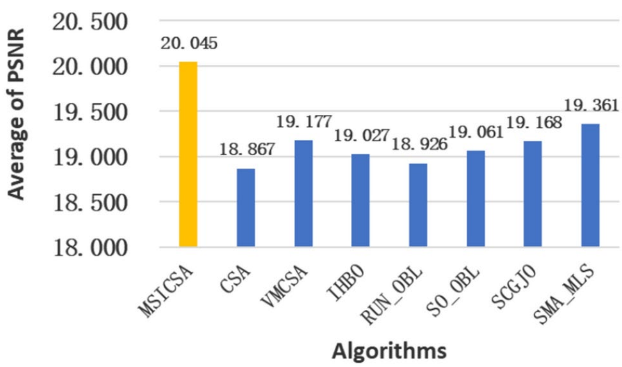


Fig. 21. Average PSNR at all threshold levels using fuzzy entropy.

Images	D	MSICSA	CSA	VMCSA	IHBO	RUN_OBL	SO_OBL	SCGJO	SMA_MLS
Img1	4	**0.5838**	0.4243	0.4429	0.4271	0.4177	0.4243	0.4243	0.4543
	8	**0.6949**	0.5649	0.5678	0.5646	0.5675	0.5645	0.5593	0.5704
	12	**0.7703**	0.6983	0.7204	0.7107	0.7147	0.7076	0.6938	0.6923
	16	**0.8209**	0.7747	0.7992	0.7616	0.7747	0.7555	0.7524	0.7510
	20	**0.8549**	0.8331	0.8176	0.8183	0.8170	0.8193	0.8079	0.8121
Img2	4	0.4564	0.4644	0.4715	0.4725	0.4723	0.4723	0.4445	**0.4731**
	8	0.2944	0.2913	0.3300	0.3195	0.3171	0.3005	**0.4840**	0.3245
	12	0.5812	0.6042	0.6116	**0.6289**	0.6011	0.6172	0.5892	0.5975
	16	0.6959	0.7031	0.7062	0.6970	0.7057	0.6565	0.6862	**0.7174**
	20	0.7536	0.7467	0.7608	0.7655	**0.7838**	0.7555	0.7413	0.7490
Img3	4	**0.3876**	0.3259	0.3259	0.3262	0.3180	0.3259	0.3259	0.3259
	8	**0.5849**	0.5169	0.5164	0.5242	0.5176	0.5167	0.5110	0.5658
	12	**0.6696**	0.6114	0.6096	0.6328	0.6241	0.6258	0.6127	0.6491
	16	**0.7469**	0.7264	0.7371	0.7227	0.7199	0.7031	0.6999	0.7429
	20	**0.8143**	0.7609	0.7918	0.7836	0.7949	0.7792	0.7787	0.8052
Img4	4	**0.5194**	0.3713	0.3713	0.3713	0.3713	0.3713	0.3713	0.4503
	8	**0.6774**	0.4829	0.4898	0.4860	0.4817	0.4854	0.5081	0.5449
	12	**0.7346**	0.6155	0.5761	0.5768	0.5687	0.5691	0.5772	0.6764
	16	**0.7692**	0.7386	0.6570	0.7466	0.6751	0.6664	0.6978	0.7514
	20	0.7918	0.7617	0.7473	**0.7869**	0.7813	0.7988	0.7798	**0.7971**
Img5	4	**0.3680**	0.2745	0.2984	0.2835	0.3120	0.3320	0.3413	0.3478
	8	**0.5642**	0.4839	0.4846	0.4822	0.4772	0.4763	0.4796	0.4990
	12	**0.7711**	0.6664	0.6754	0.6843	0.6397	0.6846	0.6851	0.6934
	16	**0.8176**	0.7556	0.7218	0.7282	0.7248	0.7331	0.7210	0.7246
	20	**0.8590**	0.8251	0.7831	0.8108	0.7889	0.7951	0.8067	0.7967
Img6	4	**0.1763**	0.1263	0.1263	0.1263	0.1253	0.1263	0.1263	0.1264
	8	**0.5012**	0.4521	0.4626	0.4515	0.4505	0.4511	0.4693	0.4795
	12	**0.6834**	0.6453	0.6164	0.6648	0.6495	0.6809	0.6474	0.6602
	16	**0.8169**	0.7918	0.7378	0.8164	0.7598	0.8055	0.7783	0.7916
	20	0.8318	0.8563	0.8239	0.8616	0.8326	0.8613	0.8326	**0.8789**
Img7	4	**0.4811**	0.4343	0.4343	0.4353	0.4343	0.4343	0.4342	0.4357
	8	**0.6017**	0.5875	0.5944	0.5772	0.5707	0.5663	0.5936	0.5815
	12	**0.7721**	0.6989	0.6813	0.7128	0.6750	0.6775	0.6795	0.7043
	16	0.8451	**0.8592**	0.8496	0.8516	0.8476	0.8498	0.8532	0.8517
	20	**0.8897**	0.8797	0.8750	0.8813	0.8820	0.8828	0.8817	0.8746
Img8	4	0.4582	0.4495	0.4569	0.4551	0.4541	0.4579	**0.5178**	**0.5465**
	8	0.4775	0.4793	0.5053	0.4914	0.5003	0.4877	0.5442	0.5717
	12	0.5998	0.5825	0.6175	0.5653	0.5754	0.5633	**0.6782**	0.6565
	16	0.7156	0.7102	0.7173	0.6539	0.6780	0.6397	0.7187	0.7325
	20	0.7521	0.7342	**0.7807**	0.7466	0.7224	0.7631	0.7708	0.7731
CX1	4	**0.4520**	0.3750	0.3911	0.3886	0.4169	0.4263	0.3804	0.4214
	8	**0.6906**	0.5364	0.5862	0.5954	0.5489	0.5843	0.5446	0.5961
	12	**0.8118**	0.7448	0.7148	0.7567	0.7851	0.7564	0.6807	0.7273
	16	**0.8777**	0.8121	0.8326	0.8469	0.8551	0.8452	0.8045	0.8252
	20	0.9357	0.9225	0.9271	0.9284	**0.9429**	0.9161	0.9135	0.8986
CX2	4	**0.5677**	0.5675	0.5674	0.5675	0.5675	0.5675	0.5619	0.5675
	8	**0.7889**	0.7653	0.7552	0.7063	0.7762	0.7502	0.7009	0.7448
	12	0.7896	**0.8074**	0.7983	0.7811	0.8069	0.7801	0.7937	0.7828
	16	0.8767	0.8447	0.8687	0.8515	0.8635	0.8608	**0.8847**	0.8771
	20	0.9037	0.9080	0.9147	0.8911	**0.9198**	0.9040	0.9140	0.8881
CX3	4	**0.5612**	0.5335	0.5368	0.5463	0.5481	0.5421	0.5317	0.5565
	8	**0.6937**	0.6103	0.6502	0.6765	0.6658	0.6388	0.6166	0.6810
	12	0.7588	**0.7639**	0.7455	0.7588	0.7603	0.7608	0.7103	0.7468
	16	**0.7787**	0.7646	0.7696	0.7647	0.7781	0.7672	0.7734	0.7678
	20	**0.8116**	0.7809	0.8077	0.7726	0.7166	0.7259	0.8074	0.7976

Continued

Images	D	MSICSA	CSA	VMCSA	IHBO	RUN_OBL	SO_OBL	SCGJO	SMA_MLS
CX4	4	**0.8455**	0.7415	0.8397	0.7946	0.7353	0.8250	0.8441	0.8144
	8	**0.6382**	0.6355	0.6373	0.6362	0.6362	0.6348	0.6184	0.6361
	12	0.7553	0.7154	0.7153	0.7380	0.6780	0.7394	0.7018	0.7222
	16	**0.7820**	0.7588	0.7365	0.7544	0.6956	0.7468	0.7231	0.7402
	20	**0.8424**	0.7775	0.8243	0.7971	0.8115	0.8145	0.7813	0.8198
Friedman average rank		1.99	5.91	4.37	4.76	5.41	4.79	5.01	3.77
Rank		1	8	3	4	6	5	7	2

Table 19. SSIM values of different algorithms using fuzzy entropy. The optimal values highlighted in bold.

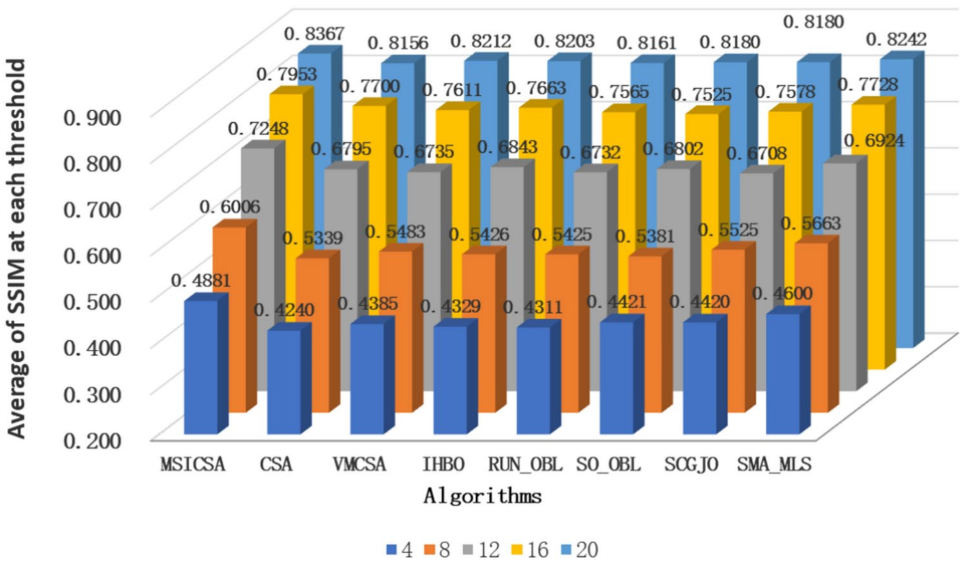


Fig. 22. Average SSIM at each threshold level using fuzzy entropy.

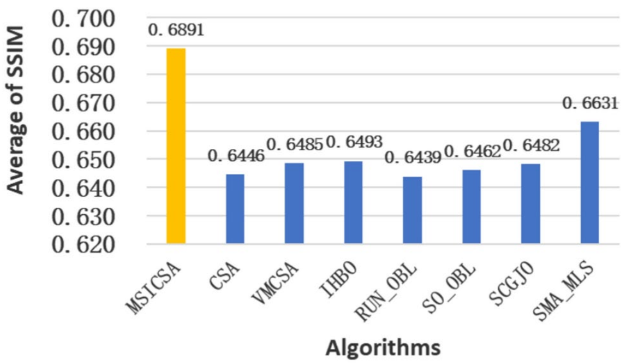


Fig. 23. Average SSIM at all threshold levels using fuzzy entropy.

Images	D	MSICSA	CSA	VMCSA	IHBO	RUN_OBL	SO_OBL	SCGJO	SMA_MLS
Img1	4	**0.7257**	0.6685	0.7046	0.7045	0.7078	0.7155	0.7085	0.7183
	8	**0.8012**	0.7381	0.7536	0.7681	0.7689	0.7683	0.7558	0.7697
	12	**0.8572**	0.8031	0.8072	0.8176	0.8229	0.8155	0.8028	0.8228
	16	**0.8941**	0.8564	0.8573	0.8549	0.8663	0.8506	0.8446	0.8628
	20	**0.9181**	0.8946	0.9014	0.8931	0.9005	0.8940	0.8833	0.9034
Img2	4	0.6918	0.6952	0.6982	0.6988	0.6987	0.6987	0.6880	**0.6991**
	8	0.6403	0.6398	0.6549	0.6586	0.6543	0.6477	**0.7377**	0.6466
	12	0.8044	0.7939	0.8032	0.8124	0.8025	**0.8141**	0.8099	0.8035
	16	0.8623	0.8586	0.8647	0.8434	0.8569	0.8261	**0.8722**	0.8695
	20	0.8986	0.8834	0.8935	0.8903	0.8937	0.8836	**0.9078**	0.8969
Img3	4	**0.5939**	0.5458	0.5660	0.5648	0.5557	0.5651	0.5689	0.5758
	8	**0.7355**	0.6785	0.6817	0.6785	0.6766	0.6778	0.6716	0.6918
	12	**0.7935**	0.7667	0.7649	0.7760	0.7793	0.7786	0.7677	0.7798
	16	**0.8429**	0.8221	0.8369	0.8341	0.8394	0.8258	0.8164	0.8322
	20	**0.8925**	0.8721	0.8766	0.8739	0.8832	0.8730	0.8624	0.8797
Img4	4	**0.8105**	0.7220	0.7188	0.7220	0.7220	0.7220	0.7220	0.7266
	8	**0.8620**	0.8282	0.8349	0.8297	0.8259	0.8294	0.8427	0.8397
	12	**0.8954**	0.8783	0.8855	0.8691	0.8656	0.8685	0.8690	0.8856
	16	**0.9220**	0.9080	0.9175	0.9013	0.9031	0.9003	0.9096	0.9164
	20	0.9369	0.9217	0.9369	0.9370	0.9367	**0.9384**	0.9370	0.9300
Img5	4	**0.6533**	0.5883	0.5880	0.5883	0.5883	0.5883	0.5883	0.5882
	8	**0.7737**	0.7166	0.7166	0.7176	0.7162	0.7158	0.7140	0.7190
	12	**0.8400**	0.8114	0.8052	0.8099	0.8116	0.8049	0.7960	0.7974
	16	**0.8783**	0.8487	0.8586	0.8528	0.8546	0.8587	0.8478	0.8454
	20	**0.9074**	0.8983	0.8960	0.8947	0.8865	0.8887	0.8870	0.8797
Img6	4	**0.6786**	0.6680	0.6677	0.6680	0.6679	0.6680	0.6680	0.6683
	8	**0.7772**	0.7666	0.7676	0.7660	0.7658	0.7672	0.7685	0.7673
	12	**0.8348**	0.7970	0.8066	0.8020	0.7997	0.8070	0.8007	0.8187
	16	**0.8575**	0.8447	0.8567	0.8540	0.8424	0.8523	0.8492	0.8528
	20	0.8798	0.8645	0.8874	0.8854	0.8746	0.8823	0.8774	**0.8886**
Img7	4	**0.7649**	0.7516	0.7509	0.7516	0.7516	0.7516	0.7516	0.7517
	8	**0.8032**	0.7801	0.7808	0.7810	0.7809	0.7796	0.7803	0.7885
	12	**0.8477**	0.8341	0.8279	0.8121	0.8209	0.8215	0.8267	0.8256
	16	0.8558	0.8524	0.8585	**0.8635**	0.8594	0.8634	0.8576	0.8560
	20	0.8732	0.8633	**0.8767**	0.8742	0.8731	0.8724	0.8752	0.8680
Img8	4	0.6946	0.6754	0.6900	0.6976	0.6636	0.6831	0.6885	**0.7053**
	8	0.7370	0.7346	0.7483	0.7352	0.7392	0.7358	0.7473	**0.7643**
	12	0.7951	0.7953	0.8032	0.7946	0.8007	0.7944	**0.8173**	0.7976
	16	0.8284	0.8376	0.8476	0.8359	**0.8479**	0.8354	0.8368	0.8439
	20	0.8680	0.8719	0.8774	0.8709	0.8721	0.8652	**0.8796**	0.8654
CX1	4	**0.6359**	0.5864	0.5858	0.5864	0.5854	0.5864	0.5864	0.5863
	8	**0.7495**	0.6840	0.7121	0.7012	0.6924	0.6977	0.6900	0.7129
	12	0.8459	0.8358	0.8278	0.8186	**0.8463**	0.8338	0.8354	0.8218
	16	**0.9141**	0.8953	0.8911	0.8989	0.8979	0.8944	0.9027	0.9071
	20	**0.9451**	0.9341	0.9385	0.9380	0.9442	0.9302	0.9293	0.9142
CX2	4	0.6643	0.6643	0.6645	0.6643	0.6643	0.6643	0.6524	**0.6666**
	8	**0.7969**	0.7760	0.7678	0.7220	0.7837	0.7586	0.7441	0.7595
	12	0.7992	0.7920	0.8054	0.7936	**0.8130**	0.7927	0.8088	0.7957
	16	0.8792	0.8575	0.8707	0.8559	0.8691	0.8649	**0.8858**	0.8723
	20	0.9079	0.9028	0.9184	0.8970	0.9045	0.9098	0.9167	**0.9227**
CX3	4	**0.7412**	0.7215	0.7133	0.7291	0.7378	0.7210	0.6946	0.7134
	8	0.7772	**0.7805**	0.7710	0.7775	0.7771	0.7784	0.7514	0.7705
	12	**0.7922**	0.7789	0.7833	0.7782	0.7873	0.7794	0.7872	0.7888
	16	**0.8297**	0.8140	0.8257	0.8082	0.8056	0.8035	0.8196	0.8201
	20	**0.8532**	0.8369	0.8449	0.8369	0.8324	0.8332	0.8434	0.8515
Continued									

Images	D	MSICSA	CSA	VMCSA	IHBO	RUN_OBL	SO_OBL	SCGJO	SMA_MLS
CX4	4	**0.7698**	0.7226	0.7218	0.7198	0.7198	0.7254	0.7584	0.7499
	8	**0.7987**	0.7967	0.7798	0.7980	0.7605	0.7966	0.7397	0.7938
	12	0.8273	0.8216	0.8265	0.8367	0.8136	0.8319	0.8369	**0.8365**
	16	**0.8868**	0.8761	0.8679	0.8636	0.8312	0.8675	0.8664	0.8473
	20	**0.9173**	0.9000	0.9100	0.8930	0.8835	0.8934	0.9066	0.9147
Friedman average rank		2.31	5.69	4.27	4.92	5.03	5.26	4.81	3.72
Rank		1	8	3	5	6	7	4	2

Table 20. The FSIM values of different algorithms using fuzzy entropy. The optimal values highlighted in bold.

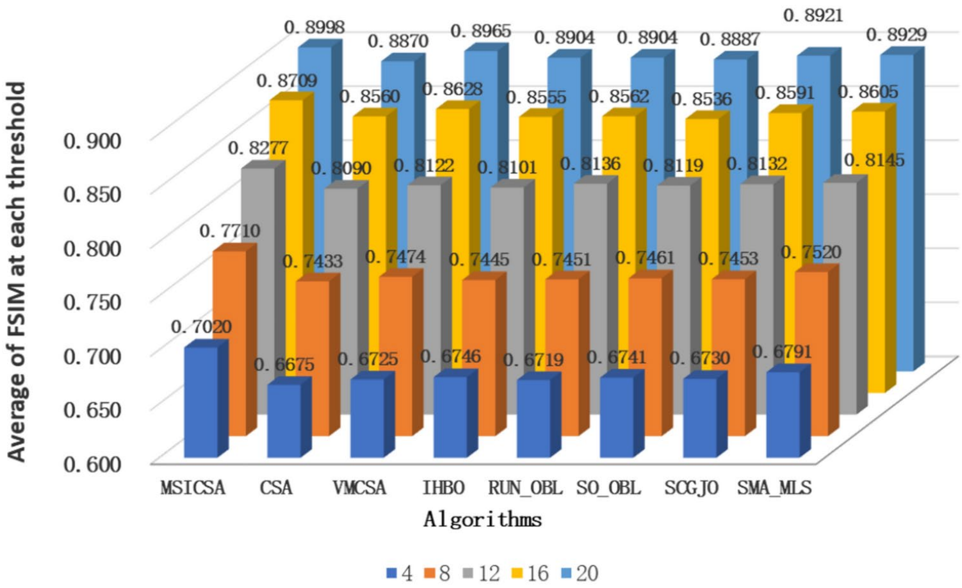


Fig. 24. Average FSIM at each threshold level using fuzzy entropy.

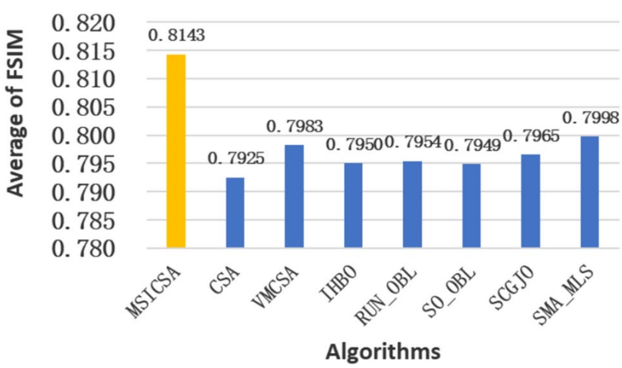


Fig. 25. Average FSIM at all threshold levels using fuzzy entropy.

Images	D	MSICSA	CSA	VMCSA	IHBO	RUN_OBL	SO_OBL	SCGJO	SMA_MLS
Img1	4	3 133 135 253	3 133 136 252	2 131 135 253	3 134 137 252	4 131 135 253	5 133 135 252	28 150 157 252	3 131 135 252
	8	3 93 96 133 136 176 180 252	5 91 96 132 137 173 177 249	6 91 97 130 136 171 177 250	5 91 98 131 135 171 176 249	5 91 96 130 135 170 176 247	8 90 96 131 135 172 180 251	36 119 121 138 148 189 191 250	6 90 96 131 135 172 180 251
	12	8 54 60 72 79 130 139 144 146 195 199 252	9 54 58 72 75 130 135 143 148 191 195 251	7 61 66 72 78 120 124 151 154 191 195 250	9 51 55 72 79 132 136 144 149 191 195 251	6 51 55 71 75 123 130 141 146 191 196 252	9 52 58 72 73 130 135 144 148 192 197 251	16 91 97 131 136 147 156 195 203 206 209 252	8 60 65 70 71 130 134 145 146 191 192 251
	16	3 40 49 63 74 103 106 135 141 146 154 192 199 200 208 253	4 50 54 62 66 101 105 124 130 149 154 191 195 201 206 252	3 52 59 64 68 105 109 135 140 143 147 191 195 201 205 252	3 43 48 71 73 101 105 141 145 154 159 196 200 210 214 252	10 42 47 54 60 101 105 123 128 150 153 190 195 205 210 253	3 50 53 55 59 94 99 120 126 150 153 200 204 209 212 251	11 52 55 70 74 115 125 137 153 190 194 200 206 220 224 252	3 42 46 61 65 93 97 131 135 140 143 191 195 200 205 252
	20	3 26 34 55 64 75 78 106 112 119 124 147 154 167 175 190 195 200 207 251	2 41 45 52 56 85 88 100 104 131 135 146 149 172 177 200 204 222 225 252	3 22 25 53 57 74 77 81 85 113 116 135 138 150 154 195 199 204 213 253	2 30 34 50 51 74 71 105 108 125 129 151 156 170 176 200 203 220 225 253	2 35 39 50 53 75 78 101 105 125 129 146 150 170 175 201 205 211 216 253	2 35 39 51 55 86 90 111 115 125 128 150 154 162 166 200 204 211 215 252	11 62 65 85 91 101 106 130 134 146 153 162 167 205 210 214 225 231 235 253	3 39 42 46 50 83 86 95 99 120 125 144 148 172 176 204 208 213 218 252
Img2	4	64 120 128 251	66 124 132 248	65 122 128 252	64 119 128 250	63 118 128 250	63 122 128 250	72 150 157 251	66 120 129 250
	8	6 114 118 123 129 194 202 248	5 113 118 124 128 193 201 248	5 114 118 123 128 193 201 247	5 112 116 122 128 193 203 249	6 115 118 121 128 196 201 247	5 114 118 123 128 193 201 247	60 126 130 138 141 204 205 251	4 114 119 123 129 190 200 252
	12	9 100 104 107 112 157 161 164 169 211 216 251	8 100 105 108 114 162 167 102 173 212 217 252	1 61 65 76 79 129 133 166 169 213 217 248	10 61 66 91 95 130 133 163 169 193 199 252	11 92 100 106 110 151 154 164 170 210 213 251	6 60 65 90 94 131 135 165 168 211 217 251	38 93 97 101 105 146 150 174 179 213 216 251	7 100 104 110 115 160 166 173 175 211 216 253
	16	2 62 65 68 73 114 119 121 126 161 166 173 177 212 215 252	2 44 48 60 65 100 104 111 115 152 156 171 175 210 215 252	3 61 64 69 71 105 109 124 128 142 146 163 167 200 205 245	4 56 60 64 67 109 113 123 127 164 168 177 180 210 213 251	3 58 63 68 72 110 115 125 129 166 169 184 185 210 214 252	4 60 63 66 72 110 114 121 126 161 166 181 186 213 217 251	28 70 74 82 86 109 115 126 136 182 188 191 194 220 224 252	2 60 63 70 75 113 120 124 128 164 169 190 194 210 214 253
	20	4 42 45 63 66 91 95 120 125 131 135 170 174 181 184 220 224 226 230 253	8 50 54 61 65 96 97 110 114 151 154 171 175 187 190 216 220 225 229 253	4 52 57 62 71 90 93 105 108 126 129 152 155 181 184 206 213 231 235 252	2 53 56 60 63 90 95 114 117 122 127 159 162 172 175 213 217 224 228 253	2 44 48 60 64 94 97 113 116 130 133 155 160 176 180 213 218 222 227 252	3 50 53 56 59 85 88 111 114 135 139 161 165 181 184 214 217 226 230 253	24 52 56 61 64 98 101 114 123 142 146 156 169 183 187 210 214 226 230 252	2 42 46 53 56 83 87 110 113 120 124 161 165 182 185 213 217 224 227 253
Img3	4	6 133 142 250	6 135 142 250	7 131 140 252	6 130 142 250	7 135 143 251	8 135 144 250	41 130 138 250	6 131 142 252
	8	4 104 108 131 137 184 192 250	5 100 108 125 131 184 190 251	8 101 108 121 135 181 194 251	10 104 108 131 135 181 192 251	10 104 108 131 135 181 192 251	3 100 104 131 137 180 186 250	36 98 101 142 153 190 193 251	3 100 104 131 137 180 186 251
	12	12 70 75 95 99 130 135 171 176 191 197 251	8 65 70 94 99 130 134 175 179 190 195 251	9 65 70 92 100 119 123 177 180 184 187 252	7 62 66 91 95 122 127 170 176 190 195 251	9 63 69 94 100 130 133 180 184 189 193 252	12 62 67 92 97 124 129 163 168 191 195 252	44 72 75 110 118 143 147 181 187 211 214 251	10 71 76 96 99 123 128 170 173 184 188 251
	16	4 51 56 59 63 101 105 109 114 150 153 181 184 191 194 251	1 55 56 57 58 104 105 109 110 151 153 181 183 193 194 252	3 55 59 60 65 99 103 114 118 144 148 183 188 191 196 252	2 46 49 71 75 103 107 131 136 161 165 187 193 197 200 252	2 50 55 64 67 100 105 111 116 150 154 183 186 194 199 252	4 51 55 62 66 95 99 121 126 150 154 185 189 200 205 252	3 52 57 60 64 102 107 125 129 162 166 185 190 214 218 251	27 64 68 87 94 108 123 140 150 182 186 210 213 221 224 252
	20	4 48 53 56 59 94 99 101 106 135 139 155 159 175 179 205 211 220 224 253	2 45 50 54 57 94 98 100 105 134 138 151 155 192 196 201 204 222 226 252	2 35 39 60 64 87 90 107 112 126 131 171 175 181 184 213 218 221 225 253	5 42 45 51 55 90 95 101 104 132 135 160 165 180 184 205 209 220 223 253	3 44 49 55 59 91 95 104 108 124 129 151 155 173 178 194 198 220 224 252	4 42 46 51 56 85 90 105 109 132 136 154 158 171 175 202 206 212 218 253	29 64 73 80 90 119 120 126 130 155 158 180 185 195 198 216 217 228 238 253	3 52 56 60 65 102 106 111 116 144 148 163 167 183 186 213 217 214 220 252
Img4	4	6 121 135 158	7 123 134 161	6 120 131 160	7 130 141 159	8 121 131 158	8 123 131 154	32 122 129 228	5 124 131 151
	8	3 80 85 123 129 160 165 252	4 82 87 125 130 160 164 251	3 82 85 126 134 160 164 252	4 84 88 125 130 158 162 251	3 80 84 124 131 160 163 252	5 82 87 125 130 158 163 250	34 111 115 155 172 191 196 252	3 80 85 125 130 160 163 252
	12	6 70 75 91 96 131 137 171 176 193 199 251	8 70 73 90 95 130 136 173 177 191 195 252	7 63 69 113 116 124 128 179 184 194 198 252	12 72 78 84 88 140 144 163 169 202 207 251	8 70 75 91 96 131 136 171 175 19 195 252	11 72 77 92 97 131 135 170 175 190 194 251	29 75 80 120 124 148 156 181 189 213 216 251	12 70 75 92 98 134 139 173 178 203 207 251
	16	3 50 53 56 60 105 109 113 117 152 155 181 185 200 204 252	3 68 71 86 90 123 127 141 145 171 175 192 196 226 229 251	2 62 66 75 80 111 116 140 145 162 166 194 198 205 210 252	3 74 77 95 99 124 127 136 140 186 191 196 200 226 230 253	4 61 65 71 75 112 116 130 134 164 168 191 195 220 223 252	4 60 64 70 74 112 116 131 135 165 169 191 194 214 218 252	5 65 69 76 80 121 124 136 140 170 174 185 191 220 225 251	31 82 87 98 105 132 140 155 158 181 186 204 208 214 217 252
	20	3 51 55 74 77 99 104 116 120 140 144 165 170 181 185 211 215 224 227 253	2 61 65 71 75 100 103 115 119 145 149 169 173 182 186 211 215 230 234 253	3 51 54 67 71 85 90 120 124 150 155 162 165 185 188 200 204 223 226 247	3 50 53 62 65 83 88 106 110 132 136 150 156 173 178 201 205 222 227 252	1 55 56 72 73 99 100 121 122 141 142 160 161 185 186 202 203 228 229 253	4 52 56 72 77 100 104 121 125 142 145 171 175 185 189 213 217 235 238 253	16 60 63 81 86 103 111 128 139 148 154 168 181 188 191 220 223 229 234 253	3 40 44 65 69 82 85 110 113 135 139 166 169 191 195 205 210 230 234 253
Continued									

Images	D	MSICSA	CSA	VMCSA	IHBO	RUN_OBL	SO_OBL	SCGJO	SMA_MLS
Img5	4	7 137 143 250	6 141 149 249	8 135 143 250	7 140 148 251	5 140 146 250	6 137 143 252	25 140 145 253	4 142 149 251
	8	5 87 93 139 144 163 168 250	4 88 92 141 145 162 167 250	4 84 90 140 144 162 165 251	5 87 92 142 145 162 169 248	4 85 91 141 145 167 172 250	5 87 94 141 146 163 168 251	29 99 102 160 163 180 183 252	4 87 92 140 145 163 167 251
	12	5 62 70 80 85 130 134 160 166 200 205 252	9 63 70 80 81 130 136 160 165 192 199 252	8 65 69 82 86 122 128 171 177 190 194 252	5 62 67 81 86 132 137 162 167 194 200 251	7 63 69 74 78 130 134 161 167 194 198 252	5 63 70 80 85 132 135 163 168 195 199 251	22 83 90 130 136 142 145 195 199 216 220 251	4 63 69 80 83 128 133 168 173 192 196 252
	16	24 63 67 85 90 101 107 140 145 156 160 193 196 200 205 252	8 71 75 95 99 112 117 141 146 153 158 194 198 203 208 251	12 71 74 87 94 135 141 146 150 176 180 185 189 215 218 241	4 60 64 90 95 105 109 141 145 152 156 184 190 206 210 252	6 70 74 90 95 120 124 140 144 164 168 192 197 204 209 253	4 70 75 90 95 105 109 140 144 151 156 195 199 206 212 251	36 61 65 98 103 119 125 150 154 155 159 181 186 204 210 252	6 71 77 90 94 111 114 140 144 161 165 196 201 204 208 252
	20	2 47 50 55 59 86 89 103 107 134 139 153 156 174 178 200 203 213 216 252	3 50 53 64 68 90 94 125 128 140 144 165 169 181 182 204 208 215 220 253	4 33 36 59 65 83 86 120 123 133 137 164 169 174 181 206 211 215 218 252	4 34 38 52 56 81 85 101 105 124 129 162 166 172 175 201 205 220 224 253	7 43 46 55 60 83 87 106 110 130 134 151 155 175 178 201 205 213 216 253	6 45 49 56 59 85 89 96 99 130 134 151 155 174 178 200 203 216 219 252	18 47 50 75 78 104 107 130 133 144 147 165 169 195 199 219 225 232 239 252	7 44 45 60 61 85 86 117 118 130 132 165 166 171 174 201 203 214 215 253
Img6	4	9 144 149 249	10 142 149 248	7 143 148 251	9 146 152 251	8 143 149 252	9 148 152 250	25 160 164 252	11 140 149 253
	8	3 96 102 147 153 173 178 252	4 97 103 147 153 172 178 252	5 98 103 148 153 174 179 251	3 95 102 146 154 172 178 250	3 93 100 142 149 175 178 251	3 97 102 148 154 177 183 252	40 122 126 147 150 192 195 253	4 96 100 145 149 174 178 251
	12	5 86 92 95 98 142 146 175 179 192 196 251	11 82 87 90 94 142 146 173 179 193 199 251	5 85 89 92 96 132 138 161 167 197 200 252	9 71 75 90 94 130 133 172 178 192 197 251	12 05 76 90 93 133 138 172 178 191 198 252	14 84 90 96 99 142 147 180 183 200 202 251	37 75 79 99 109 139 155 199 200 208 211 251	5 84 91 95 99 149 150 180 185 203 208 252
	16	4 60 65 74 78 111 115 135 138 154 159 185 190 206 210 251	6 52 56 72 76 100 105 140 144 152 156 196 201 204 208 251	3 50 54 64 69 99 103 122 126 153 157 180 185 204 208 252	5 51 55 64 68 93 97 130 133 151 155 175 179 200 203 251	5 51 52 71 75 94 98 124 129 151 154 180 185 202 206 252	4 45 49 62 66 99 103 126 130 145 149 180 185 200 204 251	11 69 72 91 94 104 107 142 151 162 166 206 215 220 224 252	4 45 49 73 77 91 96 122 126 154 158 173 177 201 204 251
	20	12 64 68 70 73 95 100 103 107 131 134 153 157 172 176 204 208 212 215 253	30 53 59 66 69 94 98 110 114 134 139 152 156 179 184 194 198 212 216 253	11 64 69 82 86 94 100 121 125 133 137 158 161 182 187 198 206 211 214 252	19 53 57 62 67 90 93 115 118 141 144 163 167 184 188 210 213 221 225 253	4 61 64 71 75 94 97 105 110 130 133 152 157 172 177 200 203 212 215 252	18 53 57 65 69 91 95 116 119 141 145 161 164 180 186 204 208 215 219 253	37 52 56 71 76 93 96 115 118 140 143 155 159 163 169 215 219 230 234 253	13 52 56 60 63 100 103 110 114 144 149 154 158 175 179 201 204 213 217 252
Img7	4	3 125 128 252	2 126 128 250	11 126 138 250	5 120 128 250	9 127 133 251	4 120 128 251	27 142 145 251	4 127 134 251
	8	2 106 109 140 145 175 178 252	2 103 108 140 144 173 178 251	3 104 108 141 147 172 178 251	3 104 110 141 146 172 178 250	4 103 109 142 146 174 178 251	3 102 109 142 146 173 179 250	33 115 119 157 164 192 196 252	3 104 109 140 146 173 178 251
	12	3 32 40 109 115 122 130 172 180 187 196 252	5 32 40 104 111 123 129 174 179 190 195 251	7 34 39 115 121 124 128 182 191 196 199 252	8 32 40 110 114 121 126 175 180 186 189 251	1 32 40 110 115 125 131 174 178 190 194 252	7 33 38 110 114 120 126 181 184 187 190 251	24 78 80 100 110 149 154 170 175 202 206 251	4 38 40 114 118 128 130 183 186 190 194 252
	16	3 30 39 70 79 95 102 124 131 142 147 176 183 196 202 253	13 35 38 75 76 94 100 130 133 140 145 190 193 202 203 251	3 35 42 71 75 103 108 131 136 150 158 182 186 201 204 253	6 36 37 71 75 100 105 130 134 150 155 185 189 201 205 252	5 32 38 70 75 101 105 131 135 151 155 190 193 201 205 251	3 33 37 70 74 104 108 126 131 151 156 190 195 200 203 251	16 37 40 73 76 92 98 134 139 154 169 183 189 224 229 252	7 34 38 75 77 91 95 123 128 141 146 180 183 192 195 251
	20	14 30 36 60 66 70 78 100 107 120 126 147 155 167 173 190 198 207 216 249	13 33 37 60 63 71 74 106 110 122 127 150 154 170 174 200 204 211 214 249	13 32 38 71 74 90 94 114 117 129 133 162 166 177 182 200 204 211 219 250	6 34 37 60 64 73 78 107 110 122 126 151 154 162 166 194 199 210 213 252	5 33 38 61 64 80 83 104 108 124 128 152 155 170 174 202 205 210 213 253	3 33 37 60 63 72 77 104 108 121 125 151 155 170 174 195 199 210 214 253	15 28 41 75 79 88 92 110 114 123 136 140 143 165 170 193 197 200 204 252	2 35 40 72 76 80 85 111 115 125 129 158 164 170 175 205 209 210 214 253
Img8	4	7 178 182 187	7 172 179 189	8 172 178 190	9 173 181 187	9 171 179 189	6 173 181 189	12 161 168 251	8 172 179 190
	8	4 97 102 173 181 185 189 250	4 101 107 175 181 185 191 251	3 97 102 172 179 183 188 251	4 100 104 173 181 186 189 250	3 101 106 173 178 183 189 250	3 98 103 176 181 184 188 251	29 120 125 157 181 191 196 253	3 99 102 177 181 184 189 252
	12	7 68 73 82 87 120 125 178 182 186 193 251	5 64 69 90 94 120 126 171 182 187 194 252	4 60 64 86 88 115 117 180 183 187 193 251	10 62 67 81 86 121 127 180 184 189 196 252	10 62 67 81 86 121 127 180 184 189 196 252	5 63 69 81 85 123 127 178 182 185 189 251	45 83 89 120 125 142 146 202 204 206 211 252	5 60 65 84 88 122 125 176 181 184 185 252
	16	3 53 59 63 66 93 100 123 126 135 140 180 183 186 190 253	2 46 49 66 69 84 88 126 129 137 138 180 184 189 194 252	3 46 49 68 72 99 101 121 125 141 143 176 179 186 192 252	2 50 53 63 68 100 105 120 123 144 147 180 184 187 192 251	2 45 50 62 67 85 88 130 135 140 145 177 181 184 189 252	3 51 56 63 67 95 98 125 128 136 139 178 181 184 188 253	23 69 76 93 101 112 115 146 148 186 189 200 205 218 224 252	2 50 53 62 65 99 102 125 129 140 146 180 183 189 192 251
	20	4 40 43 56 61 86 89 105 108 124 128 144 148 181 185 188 192 216 220 252	3 58 62 66 69 94 98 101 105 125 129 140 144 180 184 193 197 211 215 253	4 30 34 62 71 84 87 117 120 123 127 150 155 173 178 190 194 201 213 252	3 41 45 61 65 84 88 103 106 123 127 141 145 180 183 186 190 215 218 253	6 43 47 60 65 83 89 94 98 126 130 140 143 180 184 187 191 211 215 252	3 44 49 55 58 86 90 100 104 125 130 140 143 176 180 185 188 210 215 253	11 55 62 66 80 100 105 117 120 132 140 158 161 177 183 201 205 225 228 253	6 43 46 60 64 83 87 101 105 130 134 140 144 180 183 187 191 210 213 252
CX1	4	7 111 120 248	10 110 122 251	8 113 121 250	8 111 123 248	12 117 120 251	10 109 122 249	16 140 144 247	9 112 120 247
	8	5 78 82 85 89 137 142 251	7 75 82 88 92 137 142 252	5 80 85 89 95 131 137 250	6 77 81 87 93 137 141 250	7 81 85 88 92 137 142 252	6 81 85 88 93 138 142 252	21 90 94 99 100 157 160 251	10 75 81 84 88 141 146 252
	12	9 60 65 74 80 105 110 141 144 153 156 244	7 62 67 73 78 110 115 140 145 152 157 252	8 69 72 81 86 115 119 141 145 151 156 251	7 62 66 74 79 104 107 135 138 152 157 252	4 61 65 72 77 103 108 134 140 151 155 251	7 60 65 71 75 110 115 140 146 156 159 240	30 57 61 94 99 100 105 132 137 161 166 252	7 62 67 80 83 104 108 142 147 153 158 244
	16	3 55 58 62 65 95 99 105 109 124 127 151 154 162 165 243	2 53 58 61 64 95 99 104 109 122 127 151 155 161 165 249	3 50 54 60 63 85 90 100 105 121 127 145 150 160 164 211	12 50 54 58 63 85 90 100 105 125 130 142 147 153 157 251	4 42 47 52 55 84 89 93 97 124 128 151 156 160 165 252	3 42 46 60 63 80 83 99 100 116 119 142 149 155 159 253	6 50 55 60 63 90 94 111 115 123 128 151 156 162 166 252	37 47 58 66 82 101 105 121 131 146 154 160 165 172 176 251
Continued									

Images	D	MSICSA	CSA	VMCSA	IHBO	RUN_OBL	SO_OBL	SCGJO	SMA_MLS
	20	18 81 83 95 98 114 117 137 140 152 155 174 178 192 195 211 215 222 226 250	29 82 85 93 95 111 115 125 129 141 145 162 166 190 193 201 204 220 224 251	33 80 84 92 96 108 116 130 133 150 154 166 174 185 190 205 209 232 235 249	3 11 25 40 43 81 85 101 106 123 127 150 153 165 170 192 195 210 214 253	14 76 80 90 93 113 116 130 133 151 154 166 170 185 189 202 205 221 226 253	43 62 66 80 84 94 98 115 118 132 135 147 151 183 187 195 198 218 221 252	21 25 37 44 49 54 57 91 96 101 105 134 139 156 160 177 194 205 210 253	24 81 86 92 96 111 115 124 127 146 150 172 175 193 197 200 205 221 225 253
CX2	4	6 70 75 248	7 74 79 250	7 70 75 250	8 72 76 251	1 74 78 253	8 71 77 247	19 101 112 251	5 71 75 250
	8	9 62 66 100 104 132 137 252	8 71 78 113 117 200 206 216	11 62 66 110 114 135 139 251	12 66 72 113 116 198 201 215	14 68 72 110 114 200 204 215	12 71 76 110 114 198 201 215	37 90 100 142 151 199 202 252	7 70 76 115 118 200 205 217
	12	6 55 59 64 69 113 117 135 139 198 201 213	8 52 56 64 69 112 116 135 140 198 201 216	9 48 55 71 75 112 118 141 144 191 202 216	7 52 56 63 69 110 115 141 145 199 204 215	5 51 55 71 76 112 116 134 140 200 206 215	6 51 57 63 70 111 116 142 146 200 206 215	35 62 69 97 108 134 148 193 201 205 207 252	4 50 54 71 72 113 118 154 157 200 206 216
	16	4 43 50 53 57 87 92 96 99 120 124 143 148 164 169 252	7 45 49 62 67 94 98 114 118 143 147 166 169 195 201 218	4 43 48 53 59 94 99 105 109 139 143 169 174 199 203 234	3 50 54 71 75 104 109 130 135 169 173 197 201 206 209 253	3 51 55 68 69 103 107 131 135 161 165 200 203 206 209 251	3 51 55 65 69 94 98 131 136 161 164 199 201 205 209 252	19 57 66 71 93 99 107 133 141 144 160 191 201 211 215 252	5 53 58 70 74 113 116 136 139 162 167 200 203 206 209 253
	20	6 42 45 55 58 76 79 103 107 124 127 145 148 174 177 198 202 206 211 253	2 42 46 55 58 75 79 101 105 122 126 152 156 162 166 196 202 206 209 253	10 40 42457 60 76 79 103 113 131 135 146 150 169 174 190 202 209 212 249	3 35 38 51 52 72 75 104 108 115 120 146 150 170 173 201 204 207 213 252	3 40 44 50 55 75 79 105 108 121 125 145 148 171 175 200 203 206 209 253	2 40 43 60 64 85 86 102 106 126 129 150 156 165 170 201 204 207 212 253	20 42 47 66 70 81 86 107 116 129 133 160 166 180 186 199 210 215 219 253	2 50 54 58 60 82 86 94 98 130 133 146 140 161 165 200 203 206 209 253
CX3	4	90 127 128 252	90 122 128 251	83 121 128 250	90 127 128 252	90 127 138 250	87 124 128 253	96 134 139 252	90 120 128 252
	8	44 81 86 120 123 147 151 252	46 81 85 120 126 150 155 252	42 76 78 112 119 136 140 245	46 82 85 122 127 150 155 251	46 81 85 120 124 147 151 252	46 80 85 120 124 150 156 251	59 104 108 136 140 176 179 251	43 80 85 114 120 145 150 252
	12	29 80 83 86 89 120 124 155 160 170 174 248	8 72 79 84 87 122 128 154 158 165 168 252	14 69 70 79 82 105 109 150 153 161 164 233	10 52 56 73 77 111 115 151 155 161 165 248	25 71 75 90 95 121 127 154 158 171 176 252	27 79 84 92 97 122 127 160 165 171 175 251	13 52 57 98 107 133 138 180 183 189 193 251	35 70 75 91 94 123 127 153 158 170 175 248
	16	7 30 34 60 64 73 77 112 116 123 127 161 164 170 174 245	3 25 28 51 56 70 74 100 103 124 127 161 165 169 174 228	7 23 28 58 61 77 81 111 114 127 133 160 164 169 174 237	3 32 36 50 53 80 83 102 107 136 139 164 169 171 176 247	3 30 34 50 55 74 78 94 100 123 127 161 165 169 173 253	2 24 27 51 54 81 86 102 105 124 128 161 164 168 173 251	15 56 59 85 89 96 113 124 133 146 150 165 167 180 188 253	4 23 29 44 48 76 79 101 104 120 124 157 161 164 168 249
	20	7 20 25 35 40 55 58 82 86 92 96 124 128 132 136 171 175 179 184 214	8 17 22 42 45 75 77 81 89 103 107 132 136 141 145 171 174 179 185 238	8 17 22 42 45 75 79 83 89 103 107 132 136 141 145 171 174 178 184 238	3 22 25 40 43 70 74 78 83 112 116 121 125 151 155 170 173 180 184 253	3 20 25 32 36 52 57 73 78 90 94 122 126 131 134 171 174 177 185 253	2 21 24 38 41 62 66 82 85 100 103 122 126 142 146 172 176 181 185 250	11 37 44 59 61 81 89 99 107 125 132 145 148 156 161 170 188 195 199 253	2 20 24 43 46 55 60 76 80 101 105 132 135 140 144 171 176 181 185 249
CX4	4	51 152 155 251	49 152 155 253	52 153 154 253	51 154 155 253	51 154 157 253	53 150 155 251	67 163 164 242	55 152 155 252
	8	35 103 107 122 126 183 188 209	35 102 107 120 123 184 189 209	41 95 99 160 163 180 183 251	36 100 105 121 125 185 189 209	43 103 107 161 165 183 190 252	35 107 110 121 125 185 190 208	36 123 125 160 170 205 208 253	43 100 104 160 165 186 187 251
	12	43 91 94 115 119 142 147 190 194 200 204 252	36 94 98 120 126 145 149 191 194 201 205 251	34 102 105 120 123 151 155 194 197 199 202 253	38 91 94 110 114 150 153 184 188 202 206 252	32 90 94 115 120 145 148 191 194 201 204 251	33 91 96 115 118 144 149 190 194 201 206 251	19 121 124 132 135 175 180 189 192 205 210 252	33 91 95 110 115 150 155 190 194 204 208 251
	16	41 84 87 100 103 133 136 150 156 171 174 194 197 204 207 252	35 90 94 97 100 124 128 140 143 163 168 190 195 204 209 252	23 84 88 110 115 120 124 145 148 161 165 192 196 207 216 253	35 75 78 94 99 113 117 141 146 163 167 191 195 203 208 251	31 84 90 100 103 130 133 150 155 170 174 190 194 204 209 253	43 80 83 102 106 120 124 143 147 160 163 190 194 211 215 252	20 84 89 105 112 135 138 152 157 168 182 196 199 213 219 252	27 83 88 97 99 130 135 150 153 163 167 192 197 202 206 253
	20	18 81 84 94 98 113 117 137 140 150 153 174 178 192 194 211 215 222 227 250	29 80 84 93 97 111 116 125 129 144 145 162 166 190 194 201 204 220 224 251	33 80 84 92 95 108 116 130 133 151 154 166 174 185 190 205 209 231 235 249	2 11 25 41 44 81 84 101 106 123 127 150 153 167 170 190 195 210 214 253	14 76 80 91 95 113 116 130 133 151 154 166 170 185 189 201 205 221 224 253	43 62 66 80 84 94 98 114 118 132 135 145 151 183 187 195 198 217 221 252	21 23 37 44 49 53 57 91 96 101 105 135 139 155 160 177 194 205 210 252	24 81 85 92 95 111 116 122 127 148 152 171 175 190 194 201 205 221 224 253

Table 21. Te best thresholds values for different algorithms using fuzzy entropy.

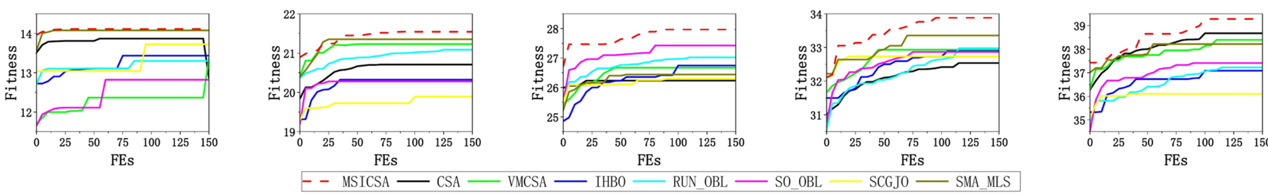


Fig. 26. Convergence curves of Img1, D = 4, 8, 12, 16 and 20 using fuzzy entropy.

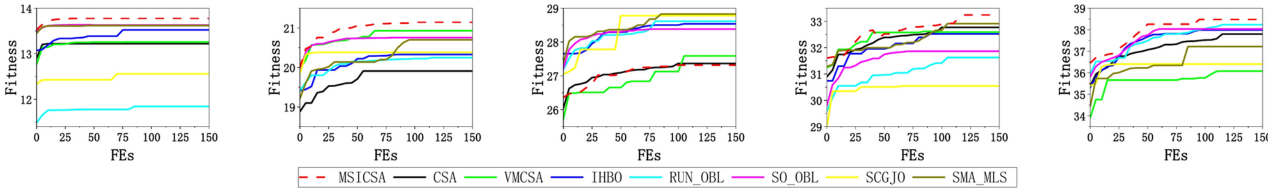


Fig. 27. Convergence curves of Img7, D = 4, 8, 12, 16 and 20 using fuzzy entropy.

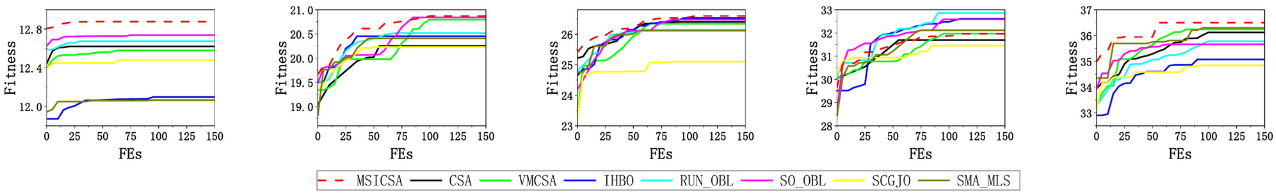


Fig. 28. Convergence curves of CX4, D = 4, 8, 12, 16 and 20 using fuzzy entropy.

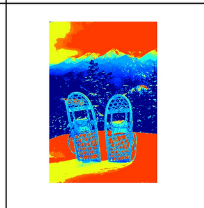
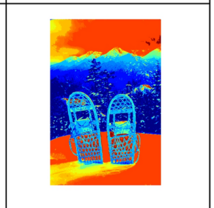
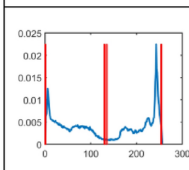
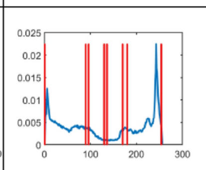
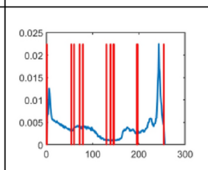
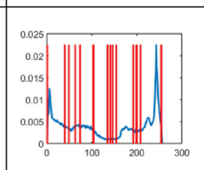
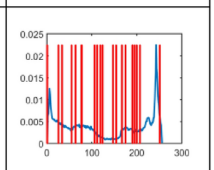
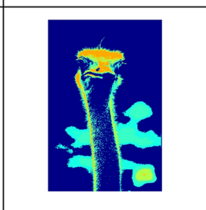
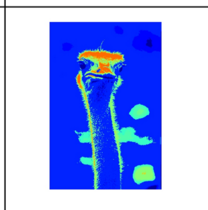
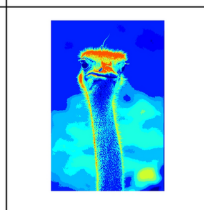
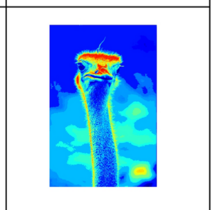
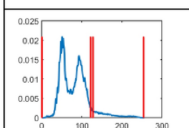
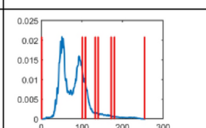
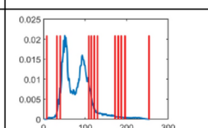
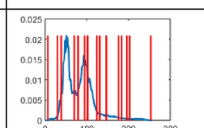
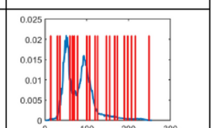
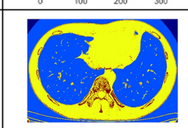
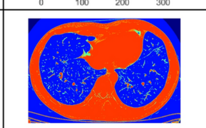
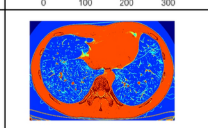
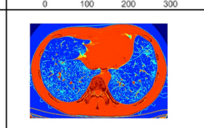
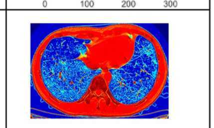
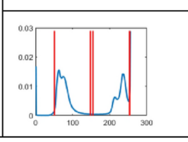
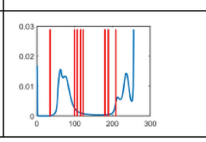
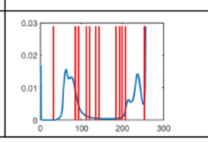
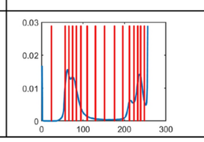
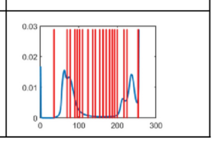
Images	D=4	D=8	D=12	D=16	D=20
Img1					
					
Img7					
					
CX4					
					

Table 22. Segmentation results of MSICSA using fuzzy entropy.

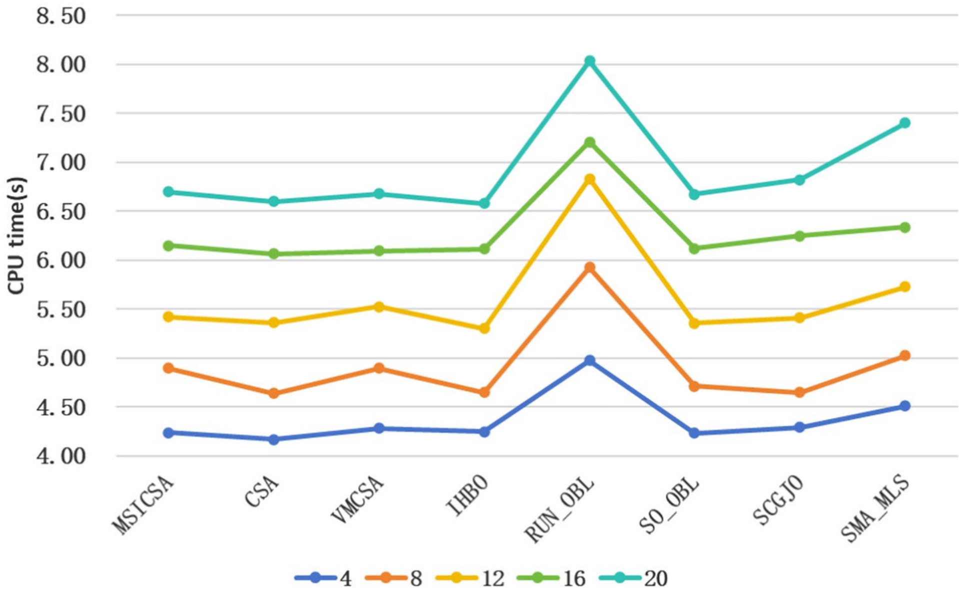


Fig. 29. Average CPU time of different algorithms at each threshold level using fuzzy entropy.

Data availability

All data generated or analysed during this study are included in this published article.

Received: 14 August 2024; Accepted: 12 March 2025
Published online: 06 June 2025

References

1. Mary, M., Padma, L. & John, M. Modified image segmentation method based on region growing and region merging. *Int. Arab J. Inf. Technol.* **13**, 899–907 (2016).
2. Yadav, N. K. & Saraswat, M. A novel fuzzy clustering based method for image segmentation in rgb-d images. *Eng. Appl. Artif. Intell.* **111**, 104709 (2022).
3. Santos, E. S. D. et al. Semi-automatic segmentation of skin lesions based on superpixels and hybrid texture information. *Med. Image Anal.* **77**, 102363 (2022).
4. Jiang, H. et al. A review of deep learning-based multiple-lesion recognition from medical images: classification, detection and segmentation. *Comput. Biol. Med.* **157**, 106726 (2023).
5. Sunaina, S. et al. Metaheuristic optimization algorithms and recent applications: A comprehensive survey. In *2023 International Conference on Computational Intelligence, Communication Technology and Networking (CICTN)* 506–511 (2023).
6. Rai, R., Das, A. & Dhal, K. G. Nature-inspired optimization algorithms and their significance in multi-thresholding image segmentation: An inclusive review. *Evol. Syst.* **13**, 889–945 (2022).
7. Liu, W. et al. Renyi's entropy based multilevel thresholding using a novel meta-heuristics algorithm. *Appl. Sci.* **10**, 3225 (2020).
8. Shivahare, B. D. et al. Survey paper: Whale optimization algorithm and its variant applications. In *2021 International Conference on Innovative Practices in Technology and Management (ICIPTM)* 77–82 (2021).
9. Khairuzzaman, A. K. M. & Chaudhury, S. Multilevel thresholding using grey wolf optimizer for image segmentation. *Expert. Syst. with Appl.* **86**, 64–76 (2017).
10. He, L. & Huang, S. An efficient krill herd algorithm for color image multilevel thresholding segmentation problem. *Appl. Soft Comput.* **89**, 106063 (2020).
11. Mousavirad, S. J. & Ebrahimpour-Komleh, H. Human mental search-based multilevel thresholding for image segmentation. *Appl. Soft Comput.* **97**, 105427 (2020).
12. Küükuurlu, B. & Gedikli, E. Symbiotic organisms search algorithm for multilevel thresholding of images. *Expert. Syst. with Appl.* **147**, 113210 (2020).
13. Shang, C., Zhang, D. & Yang, Y. A gradient-based method for multilevel thresholding. *Expert. Syst. with Appl.* **175**, 114845 (2021).
14. Mousavirad, S. J., Schaefer, G., Zhou, H. & Moghadam, M. H. How effective are current population-based metaheuristic algorithms for variance-based multi-level image thresholding?. *Knowl. Based Syst.* **272**, 110587 (2023).
15. Wu, B., Zhou, J., Ji, X., Yin, Y. & Shen, X. An ameliorated teaching–learning-based optimization algorithm based study of image segmentation for multilevel thresholding using kapur's entropy and otsu's between class variance—sciencedirect. *Inf. Sci.* **533**, 72–107 (2020).
16. Basu Dev, S. & Gupta, S. K. Multi-level image segmentation using randomized spiral-based whale optimization algorithm. *Recent Patents Eng.* **15**, 13–25 (2021).
17. Zhao, S., Wang, P., Heidari, A. A., Zhao, X. & Chen, H. Boosted crow search algorithm for handling multi-threshold image problems with application to x-ray images of COVID-19. *Expert. Syst. with Appl.* **213**, 119095 (2023).
18. Chauhan, D. & Yadav, A. A crossover-based optimization algorithm for multilevel image segmentation. *Soft Comput.* https://doi.org/10.1007/s00500-023-09398-w (2023).
19. Sayed, G. I. A novel multilevel thresholding algorithm based on quantum computing for abdominal ct liver images. *Evol. Intell.* **16**, 439–483 (2023).
20. Houssein, E. H., Mohamed, G. M., Ibrahim, I. A. & Wazery, Y. M. An efficient multilevel image thresholding method based on improved heap-based optimizer. *Sci. Rep.* **13**, 9094 (2023).
21. Casas-Ordaz, A., Oliva, D., Navarro, M. A., Ramos-Michel, A. & Perez-Cisneros, M. An improved opposition-based runge kutta optimizer for multilevel image thresholding. *J. Supercomput.* **79**, 17247–17354 (2023).

22. Chakraborty, S. & Mali, K. A multilevel biomedical image thresholding approach using the chaotic modified cuckoo search. *Soft Comput.* **28**, 5359–5436 (2024).
23. Zeng, T. et al. Differential evolutionary optimization fuzzy entropy for gland segmentation based on breast mammography imaging. *J. Radiat. Res. Appl. Sci.* **17**, 100966 (2024).
24. Houssein, E. H., Abdalkarim, N., Hussain, K. & Mohamed, E. Accurate multilevel thresholding image segmentation via oppositional snake optimization algorithm: Real cases with liver disease. *Comput. Biol. Med.* **169**, 107922 (2024).
25. Zhang, J., Zhang, G., Kong, M. & Zhang, T. SCGJO: A hybrid golden jackal optimization with a sine cosine algorithm for tackling multilevel thresholding image segmentation. *Multimed. Tools Appl.* **83**, 7681–7719 (2024).
26. Bei, J., Wang, J., Song, H. & Liu, H. Slime mould algorithm with mechanism of leadership and self-phagocytosis for multilevel thresholding of color image. *Appl. Soft Comput.* **163**, 111836 (2024).
27. Zhang, J. et al. A complex-valued encoding golden jackal optimization for multilevel thresholding image segmentation. *Appl. Soft Comput.* **165**, 112108 (2024).
28. Hashim, F. A. et al. An enhanced exponential distribution optimizer and its application for multi-level medical image thresholding problems. *Alex. Eng. J.* **93**, 142–188 (2024).
29. Houssein, E. H. et al. An improved honey badger algorithm for global optimization and multilevel thresholding segmentation: real case with brain tumor images. *Clust. Comput. J. Netw. Softw. Tools Appl.* **27**, 14315–14364 (2024).
30. Upadhyay, P. & Chhabra, J. K. Kapur's entropy based optimal multilevel image segmentation using crow search algorithm. *Appl. Soft Comput.* **97**, 105522 (2020).
31. Manimurugan, S., Majdi, A., Mohmmed, M., Narmatha, C. & Varatharajan, R. Intrusion detection in networks using crow search optimization algorithm with adaptive neuro-fuzzy inference system. *Microprocess. Microsyst.* **79**, 103261 (2020).
32. Farid, M. & Hamdi, A. A modified crow search algorithm (MCSA) for solving economic load dispatch problem. *Appl. Soft Comput.* **71**, 51–65 (2018).
33. Surendar, P. & Ponni, B. M. Diagnosis of lung cancer using hybrid deep neural network with adaptive sine cosine crow search algorithm. *J. Comput. Sci.* **53**, 101374 (2021).
34. Gupta, D., Sundaram, S., Khanna, A., Ella Hassanien, A. & De Albuquerque, V. H. C. Improved diagnosis of Parkinson's disease using optimized crow search algorithm. *Comput. Electr. Eng.* **68**, 412–424 (2018).
35. Wolpert, D. H. & Macready, W. G. No free lunch theorems for optimization. *IEEE Trans. Evol. Comput.* **1**, 67–82 (1997).
36. Otsu, N. A threshold selection method from gray-level histograms. *IEEE Trans. Syst. Man Cybern.* **9**, 62–66 (1979).
37. Naidu, M., Rajesh Kumar, P. & Chiranjeevi, K. Shannon and fuzzy entropy based evolutionary image thresholding for image segmentation. *Alex. Eng. J.* **57**, 1643–1655 (2018).
38. Sarkar, S., Paul, S., Burman, R., Das, S. & Chaudhuri, S. S. A fuzzy entropy based multi-level image thresholding using differential evolution. In *Swarm, Evolutionary, and Memetic Computing* (ed. Panigrahi, B. K.) 386–395 (Springer International Publishing, 2015).
39. Tizhoosh, H. Opposition-based learning: A new scheme for machine intelligence. In *International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'06)*, Vol. 1 695–701 (2005).
40. Dinkar, S. K., Deep, K., Mirjalili, S. & Thapliyal, S. Opposition-based laplacian equilibrium optimizer with application in image segmentation using multilevel thresholding. *Expert. Syst. with Appl.* **174**, 114766 (2021).
41. Houssein, E. H., Hussain, K., Abualigah, L., Elaziz, M. A. & Cuevas, E. An improved opposition-based marine predators algorithm for global optimization and multilevel thresholding image segmentation. *Knowl. Based Syst.* **229**, 107348 (2021).
42. Xing, Z. An improved emperor penguin optimization based multilevel thresholding for color image segmentation. *Knowl. Based Syst.* **194**, 105570 (2020).
43. Shajin, F. H., Devi, B., Prakash, N., Sreekanth, G. & Rajesh, P. Sailfish optimizer with levy flight, chaotic and oppositionbased multi-level thresholding for medical image segmentation. *Soft Comput.* **27**, 12457–12482 (2023).
44. Kennedy, J. & Eberhart, R. Particle swarm optimization. In *Proceedings of ICNN'95—International Conference on Neural Networks*, Vol. 4 1942–1948 (1995).
45. Mirjalili, S. SCA: A sine cosine algorithm for solving optimization problems. *Knowl. Based Syst.* **96**, 120–133 (2016).
46. Hashim, F. A. & Hussien, A. G. Snake optimizer: A novel meta-heuristic optimization algorithm. *Knowl. Based Syst.* **242**, 108320 (2022).
47. Chopra, N. & Mohsin Ansari, M. Golden jackal optimization: A novel nature-inspired optimizer for engineering applications. *Expert. Syst. Appl.* **198**, 116924 (2022).
48. Horé, A. & Ziou, D. Image quality metrics: PSNR vs. SSIM. In *2010 20th International Conference on Pattern Recognition* 2366–2369 (2010).
49. Wang, Z., Bovik, A., Sheikh, H. & Simoncelli, E. Image quality assessment: From error visibility to structural similarity. *IEEE Trans. Image Process.* **13**, 600–612 (2004).
50. Zhang, L., Zhang, L., Mou, X. & Zhang, D. FSIM: A feature similarity index for image quality assessment. *IEEE Trans. Image Process.* **20**, 2378–2386 (2011).
51. Shott, S. *Non-parametric Statistics* 93–108 (Springer Milan, 2009).

Author contributions

X.Z.:Architecture design and writing of the main manuscript; C.H.:Simulation experiments ,data analysis and chart drawing; W.G.:Data collection and paper revision. All authors reviewed the manuscript.

Funding

This work was supported in part by the Guangxi Key Laboratory of Big Data in Finance and Economics and in part by Guangxi First-class Discipline statistics Construction Project Fund.

Declarations

Competing interests

The authors declare that there is no Competing interests.

Additional information

Correspondence and requests for materials should be addressed to C.H.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.