

Article

Enhanced Sea Horse Optimization Algorithm for Hyperparameter Optimization of Agricultural Image Recognition

Zhuoshi Li ^{1,2} , Shizheng Qu ² , Yinghang Xu ², Xinwei Hao ² and Nan Lin ^{2,*}¹ College of Plant Protection, Jilin Agricultural University, Changchun 130118, China; zhuoshil@jlau.edu.cn² College of Information Technology, Jilin Agricultural University, Changchun 130118, China; qushizheng1211@163.com (S.Q.); 15950702593@163.com (Y.X.); lbxx521520@163.com (X.H.)

* Correspondence: nanl@jlau.edu.cn; Tel.: +86-0431-135-0088-8521

Abstract: Deep learning technology has made significant progress in agricultural image recognition tasks, but the parameter adjustment of deep models usually requires a lot of manual intervention, which is time-consuming and inefficient. To solve this challenge, this paper proposes an adaptive parameter tuning strategy that combines sine–cosine algorithm with Tent chaotic mapping to enhance sea horse optimization, which improves the search ability and convergence stability of standard sea horse optimization algorithm (SHO). Through adaptive optimization, this paper determines the best parameter configuration in ResNet-50 neural network and optimizes the model performance. The improved ESHO algorithm shows superior optimization effects than other algorithms in various performance indicators. The improved model achieves 96.7% accuracy in the corn disease image recognition task, and 96.4% accuracy in the jade fungus image recognition task. These results show that ESHO can not only effectively improve the accuracy of agricultural image recognition, but also reduce the need for manual parameter adjustment.

Keywords: sea horse optimization algorithm; chaos mapping algorithm; sine and cosine algorithm; parameter optimization; CNN

MSC: 68T07

Citation: Li, Z.; Qu, S.; Xu, Y.; Hao, X.; Lin, N. Enhanced Sea Horse Optimization Algorithm for Hyperparameter Optimization of Agricultural Image Recognition. *Mathematics* **2024**, *12*, 368. <https://doi.org/10.3390/math12030368>

Academic Editors: Kunpeng Liu, Pengfei Wang, Yifeng Gao and Yanjie Fu

Received: 6 December 2023

Revised: 6 January 2024

Accepted: 19 January 2024

Published: 23 January 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Since its inception, deep learning has garnered widespread attention for its unique advantages and has been applied across various domains. However, as societal demands continue to increase, conventional neural networks are no longer sufficient to meet people's needs. Consequently, a plethora of enhanced neural networks have emerged.

Wang et al. built a new down sampling attention module based on AlexNet, and introduced the Mish activation function. The new module of the fully connected layer also reduced the network parameters, so as to build a new model AT-AlexNet [1]. In corn disease recognition, the accuracy of AT-AlexNet was significantly higher than other models. Fan et al. studied and designed a corn disease recognition system VGNet based on pretrained VGG16 [2]. The experimental results show that the performance of the proposed model is significantly better than other models. Dai et al. proposed an accurate detection and diagnosis system for corn leaf diseases based on multitask deep learning (MTDL-EPDCLD) [3]. The experimental results show that MTDL-EPDCLD can accurately and effectively identify corn diseases. Zeng et al. proposed a lightweight Dense scale network (LDSNet) [4]. The basic module of the network is the improved Dense Dilated Convolution (IDDC) module, which is used for real-world corn leaf disease image recognition, and the accuracy can reach 95.4%. A large number of studies have proved that the optimization of deep learning models can effectively improve the performance of the model.

In recent years, the superior performance of swarm intelligence algorithms in the field of optimization has gradually attracted extensive attention, especially in the field of hyperparameter optimization [5]. Bahaa et al. used the improved swarm intelligence optimization

algorithm to optimize the convolutional neural network with hyperparameters, so as to construct a new model APSO-WOA-CNN [6]. The experimental results show that the performance of APSO-WOA-CNN is significantly better than other models. Paharia et al. improved the grey Wolf optimization algorithm, and applied the improved grey Wolf optimization algorithm to optimize the convolutional neural network [7], and the performance of the newly constructed model was significantly improved. In order to solve the problem of CNN hyperparameter configuration, Wang et al. improved the particle swarm optimization algorithm, and used the improved particle swarm optimization algorithm to optimize the hyperparameters of CNN [8]. The experimental results show that the improved particle swarm optimization algorithm can solve the problem of CNN hyperparameter optimization, and effectively improve the performance of CNN. In summary, it is feasible to use the swarm intelligence optimization algorithm to optimize the hyperparameters of the neural network model, and can effectively improve the performance of the neural network. Based on this, this paper applied the improved sea horse optimization algorithm to hyperparameter optimization of ResNet-50, and applied the newly constructed model to identify corn diseases. The rest of the paper is structured as follows.

In this paper, the improved sea horse optimization algorithm is applied to find the optimal parameter configuration of ResNet-50 adaptively, so as to improve the performance of ResNet-50.

In this paper, CEC2017 test function was used to verify the performance of the improved sea horse optimization algorithm, and the model improved by ESHO was applied to the image recognition of jade fungus and corn, respectively, in order to verify the performance of the proposed model.

The rest of this paper is structured as follows.

Section 2 details the dataset and experimental methods. Section 3 mainly introduces the experimental environment and experimental results. Section 4 mainly summarizes the overall work and states the focus of future work.

2. Materials and Methods

2.1. Data Sources

2.1.1. Jade Fungus Dataset

The obtained images of dried wood ear were acquired by the FScan2000 acquisition device. As shown in the Figure 1 below, the dimensions of the device body are 570 mm × 430 mm × 280 mm. The imaging resolution is 16 million pixels (4608 × 3456), with a camera sensor size of 1/2.3 CMOS sensor and a focal length of 8 mm for the camera lens. The light source is a 360° surround light, utilizing LED white light. The maximum shooting size is 400 mm × 300 mm, and the minimum accuracy is 0.12 mm.



Figure 1. FScan2000 Display of collection devices.

The experimental data used in this study were collected in the year 2023, from Haotian Village in Najin Town, Tao nan City, Jilin Province. The black board in the center of the apparatus was used to place the edible fungus. During each data collection, the same distance and angle were maintained with respect to each camera.

To further ensure the accuracy and reliability of the edible fungus data, this study meticulously annotated the image data of each Jade fungus sample. Following the dry Jade fungus grading standard DB22/T 2605-2016, we recorded detailed information on the size, quantity, shape, and color of Jade fungus slices, as well as the presence of spots and damage. These six indicators were used as the basis for grading Jade fungus into four levels: first-grade, second-grade, third-grade, and disqualified. The classification method is presented in the Table 1 below.

Table 1. Display of jade fungus dataset and classification criteria.

Data	Jade Fungus			
	Level One	Level Two	Level Three	Disqualified
Number	1	2	3	4
Ear color	White to light yellow	Beige white to light yellow	Light yellow to beige	
Number of ear pieces	Single	Single	Single or multiple	Multiple
Ear size	2~3	3~4	4~5	>5
Ear shape	Complete and uniform	More complete and uniform	More complete and uniform	Incomplete both
Ear condition	Healthy	Contains broken ears	Contains broken ears	Contains infestation
Image				
Image Num	320	408	210	360

2.1.2. Dataset on Corn Diseases

The public dataset (<https://www.kaggle.com/> accessed on 2 November 2023) is used for the experimental data in this paper. The dataset contains a large number of pictures of plant diseases and healthy plants, which aims at the research work of plant disease classification and recognition. The dataset uses 4187 images covering four different types of plant diseases as well as samples of healthy plants. In the dataset, there were 1146 Blight images, 1306 Common-Rust images, 573 gray spot images, and 1162 health images. The images were collected and annotated by various plant disease experts and researchers and are representative and diverse.

For accurate plant disease classification and identification, each image is carefully annotated and classified into the following categories:

Blight: This category contains 1146 images of plants affected by Blight. Blight [9–11] is a common plant disease that causes the leaves and stems of plants to atrophy and turn yellow.

Common-Rust: The dataset contains 1306 Common-Rust images that show the appearance of Common-Rust spots on plant leaves. Common-Rust [12–14] is a plant disease caused by fungi that causes rust-red spots and disease spots on plant leaves.

Grey leaf spot: The 573 grey leaf spot images in the dataset reveal the grey leaf spot on the plant leaves. Grey leaf spot [15–17] is a common disease caused by fungi, which causes dark brown spots on the surface of plant leaves.

Health: 1162 images of healthy plants are also included in the dataset, which show healthy plants that have not been affected by any visible plant diseases.

Figure 2 below shows the three diseases and healthy comparison images.

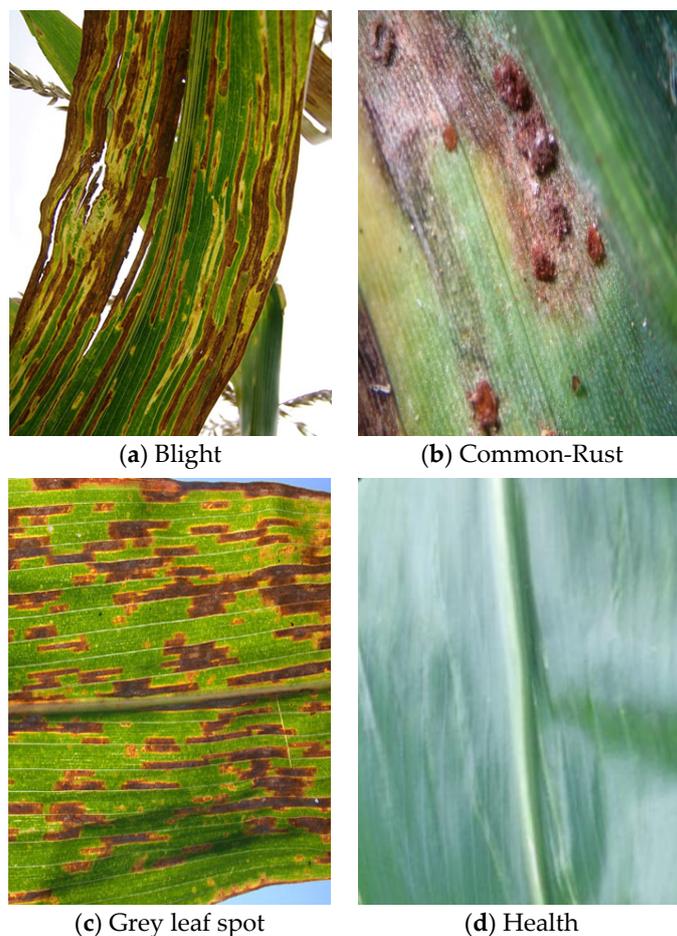


Figure 2. The three corn diseases and Health data sets show: (a) Blight data set, (b) Common-Rust data set, (c) Grey leaf spot data set, (d) Health data set.

Through the dataset, a wide range of representative samples of plant diseases and healthy plants can be obtained, which provides an important basis for subsequent research work. At the same time, in the process of constructing the dataset, we also fully consider the accuracy of annotation and the diversity of data samples to ensure the reliability and validity of the experimental results.

2.2. Experiment Method

2.2.1. Resnet-50 Model

This paper uses ResNet-50 [18], a subclass of deep neural networks, which contains two basic layers: the convolutional layer and the fully connected layer. There are 49 convolutional layers and one fully connected layer. The ResNet-50 network structure is mainly composed of five parts, of which the first part does not contain residual blocks, mainly preprocessing the input data, including: convolution, regularization, activation function, and maximum pooling calculation. The structure of the second, third, fourth, and fifth parts is similar, mainly composed of two residual blocks, namely the identity residual block and the convolutional residual block.

ResNet-50 is mainly used to solve image classification problems, and its advantage is that it can be connected across one layer to the next, which significantly reduces the overall computing time of network classification, and all five parts contain convolutional layers and pooling layers.

2.2.2. Sea Horse Algorithm

In this section, we will introduce the standard Sea horse algorithm [19] in detail. The SHO algorithm simulates the movement, predation, and reproduction of sea horse. These three behaviors are key components of SHO. In order to better balance the improvement of SHO algorithm, the global and locaters strategies are applied to the motion and predation behaviors, respectively.

Movement Behavior of the Seahorse

Seahorse locomotion behavior is divided into two situations: one is the spiral movement of the hippocampus with the vortex of the ocean, and the other is the Brownian motion of the hippocampus with the waves.

Case one: The Seahorse spirals with the whirlpool of the ocean.

The Seahorse spirals closer to its best advantage, and Lévy flight is used here to simulate the Seahorse’s moving steps. This method will avoid the strategy of SHO falling into the local optimal solution, and the unique spiral motion of the Seahorse can also make it constantly change the rotation angle, and can also expand the neighborhood of the existing local solution. This is mathematically achieved as follows:

$$X_{new}^1(t + 1) = X_i(t) + Levy(\lambda)((X_{elite}(t) - X_i(t)) \times x \times y \times z + X_{elite}(t)) \quad (1)$$

where $x, y,$ and z represent the three-dimensional vector of coordinates (x, y, z) in the spiral motion, respectively.

Case two the seahorse does Brownian motion with the waves.

$$\sigma = \left(\frac{\Gamma(1 + \lambda) \times \sin\left(\frac{\pi\lambda}{2}\right)}{\Gamma\left(\frac{1+\lambda}{2}\right) \times \lambda \times 2^{\left(\frac{\lambda-1}{2}\right)}} \right) \quad (2)$$

To the left of the r_1 cut-off point, in order to better explore the search space for SHO, Brownian motion is used to simulate the motion step size of the sea horse, which is expressed as follows:

$$X_{new}^1(t + 1) = X_i(t) + rand \times l \times \beta_t \times (X_i(t) - \beta_t \times X_{elite}) \quad (3)$$

where l is the constant coefficient (this article sets it to $l = 0.05$).

Predatory Behavior of the Seahorse

There are two situations in which seahorses are preyed: one is success and the other is failure. To simulate both cases, this article introduces a random number r_2 . In real life, the predation success rate of the seahorse is 90%, so we set the critical value to $r_2 > 0.1$, when it is proved that the seahorse finally successfully captured the prey; on the contrary, it means that the speed of the prey is faster than the speed of the seahorse when preying, let it escape, and unsuccessfully capture the prey, expressed by a mathematical model as follows:

$$X_{new}^2(t + 1) = \begin{cases} \alpha \times (X_{elite} - rand \times X_{new}^1(t)) + (1 - \alpha) \times X_{elite} & r_2 > 0.1 \\ (1 - \alpha) \times (X_{new}^1(t) - rand \times X_{elite}) + \alpha \times X_{new}^1(t) & r_2 \leq 0.1 \end{cases} \quad (4)$$

$X_{new}^1(t)$ indicates the new position of the seahorse at the time of t , r_2 is the random number $[0, 1]$, which is used to adjust the step length of the seahorse during predation, which decreases linearly as the iteration progresses.

Reproductive Behavior of the Seahorse

It is worth noting that male seahorse is reproduced in nature, so in the SHO algorithm, some of the better fitness values are used as male populations for reproduction, and the

other part is used as female populations to distinguish the next generation with better characteristics, the mathematical expression is as follows:

$$\begin{aligned} father &= X_{sort}^2(1 : pop/2) \\ mother &= X_{sort}^2(pop/2 + 1 : pop) \end{aligned} \tag{5}$$

Here, X_{sort}^2 represents the fitness value for all X_{sort}^2 's in ascending order of predation behavior, and father and mother represent the male and female populations, respectively.

To make the SHO algorithm easier to perform, it is assumed that only one offspring will be produced per pair of seahorse random mating, with the following expression:

$$X_i^{offspring} = r_3 X_i^{father} + (1 - r_3) X_i^{mother} \tag{6}$$

r_3 represents a random number between [0, 1], i is a positive integer in the range [1, $pop/2$], X_i^{father} and X_i^{mother} represent an individual randomly produced in male and female populations, respectively.

2.2.3. Sine–Cosine Algorithm Optimizes Chaotic Sea Horse Algorithm

The enhanced seahorse optimization algorithm proposed in this paper is improved in two aspects: the first point introduces the chaotic mapping algorithm in the initialization of the seahorse optimization algorithm, and the second point introduces the sine–cosine optimization algorithm in the seahorse optimization algorithm to optimize the fitness value.

Chaos Initialization and Parameter Optimization

The traditional seahorse optimization algorithm adopts the strategy of population initialization is random generation, which has the disadvantage of this method is that the randomness is large, and the quality of the initial solution cannot be guaranteed. In order to better optimize the problems of traditional seahorse optimization algorithms, in this paper, Tent chaos mapping is used to generate random chaotic sequences to generate initial sea horse populations.

The advantage of chaos mapping is that it has randomness, ergo city, and strong sensitivity to initial values, which makes the algorithm adding chaos mapping have a faster convergence speed than the original algorithm.

Tent mapping [20] is also known as tent mapping because of the function image's resemblance to the tent shape. This is shown in Figure 3.

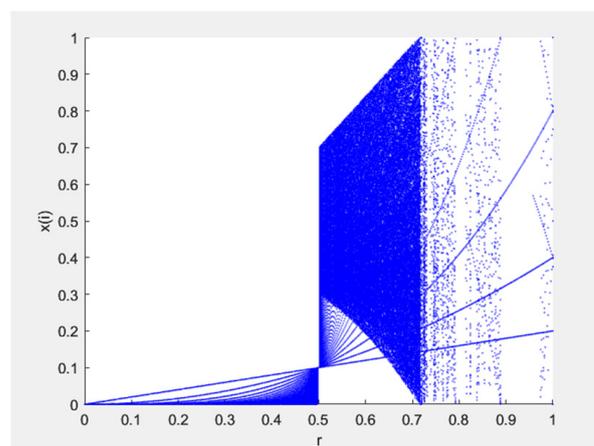


Figure 3. Image of Tent chaotic map function.

Compared with the tent map, the initial population optimization effect in the optimization algorithm is better, and the random chaotic sequence generated by the tent map is used instead of the randomly generated parameters in the original algorithm, so that the

initial solution generated in the search space has good diversity. The resulting high-quality initial solution will help the algorithm with convergence speed and accuracy of the result.

The steps are as follows:

1. Determine the parameters α (This article $\alpha = 0.7$)
2. Set the value range of the initial value x_0 sequence according to the objective function, and generate X values in this range.
3. $X_0 = x_0(n), n = 1, 2 \dots X$
4. $x(1) = X_0$

$$x(n + 1) = \begin{cases} \frac{X(n)}{\alpha} & x(n) \in [0, \alpha) \\ \frac{(1-X(n))}{(1-\alpha)} & x(n) \in [\alpha, 1) \end{cases} \tag{7}$$

Sine–cosine algorithm optimizes chaotic seahorse algorithm.

The seahorse optimization algorithm has the disadvantages of slow convergence speed, low search accuracy, and easy to fall into local optimal solution. Although the tent mapping algorithm is introduced to improve the convergence speed, the sine and cosine optimization algorithm is introduced because it greatly improves the speed of leader position update, improves the optimization speed, improves the search accuracy, and improves the local optimal solution.

Sine cosine optimization algorithm (SCA) is a global optimization algorithm proposed in recent years [21], which is realized by using the properties of sine function and cosine function in mathematics, and balances the global exploration ability and local development ability of the algorithm through amplitude, which is different from the traditional swarm intelligent optimization algorithm, the advantage structure of the algorithm is relatively simple, robust, and easy to implement.

Assuming the population size is N and the search space dimension is d , map each solution of the optimization target problem to the location of each population in the search space. Then the population position of $i(i = 1, 2 \dots N)$ populations that have undergone t iterations in the d dimensional search space can be expressed as $X_i^t = (X_{i1}^t, X_{i2}^t \dots X_{iD}^t)$.

First, N population locations are randomly initialized in the search space; then, the population fitness value is calculated according to the objective function. Sort the best and disadvantage by the fitness value of the population, and update the optimal fitness value and its corresponding position.

$$X_d^i(t + 1) = \begin{cases} X_d^i(t) + a \times \sin(r_3) \times |r_4 X^* - X_d^i(t)| & r_5 < 0.5 \\ X_d^i(t) + a \times \cos(r_3) \times |r_4 X^* - X_d^i(t)| & r_5 \geq 0.5 \end{cases} \tag{8}$$

where X_d^i represents the position of the t generation of i populations in the d dimension; X^* represents the current optimal position; The a parameter is to control the search direction of the population, and the change mode $a = (1 - \frac{t}{T})^{\frac{2t}{T}}$, r_3 is the random number in the value range of $[0, 2\pi]$ is used to control the search distance of the algorithm, r_4 and r_5 are random numbers on $[0, 2]$ and $[0, 1]$, respectively (in this paper r_3, r_4 , and r_5 random numbers are selected by Tent chaotic mapping), and the way to control the update position of the $t + 1$ generation is to use a sine function or a cosine function.

2.2.4. Enhanced Seahorse Optimization Algorithm

Because the seahorse optimization algorithm is too chaotic in the initial sequence, the chaotic mapping algorithm is introduced in the initialization to improve, and the position with the best fitness value is assigned to the seahorse leader at each iteration, which leads to the algorithm being easy to fall into the local optimal region, which often leads to the selection of the optimal fitness value by the optimization precision, and the SCA algorithm can be randomly selected to optimize the sine and cosine cross-optimization, so that the position update methods of the two complement each other. Pseudocode to enhance the seahorse optimization algorithm is shown in Algorithm 1.

Algorithm 1 Enhance sea horse optimization algorithm

Input: The population size pop , the maximum number of iterations T and the variable dimension Dim
Output: The optimal search agent X_{best} and its fitness value f_{best}
 Use Tent chaos to map the initial population X_i^j and the parameters $rand, \lambda, w,$ and k
 While ($t < T$)
 if ($r_1 > 0$)
 Update seahorse position using Equation (1)
 else if
 Update seahorse position using Equation (3)
 end if
 Update seahorse position using Equation (4)
 Calculate the fitness values for each hippocampus
 Select parents with Equation (5)
 Calculate the next generation using Equation (6)
 Update fitness values using SCA, Equation (8)
 Update the location of the seahorses
 $t = t + 1$
 End While

2.2.5. ESHO Hyperparameter Optimization Resnet-50 Model

ResNet-50 includes a number of hyperparameters that contribute to model performance, including training algorithm, momentum leaning, batch size, epoch, and validation frequency. These parameters are the main performance improvement parameters.

The main steps are as follows:

Step 1: Set the number of populations, dimensions, maximum iterations of the population, and determination of the boundary of the SHO algorithm.

Step 2: Initialization: According to the parameters of the ResNet-50 network and ESHO, the population is created through Tent chaos mapping.

Step 3: Random numbers such as $rand, \lambda, w,$ and k are uniformly generated by Tent chaos mapping.

Step 4: Fitness value: The objective function is used to evaluate each ResNet-50 network, and the required hyperparameter values are automatically updated by the ESHO algorithm, and the fitness function selected from it is also the determination of the error rate.

Step 5: Based on the target value and the generated hyperparameters, create a new network for calculation.

Step 6: Update the position of the deposited target value according to the SCA algorithm, and introduce the greedy mechanism to determine whether it is the global optimal solution.

Step 7: Steps 3, 4, 5, and 6 are recalculated until the optimal solution with the maximum number of iterations is reached.

This is shown in Figure 4 and Algorithm 2.

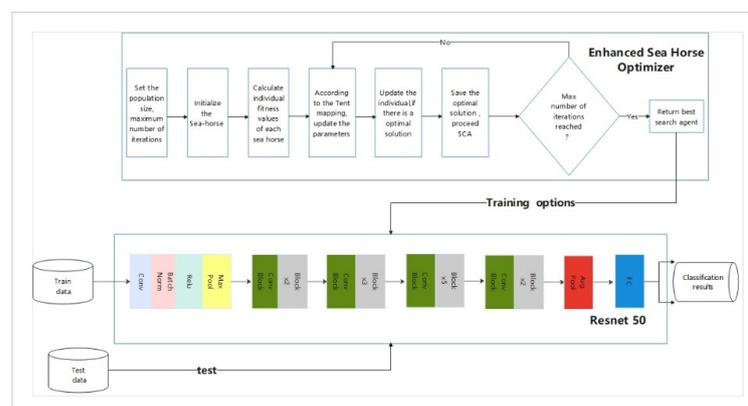


Figure 4. ESHO hyperparameter optimization flowchart.

Algorithm 2 Enhanced sea horse optimization algorithm for hyperparameter optimization of CNN

Input: dim, pop, T, K1, K2, K3, K4, Hyperparameter EvalFunction
Output: Optimized deep learning parameters w&b
 Initialize enhanced sea horse optimization algorithm population (including each individual’s position and deep learning parameters w&b, a, A, C)
 Obtain a batch of training datasets
 For each iteration $t = 1$ to T do:
 For each population member $i = 1$ to pop do:
 Calculate fitness value
 Update the current individual’s position if a better position is found
 End inner for loop
 Update algorithm control parameters a, A, C
 (K1, K2, K3, K4) = Hyperparameter EvalFunction(current optimal position L_p)
 End outer for loop
 Descent with Momentum (SGDM) optimization algorithm
 Update w&b using SGDM with hyperparameters K1, K2, K3, K4 Return w&b
 End While

2.3. Evaluation

This subsection introduces the evaluation criteria proposed in this paper as accuracy, sensitivity, precision, recall, and other evaluation criteria for verification, and ROC analysis is used to verify the experimental data results. The following Table 2 is the indicator formula for the classification criteria.

Table 2. Evaluation criteria formula.

Evaluate	Formula
Accuracy	$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$
Sensitivity	$TPR = \frac{TP}{TP+FN}$
Precision	$Precision = \frac{TP}{TP+FP}$
Recall	$Recall = \frac{TP}{TP+FN}$

TP (True Positives) is the sample is positive and the prediction result is positive. *FP* (False Positives) is the sample is negative and the prediction result is positive. *TN* (True Negatives) is the sample is negative and the prediction result is negative. *FN* (False Negatives) is the sample is positive and the prediction result is negative.

Accuracy was used to assess whether Blight, Common-Rust, gray spot, and health could be completely distinguished, i.e., the proportion of samples correctly classified in the total sample.

Sensitivity, also known as Recall, is used to assess the proportion of all Blight in the correct classification, as well as the ability to recognize Blight. It is the proportion of all samples correctly identified as Blight to the total number of samples that are truly Blight.

Precision, also known as the Positive Predictive Value, refers to the proportion of samples correctly classified as Blight to the total number of samples classified as Blight. It is used to assess the accuracy of the classification of Blight.

Recall (also known as Sensitivity) is the proportion of all samples correctly identified as Blight to the total number of samples that are truly Blight, used to assess the ability to recognize Blight.

3. Results

Experiment 1 compares the proposed improved ESHO algorithm with the traditional SHO algorithm and its two enhancement strategies in terms of their search capability, convergence speed, and accuracy. This experiment will utilize the CEC2017 test suite to evaluate the performance of these algorithms on a set of standard test functions for optimization problems. The primary aim of Experiment 2 is to assess the classification performance of a neural network optimized by the ESHO algorithm (ESH0-net) when processing the Jade fungus image dataset. The experiment utilizes the Jade fungus image

dataset, collected using an industrial camera, as input data, with the ESHO algorithm optimized neural network performing the classification task. To provide a comparative analysis, ESHO-net is compared against other classical neural network models, as well as models optimized using swarm intelligence algorithms. By comparing metrics such as classification accuracy, recall, and precision among these different models, the strengths and weaknesses of ESHO-net in handling the Jade fungus image dataset are evaluated. The aim of Experiment 3 is to evaluate the performance of the neural network optimized by the ESHO algorithm (ESHO-net) in the task of image classification using 3221 images of corn diseases. This experiment seeks to compare the classification accuracy, multiclass classification performance, and the ability to handle imbalanced data of the ESHO-net with models optimized by other classical and swarm intelligence algorithms when processing corn disease images. These performance metrics will aid in determining the superiority of ESHO-net in the task of classifying corn disease images.

The test environment is Windows 64-bit operating system, and the MATLAB is 2023. Hardware environment: The processor on the computer is AMD Ryzen 7 5800H with Radeon Graphics CPU @ 3.20 GHz. The running memory is 128 GB, the graphics card is NVIDIA Quadro RTX 3060, and the video memory (VRAM) is 16 GB.

3.1. Experiment 1 Function Test

ESHO refers to an algorithm obtained by optimizing SHO algorithm using Tent mapping and sine cosine algorithm. Tent mapping and sine cosine algorithm are used as improvement measures to improve the performance and convergence speed of the optimization algorithm to better solve the problem.

For ESHO algorithm, the Tent map was used to increase the diversity of the search space, and use the cosine algorithm with local search, further optimize the solution. These improvements help to improve the global search ability and solution quality of ESHO algorithm.

ESHO algorithm improves the initial population in SHO algorithm and adds sine cosine optimization algorithm for improvement. In order to verify the effectiveness of the algorithm improvement, the effect of the two position improvements will be compared. As shown in Table 3, the modified initial population position is denoted by "S", and the addition of the sine cosine optimization algorithm is denoted by "SIN". In the table, "1" means that this location update strategy is improved in SHO, and "0" means that it is not improved.

Table 3. Representation of the two optimizations in SHO.

Algorithm	S	SIN
SHO	0	0
SSH0	1	0
SINSHO	0	1
ESH0	1	1

Therefore, the ESHO algorithm may be better than the original SHO algorithm in terms of performance, resulting in solutions that are closer to the optimal solution and have smaller Best, Worst, Mean, and Std metrics. This means that ESHO algorithm may be improved in the ability to find the global optimal solution, the ability to avoid poor solutions, and the average quality and stability of the solution. ESHO algorithm is an optimization algorithm that improves SHO algorithm by introducing Tent mapping and sine cosine algorithm. These improvements help to improve the performance of the algorithm and the quality of the solution. The algorithm pairs are shown in Table 4 for example.

As illustrated in Figure 5, in the CEC2017 function testing experiment, we selected six distinct functions for comparison, namely F1, F2, F4, F17, F23, and F27. By comparison, it is apparent that the addition of Tent mapping has improved the selection of initial points, signifying the optimization algorithm's enhanced exploration of the search space. In our experiment, the inclusion of Tent mapping may have been observed to elevate the optimization performance for these functions. Incorporating the cosine algorithm facilitates local

search, enabling more precise adjustments and refinement near the current solution, thus further optimizing the solution. In our experiment, integrating the cosine algorithm may have rendered the algorithm more effective in the search process, thereby further enhancing its performance. The inclusion of both enhancement strategies simultaneously has led to a more pronounced improvement, indicating their complementary and reinforcing role in augmenting the performance of the SHO algorithm. They can better guide the search process, enhance the algorithm’s global search capabilities, and improve solution quality.

Table 4. Comparison of CEC2017 test function algorithms.

Fun		SHO	SSHO	SINSHO	ESHO
F1	Best	1.15×10^{10}	1.01×10^{10}	1.26×10^{10}	1.01×10^{10}
	worst	3.08×10^{10}	3.30×10^{10}	3.37×10^{10}	2.68×10^{10}
	mean	1.94×10^{10}	2.07×10^{10}	1.92×10^{10}	1.81×10^{10}
	std	5.37×10^9	5.47×10^9	4.96×10^9	4.32×10^9
F2	Best	2.87×10^{27}	9.74×10^{26}	3.09×10^{25}	4.27×10^{24}
	worst	9.61×10^{38}	3.89×10^{38}	9.78×10^{36}	7.57×10^{36}
	mean	3.31×10^{37}	2.46×10^{37}	4.55×10^{35}	3.92×10^{35}
	std	1.75×10^{38}	7.81×10^{37}	1.82×10^{36}	1.41×10^{36}
F3	Best	52,212.41	53,015.29	48,580.26	52,613.81
	worst	83,056.46	82,353.02	86,771.12	78,841.13
	mean	68,990.98	70,028.33	66,705.07	65,567.97
	std	7876.64	7256.565	8976.78	6903.744
F4	Best	1054.55	1500.277	1268.575	836.9427
	worst	6747.084	6912.596	7586.876	6463.786
	mean	3259.558	4022.42	2928.848	2541.938
	std	1525.745	1535.826	1579.358	1350.051
F5	Best	690.1524	706.2802	713.8169	709.8417
	worst	801.9082	852.4711	804.9284	843.369
	mean	756.447	759.936	750.273	750.5114
	std	28.965	32.90426	25.95798	26.68688
F6	Best	6.43×10^2	6.40×10^2	6.41×10^2	6.31×10^2
	worst	6.60×10^2	6.60×10^2	6.65×10^2	6.65×10^2
	mean	6.53×10^2	6.52×10^2	6.53×10^2	6.51×10^2
	std	4.39×10	5.99×10	5.99×10	6.32×10
F7	Best	9.18×10^2	9.66×10^2	9.55×10^2	9.37×10^2
	worst	1.05×10^3	1.08×10^3	1.06×10^3	1.06×10^3
	mean	1.01×10^3	9.99×10^2	9.95×10^2	9.95×10^2
	std	3.09×10^1	2.56×10^1	2.30×10^1	2.67×10^1
F8	Best	9.18×10^2	9.66×10^2	9.55×10^2	9.37×10^2
	worst	1.05×10^3	1.08×10^3	1.06×10^3	1.06×10^3
	mean	1006.899	999.0511	995.0073	994.866
	std	30.87526	25.63718	22.99354	26.69634
F9	Best	5045.912	4770.397	4051.609	4124.663
	worst	7286.883	9052.588	6716.671	6892.722
	mean	6038.505	6174.653	5558.307	5405.359
	std	560.4052	895.5512	697.2742	695.639
F10	Best	4816.358	4234.36	5170.172	4167.356
	worst	7052.398	6951.179	7235.78	6503.908
	mean	5985.535	5759.261	5885.718	5673.887
	std	519.5401	545.7817	499.8241	612.4502
F11	Best	1518.007	1713.981	1513.699	1561.659
	worst	8127.989	7158.733	7545.37	5901.66
	mean	3472.514	3218.312	3695.645	3402.712
	std	1504.513	1248.817	1537.379	1151.42
F12	Best	1.09×10^8	1.32×10^8	1.69×10^8	1.28×10^8
	worst	7.53×10^9	6.77×10^9	7.66×10^9	4.86×10^9
	mean	2×10^9	2.36×10^9	1.93×10^9	2×10^9
	std	1.82×10^9	2.03×10^9	1.89×10^9	1.13×10^9
F13	Best	1,936,199	5,324,252	4,639,137	4,718,375
	worst	7.62×10^9	1.04×10^{10}	7.04×10^8	5.1×10^9
	mean	9.9×10^8	7.21×10^8	1.78×10^8	6×10^8
	std	2.02×10^9	1.96×10^9	1.73×10^8	1.2×10^9
F14	Best	68,110.24	130,880	98,109.13	70,302.43
	worst	3,129,325	2,211,560	2,555,339	1,107,680
	mean	922,027.1	1,037,686	976,467.4	517,943.9
	std	649,074.7	607,791.7	624,504.3	295,845.4

Table 4. Cont.

Fun		SHO	SSHO	SINSHO	ESHO
F15	Best	14,784.28	17,108.77	11,294.54	11,057.22
	worst	6,578,499	9,345,190	20,656,181	58,669,887
	mean	942,711.7	1,513,059	2,608,761	4,129,592
	std	1,405,698	2,038,891	4,679,572	11,314,354
F16	Best	2369.129	2429.833	2473.326	2471.599
	worst	3599.571	4159.49	3831.684	4158.597
	mean	3108.508	3167.448	3130.635	3188.239
	std	266.2169	422.0315	318.3059	375.3002
F17	Best	1944.424	1839.123	1959.663	1828.769
	worst	2834.808	2963.481	2740.47	2803.459
	mean	2356.188	2293.307	2323.588	2353.132
	std	251.8807	285.3271	216.0364	219.267
F18	Best	596,877.9	448,618.8	139,801.9	128,147.2
	worst	24,946,311	21,730,056	13,947,137	1,966,4615
	mean	3,270,441	4,479,111	2,715,288	2,996,348
	std	4,749,170	5,229,032	2,816,730	3,984,972
F19	Best	32,669.16	23,875.63	9838.216	6813.085
	worst	5.28×10^8	1.23×10^8	3.86×10^8	1.31×10^8
	mean	31,618,101	5,287,557	17,247,149	8,906,206
	std	99,909,099	22,234,090	72,483,988	26,703,308
F20	Best	2304.26	2295.288	2300.131	2246.822
	worst	3019.55	3029.424	2908.584	2795.154
	mean	2593.264	2628.703	2515.603	2505.816
	std	209.162	182.0024	167.7344	146.8245
F21	Best	2474.796	2463.452	2473.199	2467.6
	worst	2621.73	2579.178	2639.677	2598.871
	mean	2525.397	2523.414	2524.254	2525.216
	std	32.47812	30.24221	36.24246	32.04138
F22	Best	3383.561	3770.807	3657.703	3822.016
	worst	8627.898	8504.043	9044.294	8403.451
	mean	6208.513	6220.16	5837.821	5750.331
	std	1477.657	1411.245	1808.969	1427.724
F23	Best	2962.408	2945.802	2917.354	2879.052
	worst	3226.973	3201.826	3093.162	3170.076
	mean	3054.79	3048.868	3015.775	3008.702
	std	64.01824	71.73463	45.61353	72.37846
F24	Best	3214.782	3246.572	3198.056	3184.568
	worst	3447.956	3481.337	3421.383	3397.219
	mean	3339.036	3353.138	3281.536	3285.132
	std	56.71183	56.07424	56.82005	49.41599
F25	Best	3168.9	3161.506	3159.44	3132.425
	worst	3909.32	4297.002	4290.112	3919.242
	mean	3436.999	3521.027	3410.432	3406.443
	std	195.7024	272.8187	264.2817	213.8743
F26	Best	6052.081	5424.125	6308.89	5125.343
	worst	8937.705	9404.311	8614.525	8872.829
	mean	7553.396	7795.786	7448.687	7287.09
	std	771.6484	919.546	603.431	882.8359
F27	Best	3399.312	3357.454	3319.294	3370.905
	worst	3734.859	3834.588	4053.098	3770.368
	mean	3545.823	3572.432	3489.797	3492.418
	std	91.95908	115.2077	137.8223	85.95479
F28	Best	3985.444	3698.788	3804.833	3699.801
	worst	5267.373	5215.807	5433.59	4982.299
	mean	4517.781	4390.196	4430.446	4288.284
	std	380.2863	387.744	479.7648	382.0062
F29	Best	3854.939	3837.67	3877.058	4156.992
	worst	5259.497	5131.16	5073.191	5077.449
	mean	4505.4	4477.765	4533.303	4523.774
	std	364.2282	277.1826	282.0671	250.438
F30	Best	3,313,653	2,518,304	1,023,277	2,937,596
	worst	8.65×10^8	44,619,145	8.6×10^8	1.08×10^8
	mean	48,992,562	18,326,447	43,953,905	18,719,231
	std	1.55×10^8	13,137,887	1.55×10^8	21,574,223

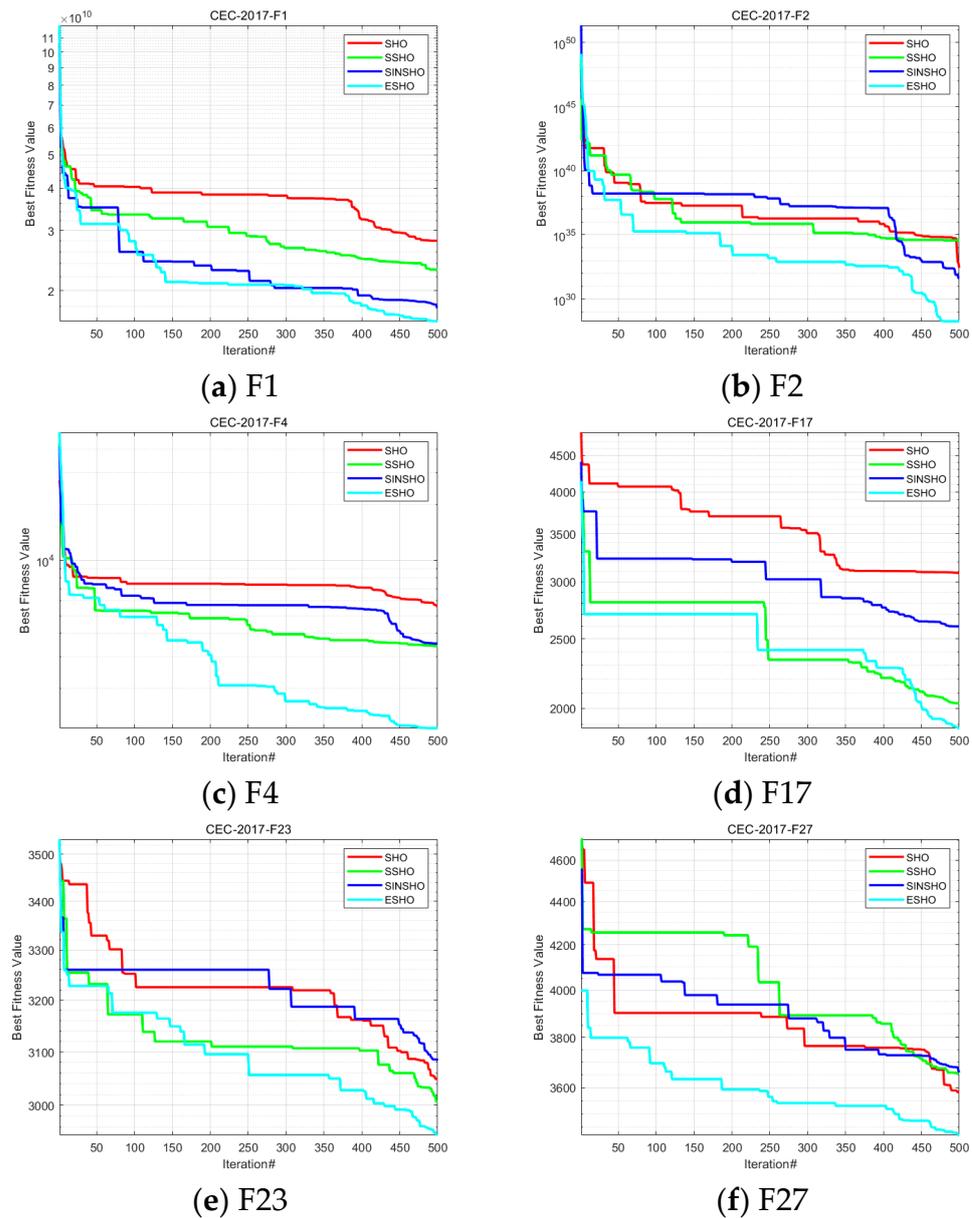


Figure 5. CEC2017 function test comparison diagram. (a) Function 1; (b) Function 2; (c) Function 4; (d) Function 17; (e) Function 23; (f) Function 27.

In summary, the experimental results demonstrate a remarkable improvement in the SHO algorithm resulting from the addition of Tent mapping and the cosine algorithm. This strongly underscores the significance of these two enhancement strategies in optimizing the SHO algorithm, particularly when dealing with functions such as F1, F2, F4, F17, F23, and F27. This bears important guiding implications for further enhancing algorithm performance and improving solution quality.

3.2. Experiment 2 Jade Fungus Classification

The aim of this study is to compare the performance of 7 different pretrained networks (Resnet 18 [22], Google net [23], Inception v3 [24], Alexnet, Densenet 201 [25], ResNet50, VGG16) in classifying images of Jade fungus, incorporating optimization algorithms (SHO, SSHO, SINSHO, WOA, GA, PSO, GWO) for fine-tuning ResNet50. Additionally, the results will be compared to ESHO-net model. Evaluation metrics include accuracy, sensitivity, precision, and recall. Figure 6 and Table 5 present the optimal parameters and confusion matrices obtained through the optimization of ESHO-net model.

94 24.2%	1 0.3%	0 0.0%	1 0.3%	97.9% 2.1%
2 0.5%	117 30.1%	6 1.5%	0 0.0%	93.6% 6.4%
0 0.0%	4 1.0%	57 14.7%	0 0.0%	93.4% 6.6%
0 0.0%	0 0.0%	0 0.0%	107 27.5%	100% 0.0%
97.9% 2.1%	95.9% 4.1%	90.5% 9.5%	99.1% 0.9%	96.4% 3.6%

Figure 6. The optimal parameter confusion matrix is selected for the jade fungus grading data set.

Table 5. Optimal parameters for classification dataset of jade fungus.

Parameter	SGDM
Momentum	0.9
Initial learning rate	0.003855
Maximum epoch	32
Validation frequency	33

3.2.1. Compared with Pretrained Neural Network

In Figure 7, a selection of 7 different pretrained networks were chosen to assess their performance in the classification of Jew’s Ear mushroom images, and these networks were compared with the ESHO-net model. To evaluate the performance of these networks, four key indicators were utilized, namely accuracy, recall, sensitivity, and precision. The ESHO-net model demonstrated improvements across all performance metrics. Specifically, the ESHO-net model showed an increase in accuracy ranging from 1.5% to 16.1%, improvements in recall and sensitivity ranging from −2.1% to 10.9%, −1.2% to 20.8%, 10.3% to 35.1%, and 1.9% to 9.8%, respectively. The precision also experienced enhancements within the ranges from −1.1% to 7.3%, 6.5% to 21.3%, 1.1% to 15.9%, and −0.9% to 9.3%.

92 23.7%	1 0.3%	0 0.0%	0 0.0%	98.9% 1.1%
2 0.5%	99 25.4%	11 2.8%	1 0.3%	87.6% 12.4%
2 0.5%	19 4.9%	52 13.4%	0 0.0%	71.2% 28.8%
0 0.0%	3 0.8%	0 0.0%	107 27.5%	97.3% 2.7%
95.8% 4.2%	81.1% 18.9%	82.5% 17.5%	99.1% 0.9%	90.0% 10.0%

(a) AlexNet

95 24.4%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
1 0.3%	109 28.0%	4 1.0%	1 0.3%	94.8% 5.2%
0 0.0%	11 2.8%	59 15.2%	1 0.3%	83.1% 16.9%
0 0.0%	2 0.5%	0 0.0%	106 27.2%	98.1% 1.9%
99.0% 1.0%	89.3% 10.7%	93.7% 6.3%	98.1% 1.9%	94.9% 5.1%

(b) DenseNet 201

Figure 7. Cont.



Figure 7. Comparison of Traditional Deep Learning Confusion Matrices (a) AlexNet Confusion matrix; (b) DenseNet201 Confusion matrix; (c) GoogleNet Confusion matrix; (d) inception V3 Confusion matrix; (e) ResNet 18 Confusion matrix; (f) ResNet 50 Confusion matrix; (g) VGG 16 Confusion matrix.

3.2.2. Optimize Resnet-50 Network with Other Optimization Algorithms

The proposed evaluation methods (accuracy, sensitivity, accuracy, recall) were compared with other optimization algorithms (SHO, SSHO, SINSHO, WOA [26], GA [27], PSO [28], GWO [29]) to optimize the network.

The confusion matrix in Figure 8 reveals that the ESHO-net model has exhibited improvements in the following aspects: an increase in accuracy ranging from 1.0% to 2.8%, and in recall and sensitivity from −2.1% to 1%, −1.2% to 3.8%, 6.1% to 12.8%, and 0% to 2.8%. Furthermore, the precision has seen enhancements from −1.1% to 2.1%, 2.5% to 7.4%,

−3.2% to 9.5%, and −0.9% to 3.7%. ESHO-net demonstrates performance enhancements across multiple evaluation metrics, indicating its potential as a promising model.



Figure 8. Other optimization algorithms were used to optimize the confusion matrix comparison of Resnet-50 network (a) SHO-net Confusion matrix; (b) SSHO-net Confusion matrix; (c) SINSHO-net Confusion matrix; (d) WOA-net Confusion matrix; (e) GA-net Confusion matrix; (f) PSO-net Confusion matrix; (g) GWO-net Confusion matrix.

3.3. Experiment 3 Classification of CORN Diseases

The optimal momentum, initial learning rate, maximum epoch, and validation frequency were selected through the ESHO-net model as presented in Table 6. The confusion matrix for the classification of Blight, Common Rust, Gray Spot, and Healthy Corn is shown in Figure 9. This includes metrics such as accuracy, sensitivity, precision, and recall, with an accuracy of 96.7% and a loss rate of 3.3%. The sensitivity is 94.7%, 99.0%, 89.1%, and 99.7%, while the precision is 94.5%, 98.5%, 90.1%, and 100%. The recall and sensitivity are 94.7%, 99.0%, 89.1%, and 99.7%, respectively.

Table 6. Optimal parameters for maize disease dataset.

Parameter	SGDM
Momentum	0.5
Initial learning rate	0.03943
Maximum epoch	32
Validation frequency	33

324 25.8%	4 0.3%	14 1.1%	0 0.0%	94.7% 5.3%
1 0.1%	386 30.7%	3 0.2%	0 0.0%	99.0% 1.0%
17 1.4%	2 0.2%	155 12.3%	0 0.0%	89.1% 10.9%
1 0.1%	0 0.0%	0 0.0%	349 27.8%	99.7% 0.3%
94.5% 5.5%	98.5% 1.5%	90.1% 9.9%	100% 0.0%	96.7% 3.3%

Figure 9. The optimal parameter confusion matrix is selected for the Maize disease classification dataset.

3.3.1. Compared with Pretrained Neural Network

The proposed evaluation methods (accuracy, sensitivity, accuracy, recall) were compared with unoptimized pretrained networks (Resnet 18, Google net, Inception v3, Alexnet, Densenet 201). The confusion matrix pair is shown in Figure 10 below:

317 25.2%	8 0.6%	32 2.5%	0 0.0%	88.8% 11.2%
5 0.4%	378 30.1%	2 0.2%	0 0.0%	98.2% 1.8%
22 1.8%	3 0.2%	138 11.0%	0 0.0%	84.7% 15.3%
0 0.0%	3 0.2%	0 0.0%	349 27.8%	99.1% 0.9%
92.2% 7.8%	96.4% 3.6%	80.2% 19.8%	100% 0.0%	94.0% 6.0%

(a) AlexNet

321 25.5%	3 0.2%	20 1.6%	0 0.0%	93.3% 6.7%
2 0.2%	389 30.9%	3 0.2%	0 0.0%	98.7% 1.3%
19 1.5%	0 0.0%	149 11.9%	1 0.1%	88.2% 11.8%
2 0.2%	0 0.0%	0 0.0%	348 27.7%	99.4% 0.6%
93.3% 6.7%	99.2% 0.8%	86.6% 13.4%	99.7% 0.3%	96.0% 4.0%

(b) DenseNet

319 25.4%	9 0.7%	41 3.3%	1 0.1%	86.2% 13.8%
10 0.8%	380 30.2%	3 0.2%	0 0.0%	96.7% 3.3%
12 1.0%	3 0.2%	128 10.2%	0 0.0%	89.5% 10.5%
3 0.2%	0 0.0%	0 0.0%	348 27.7%	99.1% 0.9%
92.7% 7.3%	96.9% 3.1%	74.4% 25.6%	99.7% 0.3%	93.5% 6.5%

(c) GoogleNet

313 24.9%	8 0.6%	37 2.9%	0 0.0%	87.4% 12.6%
10 0.8%	382 30.4%	7 0.6%	0 0.0%	95.7% 4.3%
18 1.4%	2 0.2%	128 10.2%	0 0.0%	86.5% 13.5%
3 0.2%	0 0.0%	0 0.0%	349 27.8%	99.1% 0.9%
91.0% 9.0%	97.4% 2.6%	74.4% 25.6%	100% 0.0%	93.2% 6.8%

(d) inception V3

301 23.9%	7 0.6%	28 2.2%	0 0.0%	89.6% 10.4%
4 0.3%	382 30.4%	1 0.1%	0 0.0%	98.7% 1.3%
38 3.0%	3 0.2%	143 11.4%	0 0.0%	77.7% 22.3%
1 0.1%	0 0.0%	0 0.0%	349 27.8%	99.7% 0.3%
87.5% 12.5%	97.4% 2.6%	83.1% 16.9%	100% 0.0%	93.5% 6.5%

(e) ResNet 18

Figure 10. Comparison of Traditional Deep Learning Confusion Matrices (a) AlexNet Confusion matrix; (b) DenseNet Confusion matrix; (c) GoogleNet Confusion matrix; (d) inception V3 Confusion matrix; (e) ResNet 18 Confusion matrix.

Through comparative analysis of confusion matrix, we evaluate five kinds of popular neural networks with strong classification ability. The results show that the accuracy index of these neural networks has been improved in different degrees, the improvement range is between 0.7% and 3.5%, among which DenseNet network has the best performance, reaching 96.0%, while the network with low accuracy is Inception V3, which is 93.2%.

Further comparing sensitivity, accuracy, and recall rates, we found that in the classification of blight, rust, gray spot, and health, sensitivity and recall rates increased by 1.4% to 8.5% and 0.3% to 3.3%, respectively, and the relatively improved ESHO-net model improved sensitivity and recall rates to varying degrees. In addition, the improvement of comparison accuracy ranges from 1.2% to 7%, $-0.7%$ to 2.1%, 3.5% to 15.7%, and 0 to 0.3%. Based on the above comparison results, it can be concluded that the ESHO-net model has excellent performance.

3.3.2. Optimize Resnet-50 Network with Other Optimization Algorithms

The proposed evaluation methods (accuracy, sensitivity, accuracy, recall) were compared with other optimization algorithms (SHO, SSHO, SINSHO, WOA, GA, PSO, GWO) to optimize the network. The confusion matrix pair is shown in Figure 11 below:

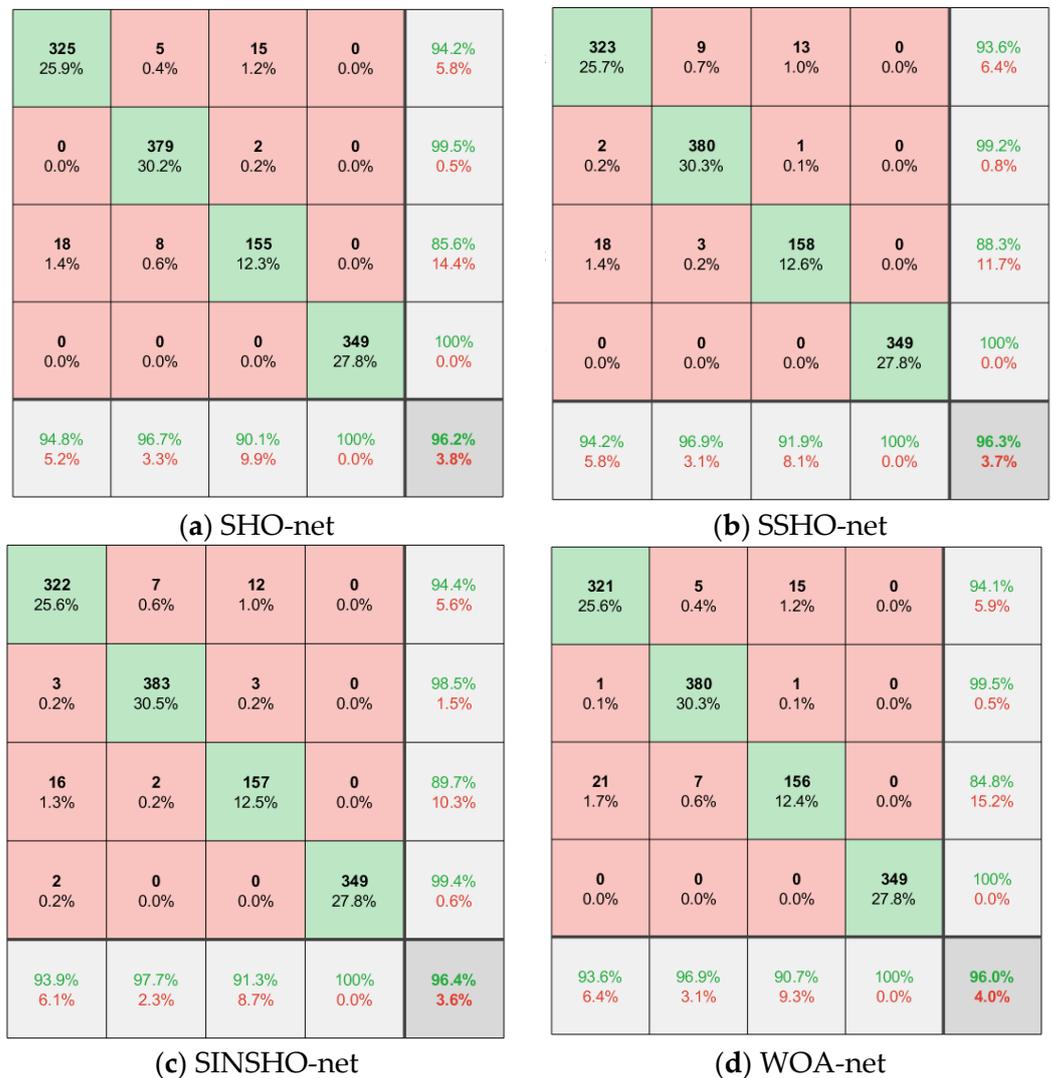


Figure 11. Cont.



Figure 11. Other optimization algorithms were used to optimize the confusion matrix comparison of Resnet-50 network (a) SHO-net Confusion matrix; (b) SSHO-net Confusion matrix; (c) SINSHO-net Confusion matrix; (d) WOA-net Confusion matrix; (e) GA-net Confusion matrix; (f) PSO-net Confusion matrix; (g) GWO-net Confusion matrix.

The comparison SHO, SSHO, SINSHO, and four optimization algorithms with strong optimization ability WOA, GA, PSO, and GWO are optimized for ResNet-50 at the same time to obtain the confusion matrix in Figure 6. Specifically, the ESHO algorithm improves the accuracy by 0.3% to 2.9%. In the classification of four diseases (Blight, Common-Rust, Gray spot, and Health), ESHO algorithm has improved sensitivity, accuracy, and recall rate compared with other optimization algorithms. Among them, the sensitivity of Blight, Gray-Leaf-Spot, and Health was increased by -0.7 – 2.6% . The accuracy is increased by -0.5% to 0.5% . The increase in recall is -0.6% to 7.3% . For the classification of Common-Rust, the sensitivity was increased by -0.3% to 0.6% . The accuracy is increased by 0.3% to 1.9% . The increase in recall is -2.3% to 6.4% . Considering the performance of various indicators and confusion matrix, it can be concluded that compared with other optimization algorithms, ESHO-net network has better classification effect and optimization performance for ResNet-50 network.

4. Conclusions

This study successfully demonstrated the strong potential of enhanced sea horse optimization algorithm (ESHO) and its optimized ResNet50 model (ESHO-NET) in agricultural

image recognition, especially in maize leaf disease and jade fungus image classification. The SHO algorithm is enhanced by chaotic map and sine cosine algorithm to achieve a balance of search behavior and solve the problem of excessive randomness and imbalance between exploration and exploitation in the original algorithm. The biggest advantage of the ESHO algorithm is reflected in the adaptive adjustment of ResNet50 parameters, which significantly reduces the burden of manual parameter tuning.

Specifically, Experiment I verifies that the ESHO algorithm outperforms the original and its optimization strategy optimized SHO algorithm in several performance metrics. In the second experiment, ESHO-net shows superior classification accuracy on the jade auricularia image dataset, which surgoes multiple comparison neural network models optimized by classical and swarm intelligence algorithms. In addition, the results of Experiment 3 prove that ESHO-net is significantly superior to other state-of-the-art models in terms of accuracy, sensitivity, and recall in the classification task of 3221 corn disease images dataset, which has a significant accuracy of 96.7% and a loss rate as low as 3.3%.

Although the above findings confirm the practical value of ESHO-net for image recognition tasks, the limitations of this study should also be noted, such as the range of disease types and the diversity of data sources. Future work will focus on these limitations and strengthen the generalization ability and practical application of the model by expanding the collection of disease types and multisource data. The conclusion of this study provides a solid foundation for further using sea horse optimization algorithm to promote intelligent image recognition in the agricultural field, and indicates a broad application prospect in the field of intelligent agriculture and precision agriculture.

Author Contributions: Conceptualization, S.Q.; methodology, S.Q.; software, S.Q.; validation, S.Q.; formal analysis, N.L.; investigation, N.L.; resources, Y.X.; data curation, X.H.; writing—original draft preparation, N.L.; writing—review and editing, N.L.; visualization, N.L.; supervision, Z.L.; project administration, Z.L.; funding acquisition, Z.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Scientific Research Project of the Department of Education of Jilin Province, grant number JJKH20210331KJ; “14th Five-Year Plan” national key Research and development Plan, grant number: 2023YFD1201602-1.

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Li, X.S.; Li, J.Y.; Yang, H.C.; Wang, Y.R.; Gao, S.C. Population interaction network in representative differential evolution algorithms: Power-law outperforms Poisson distribution. *Phys. A Stat. Mech. Its Appl.* **2022**, *603*, 127764. [[CrossRef](#)]
2. Fan, X.P.; Guan, Z.B. VGNet: A Lightweight Intelligent Learning Method for Corn Diseases Recognition. *Agriculture* **2023**, *13*, 1606. [[CrossRef](#)]
3. Dai, D.K.; Xia, P.W.; Zhu, Z.Y.; Che, H.L. MTDL-EPDCLD: A Multi-Task Deep-Learning-Based System for Enhanced Precision Detection and Diagnosis of Corn Leaf Diseases. *Plants* **2023**, *12*, 2433. [[CrossRef](#)]
4. Zeng, W.H.; Li, H.D.; Hu, G.S.; Liang, D. Lightweight dense-scale network (LDSNet) for corn leaf disease identification. *Comput. Electron. Agric.* **2022**, *197*, 106943. [[CrossRef](#)]
5. Emambocus, B.A.S.; Jasser, M.B.; Amphawan, A. A Survey on the Optimization of Artificial Neural Networks Using Swarm Intelligence Algorithms. *IEEE Access* **2023**, *11*, 1280–1294. [[CrossRef](#)]
6. Bahaa, A.; Sayed, A.; Elfangary, L.; Fahmy, H. A novel hybrid optimization enabled robust CNN algorithm for an IoT network intrusion detection approach. *PLoS ONE* **2022**, *17*, e0278493. [[CrossRef](#)] [[PubMed](#)]
7. Paharia, N.; Jadon, R.S.; Gupta, S.K. Optimization of convolutional neural network hyperparameters using improved competitive gray wolf optimizer for recognition of static signs of Indian Sign Language. *J. Electron. Imaging* **2023**, *32*, 023042. [[CrossRef](#)]
8. Wang, C.X.; Shi, T.T.; Han, D.N. Adaptive Dimensional Gaussian Mutation of PSO-Optimized Convolutional Neural Network Hyperparameters. *Appl. Sci.* **2023**, *13*, 4254. [[CrossRef](#)]
9. Xiong, C.Y.; Mo, H.W.; Fan, J.S.; Ren, W.C.; Pei, H.; Zhang, Y.H.; Ma, Z.W.; Wang, W.Y.; Huang, J. Physiological and Molecular Characteristics of Southern Leaf Blight Resistance in Sweet Corn Inbred Lines. *Int. J. Mol. Sci.* **2022**, *23*, 10236. [[CrossRef](#)]

10. Liu, H.; Guo, F.F.; Chen, X.L.; Wu, B.M. Temporal Progress and Spatial Patterns of Northern Corn Leaf Blight in Corn Fields in China. *Phytopathology* **2022**, *112*, 1936–1945. [[CrossRef](#)]
11. Ahangar, M.A.; Wani, S.H.; Dar, Z.A.; Roohi, J.; Mohiddin, F.; Bansal, M.; Choudhary, M.; Aggarwal, S.K.; Waza, S.A.; Dar, K.A.; et al. Distribution, Etiology, Molecular Genetics and Management Perspectives of Northern Corn Leaf Blight of Maize (*Zea mays* L.). *Phyton-Int. J. Exp. Bot.* **2022**, *91*, 2111–2133. [[CrossRef](#)]
12. Iseghohi, I.; Abe, A.; Meseka, S.; Mengesha, W.; Gedil, M.; Job, A.; Menkir, A. Reactions of provitamin-A-enriched maize to foliar diseases under field conditions in Nigeria. *Cereal Res. Commun.* **2023**. [[CrossRef](#)]
13. Micca-Ramerez, M.V.; Andrada, N.R. Damage function of common rust corn, *Puccinia sorghi*, applicable in the semiarid area. *Phytopathology* **2022**, *112*, 10.
14. Holan, K.; Whitham, S.A. Long-read draft genome assembly of *Puccinia sorghi*, the common rust pathogen of maize. *Phytopathology* **2022**, *112*, 79.
15. Nikzainalalam, N.; Copeland, D.; Wiggins, M.; Telenko, D.E.P.; Wise, K.A.; Jacobs, J.L.; Chilvers, M. Fungicide sensitivity of *Cercospora* spp, the causal agent of grey leaf spot disease on corn (*Zea-mays*). *Phytopathology* **2022**, *112*, 109.
16. Luis, J.M.S.; Nicolli, C.P.; Duffeck, M.R.; Robertson, A.E.; Smith, D.L.; Allen, T.; Bissonnette, K.; Sjarpe, D.; Check, J.C.; Chilvers, M.; et al. Validation of binary logistic regression models for assessing the risk for gray leaf spot of maize prior to planting. *Phytopathology* **2022**, *112*, 38.
17. Luis, J.M.S.; Duffeck, M.R.; Frey, T.S.; Nicolli, C.P.; Robertson, A.E.; Smith, D.L.; Allen, T.; Bissonnette, K.; Chilvers, M.; Check, J.C.; et al. Vertical progress of gray leaf spot of maize from an in-field source of inoculum as influenced by environment and hybrid resistance. *Phytopathology* **2022**, *112*, 126.
18. Wen, L.; Li, X.Y.; Gao, L. A transfer convolutional neural network for fault diagnosis based on ResNet-50. *Neural Comput. Appl.* **2020**, *32*, 6111–6124. [[CrossRef](#)]
19. Zhao, S.; Zhang, T.; Ma, S.; Wang, M. Sea-horse optimizer: A novel nature-inspired meta-heuristic for global optimization problems. *Appl. Intell.* **2023**, *53*, 11833–11860. [[CrossRef](#)]
20. Zhao, B.X.; Zhu, J.Z.; Hu, Y.B.; Liu, Q.M.; Liu, Y. Mapping Landslide Sensitivity Based on Machine Learning: A Case Study in Ankang City, Shaanxi Province, China. *Geofluids* **2022**, *2022*, 2058442. [[CrossRef](#)]
21. Mirjalili, S. SCA: A Sine Cosine Algorithm for solving optimization problems. *Knowl.-Based Syst.* **2016**, *96*, 120–133. [[CrossRef](#)]
22. Huang, B.; Liu, J.H.; Zhang, Q.; Liu, K.; Li, K.; Liao, X.Y. Identification and Classification of Aluminum Scrap Grades Based on the Resnet18 Model. *Appl. Sci.* **2022**, *12*, 11133. [[CrossRef](#)]
23. Fu, Y.S.; Song, J.; Xie, F.X.; Bai, Y.; Zheng, X.; Gao, P.; Wang, Z.T.; Xie, S.Q. Circular Fruit and Vegetable Classification Based on Optimized GoogLeNet. *IEEE Access* **2021**, *9*, 113599–113611. [[CrossRef](#)]
24. Wang, C.; Chen, D.L.; Hao, L.; Liu, X.B.; Zeng, Y.; Chen, J.W.; Zhang, G.K. Pulmonary Image Classification Based on Inception-v3 Transfer Learning Model. *IEEE Access* **2019**, *7*, 146533–146541. [[CrossRef](#)]
25. Zhao, C.; Shuai, R.J.; Ma, L.; Liu, W.J.; Hu, D.; Wu, M.L. Dermoscopy Image Classification Based on StyleGAN and DenseNet201. *IEEE Access* **2021**, *9*, 8659–8679. [[CrossRef](#)]
26. Mirjalili, S.; Lewis, A. The Whale Optimization Algorithm. *Adv. Eng. Softw.* **2016**, *95*, 51–67. [[CrossRef](#)]
27. Drezner, Z.; Drezner, T.D. Biologically Inspired Parent Selection in Genetic Algorithms. *Ann. Oper. Res.* **2020**, *287*, 161–183. [[CrossRef](#)]
28. Kennedy, J.; Eberhart, R. Particle swarm optimization. In Proceedings of the ICNN'95—International Conference on Neural Networks, Perth, WA, Australia, 27 November–1 December 1995; Volume 1944, pp. 1942–1948.
29. Wang, Z.D.; Xie, H.M. Wireless Sensor Network Deployment of 3D Surface Based on Enhanced Grey Wolf Optimizer. *IEEE Access* **2020**, *8*, 57229–57251. [[CrossRef](#)]

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