Computer Fraud and Security
ISSN (online): 1873-7056

Research on Image Segmentation Algorithm based on Improved SGDF-SSA Algorithm of Chaotic Map and Gaussian Cloud

Yan Chai*, Xiang Yu
College of Science, Liaoning Technical University, Fuxin Liaoning 123000, China
**Corresponding Author.*

Abstract:
Leaf image segmentation is of great significance in automatically segmenting foreground leaves from noisy background, and the accuracy of image segmentation requires high. Through the improvement of the algorithm, the accuracy of leaf extraction and crop analysis can be greatly improved. The Sparrow Search Algorithm (SSA) is a commonly used image segmentation method. However, the traditional SSA suffers from issues such as a tendency to fall into local optima and insufficient search capability during the optimization process. To address these shortcomings, we propose an improved SSA model, the SGDF-SSA algorithm. First, we use the chaotic phenomenon generated by the SPM chaotic map to initialize the particle population, enhancing the randomness and traversal ability of particles and thereby improving global search capability. Second, we introduce an adaptive Gaussian cloud mutation strategy in the discoverer position update process to further enhance global search ability during iterations. Additionally, we design a sinecosine optimization and inertia weight-based discoverer update mechanism to improve the discoverers' global search capability. Moreover, we propose a follower update mechanism based on Cauchy chaotic mutation, which combines chaotic mapping and Cauchy mutation to prevent the algorithm from falling into local optima. The improved algorithm outperforms various comparative algorithms in terms of average performance on the CEC2017 benchmark test set, achieving superior results in 11 test functions. It demonstrates better performance in different fidelity parameters and computational time, showing promising potential for plant leaf image segmentation. This advancement is expected to contribute to the progress of leaf pathology analysis and precision agriculture mechanized batch processing technology.

Keywords: leaf segmentation, sparrow search algorithm, swarm intelligence optimization, SPM chaotic map, adaptive gaussian cloud, sine cosine optimization, cauchy chaotic mutation

INTRODUCTION

With the continuous advancements in computer technology, solving various problems using image processing and computer vision techniques has become feasible. Increasing attention has been given to applying digital image processing techniques to plant segmentation and classification studies. Plants are one of the essential resources for human production and daily life, representing the most widely distributed life forms on Earth. Analyzing, classifying, and identifying plants can facilitate better plant conservation, maintain species diversity, aid in disaster prevention and mitigation, and provide convenience for various industries such as urban environmental planning, tourism, and photography education. Although there are numerous methods for plant research, traditional botanical research techniques are primarily mastered by a small group of professionals, making it extremely difficult for nonexperts to acquire these skills.

Plants are generally composed of organs such as roots, stems, leaves, and fruits, all of which serve as taxonomic criteria in botany. During plant growth, roots and stems undergo changes based on environmental conditions, and even within the same species, the shape, size, and other characteristics of roots and stems can vary significantly [1]. Consequently, identifying plants based on root or stem images is challenging. In contrast, leaves, as crucial plant organs composed of the leaf blade, petiole, and stipules, provide important morphological, textural, and color information that can be directly observed. Unlike other plant features, leaf shape and color characteristics remain relatively stable throughout the seasons, making them reliable criteria for plant classification and identification [2]. For a given leaf image, effectively segmenting the complete target leaf region is critical for providing reliable data for subsequent research [3].

In plant leaf segmentation using computer vision, existing segmentation approaches primarily fall into two categories: one involves extracting leaves after they are plucked and photographed against a uniform background, while the other focuses on segmenting leaves from images captured in natural scenes [4]. The latter approach is more versatile but presents challenges such as complex backgrounds, monochromatic tones, and leaf occlusion.

To accurately identify and analyze target leaves in plant leaf images, it is necessary to segment them from these complex backgrounds. Existing methods, such as single-threshold-based segmentation, edge-based segmentation, and clustering-based segmentation, often suffer from inefficiencies or excessive computational time when dealing with such complex images [5]. Since subsequent plant classification and identification rely on the accuracy of segmentation, there is a pressing need for a more suitable and efficient segmentation method to enhance the accuracy of leaf extraction in images [6]. This will provide more objective and precise experimental materials for subsequent classification and identification tasks.

Image segmentation is a crucial step in image processing, and thresholding segmentation is one of the most commonly used techniques, particularly in plant leaf-related research [7]. However, existing thresholding segmentation methods for plant leaves are predominantly based on single-threshold segmentation [8]. When applied to leaf image segmentation, single-threshold methods can only divide the image into two grayscale regions. Due to the complexity of leaf images in natural scenes, even after preprocessing, single-threshold segmentation often fails to completely distinguish between the foreground and background. This not only affects the accuracy of leaf region segmentation but also increases overall processing time, reducing efficiency [9].

Compared to single-threshold segmentation, multi-threshold segmentation achieves better results. To enhance the accuracy of plant leaf segmentation, this study explores the use of multi-threshold segmentation to refine segmentation results for leaf images in natural scenes, effectively distinguishing between foreground and background regions [10]. However, multi-threshold segmentation involves several challenges, such as prolonged computational time and difficulty in determining the optimal number of segmentation thresholds for different images [11].

LITERATURE REVIEWS

Image segmentation is a crucial step in image processing and, fundamentally, a classification problem. It involves partitioning an image into several nonoverlapping regions based on characteristics such as grayscale intensity, color, shape, and texture [12]. Regions within the same segment exhibit similarity or consistency, while different regions display significant differences. Research on image segmentation has long been a focal point in the field of digital image processing, as the quality of segmentation directly impacts subsequent image processing tasks [13]. Due to the wide range of segmentation methods available, selecting an appropriate method largely depends on the specific characteristics of the target image [14]. As a key stage in image processing, image segmentation also remains one of the most challenging problems in computer vision, making it difficult to develop a universal segmentation technique applicable across all domains [15]. Existing image segmentation methods can be broadly classified into supervised and unsupervised approaches [16]. Supervised segmentation methods typically rely on neural networks, requiring extensive labled datasets and significant training time to classify pixels or entire images [17]. These methods generally achieve high segmentation accuracy in specific scenarios. In contrast, unsupervised segmentation methods operate on individual images, leveraging pixel-level at-tributes such as colour, texture, and spatial position to segment meaningful regions based on various strategies. These methods generally offer better real-time performance. Among unsupervised methods, clustering-based segmentation is particularly notable for its speed, broad applicability, and independence from prior knowledge [18]. As a representative unsupervised approach, clustering-based segmentation has been widely applied in the field of image segmentation and holds significant research value [19].

When applying clustering algorithms to image segmentation, the meaningful subregions in the segmentation results are considered natural groupings discovered within the image [20]. Clustering-based image segmentation methods divide an image into several meaningful regions based on each pixel’s colour and spatial information [21]. In this context, image segmentation is treated as an optimization problem, where the objective is to ensure that pixels representing the same object belong to the same cluster, while pixels representing different objects are assigned to different clusters [22]. At the same time, the similarity among pixels within the same cluster should be maximized, while the differences between pixels in different clusters should be clearly distinguished.

Traditional clustering algorithms iteratively update cluster centers during segmentation. Once the stopping criteria are met, pixels are assigned to clusters based on their distance to the cluster centers. For example, Das et al. proposed the kernel fuzzy c-means (KFCM) clustering method for image segmentation, which integrates adaptive

thresholding, morphological processing, and kernel functions to enhance segmentation quality [23]. K-means clustering algorithm combined with morphological processing was used to solve the problem that the target contour of psoriasis could not be accurately delineated in colour skin images [24]. Wang Qinqin applied a median filter for image preprocessing before using the fuzzy c-means (FCM) clustering algorithm to segment target objects from the background, followed by a random walk algorithm to further refine tumour segmentation [25]. When using density-based clustering algorithms for image segmentation, the process begins by selecting the pixel with the highest density—i.e., the one surrounded by the most densely distributed neighbouring pixels. The algorithm then iteratively merges nearby high-density pixels until all pixels are assigned to a cluster [26].

Supervised image segmentation methods are typically implemented using neural networks. A neural network is a mathematical model inspired by the functioning of biological neural systems, utilizing a distributed and parallel approach for information processing [27]. By continuously adjusting its hierarchical structure and the connections between numerous nodes, a neural network effectively processes information [28]. Although supervised image segmentation methods can achieve high segmentation accuracy in specific scenarios, they require extensive labeled datasets and significant computational time for model training [29]. Additionally, model accuracy must be validated on test datasets, followed by parameter finetuning based on the application context before deployment. This lengthy preparation process limits the real-time performance and generalizability of supervised segmentation methods [30].

In recent years, researchers have conducted extensive studies on clustering-based unsupervised image segmentation methods, achieving significant progress. However, commonly used clustering-based segmentation algorithms still have several limitations: One major issue is the presence of noise in images obtained through various sources [31]. In image segmentation tasks, noise can severely impact segmentation results, making noise reduction a crucial consideration in all segmentation methods [32]. A popular approach to mitigating noise effects is incorporating neighbourhood information of each pixel into the objective function [33]. By considering a pixel's surrounding context, the impact of noise can be reduced. Several representative methods have been proposed in this regard [34]. For example, Ahmed et al. incorporated pixel neighbourhood information into the clustering objective function, ensuring that each pixel’s label is influenced by the labels of its surrounding pixels, thereby reducing noise interference [35]. Chen and Zhang introduced two improved FCM-based methods, FCM_S1 and FCM_S2, which define local information terms based on mean-filtered and median-filtered images in the objective function to minimize noise effects [36]. However, these methods often come with increased computational costs due to the additional complexity in the objective function [37]. Another effective strategy is using superpixels for image preprocessing. By grouping adjacent pixels with similar properties into superpixels, the segmentation algorithm can reduce noise influence while also decreasing data complexity [38]. For instance, Lei et al. proposed an Automatic Fuzzy Clustering Framework (AFCF) based on superpixels and density peak clustering [39]. Their method first applies superpixel segmentation to reduce the size of the distance matrix in subsequent density peak clustering while simultaneously minimizing noise interference. Then, it utilizes a density-balancing algorithm and prior entropy knowledge to complete the image segmentation process.

Plant leaf image segmentation plays a crucial role in precision agriculture, disease detection, and growth monitoring. However, due to complex environmental backgrounds, noise interference, and the diverse morphology of leaves, traditional segmentation methods often struggle to balance high accuracy and robustness [40]. In recent years, image segmentation methods based on intelligent optimization algorithms have gained significant attention, with the Sparrow Search Algorithm (SSA) being particularly notable for its strong global optimization capability and adaptability [41]. However, conventional SSA tends to fall into local optima during the optimization process, leading to low search efficiency and limiting further improvements in segmentation accuracy [42]. Therefore, enhancing SSA’s global search ability and optimization stability is essential for improving the precision and reliability of leaf image segmentation [43].

To address these challenges, this study proposes an improved SSA, the SGDF-SSA algorithm, designed to enhance search performance and improve leaf extraction under complex backgrounds. The significance of this research is reflected in several key aspects. First, the SGDF-SSA algorithm utilizes chaotic phenomena derived from a novel one-dimensional chaotic system (SPM) for initialization, significantly enhancing particle diversity and randomness while improving large-scale search capability. Second, an adaptive Gaussian cloud mutation

strategy is incorporated to refine the position update mechanism of discoverers, allowing the algorithm to dynamically adjust search step sizes during iterations, accelerating convergence while maintaining diversity to prevent premature convergence. Additionally, Cauchy chaotic mutation is introduced to strengthen the search capability of followers, ensuring superior performance across different datasets and demonstrating strong generalization ability. This study not only optimizes the theoretical performance of the algorithm but also achieves promising results in plant leaf segmentation applications. The proposed approach provides a more efficient computational tool for disease analysis and precision crop monitoring, contributing to the advancement of agricultural mechanization and fostering further development in agricultural intelligence.

METHODOLOGY

Basic Sparrow Search algorithm

The Sparrow Search Algorithm (SSA) is a swarm intelligence optimization algorithm inspired by the foraging behaviour and predator-avoidance strategies of sparrows [44]. It simulates the foraging process of a sparrow population by establishing a discoverer-joiner model and incorporating a SGDF-SSA warning mechanism [45]. Discoverers guide the population by identifying optimal foraging areas and directions, possessing higher fitness values [46]. Joiners, with relatively lower fitness, improve their fitness by following discoverers in the foraging process [47]. During the simulation, the proportion of discoverers and joiners within the population remains constant [48]. To enhance their foraging efficiency and compete for food resources, some joiners continuously monitor discoverers. If a sparrow perceives danger or encounters a predator threat, it immediately takes safety measures [49]. In a d-dimensional search space with N sparrows, the position of the i-th sparrow is represented as:

$$X_i = [x_{i1}, \ldots, x_{id}] \tag{1}$$

where x_{id} denotes the position of the i-th sparrow in the d-th dimension, with $i = 1,2, \ldots, N$. The fitness values of all sparrows are given by:

$$F_x = \begin{bmatrix} f(x_1^1, x_1^2, \ldots, x_1^d) \\ f(x_2^1, x_2^2, \ldots, x_2^d) \\ \vdots \\ f(x_N^1, x_N^2, \ldots, x_N^d) \end{bmatrix} \tag{2}$$

where f represents the fitness function. Within the sparrow population, high-fitness discoverers are responsible for locating food for the entire group and have priority access to it. They also guide the joiners in the foraging direction while maintaining a broad search range [47]. In each iteration, discoverers typically constitute 10% to 20% of the population. The position update equation for discoverers is as follows:

$$x_{id}^{(t+1)} = \begin{cases} x_{id}^t \exp\left(\dfrac{-i}{\alpha T}\right), & R_2 < ST \\ x_{id}^t + QL, & R_2 \geq ST \end{cases} \tag{3}$$

x_{id}^t represents the position of the i-th sparrow in the d-th dimension at iteration t. T is the total number of iterations. $L = 1 \times d$, where each element in this matrix is 1. α is a random number in the range $(0,1)$. Q follows a normal distribution $N(0,1)$. The warning value R_2 is randomly selected from $[0,1]$. The safety threshold ST is within the range $[0.5,1]$. When $R_2 < ST$, the search environment is safe, allowing discoverers to conduct a wide-range search and improve the population's fitness. However, when $R_2 \geq ST$, the search environment becomes dangerous. Sparrows performing SGDF-SSA functions promptly emit danger signals to the group, prompting them to adopt antipredation behaviours, adjust their search routes, and swiftly move to safer locations for foraging.

The position update equation for joiners is given by:

$$x_{id}^{(t+1)} = \begin{cases} Q\exp\left(\dfrac{x_{w_d}^t - x_{id}^t}{i^2}\right), & i > n/2 \\ x_{b_d}^{(t+1)} + \left|x_{id}^t - x_{b_d}^{(t+1)}\right| A^+ L, & \text{otherwise} \end{cases} \tag{4}$$

$x_{w_d}^t$ represents the worst position in the d-th dimension at iteration t. $x_{b_d}^{(t+1)}$ denotes the best position in the d-th dimension at iteration $t+1$. A is a $1 \times d$ matrix with each element taking values of 1 or -1, and $A^+ = A^T(AA^T)^{-1}$, allowing for slight position perturbations. To further refine the model, we introduce a random factor to modify the update equation as follows:

$$x_{id}^{(t+1)} = \begin{cases} Q\exp\left(\dfrac{x_{w_d}^t - x_{id}^t}{i^2}\right), & i > n/2 \\ x_{b_d}^{(t+1)} + \dfrac{1}{D}\sum_{d=1}^{D} \text{rand}[-1,1] \times \left|x_{id}^t - x_{b_d}^{(t+1)}\right|, & i \le n/2 \end{cases} \tag{5}$$

When $i > n/2$, the joiner has a low fitness value, indicating it has failed to obtain food and is in a state of hunger. In this case, it needs to change its path and fly elsewhere to forage. When $i \le n/2$, the joiner is near the current optimal position x_b and can search for food in an arbitrary nearby location. Position Update for Sparrows with SGDF-SSA and Warning Functions. Sparrows with SGDF-SSA and warning abilities typically make up 10%-20% of the population. Their position updates are governed by the following equation:

$$x_{id}^{t+1} = \begin{cases} xb_d^t + \beta(x_{id}^t - xb_d^t), & f_i \neq f_g \\ x_{id}^t + K\left(\dfrac{x_{id}^t - xw_d^t}{|f_i - f_w| + e}\right), & f_i = f_g \end{cases} \tag{6}$$

K is a random number in the range $[-1,1]$, controlling the step size and movement direction. β is a step size control parameter following a normal distribution with a mean of 0 and variance of 1. ϵ is a very small constant to prevent division by zero. f_i represents the fitness value of the i-th sparrow. f_g and f_w denote the best and worst fitness values in the current sparrow population, respectively.

SFM Chaotic Map

A chaotic state refers to a condition in which, given an initial value, a special function iteratively updates the value to form a set of multiple elements. Within this finite set space, a one-dimensional particle exhibits asymmetric, aperiodic, and unstable motion, known as chaotic motion. Bucolo et al. proposed that the randomness of chaotic motion is superior to that of traditional uniform distribution random number generators [50]. This advantage is particularly significant in population-based search spaces with multiple local solutions, where finding the global optimum becomes increasingly difficult [51]. However, by incorporating chaotic sequences into the algorithm during population initialization, position, and velocity updates, the entire iterative process can be positively influenced. Generally, using chaotic sequences is more effective than using pseudo-random number generators, as it enhances the algorithm's performance and facilitates the discovery of the global optimal solution [52].

Chaotic mapping is a method used to generate chaotic sequences, which can be either one dimensional or multi-dimensional [53]. Common chaotic mappings include the Chebyshev chaotic map, Henon chaotic map, and Logistic chaotic map, Among them, the SPM (Stochastic Perturbation Map) chaotic mapping exhibits excellent variable random distribution and strong ergodicity. Therefore, in this study, the chaotic state characteristics derived from SPM chaotic mapping are utilized to enhance the randomness of the initial particle position distribution. The SPM chaotic mapping expression is defined as:

$$x_{i+1}=\begin{cases}\mathrm{mod}\left(\dfrac{x_i}{\eta}+\mu\sin\left(\pi x_i\right)+g_{\mathrm{tur}},1\right), & 0\le x_i<\eta\\ \mathrm{mod}\left(\dfrac{(x_i/\eta)}{(0.5-\eta)}+\mu\sin\left(\pi x_i\right)+g_{\mathrm{tur}},1\right), & \eta\le x_i<0.5\\ \mathrm{mod}\left(\dfrac{(1-x_i)}{\eta}+\mu\sin\left(\pi(1-x_i)\right)+g_{\mathrm{uur}},1\right), & 0.5\le x_i<1\end{cases} \tag{7}$$

where $\eta\in(0,1),\mu\in(0,1)$, and under these conditions, the chaotic system transitions into a chaotic state. g_{tur} represents Gaussian perturbation following a standard normal distribution, while the initial sequence value x_1 is a randomly selected number within the range $[0,1)$. The function mod denotes the modulus operation (i.e., the remainder after division).

SPM Chaotic Mapping-Based Particle Swarm Initialization Steps

Step 1: Generate an initial random number x_1 less than 1. According to reference [47], the parameters influencing the chaotic state are set as $\eta=0.4$ and $\mu=0.3$. The number of sequence elements, n, is determined, and Gaussian perturbation $g_{\mathrm{ur}}\sim N(0,1)$ is introduced.

Step 2: Set the loop count to $n-1$. Using the initial value x_1, update each subsequent value x_i iteratively based on the SPM chaotic mapping equation (7), generating a complete chaotic sequence $x_i=[x_1,x_2,\ldots,x_n]$.

Step 3: Construct the actual grayscale chaotic sequence using $X_i=255\cdot x_i$, ensuring that the boundary condition $0\le X_i\le 255$ is satisfied. This guarantees that the population size is within the grayscale range of $0-255$, completing the particle swarm initialization.

When the sequence length is set to 500, a comparison between the SPM chaotic mapping and the Logistic chaotic mapping is shown in Figure 1.

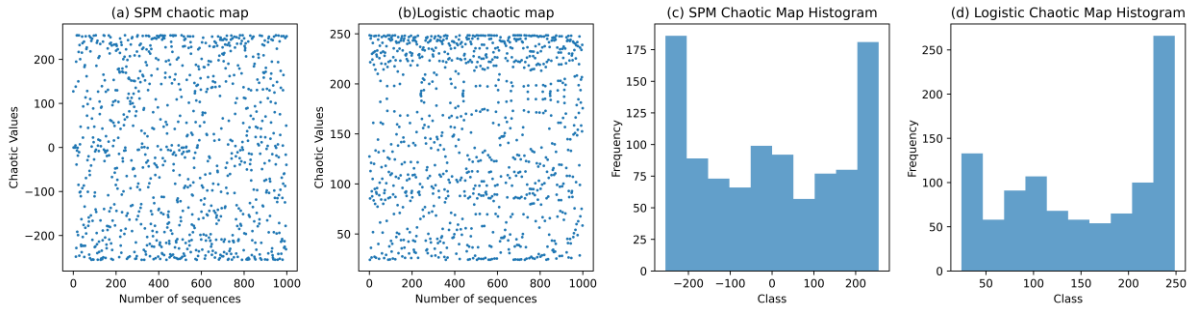

Figure 1. Histogram comparison of SPM and logistic chaotic map values

From the chaotic grayscale values and histograms of both mappings, it can be observed that within the entire range of chaotic mapping values, the Logistic chaotic mapping exhibits a biased distribution tendency. In contrast, the trajectory of the SPM chaotic mapping is more evenly distributed across the full grayscale range of 0 to 255. This indicates that the values derived from the SPM chaotic mapping possess superior randomness, which enhances the traversal capability during the particle swarm initialization stage.

Adaptive Gaussian Cloud Mutation Strategy

In the original SSA, the discoverer updates its position when $R_2<ST$ based on its current position. The movement range is given by $y=\exp\left(\frac{-x}{\alpha T}\right)$, x is the iteration number and y is the position change value. As shown in Figure 2, as x increases, y gradually decreases from the range $(0,1)$ to $(0,0.4)$, meaning the sparrow's position in each dimension gets smaller. In SSA, the discoverer, as the dominant individual, guides the population in the search process, and its search range determines the optimization accuracy of the SSA. However, in the original algorithm, the discoverer's search range shrinks as the iteration progresses, which can lead to a decrease in population diversity during the later stages of the algorithm, causing it to easily get stuck in local optima.

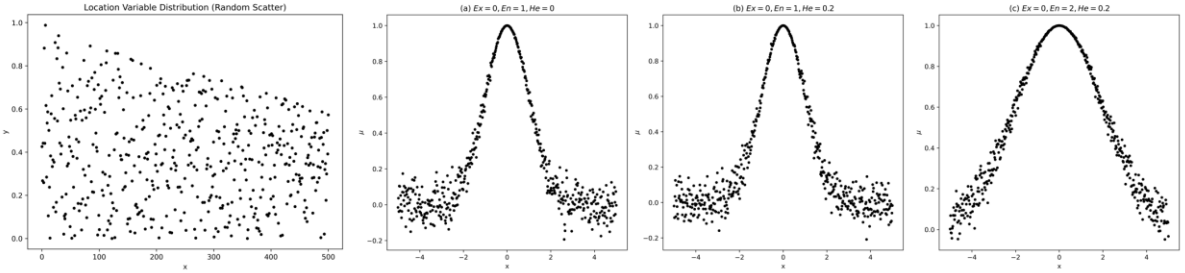

Figure 2. Location variable distribution and gaussian cloud distribution

The Gaussian cloud consists of numerous cloud droplets. The model can be characterized by three parameters: expectation E_x, entropy E_n, and hyperentropy H_e. Among them, the expectation E_x reflects the distribution expectation of the cloud droplets in space, entropy E_n reflects the uncertainty of the droplets, and hyperentropy H_e reflects the uncertainty of the entropy E_n. The function is defined by $\mu = \exp\left(-\frac{(x-E_x)^2}{2(E_n')^2}\right)$, Figure 2 shows the Gaussian cloud models generated under three different parameter conditions, it can be observed that, when the expectation E_x is constant, as H_e increases, the Gaussian cloud distribution becomes more dispersed; as E_n increases, the value range of the Gaussian cloud becomes wider.

By leveraging the randomness and fuzziness characteristics of the Gaussian cloud model, an adaptive Gaussian cloud mutation strategy is proposed to improve SSA. This strategy generates a mutation position that allows SSA to perform local searches within a small range, while with a certain probability, a new position far from the current one is generated, enabling SSA to search in a larger area. This improves population diversity, enhances the algorithm's ability to escape from local optima, and allows better global search. Consequently, the search speed is increased, and the algorithm's convergence rate is accelerated. The position update formula is given as:

$$x_i^{(t+1)} = \begin{cases} x_i^t \cdot \left(1 + \text{Gauss}\left(0, \delta^2, 0.1\delta^2\right)\right) & R_2 < ST \\ x_i^t + Q \cdot L & R_2 \geq ST \end{cases} \tag{8}$$

where $\text{Gauss}(0, \delta^2, 0.1\delta^2)$ represents the Gaussian cloud random number with parameters $(E_x, E_n, H_e) = (0, \delta^2, 0.1\delta^2)$, and δ^2 is defined as follows:

$$\delta^2 = \begin{cases} 1, & f_i \leq f_s \\ \exp\left(\frac{(f_s - f_i)}{|f_i| + e}\right), & f_i > f_s \end{cases} \tag{9}$$

where f_i is the fitness of the i-th sparrow, f_s is the fitness of a randomly selected discoverer, and e is an infinitesimally small constant. From the above equation, when the fitness of the i-th discoverer is better than that of the s-th discoverer ($f_i \leq f_s$), the sparrow will explore a broader area to avoid getting trapped in local extrema. Otherwise, it will narrow the search area to focus on refining the search in the local region. The hyperentropy term $0.1\delta^2$ ensures the fuzziness of the mutation solutions, further enhancing the search capability of the algorithm.

Finder Update Mechanism Based on Sine Cosine and Inertia Weight

When $R_2 < ST$, discoverers can perform extensive searches within the safe zone. However, as iterations progress, the dimensional values of discoverer sparrows decrease, leading to a gradual reduction in search space. This insufficient search space, along with decreased population diversity, results in reduced search accuracy and an increased likelihood of falling into local optima.

To improve the position update mechanism of discoverers and enhance their global search capability, SGDF-SSA introduces the sine cosine optimization algorithm (SCA) and an inertia weight mechanism for position updates. SCA is a novel heuristic search algorithm that updates individuals and optimizes targets based on the mathematical properties and variations of sine and cosine functions. It has strong global optimization capabilities, and its particle position update formula is given by:

$$x_{i,j}(t+1)=\begin{cases}x_{i,j}(t)+r_1\cdot\sin(r_2)\cdot\left|r_3\cdot x_{\text{best},j}-x_{i,j}(t)\right|, & r_4<0.5\\ x_{i,j}(t)+r_1\cdot\cos(r_2)\cdot\left|r_3\cdot x_{\text{best},j}-x_{i,j}(t)\right|, & r_4\geq 0.5\end{cases} \tag{10}$$

where $x_{\text{best},j}$ represents the j-th dimension of the current global optimal solution, r_1 is the amplitude adjustment factor, $r_2 \in [0,2\pi]$, $r_3 \in [-2,2]$, and $r_4 \in [0,1]$ are uniformly distributed random variables. This position update process moves particles towards the current global best position, which enhances local exploitation. However, it may lead to premature convergence and local optima. To address this issue, SGDF-SSA incorporates an inertia weight mechanism to strengthen global search capability. In the early stages of iteration, a larger inertia weight is used to improve global exploration, while in later iterations, a smaller inertia weight is applied to enhance local exploitation and accelerate convergence. The inertia weight is defined $w(t)=w_{\min}+(w_{\max}-w_{\min})\cdot e^{-t/T_{\max}}$, where $w_{\max}$ and $w_{\min}$ are the maximum and minimum inertia weights, respectively. Thus, the updated position formula for SGDF-SSA discoverers is given by:

$$x_{i,j}(t+1)=\begin{cases}w(t)\cdot x_{i,j}(t)+r_1\cdot\sin(r_2)\cdot\left|r_3\cdot x_{\text{best},j}-x_{i,j}(t)\right|, & R_2<ST\\ w(t)\cdot x_{i,j}(t)+r_1\cdot\cos(r_2)\cdot\left|r_3\cdot x_{\text{best},j}-x_{i,j}(t)\right|, & R_2\geq ST\end{cases} \tag{11}$$

By integrating the inertia weight mechanism, SGDF-SSA enhances global exploration in early iterations and improves local exploitation in later stages, ultimately increasing search efficiency and segmentation accuracy.

Follower Update Mechanism Based on Cauchy Chaotic Mutation

In SSA, followers will move closer around the current global optimal solution. Although this method can accelerate the algorithm's convergence by using elite individuals as guides, it leads to a loss of population diversity. This is particularly problematic in the case of multimodal optimization, where the algorithm is prone to getting stuck in local optima. To ensure that followers move effectively toward the discoverer and guarantee global convergence while enhancing population diversity, SGDF-SSA introduces an individual perturbation mechanism with a certain probability using the Cauchy chaotic mutation. The Cauchy mutation perturbs the current global optimum, preventing local optima, and the chaotic mapping enhances population diversity, thereby expanding the search space.

Chaos possesses characteristics of randomness, regularity, and ergodicity, which help improve population diversity. Among the commonly used chaotic mapping systems, the Tent chaotic map demonstrates better uniformity than the Logistic chaos. Therefore, the Tent chaotic system is utilized for the mutation operation. The Tent chaotic value can be defined as:

$$\psi(t+1)=\begin{cases}\dfrac{\psi(t)}{\mu}, & 0\leq\psi(t)<\mu\\ \dfrac{1-\psi(t)}{1-\mu}, & \mu\leq\psi(t)\leq 1\end{cases} \tag{12}$$

where μ is the Tent chaos parameter. Based on the Tent chaotic value, the individual mutation operation is as follows $x_{\text{new}}=UL_{\min}+\psi\times(UP_{\max}-UL_{\min})$, Where $[UL_{\min},UP_{\max}]$ are the lower and upper bounds of the individual's position, and ψ is the Tent chaotic value. Cauchy mutation originates from the Cauchy distribution, a continuous probability distribution. The probability density function of the one-dimensional Cauchy distribution is expressed as $f(x)=\frac{1}{\pi(x^2+1)}$. From the curve of the probability density function of the Cauchy distribution, it is evident that the Cauchy distribution has longer tails at both ends, which gives individuals a higher probability of jumping to better positions and escaping from local optima. Furthermore, the peak at the center (0) of the Cauchy distribution is smaller, and the descent from the peak to zero is smooth, resulting in a more uniform mutation range. In this regard, the Cauchy mutation has a stronger perturbation ability on individuals compared to the standard Gaussian mutation. SGDF-SSA introduces the Cauchy mutation operator into the position mutation of the current optimal solution, using the adjustment function of the Cauchy operator to help the algorithm escape from local optima. $x_{\text{new}}=x_{\text{best}}+\text{Cauchy}(0,1)\cdot x_{\text{best}}$, Where Cauchy (0,1) is the Cauchy operator that follows

the Cauchy distribution. SGDF-SSA performs the choice between chaotic Tent mutation or Cauchy mutation with equal probability, as shown by:

$$x_{new} = \begin{cases} UL_{\min} + \psi \times (UP_{\max} - UL_{\min}), & \text{rand}_1 > 0.5 \\ x_{best} + \text{Cauchy}(0,1) \square x_{best}, & \text{rand}_1 \le 0.5 \end{cases} \tag{13}$$

where rand_1 is a random variable uniformly distributed in the range $[0,1]$. Since SGDF-SSA also needs to maintain a state of rapid convergence while moving toward global optimality, a certain probability of preserving the original state is required. This means that the original position update method of the followers may be kept without perturbing the current global optimal individual. The mutation probability (MP) is defined as:

$$MP = -e^{\left(1-\frac{t}{T_{max}}\right)^2} \tag{14}$$

The specific mutation process is as follows: Define a random variable rand_2. If $\text{rand}_2 < MP$, the algorithm selects not to perform mutation; if $\text{rand}_2 \ge MP$, the mutation operation described is performed.

SGDF-SSA Algorithm Path

The pseudocode of SGDF-SSA based on the improved sparrow algorithm is as follows in Table 1.

Table 1. SGDF-SSA pseudocode

No.	Step Description
1	Initialize parameters: Set population size n, the ratio of protectors (PD) to scouts (SD), vigilance threshold, maximum iterations Max_T, and other parameters such as μ, η, c_1, c_2, R_2, ST, etc.
2	Generate initial population: Use SPM chaotic mapping to initialize the particle swarm, ensuring strong randomness and good traversal properties. Initialize the position and velocity of each particle, and set boundary conditions and velocity constraints.
3	Compute initial fitness values: Determine the positions of the sparrow population with the best and worst fitness values.
4	**for** $i = 1$ to Max_T :
5	**for** $j = 1$ to $PD \times N$:
6	Compute the position of discoverers.
7	**end for**
8	**for** $j = PD \times N$ to N :
9	Apply the Cauchy chaotic mutation mechanism to mutate elite individuals and update the position of followers.
10	**end for**
11	Randomly select $SD \times N$ sparrows as scouts.
12	**for** $j = 1$ to $SD \times N$:
13	Compute the position of protectors.
14	**end for**
15	Compute the fitness values of each individual and sort them to identify the current best and worst individuals.
16	Compute the mutation position of the current best individual.
17	Perform greedy selection.
18	**end for**

Convergence Analysis of the SGDF-SSA Algorithm

By combining the position information of all sparrows in the SGDF-SSA population, a sparrow state can be defined as $s = (x_1, x_2, \dots, x_N)$, where x_i represents the position information of the i-th sparrow, and N denotes the total number of sparrows in the population. The state space of the sparrow population, consisting of all possible states, is given by $S = \{s = (x_1, x_2, \dots, x_N) \mid x_i \in A, 1 \le i \le N\}$, where A represents the feasible solution space.

In SGDF-SSA, the probability of a sparrow moving from position x_i to position x_j can be expressed as $P(T(x_i) = x_j)$. Consequently, the probability of a sparrow state transitioning from s_i to s_j is given by:

$$P(T(s_i) = s_j) = \prod_{k=1}^{N} P(T(x_{ik}) = x_{jk}) \tag{15}$$

From the position update formula in SGDF-SSA, the transition probability of an individual sparrow's position state, $P(T(x(t)) = x(t+1))$, depends only on the sparrow's position at time t, the number of sparrows i, the maximum number of iterations iter $_{\max}$, the current best sparrow position x_p, the current worst sparrow position x_{worst}, and a random number Q. It does not depend on the specific time step t.

Therefore, we can conclude that the sequence of sparrow states $s(t)$ generated by SGDF-SSA exhibits Markovian properties and homogeneity. Furthermore, since the state space S is finite, the sequence of sparrow population states $s(t), t \geq 0$ forms a finite homogeneous Markov chain.

In SGDF-SSA, after each iteration, the fitness value of each sparrow is recalculated. If a sparrow's updated position yields a fitness value superior to the current best sparrow in the population, it replaces the best solution and becomes the new optimal position. As a result, the optimal solution in SGDF-SSA continuously evolves and improves over the course of iterations. This iterative improvement process ensures that SGDF-SSA satisfies Condition of the convergence criteria for stochastic search algorithms, confirming that SGDF-SSA is capable of converging towards an optimal solution.

If we assume that the optimal solution in the population is x^*, then the optimal position state set of the entire sparrow population can be defined as:

$$G = \{s = x | \ f(x) = f(x^*), s \in S\} \tag{16}$$

where $G \subseteq S$. For any optimal position state set s_i and non-optimal position state set s_j, the probability of transitioning from s_i to s_j can be expressed as:

$$P\left(T(s_i) = s_j\right) = \prod_{k=1}^{N} P\left(T(x_{ik}) = x_{jk}\right) \tag{17}$$

Within the optimal position state set G, if at least one position state has reached the optimal solution x^*, then there exists a transition probability $P\left(T(x^*) = x_{jk}\right) = 0$. Substituting this into the above equation yields $P\left(T(s_i) = s_j\right) = 0$. From this result, we can conclude that the optimal position state set G is a closed set in the state space S. Irreducibility of the SGDF-SSA State Space. Now, suppose there exists a nonempty closed set M in the state space S such that $M \cap G = \emptyset$. That is, the states in M do not overlap with the optimal position state set G. For a state $s_i = (x^*)$ in G and a state $s_j = \left(x_{jk}\right)$ in the nonempty closed set M, we have $f(x^*) < f\left(x_{jk}\right)$, Since the transition probability from state s_j to state s_i is greater than zero, it follows that M is not actually a closed set, which contradicts the initial assumption. Therefore, the assumption does not hold, meaning that the only closed set in the SGDF-SSA state space S is the optimal position state set G. This result indicates that the Markov chain of the sparrow population state in SGDF-SSA is irreducible, as every state can eventually reach G.

Furthermore, if there exists a nonempty closed set E and no other nonempty closed set O such that $E \cap O = \emptyset$, then for any $j \in E$, $\lim_{k\to\infty} P(x_k = j) = \pi_j$ and for any $j \notin E$, $\lim_{k\to\infty} P(x_k = j) = 0$,This leads to the conclusion that as the number of SGDF-SSA iterations approaches infinity, the sparrow population state will inevitably enter the optimal state set G. Consequently, the probability that SGDF-SSA fails to find the global optimal solution is zero. Thus, SGDF-SSA satisfies Condition 2 of the convergence criteria for stochastic search algorithms. Since SGDF-SSA has already been shown to satisfy Condition 1, it meets both necessary conditions for global convergence. According to the definition of the convergence theorem, we can conclude that SGDF-SSA is a globally convergent algorithm.

RESULTS AND DISCUSSION

Experimental design

Table 2. Comparative experimental results of SGDF-SSA and SSA

Function	Categories	Dimension 50		Dimension 100	
		SGDF-SSA	SSA	SGDF-SSA	SSA
F1	Average value	6.89E+03	2.08E+04	1.24E+04	1.36E+04
	Standard deviation	6.08E+03	2.01E+04	1.61E+04	1.71E+04
F2	Average value	4.36E+61	3.45E+32	4.19E+58	1.72E+15
	Standard deviation	2.39E+62	1.89E+33	2.29E+59	8.07E+15
F3	Average value	2.26E+03	7.81E+02	3.39E+02	3.13E+02
	Standard deviation	1.08E+03	2.18E+02	2.22E+01	9.87E+00
F4	Average value	5.51E+02	6.68E+02	5.49E+02	6.46E+02
	Standard deviation	5.15E+01	4.41E+01	4.34E+01	3.27E+01
F5	Average value	1.02E+03	1.08E+03	1.02E+03	1.02E+03
	Standard deviation	5.81E+01	1.01E+02	5.69E+01	8.64E+01
F6	Average value	6.10E+02	6.28E+02	6.07E+02	6.23E+02
	Standard deviation	3.33E+00	6.32E+00	3.91E+00	5.76E+00
F7	Average value	1.60E+03	1.56E+03	1.55E+03	1.51E+03
	Standard deviation	1.09E+02	1.18E+02	9.17E+01	1.17E+02
F8	Average value	1.30E+03	1.34E+03	1.29E+03	1.30E+03
	Standard deviation	3.80E+01	9.08E+01	5.22E+01	8.28E+01
F9	Average value	1.77E+04	2.50E+04	1.77E+04	2.26E+04
	Standard deviation	7.20E+03	3.56E+03	7.53E+03	4.89E+03
F10	Average value	1.34E+04	1.47E+04	1.37E+04	1.50E+04
	Standard deviation	1.11E+03	1.16E+03	1.67E+03	1.55E+03
F11	Average value	1.85E+03	2.41E+03	1.84E+03	2.54E+03
	Standard deviation	1.18E+02	1.80E+02	1.68E+02	2.29E+02
F12	Average value	1.90E+07	2.31E+07	1.87E+07	1.24E+07
	Standard deviation	1.48E+07	9.67E+06	1.80E+07	6.46E+06
F13	Average value	1.37E+04	3.91E+04	2.83E+04	3.27E+04
	Standard deviation	9.77E+03	1.17E+04	6.77E+04	1.24E+04
F14	Average value	2.21E+05	4.86E+05	1.03E+05	3.27E+05
	Standard deviation	1.09E+05	2.44E+05	4.66E+04	1.56E+05
F15	Average value	6.03E+03	2.03E+04	6.59E+03	1.65E+04
	Standard deviation	7.03E+03	1.31E+04	7.02E+03	1.10E+04
F16	Average value	4.91E+03	5.26E+03	4.95E+03	4.88E+03
	Standard deviation	7.04E+02	6.58E+02	6.34E+02	5.72E+02
F17	Average value	4.68E+03	5.26E+03	4.71E+03	5.12E+03
	Standard deviation	4.87E+02	6.12E+02	5.20E+02	5.77E+02
F18	Average value	5.82E+05	1.44E+06	5.66E+05	9.12E+05
	Standard deviation	2.83E+05	6.19E+05	9.63E+05	4.59E+05
F19	Average value	7.63E+03	2.75E+04	7.64E+03	2.46E+04
	Standard deviation	5.87E+03	1.43E+04	6.46E+03	1.38E+04
F20	Average value	4.33E+03	4.91E+03	4.28E+03	4.66E+03
	Standard deviation	4.95E+02	5.05E+02	5.27E+02	5.53E+02
F21	Average value	2.82E+03	2.87E+03	2.77E+03	2.83E+03
	Standard deviation	4.29E+01	9.02E+01	5.35E+01	7.96E+01
F22	Average value	2.30E+03	1.75E+04	2.30E+03	1.67E+04
	Standard deviation	1.24E+00	1.20E+03	1.42E+01	1.34E+03
F23	Average value	3.25E+03	3.25E+03	3.23E+03	3.21E+03
	Standard deviation	5.56E+01	5.76E+01	9.85E+01	6.49E+01
F24	Average value	3.86E+03	3.81E+03	3.84E+03	3.79E+03
	Standard deviation	1.12E+02	7.71E+01	8.87E+01	6.29E+01
F25	Average value	3.27E+03	3.29E+03	3.27E+03	3.28E+03
	Standard deviation	3.72E+01	6.86E+01	3.04E+01	6.93E+01
F26	Average value	1.20E+04	1.14E+04	1.19E+04	1.11E+04
	Standard deviation	1.10E+03	6.93E+02	1.19E+03	1.01E+03
F27	Average value	3.20E+03	3.49E+03	3.20E+03	3.48E+03
	Standard deviation	2.05E-04	4.88E+01	2.38E-04	5.69E+01
F28	Average value	3.30E+03	3.38E+03	3.30E+03	3.38E+03
	Standard deviation	1.88E-04	3.81E+01	2.43E-04	3.24E+01
F29	Average value	6.13E+03	6.96E+03	6.08E+03	6.58E+03
	Standard deviation	4.81E+02	6.33E+02	4.67E+02	4.39E+02
F30	Average value	1.11E+04	7.19E+04	9.17E+03	3.57E+04
	Standard deviation	8.13E+03	2.19E+04	8.35E+03	9.97E+03
Average value		1.25	1.85	1.36	1.74
+/-/ =		~	23/3/4	~	18/3/9

To verify the performance of SGDF-SSA, this study conducts simulation experiments using 29 benchmark functions from the CEC2017 test set, covering four types of functions: unimodal functions (F1, F3), multimodal functions (F4-F10), hybrid functions (F11-F19), and composite functions (F20~F29). To investigate the performance improvement of SGDF-SSA over SSA, this section presents scalability testing experiments comparing SGDF-SSA and SSA. In the experiment, the primary goal is to set the dimensions of the benchmark functions to 50 and 100, create problems with different complexities, and compare SGDF-SSA and SSA on these functions. Table 2 presents the mean, standard deviation, and results of the Wilcoxon signed-rank test for SGDF-SSA and SSA across 30 independent comparison experiments. "AVG" represents the average value of the function results after 30 independent runs, "STD" denotes the corresponding standard deviation, "+" indicates that SGDF-SSA outperforms SSA, "-" indicates that SGDF-SSA performs worse than SSA, and "=" means SGDF-SSA and SSA have equivalent performance. When the dimension of the benchmark functions is 50, SGDF-SSA achieved the lowest mean in 24 functions and the lowest standard deviation in 22 functions. Based on the analysis of the mean and standard deviation, SGDF-SSA achieved the smallest mean and standard deviation in 21 and 17 functions, respectively. When the dimension of the benchmark functions is 100, SGDF-SSA significantly outperforms SSA in discovering the ideal solution, as evidenced by its performance in obtaining the smallest mean and standard deviation under different complexities. Additionally, SGDF-SSA outperformed SSA in all three functions tested with the Wilcoxon signedank test, demonstrating that SGDF-SSA is significantly superior to SSA in finding the ideal solution.

Due to the different search strategies, the optimization performance of various algorithms is different. To show that the proposed improved algorithm is superior to other algorithms, four metaheuristics are selected for comparison. These algorithms include particle Swarm optimization (PSO), Grey Wolf Optimizer (GWO), Dragonfly Algorithm (DBO), and Salp Swarm Algorithm (SSA). The following are the evaluation index graphs of the test functions of different algorithms, as shown in Figure 3, Figure 4 and the table of the algorithm result index in Table 3.

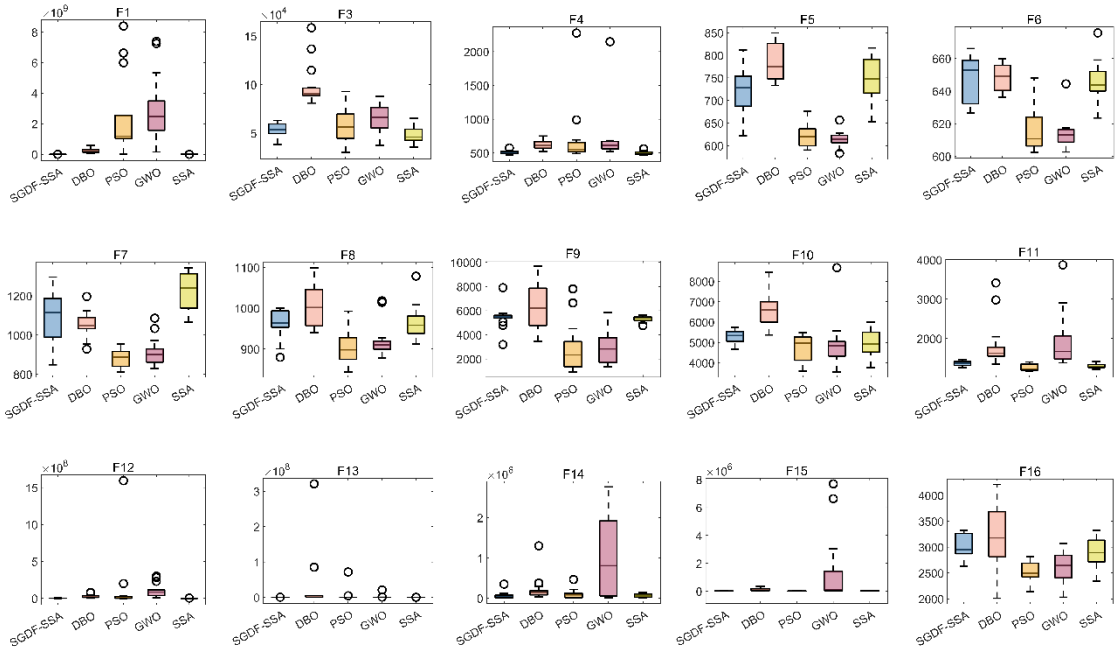

Figure 3. Box plots of F1-F16 test function metrics

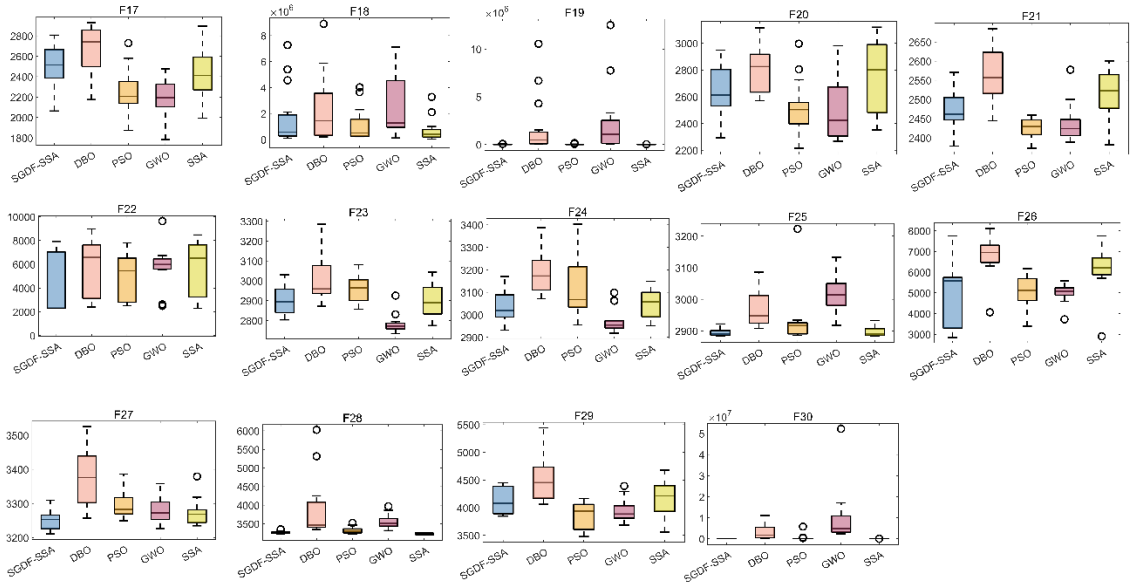

Figure 4. Box plots of F17-F30 test function metric

Table 3. Test function evaluation index plots for different algorithms

Function	Evaluation method	DBO	GWO	PSO	SSA	SGDF-SSA
F1	Minimum value	1.37E+03	4.18E+05	1.13E+05	1.10E+02	1.45E+08
F1	Standard deviation	5.26E+04	2.79E+08	9.96E+08	6.34E+03	1.96E+09
F1	Average value	4.34E+04	2.67E+08	5.77E+08	5.58E+03	3.13E+09
F1	median	2.71E+04	1.94E+08	7.84E+05	3.27E+03	2.93E+09
F1	Worst value	2.64E+05	1.39E+09	4.69E+09	2.10E+04	7.73E+09
F3	Minimum value	3.95E+04	5.84E+04	3.68E+04	3.45E+04	4.37E+04
F3	Standard deviation	5.98E+03	1.55E+04	2.44E+04	7.98E+03	1.02E+04
F3	Average value	5.13E+04	9.22E+04	7.12E+04	4.86E+04	6.23E+04
F3	median	5.03E+04	9.13E+04	7.13E+04	4.81E+04	6.25E+04
F3	Worst value	6.30E+04	1.45E+05	1.39E+05	6.68E+04	9.35E+04
F4	Minimum value	4.70E+02	5.26E+02	4.93E+02	4.70E+02	5.31E+02
F4	Standard deviation	4.07E+01	1.60E+02	2.57E+02	2.35E+01	1.24E+02
F4	Average value	5.17E+02	6.78E+02	6.58E+02	5.04E+02	6.48E+02
F4	median	5.15E+02	6.41E+02	5.43E+02	5.10E+02	6.21E+02
F4	Worst value	6.97E+02	1.32E+03	1.61E+03	5.80E+02	1.22E+03
F5	Minimum value	6.08E+02	6.64E+02	5.63E+02	6.08E+02	5.88E+02
F5	Standard deviation	5.05E+01	5.20E+01	3.37E+01	5.25E+01	4.46E+01
F5	Average value	7.23E+02	7.46E+02	6.16E+02	7.55E+02	6.35E+02
F5	median	7.32E+02	7.31E+02	6.09E+02	7.60E+02	6.27E+02
F5	Worst value	7.94E+02	8.48E+02	7.08E+02	8.22E+02	7.73E+02
F6	Minimum value	6.20E+02	6.23E+02	6.02E+02	6.18E+02	6.05E+02
F6	Standard deviation	1.44E+01	1.37E+01	1.19E+01	9.78E+00	5.31E+00
F6	Average value	6.42E+02	6.50E+02	6.16E+02	6.49E+02	6.14E+02
F6	median	6.46E+02	6.49E+02	6.14E+02	6.49E+02	6.13E+02
F6	Worst value	6.65E+02	6.77E+02	6.56E+02	6.66E+02	6.27E+02
F7	Minimum value	8.85E+02	8.65E+02	8.12E+02	9.60E+02	8.28E+02
F7	Standard deviation	1.11E+02	7.05E+01	4.74E+01	1.06E+02	5.89E+01
F7	Average value	1.07E+03	1.01E+03	8.88E+02	1.23E+03	9.12E+02
F7	median	1.06E+03	1.00E+03	8.85E+02	1.25E+03	9.05E+02
F7	Worst value	1.34E+03	1.17E+03	9.77E+02	1.35E+03	1.07E+03
F8	Minimum value	8.98E+02	9.56E+02	8.68E+02	9.22E+02	8.55E+02
F8	Standard deviation	3.08E+01	4.59E+01	2.71E+01	2.85E+01	4.04E+01
F8	Average value	9.63E+02	1.02E+03	9.15E+02	9.71E+02	9.14E+02

F8	median	9.69E+02	1.01E+03	9.15E+02	9.71E+02	9.02E+02
F8	Worst value	1.01E+03	1.11E+03	9.58E+02	1.03E+03	1.04E+03
F9	Minimum value	3.84E+03	2.63E+03	9.68E+02	4.56E+03	1.60E+03
F9	Standard deviation	6.86E+02	2.21E+03	2.74E+03	2.13E+02	1.02E+03
F9	Average value	5.25E+03	6.09E+03	3.22E+03	5.31E+03	2.70E+03
F9	median	5.26E+03	5.74E+03	1.81E+03	5.37E+03	2.45E+03
F9	Worst value	6.92E+03	1.19E+04	1.07E+04	5.56E+03	6.76E+03
F10	Minimum value	4.11E+03	4.32E+03	3.51E+03	4.49E+03	3.41E+03
F10	Standard deviation	6.80E+02	1.21E+03	8.94E+02	4.60E+02	1.73E+03
F10	Average value	5.38E+03	6.63E+03	4.87E+03	5.42E+03	5.41E+03
F10	median	5.27E+03	6.38E+03	4.54E+03	5.36E+03	4.77E+03
F10	Worst value	7.10E+03	9.29E+03	7.56E+03	6.21E+03	9.24E+03
F11	Minimum value	1.17E+03	1.36E+03	1.17E+03	1.17E+03	1.30E+03
F11	Standard deviation	1.04E+02	1.44E+03	2.86E+02	6.82E+01	9.72E+02
F11	Average value	1.37E+03	2.22E+03	1.35E+03	1.30E+03	2.31E+03
F11	median	1.34E+03	1.74E+03	1.29E+03	1.30E+03	1.97E+03
F11	Worst value	1.60E+03	8.32E+03	2.71E+03	1.49E+03	4.66E+03
F12	Minimum value	4.06E+05	3.37E+06	5.57E+05	1.08E+05	6.94E+06
F12	Standard deviation	2.08E+06	4.55E+07	4.45E+08	8.80E+05	9.59E+07
F12	Average value	2.69E+06	4.18E+07	1.26E+08	8.88E+05	8.69E+07
F12	median	2.23E+06	2.25E+07	8.85E+06	5.83E+05	5.47E+07
F12	Worst value	8.88E+06	1.88E+08	2.40E+09	3.85E+06	4.30E+08
F13	Minimum value	1.14E+04	6.30E+04	3.09E+03	3.22E+03	6.88E+04
F13	Standard deviation	4.29E+04	7.61E+07	1.35E+06	8.15E+05	3.28E+07
F13	Average value	5.67E+04	3.42E+07	4.80E+05	1.70E+05	1.16E+07
F13	median	3.68E+04	4.58E+06	3.05E+04	1.32E+04	2.14E+05
F13	Worst value	1.74E+05	3.65E+08	4.51E+06	4.48E+06	1.46E+08
F14	Minimum value	5.05E+03	7.98E+03	4.67E+03	2.70E+03	7.62E+03
F14	Standard deviation	4.80E+04	8.57E+05	6.45E+04	4.81E+04	6.06E+05
F14	Average value	5.41E+04	3.95E+05	8.10E+04	6.45E+04	4.97E+05
F14	median	3.93E+04	1.40E+05	6.43E+04	6.06E+04	2.03E+05
F14	Worst value	2.03E+05	4.65E+06	1.99E+05	1.92E+05	2.37E+06
F15	Minimum value	2.12E+03	4.89E+03	2.14E+03	2.36E+03	2.15E+04
F15	Standard deviation	3.85E+04	9.31E+05	1.77E+04	1.46E+04	1.53E+06
F15	Average value	2.43E+04	2.75E+05	1.97E+04	1.50E+04	9.46E+05
F15	median	1.05E+04	5.66E+04	1.27E+04	8.74E+03	1.29E+05
F15	Worst value	1.71E+05	5.17E+06	6.13E+04	4.70E+04	5.48E+06
F16	Minimum value	2.25E+03	2.52E+03	2.26E+03	2.17E+03	2.06E+03
F16	Standard deviation	3.72E+02	4.08E+02	2.76E+02	3.89E+02	3.64E+02
F16	Average value	2.96E+03	3.48E+03	2.72E+03	2.96E+03	2.71E+03
F16	median	2.98E+03	3.49E+03	2.74E+03	3.05E+03	2.63E+03
F16	Worst value	3.81E+03	4.26E+03	3.15E+03	3.60E+03	3.50E+03
F17	Minimum value	2.04E+03	1.95E+03	1.79E+03	1.89E+03	1.81E+03
F17	Standard deviation	2.93E+02	3.09E+02	2.12E+02	3.06E+02	1.89E+02
F17	Average value	2.49E+03	2.67E+03	2.17E+03	2.47E+03	2.17E+03
F17	median	2.47E+03	2.69E+03	2.18E+03	2.42E+03	2.18E+03
F17	Worst value	3.15E+03	3.24E+03	2.56E+03	3.21E+03	2.69E+03
F18	Minimum value	5.75E+04	6.80E+04	7.05E+04	8.18E+04	6.13E+04
F18	Standard deviation	1.29E+06	5.56E+06	1.27E+06	1.07E+06	7.12E+06
F18	Average value	1.04E+06	3.67E+06	1.27E+06	7.83E+05	4.62E+06
F18	median	6.31E+05	1.14E+06	6.65E+05	4.37E+05	2.34E+06
F18	Worst value	5.58E+06	2.40E+07	4.48E+06	5.17E+06	2.84E+07
F19	Minimum value	2.22E+03	1.10E+04	2.18E+03	2.19E+03	9.35E+03
F19	Standard deviation	1.15E+04	1.12E+07	5.34E+04	1.35E+04	5.90E+06
F19	Average value	1.08E+04	5.99E+06	3.11E+04	9.00E+03	2.20E+06
F19	median	5.53E+03	2.63E+06	8.09E+03	4.31E+03	5.05E+05
F19	Worst value	5.67E+04	5.81E+07	2.01E+05	5.59E+04	3.23E+07

F20	Minimum value	2.34E+03	2.23E+03	2.17E+03	2.42E+03	2.29E+03
F20	Standard deviation	2.00E+02	2.20E+02	1.82E+02	2.36E+02	1.39E+02
F20	Average value	2.73E+03	2.81E+03	2.46E+03	2.73E+03	2.51E+03
F20	median	2.70E+03	2.82E+03	2.43E+03	2.66E+03	2.48E+03
F20	Worst value	3.20E+03	3.26E+03	2.90E+03	3.36E+03	2.80E+03
F21	Minimum value	2.41E+03	2.46E+03	2.35E+03	2.38E+03	2.35E+03
F21	Standard deviation	5.07E+01	5.20E+01	3.63E+01	6.14E+01	3.43E+01
F21	Average value	2.48E+03	2.55E+03	2.42E+03	2.53E+03	2.42E+03
F21	median	2.48E+03	2.55E+03	2.41E+03	2.53E+03	2.41E+03
F21	Worst value	2.63E+03	2.65E+03	2.52E+03	2.64E+03	2.49E+03
F22	Minimum value	2.30E+03	2.40E+03	2.30E+03	2.30E+03	2.52E+03
F22	Standard deviation	2.18E+03	2.57E+03	1.79E+03	1.82E+03	1.81E+03
F22	Average value	3.69E+03	5.57E+03	3.94E+03	6.07E+03	5.19E+03
F22	median	2.30E+03	6.65E+03	2.91E+03	6.63E+03	5.66E+03
F22	Worst value	7.64E+03	9.46E+03	6.79E+03	8.56E+03	9.05E+03
F23	Minimum value	2.78E+03	2.87E+03	2.76E+03	2.78E+03	2.70E+03
F23	Standard deviation	6.31E+01	9.23E+01	1.02E+02	9.39E+01	4.42E+01
F23	Average value	2.87E+03	2.98E+03	2.95E+03	2.93E+03	2.78E+03
F23	median	2.87E+03	2.97E+03	2.93E+03	2.94E+03	2.79E+03
F23	Worst value	3.02E+03	3.27E+03	3.20E+03	3.10E+03	2.94E+03
F24	Minimum value	2.94E+03	3.02E+03	2.96E+03	2.92E+03	2.89E+03
F24	Standard deviation	8.87E+01	8.02E+01	7.23E+01	8.51E+01	6.53E+01
F24	Average value	3.05E+03	3.16E+03	3.10E+03	3.06E+03	2.97E+03
F24	median	3.02E+03	3.14E+03	3.09E+03	3.05E+03	2.93E+03
F24	Worst value	3.26E+03	3.33E+03	3.25E+03	3.24E+03	3.11E+03
F25	Minimum value	2.88E+03	2.89E+03	2.89E+03	2.88E+03	2.93E+03
F25	Standard deviation	9.51E+00	6.94E+01	5.09E+01	1.46E+01	5.20E+01
F25	Average value	2.89E+03	3.00E+03	2.93E+03	2.89E+03	3.01E+03
F25	median	2.89E+03	2.99E+03	2.92E+03	2.89E+03	3.01E+03
F25	Worst value	2.93E+03	3.17E+03	3.10E+03	2.94E+03	3.13E+03
F26	Minimum value	2.80E+03	3.81E+03	2.82E+03	2.90E+03	3.77E+03
F26	Standard deviation	1.59E+03	8.81E+02	9.80E+02	9.12E+02	3.63E+02
F26	Average value	4.51E+03	6.99E+03	4.93E+03	6.55E+03	4.96E+03
F26	median	4.29E+03	6.99E+03	4.95E+03	6.66E+03	4.95E+03
F26	Worst value	7.44E+03	8.75E+03	6.85E+03	7.99E+03	5.56E+03
F27	Minimum value	3.21E+03	3.26E+03	3.22E+03	3.20E+03	3.24E+03
F27	Standard deviation	3.09E+01	7.69E+01	6.23E+01	3.94E+01	3.44E+01
F27	Average value	3.25E+03	3.34E+03	3.28E+03	3.26E+03	3.28E+03
F27	median	3.25E+03	3.33E+03	3.27E+03	3.26E+03	3.27E+03
F27	Worst value	3.33E+03	3.64E+03	3.50E+03	3.38E+03	3.36E+03
F28	Minimum value	3.21E+03	3.26E+03	3.22E+03	3.20E+03	3.31E+03
F28	Standard deviation	3.16E+01	6.48E+02	2.24E+02	2.26E+01	1.49E+02
F28	Average value	3.26E+03	3.58E+03	3.35E+03	3.23E+03	3.49E+03
F28	median	3.26E+03	3.43E+03	3.28E+03	3.22E+03	3.45E+03
F28	Worst value	3.36E+03	6.72E+03	4.44E+03	3.27E+03	3.94E+03
F29	Minimum value	3.70E+03	3.61E+03	3.50E+03	3.75E+03	3.61E+03
F29	Standard deviation	2.71E+02	4.27E+02	2.61E+02	2.70E+02	1.66E+02
F29	Average value	3.11E+03	4.42E+03	3.88E+03	4.19E+03	3.98E+03
F29	median	4.04E+03	4.39E+03	3.82E+03	4.12E+03	3.97E+03
F29	Worst value	3.80E+03	5.34E+03	4.50E+03	4.86E+03	4.28E+03
F30	Minimum value	6.95E+03	2.15E+04	7.44E+03	6.80E+03	8.94E+05
F30	Standard deviation	7.54E+04	8.23E+06	1.80E+06	9.36E+03	8.36E+06
F30	Average value	3.81E+04	4.39E+06	4.18E+05	1.74E+04	1.04E+07
F30	median	2.18E+04	1.55E+06	2.40E+04	1.59E+04	7.66E+06
F30	Worst value	4.31E+05	4.17E+07	9.91E+06	5.19E+04	3.17E+07

The improved algorithm has smaller difference value, more stable optimization ability, less special values in the data set, and stronger convergence compared with other algorithms. We calculated all the functions by running the optimization program and obtained the iterative line chart, as shown in Figure 5 and Figure 6.

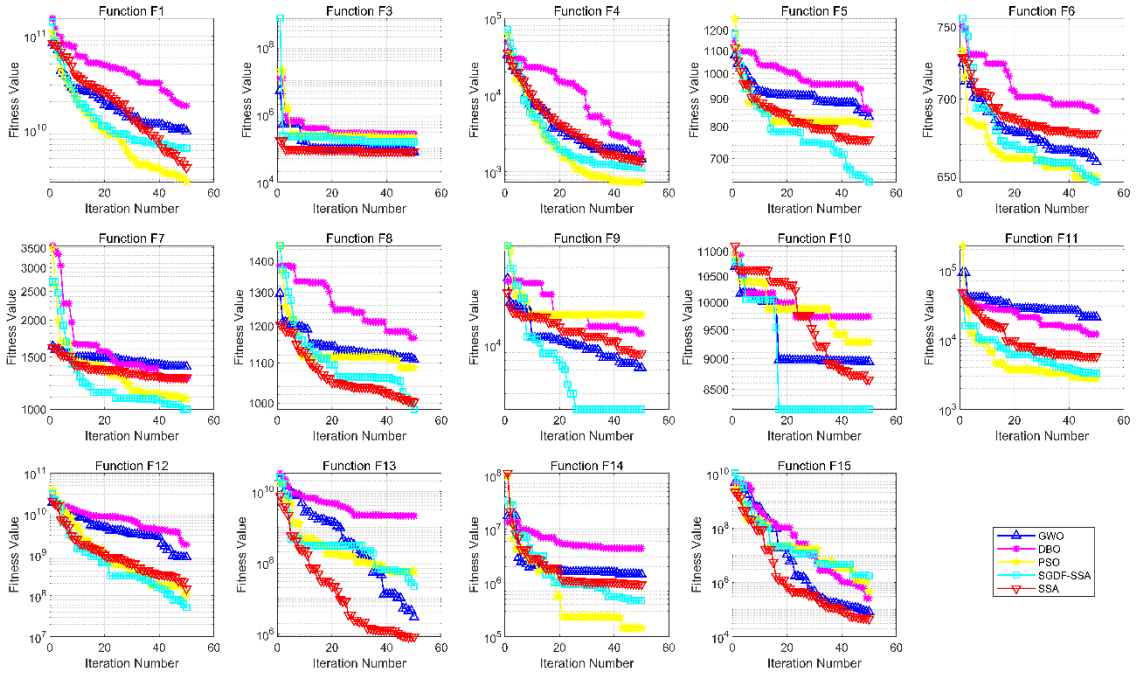

Figure 5. Line chart of function iteration data for F1-F15

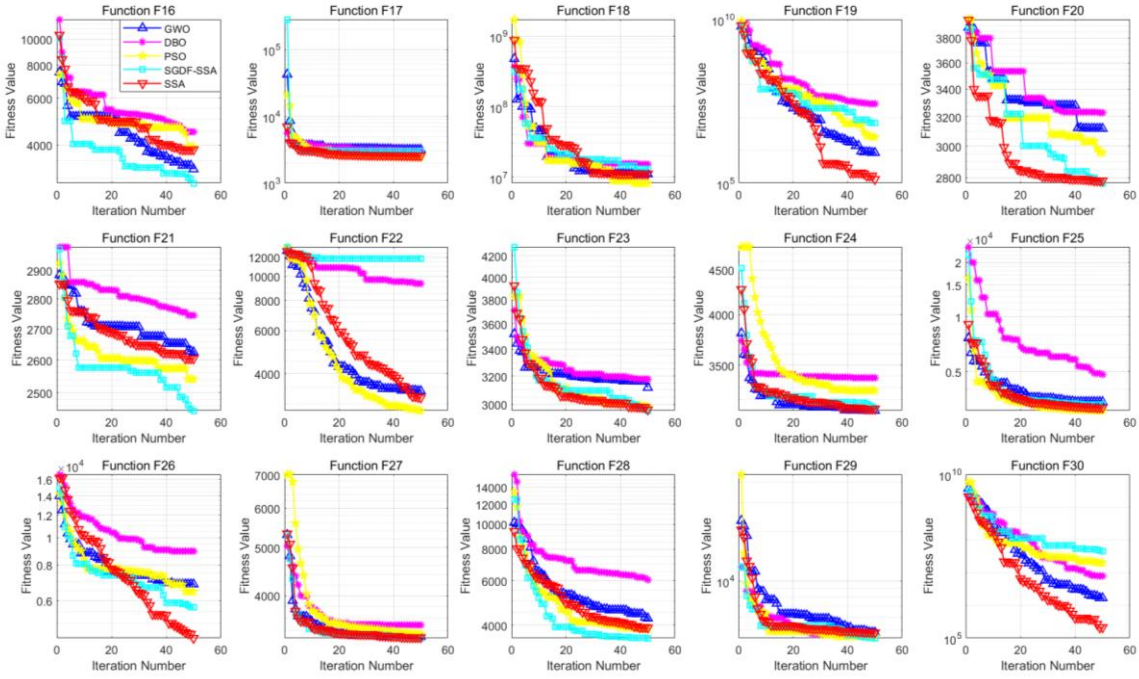

Figure 6. Line chart of function iteration data for F16-F30

For the above images, the effect ranking of the final convergence state, as shown in Table 4, and the average ranking statistics, as shown in Table 5, can be used to determine the ranking of the convergence ability of the algorithm.

Table 4. Ranking of different algorithms on the 29 benchmark functions

Functions	GWO	DBO	PSO	SSA	SGDF-SSA
F1	4	5	3	1	2
F3	5	5	4	1	3
F4	5	4	1	3	2
F5	4	5	3	1	2
F6	5	4	2	3	1
F7	4	3	1	5	2
F8	3	5	4	2	1
F9	5	4	3	2	1
F10	1	3	5	4	2
F11	5	4	2	3	1
F12	5	4	1	2	3
F13	5	2	4	1	3
F14	3	5	2	4	1
F15	1	2	4	3	5
F16	4	5	3	1	2
F17	2	3	1	4	5
F18	2	5	4	1	3
F19	4	3	5	1	2
F20	1	5	2	4	3
F21	4	5	3	2	1
F22	3	1	5	4	2
F23	2	5	4	3	1
F24	4	5	3	2	1
F25	5	4	3	2	1
F26	1	5	3	4	2
F27	4	5	3	1	2
F28	4	5	2	3	1
F29	5	4	2	3	1
F30	3	2	4	1	5

Table 5. Ranking statistics and average ranking of different algorithms on benchmark functions

Algorithm	average	ranking 1	ranking 2	ranking 3	ranking 4	ranking 5	rank average
GWO	3.55	4	3	4	9	9	4
DBO	4.03	1	3	4	7	14	5
PSO	2.97	4	6	9	7	3	3
SSA	2.45	9	6	7	6	1	2
SGDF-SSA	2.10	11	10	5	0	3	1

Table 4 shows the test results of different algorithms, and Table 5 shows the ranking results of each algorithm. SGDF-SSA ranks in the top three of the 26 benchmark functions, among which the first rank accounts for 11 times and the second rank accounts for 10 times, which indicates that the optimization ability of SGDF-SSA is significantly better than that of these basic algorithms. the Wilcoxon signedrank test analysis results are presented, and SGDF-SSA ranks first with 2.10, which again shows that the optimization ability of SGDF-SSA is much better than these base algorithms.

Application of Image Leaf Segmentation

In the leaf image processing, data set is relatively common data set of Plant Doc (https://github.com/pratikkayal/PlantDoc-Object-Detection-Dataset), This is a dataset of 2,598 images of 13 plant species used for image classification and object detection. We select 8 representative images from them for testing. The size of the test image is 512×512 and the histogram is shown in Figure 7. To evaluate the performance of the algorithm, the main parameters considered in Table 6:

Table 6. Parameter Settings for different algorithms

PSO	Group size	20
	Number of iterations	300
	Cognitive, social, and Neighbourhood acceleration	3,2,1
	Lower and upper bounds	1 and 256
	Error targets and maximum trial limits	1e-7 and 500
	Initial velocity weight value	0.95
	Velocity weight value at the end of PSO iteration	0.4
	Proportion of the maximum number of iterations for which W varies linearly	0.7
	Maximum speed step, shrinkage factor, and neighbourhood size	1
	Global minimum	0
GWO	Number of population	60
	Maximum number of iterations	200
DBO	The strength of the local search	0.1
	Balancing the weights of different search strategies	0.1
	Diversity of solutions	0.3
	Step size control parameter	0.3
	Population size	0.5
SSA	The number of sparrow populations	50
	Number of discoverers	10
	Number of followers	40
	Number of warning	10
	Initialize the upper and lower bounds of the population position	[-3,3]
	Dimension of the fitness function	337
	Maximum number of iterations	30
	Warning value	random
	Safety threshold	0.8
SSA/SGDF-SSA	Number of fireflies	20
	Maximum number of iterations	300
	Initial firefly size	20
	Randomization parameter	0.5
	Optical absorption coefficient at the source	1
	Inertial attraction	0.2

Since optimization algorithms exhibit stochastic behavior, all experiments are repeated 20 times for each image and threshold level to ensure high-fidelity evaluation. Figure 8 provides a visual assessment of the overall segmentation results. As confirmed by the experiments in Table 7, the proposed algorithm achieves the highest PSNR and the lowest MSE values. SGDF-SSA delivers reasonable, reliable, and the fastest output while maintaining the best segmentation quality. SSA also exhibits the shortest computation time at higher threshold levels (m = 5 and 8). SGDF-SSA performs accurately across both lower and higher threshold levels, offering superior segmentation results. This indicates that as the threshold level increases, the quantitative segmentation results improve. Therefore, SGDF-SSA is suitable for applications requiring rapid segmentation of leaf images or other natural images. The results demonstrate that, compared to other competing algorithms, SGDF-SSA achieves higher precision and accuracy. Segmented images based on SGDF-SSA exhibit detailed and accurate image information across all threshold levels.

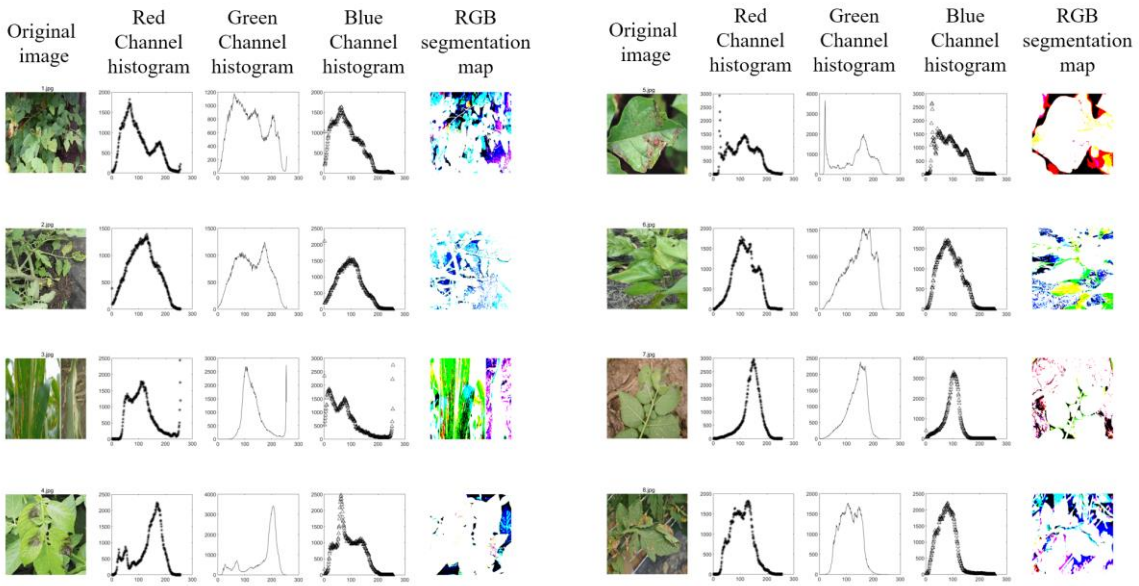

Figure 7. Image selection and multi-channel histogram

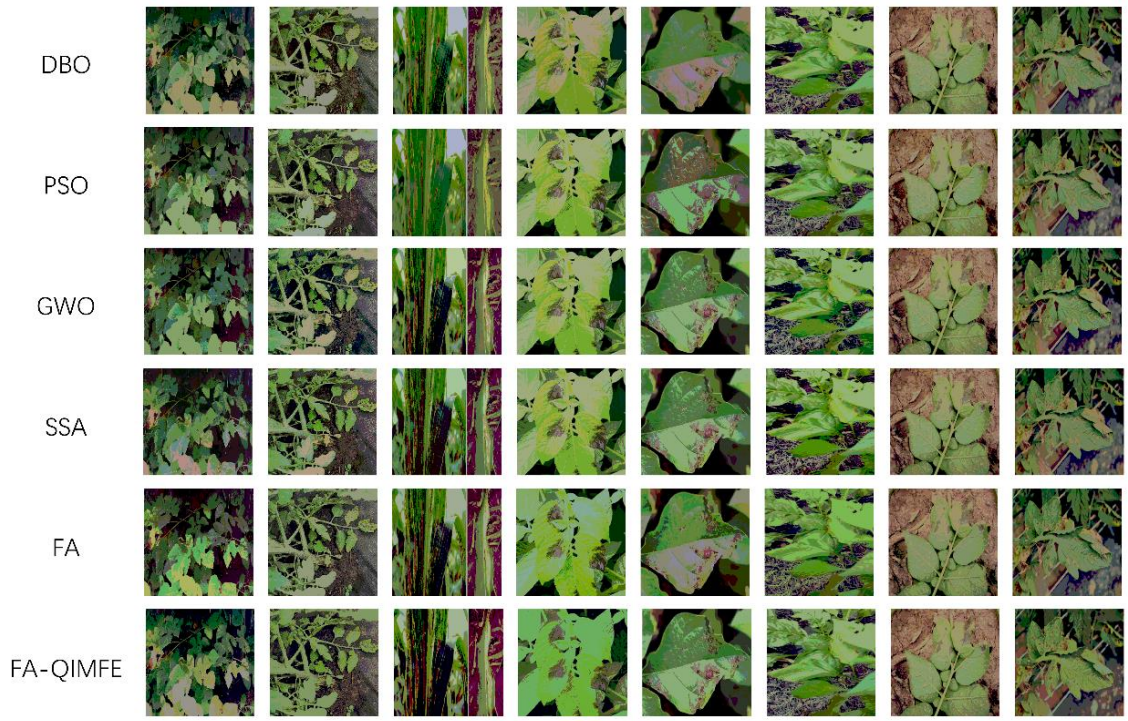

Figure 8. Test image segmentation results

Table 7. Comparison of PSNR and MSE of different algorithms

Image	Threshold	DBO		PSO		GWO		SSA		SGDF-SSA	
		PSNR	MSE	PSNR	MSE	PSNR	MSE	PSNR	MSE	PSNR	MSE
1	2	8.59	8711.89	9.66	5952.26	12.26	5872.07	12.06	5521.99	13.68	3364.73
	5	15.80	2089.47	16.51	1275.35	16.38	1853.10	16.78	1407.31	18.19	1203.02
	8	22.21	409.03	22.54	383.20	20.92	666.43	22.80	359.37	23.36	347.18
	12	25.84	279.95	24.42	203.08	25.33	133.32	27.02	157.20	27.81	125.04
2	2	13.18	3114.72	12.64	3716.66	12.42	4324.27	11.17	4937.06	13.81	3081.39
	5	20.21	709.62	19.66	640.73	18.32	906.65	19.54	694.31	20.35	672.46
	8	23.92	295.94	23.80	276.09	23.30	270.39	23.06	288.32	23.10	266.44
	12	27.54	158.61	25.99	158.02	25.51	214.29	27.45	128.86	26.16	118.77
3	2	12.44	3747.35	12.00	3760.56	13.18	3398.79	11.89	4419.79	14.13	3874.80
	5	19.00	936.93	19.41	667.32	19.08	746.92	20.45	682.37	19.09	604.00
	8	21.77	348.12	23.00	362.15	23.08	385.30	21.94	362.26	22.89	232.74
	12	27.53	153.49	27.44	190.74	25.27	156.36	27.07	151.28	25.88	132.34
4	2	14.12	2755.49	15.79	2206.54	13.74	3036.00	12.94	2998.45	14.80	2424.55
	5	20.06	664.58	22.36	483.20	18.75	773.13	18.74	1237.54	21.68	431.50
	8	20.30	573.76	21.56	401.76	19.64	646.89	20.44	510.80	21.79	383.94
	12	24.44	259.76	25.69	141.39	25.35	209.19	24.20	337.56	27.79	110.19
5	2	12.39	555.62	13.81	3130.08	12.42	3581.29	12.05	4033.74	14.75	2691.46
	5	21.14	324.74	18.43	778.92	19.70	716.73	20.72	683.91	19.80	583.83
	8	22.92	321.89	23.49	339.67	21.92	364.86	23.92	290.52	25.30	248.90
	12	26.27	133.41	25.47	143.96	25.94	174.92	24.66	183.45	26.83	122.79
6	2	12.94	3914.54	12.42	3517.01	13.61	3644.66	13.56	3233.37	14.12	3024.32
	5	19.88	846.49	20.26	3503.42	18.83	697.14	18.98	764.54	20.07	697.28
	8	22.29	397.64	23.86	321.54	22.97	345.39	23.67	290.14	23.79	252.77
	12	26.40	108.57	27.32	140.45	24.96	199.73	26.13	149.84	27.29	100.35
7	2	12.88	3078.89	13.48	2816.68	14.60	2070.78	13.86	2469.81	15.93	1186.63
	5	20.07	759.62	20.48	726.27	19.43	774.27	19.40	744.90	19.02	635.14
	8	22.60	347.69	23.85	333.65	22.52	443.43	21.98	318.20	22.67	333.03
	12	26.02	203.42	26.14	153.38	23.81	212.31	27.69	135.22	27.56	99.92
8	2	10.96	4976.56	12.85	3917.21	13.02	2968.50	10.97	5017.98	12.80	3865.93
	5	19.00	767.80	17.14	1420.86	19.80	809.56	19.71	716.50	21.60	748.00
	8	21.84	651.38	23.79	290.70	23.68	331.89	25.78	229.56	25.70	215.38
	12	27.32	162.81	27.03	113.85	25.66	178.29	27.41	152.34	29.24	108.96

Table 8 SSIM and FSIM comparison of different algorithms

Image	Threshold	DBO		PSO		GWO		SSA		SGDF-SSA	
		SSIM	FSIM	SSIM	FSIM	SSIM	FSIM	SSIM	FSIM	SSIM	FSIM
1	2	0.795	0.567	0.891	0.627	0.871	0.568	0.740	0.566	0.900	0.666
	5	0.945	0.773	0.967	0.749	0.955	0.791	0.926	0.813	0.980	0.775
	8	0.995	0.908	0.952	0.909	0.961	0.868	0.985	0.893	0.994	0.895
	12	0.961	0.929	0.967	0.938	0.977	0.941	0.964	0.947	0.971	0.974
2	2	0.914	0.761	0.890	0.742	0.884	0.710	0.866	0.703	0.901	0.760
	5	0.977	0.903	0.963	0.910	0.941	0.860	0.973	0.927	0.974	0.897
	8	0.994	0.940	0.982	0.970	0.953	0.961	0.960	0.938	0.976	0.938
	12	0.964	0.961	0.962	0.987	0.981	0.929	0.978	0.980	0.967	0.954
3	2	0.880	0.712	0.873	0.707	0.894	0.762	0.832	0.714	0.852	0.772
	5	0.961	0.921	0.941	0.925	0.985	0.928	0.978	0.917	0.950	0.945
	8	0.984	0.930	0.979	0.978	0.970	0.936	0.953	0.961	0.963	0.951
	12	0.965	0.938	0.968	0.975	0.996	0.963	0.975	0.980	0.971	0.974
4	2	0.912	0.673	0.913	0.701	0.851	0.614	0.871	0.616	0.921	0.676
	5	0.960	0.783	0.953	0.862	0.960	0.790	0.925	0.805	0.955	0.876
	8	0.957	0.831	0.951	0.895	0.979	0.920	0.943	0.845	0.987	0.906
	12	0.958	0.923	0.986	0.908	0.994	0.949	0.956	0.921	0.973	0.942
5	2	0.914	0.768	0.884	0.771	0.913	0.753	0.873	0.738	0.922	0.780
	5	0.984	0.911	0.989	0.917	0.960	0.887	0.992	0.880	0.963	0.901
	8	0.980	0.923	0.992	0.937	0.985	0.917	0.995	0.952	0.970	0.925
	12	0.994	0.943	0.999	0.964	0.981	0.931	0.974	0.946	0.959	0.977
6	2	0.893	0.672	0.906	0.724	0.897	0.692	0.916	0.730	0.887	0.724
	5	0.953	0.854	0.954	0.837	0.960	0.838	0.944	0.861	0.973	0.856
	8	0.980	0.906	0.952	0.912	0.972	0.872	0.972	0.912	0.996	0.927
	12	0.978	0.959	0.990	0.939	0.967	0.914	0.969	0.952	0.967	0.926
7	2	0.911	0.714	0.924	0.719	0.921	0.753	0.920	0.737	0.943	0.745
	5	0.967	0.849	0.982	0.843	0.954	0.851	0.962	0.876	0.975	0.878
	8	0.954	0.884	0.970	0.912	0.970	0.887	0.960	0.922	0.978	0.887
	12	0.974	0.957	0.979	0.945	0.955	0.910	0.980	0.943	0.978	0.918
8	2	0.903	0.724	0.951	0.728	0.928	0.780	0.923	0.710	0.956	0.716
	5	0.978	0.868	0.965	0.819	0.956	0.867	0.983	0.891	0.986	0.864
	8	0.979	0.887	0.987	0.925	0.968	0.900	0.966	0.922	0.986	0.915
	12	0.991	0.921	0.986	0.936	0.957	0.938	0.971	0.966	0.993	0.949

Table 9. Objective function values and CPU time calculated by different algorithms

Image	Threshold	DBO		PSO		GWO		SSA		SGDF-SSA	
		Objective function value	CPU time	Objective function value	CPU time	Objective function value	CPU time	Objective function value	CPU time	Objective function value	CPU time
1	2	0.0393	14.1745	0.0209	17.2574	0.1752	28.8856	0.0004	23.0754	0.0001	1.7068
	5	0.2422	27.2921	0.3377	20.7623	0.8819	76.7834	0.0045	29.0776	0.0147	2.5809
	8	0.9562	42.0414	0.7362	26.0491	1.8523	128.0183	0.0390	38.6054	0.2942	3.2666
	12	1.5069	58.8896	1.3035	34.1580	3.4614	188.5272	0.2708	45.7377	0.9169	4.0004
2	2	0.0214	13.8689	0.0558	15.9397	0.2449	28.8731	0.0001	22.6096	0.0009	2.0462
	5	0.1543	28.1693	0.2735	20.8500	1.6496	77.1105	0.0046	29.9755	0.0121	2.5775
	8	1.0339	45.9908	0.7374	26.0171	1.0300	125.9424	0.0286	37.2130	0.1598	3.5537
	12	1.9922	57.4400	1.4915	34.2776	3.6498	195.7756	0.2691	47.5249	2.2577	4.5133
3	2	0.1904	14.4707	0.0371	15.8143	0.3988	26.4868	0.0004	22.5349	0.0010	1.9700
	5	0.4948	31.9433	0.7353	20.4357	1.2818	73.9145	0.0055	28.5648	0.0089	2.7257
	8	1.0548	42.3640	0.9791	27.4164	2.0456	116.8846	0.0216	35.4639	0.2178	3.5798
	12	1.3549	61.9317	1.4724	34.5811	3.2175	183.8036	0.0872	48.3412	1.1703	4.3069
4	2	0.2105	13.7243	0.3451	16.1178	0.3719	27.8334	0.0001	22.8843	0.0048	1.8984
	5	1.0445	27.9589	1.3173	21.3846	1.5335	75.3848	0.0043	30.3266	0.0478	2.6471
	8	0.7356	42.2527	1.0174	27.2204	1.6922	127.1122	0.0195	36.4064	0.8392	3.5079
	12	1.8735	61.2065	2.3110	35.9459	3.3645	194.0219	0.1294	49.0957	2.1549	4.4571
5	2	0.0373	17.3237	1.1961	15.7379	1.1474	28.5103	0.0005	22.4818	0.0017	1.8687
	5	1.0222	27.5461	1.2028	20.8926	1.4977	76.5545	0.0059	30.6331	0.0292	2.5366
	8	0.8272	43.5906	1.2781	27.2096	2.1244	121.9852	0.0514	37.4528	0.7443	3.3580
	12	1.7124	59.6043	1.7649	35.6705	3.3348	187.2812	0.3651	47.4351	1.5844	4.0141
6	2	0.8950	14.9548	0.7615	16.0374	0.5252	27.2297	0.0002	22.3901	0.0001	2.0152
	5	0.4681	30.2520	0.7526	21.6847	1.0712	75.0255	0.0126	30.9207	0.0153	2.6092
	8	0.8674	42.8311	1.1033	26.9904	2.2753	125.1377	0.0483	38.8357	0.5291	3.2313
	12	1.3207	63.5016	2.0961	34.2300	3.3999	162.3074	0.1917	49.8654	1.7290	4.4521
7	2	0.0653	14.7203	0.2571	16.3932	0.3163	26.8256	0.0004	22.3597	0.0017	1.9736
	5	0.3945	30.2330	1.0401	21.3951	0.8089	72.7517	0.0067	29.9735	0.2935	2.5860
	8	0.9312	44.2493	1.2075	26.6713	2.1316	124.4328	0.1686	39.7547	0.9106	3.3896
	12	1.7480	65.7801	1.8398	34.2869	3.7055	186.7188	0.2037	47.1962	3.3303	4.4503
8	2	0.5700	16.2514	0.8934	16.1897	0.9059	26.5006	0.0003	22.6704	0.0035	1.9881
	5	0.2840	30.7694	1.2898	21.6984	1.4387	75.0573	0.0051	29.5241	0.6865	2.5419
	8	1.1337	41.3836	1.2963	27.3652	1.9527	115.1261	0.1729	36.3562	0.8153	3.3694
	12	2.5334	59.1064	2.2002	35.2254	3.5908	188.9020	0.1735	47.1825	2.9410	4.4827

It can be seen from Table 8 and Table 9, In experiments using MFE functions to deal with uncertainty and complexity in color images, in order to achieve effective and efficient multi-level threshing-based image segmentation, the best combination of all blurring parameters and thresholds must be selected to minimize the MFE function. Therefore, SSA has been assisted with the proposed entropy function to find the optimal vector of intensity levels, and SGDF-SSA algorithm based multi-threshold seg-mentation of color images at different segmentation levels (m= 2,5,8, and 12) has been performed in this work. SGDF-SSA outperforms all other algorithms in terms of PSNR, MSE, SSIM and FSIM, and is computationally fast. Compared to Kapur's entropy-based segmentation technique, which is the most widely used, MFE achieves better performance.

CONCLUSIONS

Contributions of this Paper

Focusing on the problem of high-precision image segmentation of plant leaves, this study proposes an improved SGDF-SSA algorithm to solve the problems of local optimum traps and insufficient search ability in the optimization process of traditional Sparrow Search algorithm (SSA). In order to enhance the diversity and ergodicity of the population, SPM chaotic mapping mechanism is introduced to perform global uniform sampling of the initial population, so as to improve the quality of the initial solution and the global search ability. In the finder location update strategy, the adaptive Gaussian cloud mutation mechanism is combined to further optimize the search path of the population to reduce the possibility of premature convergence and improve the global search ability of the algorithm. In addition, the finder update strategy based on sine cosine optimization function and dynamic inertia weight enables the finder to explore the search space more efficiently, thereby enhancing the ability to locate the optimal solution. In the follower update stage, this paper innovativly introduces the Cauchy chaotic mutation strategy, which combines the ergodicity of chaotic map and the long tail characteristics of

Cauchy mutation to further improve the local search ability of the algorithm and effectively avoid the interference of local optimal solutions.

In terms of experimental evaluation, this paper uses the CEC2017 standard test function set to systematically verify the optimization performance of SGDF-SSA algorithm. The experimental results show that the optimization accuracy and convergence speed of the SGDF-SSA algorithm are better than those of the existing comparison algorithms on multiple test functions, and the best performance is achieved especially on 11 test functions, showing excellent global optimization ability and robust optimization characteristics. At the same time, in terms of computational efficiency, the SGDF-SSA algorithm shows better optimization efficiency under different fidelity parameter Settings, which effectively reduces the computational cost while ensuring the segmentation accuracy. Furthermore, in the application of plant leaf image segmentation, the SGDF-SSA algorithm shows high segmentation accuracy and robustness in tasks such as complex background interference and pathological leaf detection, which significantly improves the accuracy and robustness of leaf extraction and provides strong support for leaf pathological analysis, crop health monitoring and automatic processing of precision agriculture.

Research Prospects

According to the summary and analysis of the research content in this paper, further research work can be done on this basis in the future:

It tries to reduce the time complexity of SGDF-SSA algorithm. Because the initial strategy updates its population and disturbs the optimal sparrow, the complexity of the algorithm is improved and the optimization time is increased to a certain extent. Therefore, the subsequent research will try to further update and improve the initial population in the SGDF-SSA algorithm to reduce the actual complexity of the algorithm, so as to reduce the running time of the algorithm to optimize the image threshold segmentation.

The image segmentation task completed by the algorithm will make the segmentation results weak on the semantic level. In the future, the idea of selfsupervised learning can be combined on this basis, and the nature of the segmentation results unchanged after the image space and color transformation can be used to improve the understanding of the semantic level in the whole segmentation process.

Broaden the application field of SGDF-SSA algorithm. In this paper, standard images are used to test the performance of the algorithm to apply to medicine, agriculture and other fields, and the algorithm is applied to other fields to further test its segmentation performance.

PATENTS

Funding: This research was funded by Youth Program of Ministry of Education, China, grant number 21YJCZH204; Natural Science Foundation of Liaoning Province, China, grant number 2020-MS-301, the Liaoning Provincial Department of Education, China, grant number LJKMZ20220694.

Data Availability Statement: Data are contained within the article.

Acknowledgments: The research of this paper needs to acknowledge the information and financial support provided by the Department of Education of Liaoning Province and the Ministry of National Education for this research.

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

Abbreviations

The following abbreviations are used in this manuscript:

SSA	The Sparrow Search Algorithm
SPM	Stochastic Perturbation Map
STD	Standard deviation
AVG	Average value of the function
SCA	Sine cosine optimization algorithm

FCM Fuzzy c-means

REFRENCES

[1] Ramachandran, P.; Ramirez, A.; Dinneny, J. R. Rooting for survival: how plants tackle a challenging environment through a diversity of root forms and functions. Plant Physiology 2024, kiae586. DOI: 10.1093/plphys/kiae586.

[2] Xu, W.; Chen, L.; Hu, X.; Zhang, L.; Huang, D.; Li, J.; Xiong, R.; Huang, C.; Zhu, M. Botanic signal monitor: advanced wearable sensor for plant health analysis. Advanced Functional Materials 2024, 34 (51), 2410544. DOI: 10.1002/adfm.202410544.

[3] Guo, R.; Qu, L.; Niu, D.; Li, Z.; Yue, J. Leafmask: Towards greater accuracy on leaf segmentation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021; pp 1249-1258. DOI: 10.1109/ICCVW54120.2021.00145.

[4] Xu, R. Robotic Systems for Field-Based High-Throughput Plant Phenotyping. University of Georgia, 2021.

[5] Murugan, P. R.; Thiyagarajan, A. Review on Brain MRI Segmentation. Advances in Automation, Signal Processing, Instrumentation, and Control: Select Proceedings of i-CASIC 2020 2021, 700, 351. DOI: 10.1007/978-981-15-8221-9_33.

[6] Homan, D. Tree species identification and leaf segmentation from natural images using deep semi-supervised learning. Stellenbosch: Stellenbosch University, 2022.

[7] Toro, M. L. A REVIEW OF IMAGE SEGMENTATION TECHNIQUES. Прикладная математика и информатика: современные исследования в области естественных и технических наук 2021, 86-91.

[8] Laborde, V. Progress in Hybrid Difractive/Refractive Solutions for Compact Space IR Imager. Universite de Liege (Belgium), 2022.

[9] Shoaib, M.; Hussain, T.; Shah, B.; Ullah, I.; Shah, S. M.; Ali, F.; Park, S. H. Deep learning-based segmentation and classification of leaf images for detection of tomato plant disease. Frontiers in plant science 2022, 13, 1031748. DOI: 10.3389/fpls.2022.1031748.

[10] Jiang, D.; Sun, M.; Li, S.; Yang, Z.; Cao, L. EFS-Former: An Efficient Network for Fruit Tree Leaf Disease Segmentation and Severity Assessment. Agronomy 2024, 14 (9), 1992. DOI: 10.3390/agronomy14091992.

[11] Abualigah, L.; Almotairi, K. H.; Elaziz, M. A. Multilevel thresholding image segmentation using meta-heuristic optimization algorithms: Comparative analysis, open challenges and new trends. Applied Intelligence 2023, 53 (10), 11654-11704. DOI: 10.1007/s10489-022-04064-4.

[12] Sujarani, P.; Kalaiselvi, K. Image Analysis for Health Prediction. Artificial Intelligence‐Based System Models in Healthcare 2024, 205-228. DOI: 10.1002/9781394242528.ch9.

[13] Xiao, T.; Reed, C. J.; Wang, X.; Keutzer, K.; Darrell, T. Region similarity representation learning. In Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021; pp 10539-10548. DOI: 10.1109/ICCV48922.2021.01037.

[14] Ramesh, K.; Kumar, G. K.; Swapna, K.; Datta, D.; Rajest, S. S. A review of medical image segmentation algorithms. EAI Endorsed Transactions on Pervasive Health & Technology 2021, 7 (27). DOI: 10.4108/eai.12-4-2021.169184.

[15] Siddique, N.; Paheding, S.; Elkin, C. P.; Devabhaktuni, V. U-net and its variants for medical image segmentation: A review of theory and applications. IEEE access 2021, 9, 82031-82057. DOI: 10.1109/ACCESS.2021.3086020.

[16] Subudhi, S.; Patro, R. N.; Biswal, P. K.; Dell'Acqua, F. A survey on superpixel segmentation as a preprocessing step in hyperspectral image analysis. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing 2021, 14, 5015-5035. DOI: 10.1109/JSTARS.2021.3076005.

[17] Brito, R. Automatic Classification and Segmentation of Patterned Martian Ground Using Deep Learning Techniques. The University of Western Ontario (Canada), 2023.

[18] Shafi, I.; Chaudhry, M.; Montero, E. C.; Alvarado, E. S.; Diez, I. D. L. T.; Samad, M. A.; Ashraf, I. A Review of Approaches for Rapid Data Clustering: Challenges, Opportunities and Future Directions. IEEE Access 2024.

[19] Militello, C.; Ranieri, A.; Rundo, L.; D'Angelo, I.; Marinozzi, F.; Bartolotta, T. V.; Bini, F.; Russo, G. On unsupervised methods for medical image segmentation: Investigating classic approaches in breast cancer dce-mri. Applied Sciences 2021, 12 (1), 162. DOI: 10.3390/app12010162.

[20] Lei, T.; Nandi, A. K. Image segmentation: principles, techniques, and applications; John Wiley & Sons, 2022. DOI: 10.1002/9781119859048.
[21] Zhang, Q.; Ying, Z.; Shen, J.; Kou, S.-K.; Sun, J.; Zhang, B. Unsupervised color-based nuclei segmentation in histopathology images with various color spaces and K values selection. International Journal of Image and Graphics 2024, 2550061. DOI: 10.1142/S0219467825500615.
[22] Behura, A. The cluster analysis and feature selection: Perspective of machine learning and image processing. Data Analytics in Bioinformatics: A Machine Learning Perspective 2021, 249-280. DOI: 10.1002/9781119785620.ch10.
[23] Wu, C.; Zhang, J.; Huang, C. Robust dynamic semi-supervised picture fuzzy local information clustering with kernel metric and spatial information for noisy image segmentation. Multimedia Tools and Applications 2023, 82 (21), 31869-31911. DOI: 10.1007/s11042-023-14703-8.
[24] Juang L H, Wu M N. Psoriasis image identification using k-means clustering with morphological processing. Measurement, 2011, 44(5): 895-905.
[25] Zhong, L.; Wu, X.; Ding, R.; Wang, C.; Li, L.; Zhang, J.; Zhang, J.; Gu, R.; Zhong, S. Perspectives on the application of remote sensing technology in the cultivation of medicinal plants. International Journal of Remote Sensing 2025, 1-34. DOI: 10.1080/01431161.2024.2443620.
[26] Bhattacharjee, P.; Mitra, P. A survey of density based clustering algorithms. Frontiers of Computer Science 2021, 15, 1-27. DOI: 10.1007/s11704-019-9059-3.
[27] Zhang, J.; Li, C.; Yin, Y.; Zhang, J.; Grzegorzek, M. Applications of artificial neural networks in microorganism image analysis: a comprehensive review from conventional multilayer perceptron to popular convolutional neural network and potential visual transformer. Artificial Intelligence Review 2023, 56 (2), 1013-1070. DOI: 10.1007/s10462-022-10192-7.
[28] Han Z, Li Z, Liu K, et al. Named data networking with neural networks for intelligent image processing information systems. Enterprise Information Systems, 2022, 16(10-11): 1527-1542.
[29] Peng, J.; Wang, Y. Medical image segmentation with limited supervision: a review of deep network models. Ieee Access 2021, 9, 36827-36851. DOI: 10.1109/ACCESS.2021.3062380.
[30] Mahmood, T.; Rehman, A.; Saba, T.; Nadeem, L.; Bahaj, S. A. O. Recent advancements and future prospects in active deep learning for medical image segmentation and classification. IEEE Access 2023, 11, 113623-113652. DOI: 10.1109/ACCESS.2023.3313977.
[31] Rajarajeshwari, G.; Selvi, G. C. Application of artificial intelligence for classification, segmentation, early detection, early diagnosis, and grading of diabetic retinopathy from fundus retinal images: A comprehensive review. IEEE Access 2024. DOI: 10.1109/ACCESS.2024.3494840.
[32] Liu, X.; Song, L.; Liu, S.; Zhang, Y. A review of deep-learning-based medical image segmentation methods. Sustainability 2021, 13 (3), 1224. DOI: 10.3390/su13031224.
[33] Alhatami, E.; Bhatti, U. A.; MengXing, H.; Feng, S. Advanced fuzzy denoising technique for agricultural remote sensing: Modified partition filter for suppressing impulsive. IEEE Access 2024. DOI: 10.1109/ACCESS.2024.3447704.
[34] Aghili, M. E.; Imani, M.; Ghassemian, H. Clustering based background learning for hyperspectral anomaly detection. The Egyptian Journal of Remote Sensing and Space Sciences 2023, 26 (3), 477-489. DOI: 10.1016/j.ejrs.2023.06.001.
[35] Maheshwari R, Mohanty S K, Mishra A C. DCSNE: Density-based clustering using graph shared neighbors and entropy. Pattern Recognition, 2023, 137: 109341.
[36] Qi, Y.; Zhang, A.; Wang, H.; Li, X. An efficient FCM-based method for image refinement segmentation. The Visual Computer 2022, 38 (7), 2499-2514. DOI: 10.1007/s00371-021-02126-1.
[37] Zhan, Z.-H.; Shi, L.; Tan, K. C.; Zhang, J. A survey on evolutionary computation for complex continuous optimization. Artificial Intelligence Review 2022, 55 (1), 59-110. DOI: 10.1007/s10462-021-10042-y.
[38] Chai D. Rooted spanning superpixels. International Journal of Computer Vision, 2020, 128(12): 2962-2978.
[39] Herng, O. W.; Nasir, A. S. A.; Shaari, F. N.; Sukor, A. S. A. Superpixels-based automatic density peaks and fuzzy clustering approach in covid-19 lung segmentation. In 2023 IEEE 2nd National Biomedical Engineering Conference (NBEC), 2023; IEEE: pp 152-157. DOI: 10.1109/NBEC58134.2023.10352614.

[40] Friha, O.; Ferrag, M. A.; Kantarci, B.; Cakmak, B.; Ozgun, A.; Ghoualmi-Zine, N. Llm-based edge intelligence: A comprehensive survey on architectures, applications, security and trustworthiness. IEEE Open Journal of the Communications Society 2024. DOI: 10.1109/OJCOMS.2024.3456549.

[41] Dong, Y.; Li, M.; Zhou, M. Multi-threshold image segmentation based on the improved Dragonfly algorithm. Mathematics 2024, 12 (6), 854. DOI: 10.3390/math12060854.

[42] Wang, J.; Tan, Y.; Bo, X.; Li, G. Image Segmentation Method with Improved GA Optimization of Two-Dimensional Maximum Entropy. IEEE Access 2024. DOI: 10.1109/ACCESS.2024.3508796.

[43] Yu, H.; Song, J.; Chen, C.; Heidari, A. A.; Liu, J.; Chen, H.; Zaguia, A.; Mafarja, M. Image segmentation of Leaf Spot Diseases on Maize using multi-stage Cauchy-enabled grey wolf algorithm. Engineering Applications of Artificial Intelligence 2022, 109, 104653. DOI: 10.1016/j.engappai.2021.104653.

[44] Zhu, J.; Ning, W.; Yuan, Y.; Chen, H.; Zhou, W.; Tan, Y.; He, S.; Hu, J.; Fan, Z. Hand-Eye Calibration of Surgical Robots Based on a BP Neural Network Optimized by Using an Improved Sparrow Search Algorithm. In 2022 17th International Conference on Control, Automation, Robotics and Vision (ICARCV), 2022; IEEE: pp 856-861. DOI: 10.1109/ICARCV57592.2022.10004291.

[45] Yurek, S.; DeAngelis, D. L.; Lee, H. W.; Tennenbaum, S. Visualizing wading bird optimal foraging decisions with aggregation behaviors using individual-based modeling. Ecological Modelling 2024, 493, 110702. DOI: 10.1016/j.ecolmodel.2024.110702.

[46] Zhou, Y.; Li, C.; Pang, R.; Li, Y.; Xu, Y.; Chen, J. A new approach for seepage parameter inversion of earth–rockfill dams based on an improved sparrow search algorithm. Computers and Geotechnics 2024, 167, 106036. DOI: 10.1016/j.compgeo.2023.106036.

[47] Yue, Y.; Cao, L.; Lu, D.; Hu, Z.; Xu, M.; Wang, S.; Li, B.; Ding, H. Review and empirical analysis of sparrow search algorithm. Artificial Intelligence Review 2023, 56 (10), 10867-10919. DOI: 10.1007/s10462-023-10435-1.

[48] Zhu, Z.; Zheng, Q.; Liu, H.; Zhang, J.; Wu, T.; Qu, X. Prediction model for pipeline pitting corrosion based on multiple feature selection and residual correction. Journal of Marine Science and Application 2024, 1-11. DOI: 10.1007/s11804-024-00468-5.

[49] Österblom, H. The sounds of science: orchestrating stewardship in the seafood industry; Elsevier, 2023.

[50] Dai, L.; Lei, H.; Chen, L.; Wang, C.; Feng, L. An Image Double Encryption Based on Improved GAN and Hyper Chaotic System. IEEE Access 2024. DOI: 10.1109/ACCESS.2024.3462547.

[51] Li, Z.; Tam, V.; Yeung, L. K. An adaptive multi-population optimization algorithm for global continuous optimization. IEEE Access 2021, 9, 19960-19989. DOI: 10.1109/ACCESS.2021.3054636.

[52] Wei, J.; Gu, Y.; Law, K. E.; Cheong, N. Adaptive Position Updating Particle Swarm Optimization for UAV Path Planning. In 2024 22nd International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt), 2024; IEEE: pp 124-131.

[53] Cao, W.; Cai, H.; Hua, Z. n-Dimensional Chaotic Map with application in secure communication. Chaos, Solitons & Fractals 2022, 163, 112519. DOI: 10.1016/j.chaos.2022.112519.