Journal of Network Intelligence
Taiwan Ubiquitous Information
 ISSN 2414-8105
Volume **9**, Number **3**, August **2024**

Lawn Plant Identification Based on Meta-Heuristic Neural Network

Xiao-Ning Tao*

Changsha Environmental Protection Vocational College, Hunan 410004, P. R. China
13975197607@163.com

Ying Huang

School of Information Technology and Engineering
St. Paul University Philippines, Tuguegarao 3500, Philippines
xa3506@163.com

*Corresponding author: Xiao-Ning Tao
Received November 22, 2023, revised February 26, 2024, accepted May 5, 2024.

Abstract. *The lawn plant identification system can be combined with automatic weeding robots and other equipment to achieve fully automated lawn maintenance and greatly reduce labour costs. Through automatic recognition of lawn vegetation, the health condition of lawn can be judged more accurately and the presence of noxious weeds can be detected in a timely manner, thus guiding a more scientific and reasonable weeding and fertilising work and reducing unnecessary waste of resources. Therefore, this work proposes to use the advantages of Convolutional Neural Network (CNN) to extract image features and combine with Support Vector Machine (SVM) to reduce the time of image training and complete the classification and recognition of lawn plant knowledge. Firstly, the images of common lawn plants are collected through the methods of downloading the data set publicly available on the Internet and taking photos in the field, and the images are enhanced and preprocessed to establish the lawn plant image data set. Then, the advantages of different convolutional neural networks, such as LeNet, AlexNet, and VGG, are compared, and the AlexNet network model is selected to extract image features. Finally, a new activation function is proposed and the parameters of the CNN network are optimised using the Atomic Search Optimisation (ASO) algorithm to further improve the accuracy and efficiency of image recognition. The optimised CNN-SVM model is trained and tested, and comparison experiments are done with the original CNN-SVM model. The experimental results show that the optimised CNN-SVM model has faster convergence speed and higher recognition accuracy and efficiency, which helps to improve lawn management and reduce maintenance cost.*
Keywords: convolutional neural network; lawn plants; image recognition; support vector machine; activation function; heuristic optimisation algorithm

1. **Introduction.** Research on lawn plant identification can help to improve lawn management and reduce maintenance costs, and is also conducive to the development of intelligent lawn systems and computer vision technology, which has important application and academic values [1, 2].

Computer vision includes many aspects and image classification is one of the hot topics. Image classification can be understood as the process of using computers and using algorithms to find out the category labels of images and distinguish the categories of different images. Image classification has the following steps: image preprocessing, feature

extraction, and classifier classification. Among these steps, feature extraction is particularly important and determines the accuracy of image classification. In the traditional image classification task, the texture and colour of the image are often extracted to make judgments, this method has a more satisfactory effect on the classification of simple images, but for the classification of complex images can not be achieved. Therefore, image classification algorithms based on convolutional neural nets, which are more effective in classifying complex images, have gradually become a hot spot.

Early classification of plant images is firstly to segment the image region of plants individually [3], to eliminate the irrelevant background information, leaving only the plant information region which is valuable for classification, and then extract the feature information of the plant information region with artificially designed feature extraction algorithms, and finally send the information extracted with feature extraction algorithms to the classifier to get the classification results, to complete the whole plant image classification and recognition work [4, 5]. This widely used plant classification and recognition technology requires human segmentation of plant images, and the feature extraction of plant images also requires human intervention, such as designing algorithms to extract the colour and outline of plants, which is more complicated. Traditional plant image segmentation methods are based on image segmentation techniques, which is the process of dividing a complete image into smaller pieces, each of which should represent some meaningful part of the image. Although a certain degree of classification can be achieved by using artificially designed feature extraction methods to classify plant images, their classification effectiveness and efficiency are greatly limited. This is because the extraction of features from plant images relies on the segmentation of plant images, and the quality of segmentation depends on human control, with uncertainty, which directly affects the accuracy of feature extraction, and thus the final classification accuracy [6]. In addition, feature extraction of plant images requires human design, which requires experience and lacks strong generalisation ability.

Currently, with the explosive development of machine learning technology, machine learning-based feature methods have become mainstream [7, 8, 9]. Using machine learning algorithms, feature rules are learnt based on the training dataset and then the learnt rules are applied to new images. The method requires a large number of training datasets and computational resources, but it is more effective and can be adapted to different fields of image classification. Image classification algorithms based on convolutional neural nets are more effective in classifying complex images and have gradually become a hot spot in related fields [10]. Therefore, the research objective of this work is to extract image features using convolutional neural net technique and to design a new hybrid model for automatic identification of lawn plant species by combining with support vector machine.

1.1. **Related Work.** Convolutional neural networks in deep learning have been developing rapidly in the image field in recent years and are undergoing unprecedented technological changes. Compared with traditional image extraction algorithms, convolutional neural networks have the unique advantage of being able to extract deep information from images.

Convolutional neural networks generally consist of an input layer to input the image, a convolutional layer to extract features from the image, a pooling layer to downscale the data extracted from the previous convolutional layer to prevent overfitting of the results obtained, a fully connected layer to spread and selectively classify the final extracted data, and a final output layer to output the results. Convolutional neural networks use local connectivity and weight sharing, which helps to reduce the number of weights and improve computational efficiency, as well as reduce model complexity and avoid overfitting. As

a result, it can directly take the image as input, thus avoiding the feature extraction process, which is both complex and time-consuming in traditional recognition algorithms, thus simplifying the whole algorithmic process. Currently, the structure of convolutional neural networks is divided into three main types [11]: the LeNet, AlexNet and VGG.

LeNet consists of two convolutional layers, two pooling layers and three fully connected layers. Its advantage lies in the fact that it adopts convolutional layers and pooling layers to extract the features of the image, which can effectively reduce the number of parameters in the network to avoid the problem of overfitting. Ma et al. [12] proposes a 1D-LeNet model for hyperspectral image classification. By using 1D convolutions, the model can better capture spectral features. Experiments on two datasets show 1D-LeNet outperforms the baseline LeNet and achieves state-of-the-art results. The study provides a good reference for adapting LeNet to hyperspectral data. Zaibi et al. [13] proposes a lightweight CNN based on LeNet for traffic sign recognition. By reducing parameters and depth, the model achieves real-time performance on an embedded device with only 1.5% drop in accuracy compared to the original LeNet. This demonstrates the potential of applying a compact LeNet model for automated driving systems.

The AlexNet model structure consists of 5 convolutional layers and 3 fully connected layers.AlexNet uses ReLU activation function to speed up the training process and improve the accuracy. In addition, AlexNet uses Dropout technique and Softmax classifier for classification. Lee et al. [14] fuses LeNet and AlexNet models for classifying digestive organs from endoscopic images. The hybrid model inherits simple structure from LeNet and deeper features from AlexNet. On a capsule endoscopy dataset, it delivers 96.3% accuracy for four-class classification, showing advantages over both individual models. Kaur and Gandhi [15] provides an empirical analysis of image recognition and classification using machine learning and transfer learning approaches, focusing on the accuracy rates of three architectures in Convolutional Neural Networks (CNNs): AlexNet, GoogleNet, and ResNet. The study evaluates the accuracy rates of these architectures trained with a specific number of resources and compares their accuracy after each training day. The research employs regression analysis to identify how one-day training improves the accuracy of these architectures. The findings suggest that AlexNet is more accurate in machine learning image recognition and classification and requires less time for training than GoogleNet and ResNet. However, due to having fewer layers, AlexNet is less accurate in transfer learning. This analysis provides valuable insights into the performance of AlexNet compared to other CNN architectures in image recognition and classification tasks. Ye et al. [16] proposes a lightweight model that combines VGG-16 and U-Net networks for remote sensing image classification. The study focuses on improving model accuracy while reducing memory size. The authors report an improved model accuracy of 98% with a reduced memory size of 3.4 MB. The proposed model demonstrates enhanced classification and convergence speed, making it suitable for remote sensing images with few target feature points and low accuracy. The study highlights the potential application of the model in remote sensing image classification, particularly for images with limited target feature points and low pixels. Saju and Rajesh [17] focuses on classifying cataracts into two categories: normal cataracts and cataracts. The authors process retinal fundus image data and employ VGG-19, ResNet-50, and ResNet-101 for classification. The results indicate average accuracies of 91.06% for VGG19, 93.50% for ResNet-50, and 93.50% for ResNet-101 across all retinal classes. The study demonstrates the potential of deep learning models, particularly VGG-19 and ResNet, in cataract detection, highlighting their application in medical image analysis.

Meta-inspired optimization algorithms are a class of optimization algorithms that mimic natural phenomena or human intelligence [18,19,20], which can find globally optimal or

near-optimal solutions in a large search space, such as Particle Swarm Optimization (PSO) algorithms [21], Ant Colony Optimization (ACO) algorithms [22], and Bee Colony Optimization (ABC) algorithms [23]. Meta-inspired optimisation algorithms can be used to optimise the structure and parameters of the convolutional neural network, such as the size, number, and shape of the convolutional kernel, the type and step size of the pooling layer, the number of nodes in the fully-connected layer, the choice of activation function, and the adjustment of the learning rate. This can improve the generalisation ability and accuracy of the convolutional neural network, and at the same time reduce the risk of overfitting and underfitting. Sun et al. [24] proposed a method to optimise the CNN architecture with a genetic algorithm, called GA-CNN.This method encodes the number of layers, convolutional kernel sizes, activation functions, etc., and then selects, crossovers, and mutates them with a genetic algorithm, so as to obtain the optimal CNN architecture.

1.2. **Motivation and contribution.** The appearance characteristics of lawn plants are often affected by environmental factors, such as light, temperature, moisture, soil, pests and diseases, etc., resulting in the same kind of lawn plants may show different colours, shapes, sizes, densities, etc. under different conditions, which increases the difficulty of identification. There may be crossbreeding, mutation, hybrids and other phenomena among lawn plants, resulting in certain lawn plant characteristics not meeting the original classification standards or the emergence of new characteristics, making identification more complex and difficult.

Therefore, in order to improve the accuracy and recognition efficiency of lawn plant classification, this work implements the extraction of plant image features through convolutional neural networks and feeds the features into a support vector machine (SVM) to achieve classification and recognition of lawn plants. The main innovations and contributions of this work include:

(1) Images of common lawn plants were collected by methods such as downloading publicly available datasets from the Internet and taking photographs in the field, and the images were enhanced and pre-processed to create a lawn plant image dataset.

(2) The advantages of different convolutional neural networks such as LeNet, AlexNet, and VGG are compared and the AlexNet network model is selected to extract image features.

(3) A new activation function is proposed and an atomic search optimisation (ASO) algorithm is used to optimise the parameters of the CNN network to further improve the accuracy and efficiency of image recognition. The ASO-CNN-SVM model is trained and tested, and comparison experiments with the original CNN-SVM model are done.

2. Establishment of a lawn plant image data set.

2.1. **Plant species selection.** There are many kinds of common lawn plants, and different lawn plants have different characteristics and adaptability. In this work, four common lawn plant species and their related information [25] were selected as shown in Table 1.

Looking at the shape and colour of the blades, Zoysia has flat or slightly involute blades that are dark green or grey-green; Axonopus compressus has opposite, ovate or heart-shaped blades that are light green or purplish-red; Golf green has elongated, linear blades that are dark green or blue-green; and Bermudagrass has flat, lance-shaped blades that are light green or grey-green. In terms of stem morphology and length, the stems of Zoysia and Bermudagrass have creeping rhizomes that can extend above ground to form dense turf; the stems of Axonopus compressus also have creeping stems but are shorter

Table 1. Types of common lawn plants and their related information.

Lawn plant	Genus (taxonomy)	Morphological characteristic	Use
Zoysia	Gramineae Knotweed	Cross-travelling rhizomes, flattened or slightly involute leaf blades, spikelike inflorescences	Sports grounds and waterfront lawns
Axonopus compressus	Carpetweed, Labiatae	Creeping stems, opposite ovate or heart-shaped leaves, purple, white or pink spikes	Groundcover or border plants for flower beds
Golf green	Ryegrass, family Gramineae	Erect stems, slender linear leaf blades, conical inflorescences, ovate caryopsis	Golf course greens or premium turf grasses
Bermudagrass	Bermuda, family Graminae	Creeping stems, flattened lanceolate leaf blades, conical inflorescences, ovate caryopsis	Lawns in parks, courtyards or sports grounds

and less likely to form turf; and the stems of Bermudagrass are upright and longer, forming loose turf. Examples of the four common lawn plants are shown in Figure 1.

(a) Zoysia

(b) Axonopus compressus

(c) Golf green

(d) Bermudagrass

Figure 1. Examples of 4 common lawn plants

2.2. **Image data acquisition.** For the four common lawn plants, the method used was on-site field photography. This method is time-consuming, labour-intensive and less rewarding. Since taking photographs may suffer from the problem of under-representation and broadness of the collected data samples, which makes the model overfitting during the training process, the use of data enhancement techniques is needed to improve this problem.

A search engine can be used to find information about the four common lawn plants on the Internet, download the relevant images, and screen and organise them to create a dataset, a method that ensures that the dataset comes from a wide range of sources and saves time.

This paper focuses on collecting images of four common lawn plants using web search and manual photography. The collection methods include (1) using a smartphone to take images of lawn plants from multiple angles in different backgrounds, and (2) selecting images of lawn plants from a variety of venues and different seasons through a search engine, and then downloading the images using a download tool.

After the above steps, we need to classify these 4 lawn plants obtained by proper manual screening, this is to avoid image duplication and misclassification which leads to unsatisfactory training results. Through our screening, about 1000 images of the original plant dataset were obtained. After the classification and screening process, we obtained four image categories, each containing 50 to 200 images. However, the distribution of these data samples is uneven. Therefore, during the training process of classifying and recognising the images, it is necessary to ensure that the number of samples per sample tends to be the same, which is referred to as balancing the data. The main purpose of this is to avoid the model training effect being affected by a larger number of samples and to prevent the model from over-fitting one of the classifications during the training process of performing image classification. In addition, due to the overall small size of the collected image dataset, the desired training effect may not be achieved and model underfitting may occur. All these problems can lead to inaccurate parameters of the trained model, which in turn affects the classification ability and effectiveness of the model. To solve these problems, the dataset needs to be balanced and expanded. We can choose to add a smaller number of sample images while deleting a larger number of sample images to ensure that the number of samples in the dataset for each category reaches a certain balance.

In this study, image enhancement techniques were used to expand the dataset by selecting 6 image enhancement methods: random rotation, random distortion, Gaussian noise, motion blur, random brightness, and random saturation. These 6 methods were grouped into 3 categories: graphic change, image blur, and colour change. Each image has several more copies of different types because of image enhancement, the sample size of a single class of image is increased dramatically, which improves the generalisation of the model, and can effectively avoid the model from model training underfitting. The robustness and size of the dataset are effectively improved by the image enhancement technique. The number of images of each lawn plant is increased and the size of the image dataset becomes about 5000 images.

The dataset of 5000 lawn plant images obtained after image enhancement is named Lawn Plants DataBase (LPDB). In this study, the LPDB dataset was managed with the method of image sequence tag and file type tag management, which facilitates the judgement of image training right and wrong.

2.3. **Dataset preprocessing.** After getting the LPDB dataset the images need to be preprocessed for training. The preprocessing of the dataset has two main purposes. On

the one hand, it can unify the format of the images so as to improve the success rate of model training. On the other hand, it can make the size of the original images uniform to conform to the standards of the model training and testing input sets, thus improving the effectiveness of image feature extraction. Now, the following preprocessing is done on the images in the dataset.

(1) Training and testing datasets: two kinds of non-repeating datasets, LPDB_train training set and LPDB_test testing set, are obtained using the random allocation method. Ninety per cent of the number of images of each category was randomly selected as the LPDB_train training set (4500), and ten per cent of it was selected as the LPDB_test test set (500).

(2) Image geometric transformation: Convert all the dataset image formats to RGB three-channel format uniformly. The size of the dataset was changed to 128×128×3. This adjustment reduces the computation of the computer during training.

(3) Image data feature scaling: when we read the image format as input during training data, we need to transform the image data into machine learning input data format Tensor (tensor) format for feature extraction. In this study a normalisation method is used to scale the input of training data in the interval [-1, 1].

$$O = \frac{I - E(I)}{S(I)} \tag{1}$$

where I is the input pixel value, $E(I)$ denotes the pixel mean of the input image and $S(I)$ denotes the standard deviation of the input pixels.

$$E(I) = \frac{1}{n}\sum_{i=1}^{n} I_i \tag{2}$$

$$S(I) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(I_i - E(I))^2} \tag{3}$$

3. **Comparison of CNN models.** In this work, a simple comparison of three network models, LeNet [26], AlexNet [27], and VGG-16 [28], is shown in Table 2.

Table 2. Comparison of CNN Models

	Input image	**Convolution kernel**	**Pooling**	**Activation function**	**Network depth**
LeNet	32 x 32 x 1	5 x 5 x 1	2×2 Maximum Pooling	Sigmoid	2+2+2+1
AlexNet	224 x 224 x 3	11 x 11 x 3 5 x 5 x 3 3×3×3	3×3 Maximum Pooling	ReLU	5+3+2+1
VGG-16	224 x 224 x 3	2×2×3	3×3 Maximum Pooling	ReLU	13+5+3+1

In this paper, three network models, LeNet, AlexNet, and VGG, are compared and experimented, and the output categories are uniformly changed to the names of four lawn plants, and other network parameters remain unchanged. The network is built according to the structure of LeNet, AlexNet, and VGG-16 models, and the empirical parameters other than network parameters are determined, i.e., learning rate, sample

size, and training batch. The pre-processed LPDB_train training set is input into the pre-training model for training, and the training results are iterated. The LPDB_test test set is put into the trained model for testing and analysing the results. The LeNet network model, AlexNet network model, and VGG-16 network model are built using Python's Pytorch framework.

After several experiments, it is found that the training process and test results tend to be optimal when the empirical parameters of the three networks are set as shown in Table 3, Table 4 and Table 5, respectively. A comparison of the test results of the three networks is shown in Figure 2. The combined training time and recognition accuracy can be initially found that using AlexNet network for training our dataset is the best choice, so in this work, AlexNet network is used as an image feature extractor for LPDB dataset.

Table 3. Empirical parameter settings for LeNet network models

Parameters	**Numerical value**
learning-rate	0.004
batch-size	24
epochs	30

Table 4. Empirical parameter settings for the AlexNet network model

Parameters	**Numerical value**
learning-rate	0.0004
batch-size	24
epochs	30

Table 5. Empirical parameter settings for the VGG-16 network model

Parameters	**Numerical value**
learning-rate	0.0002
batch-size	24
epochs	30

The convolutional layer in AlexNet network performs feature extraction of the input image using the convolution operation, which is computed as shown below:

$$C_i = \sigma\left(\sum_{j=1}^{m} W_{ij} * I_j + b_i\right) \tag{4}$$

where C_i denotes the i-th convolutional layer activation value, I_j denotes the j-th input image, W_{ij} denotes the convolutional kernel parameters and b_i denotes the bias term.

A non-linear mapping of the output of the convolutional layer to increase the expressive power of the network is often used with the ReLU (Rectified Linear Unit) activation function.

$$f(x) = \max(0, x) \tag{5}$$

A Local Response Normalisation (LRN) technique is used between the convolutional and pooling layers, which is used to suppress the response of neighbouring neurons, enhancing the larger response of the neurons and improving the generalisation ability of the network.

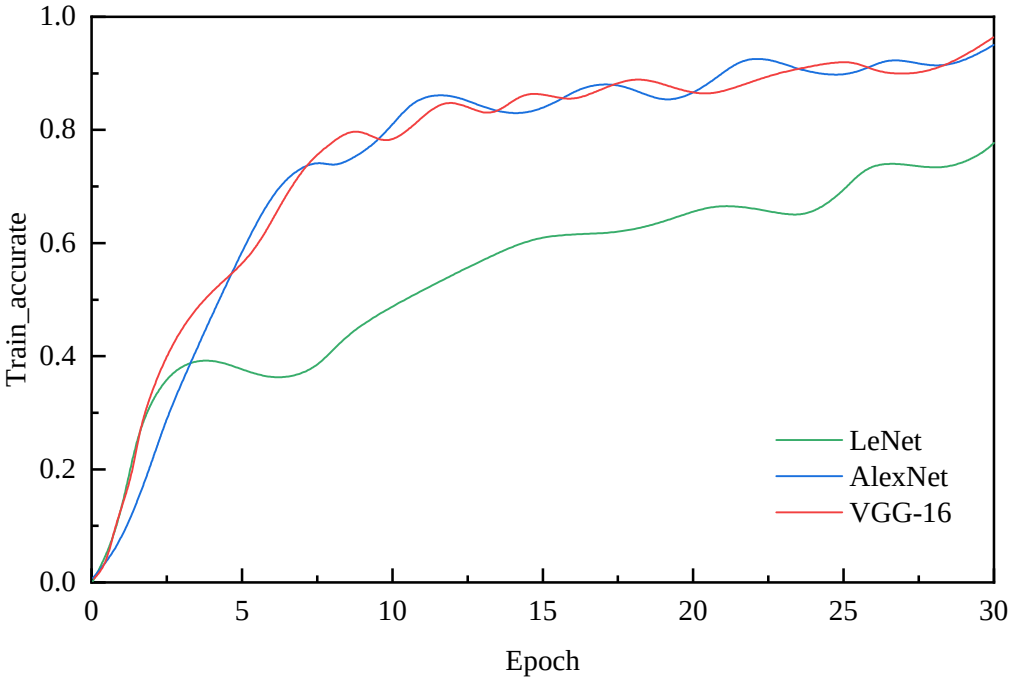

Figure 2. Comparison of the results of 3 network model training tests

$$b_{x,y}^{i} = \frac{a_{x,y}^{i}}{\left(k + \alpha \sum_{j=\max(0,i-n/2)}^{\min(N-1,i+n/2)} \left(a_{x,y}^{j}\right)^{2}\right)^{\beta}} \tag{6}$$

where $b_{x,y}^{i}$ denotes the normalised value of the i-th feature map at position (x, y), $a_{x,y}^{i}$ denotes the original value of the i-th feature map at position (x, y), n denotes the neighbourhood size, and N denotes the number of feature maps. k, α, and β are hyper-parameters which are usually taken as $k = 2$, $\alpha = 10^{-4}$ and $\beta = 0.75$.

The feature map output from the pooling layer is spread as a one-dimensional vector, connected to a fully connected layer, and the high-level semantic features of the image are extracted through multiple fully connected layers. The structure of AlexNet network is shown in Figure 3.

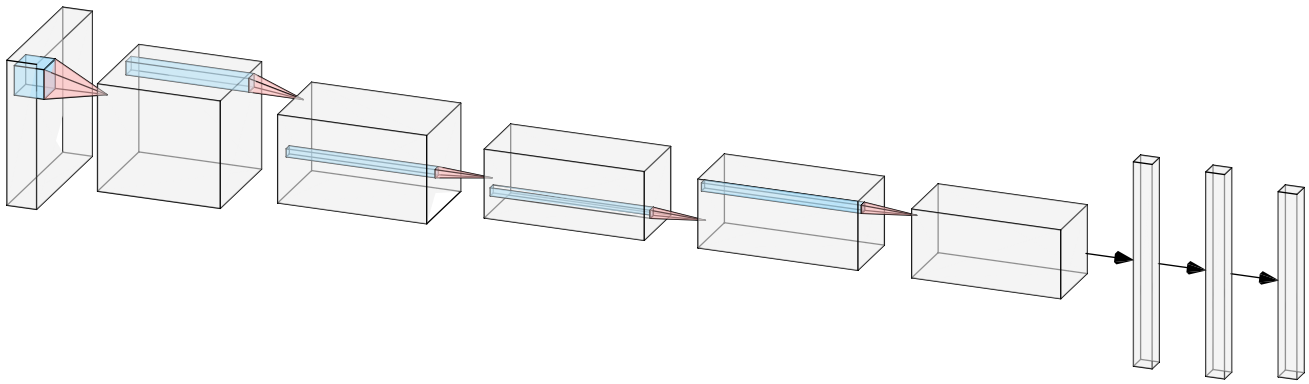

Figure 3. AlexNet network structure

4. ASO-CNN-SVM model construction.

4.1. Optimisation of activation function.

In order to improve the robustness and generalisation of convolutional neural networks, the activation function used in CNNs is improved in this work. The earlier activation function used is the Sigmoid function whose function image is shown in Figure 4.

$$f(x) = \frac{1}{1+e^{-x}} \tag{7}$$

From the image of the Sigmoid function, it can be seen that after the input values are

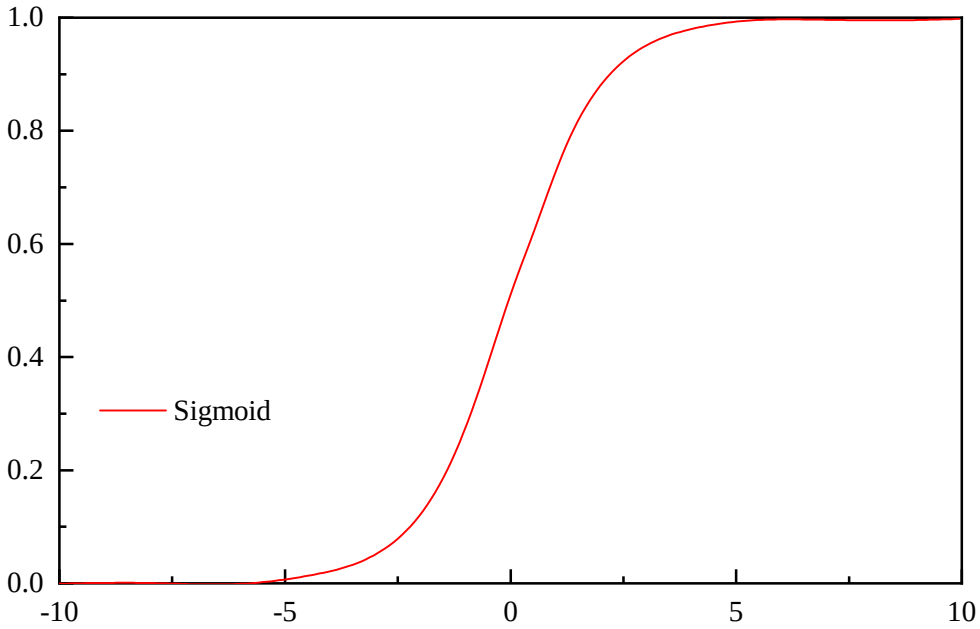

Figure 4. Sigmoid function

input to the Sigmoid function, the output values will be compressed in the range of 0 to 1. When the Sigmoid function is back-propagating the gradient update during the training process, because the output values are compressed in the range of 0 to 1, it results in the gradient seeking when some of the gradient parameters will be close to 0, which slows down the updating of the network's weighting parameters, and makes it difficult for the network to be training.

The present study is inspired by the ReLU function and the Tanh function, using the sparse non-saturation property of the positive half-axis of the ReLU [29] and the saturation property of the negative half-axis of the Tanh function [30, 31], which are combined into a new activation function named R-Tanh activation function, which is computed as shown in Equation (8), and its image is shown in Figure 5.

$$f(x) = \begin{cases} \frac{1-e^{-2x}}{1+e^{-2x}} & x < 0 \\ x & x \geq 0 \end{cases} \tag{8}$$

The proposed R-Tanh activation function mitigates the vanishing gradient. When the input value is greater than 0, the derivative of the R-Tanh activation function is 1, which is both incompressible and sparse. The R-Tanh activation function solves the problem by suppressing the gradient decay when the input value is less than 0. The R-Tanh activation function can still obtain activation values even when the value falls on the negative semiaxis, thus maintaining a part of the feature information of the image to be

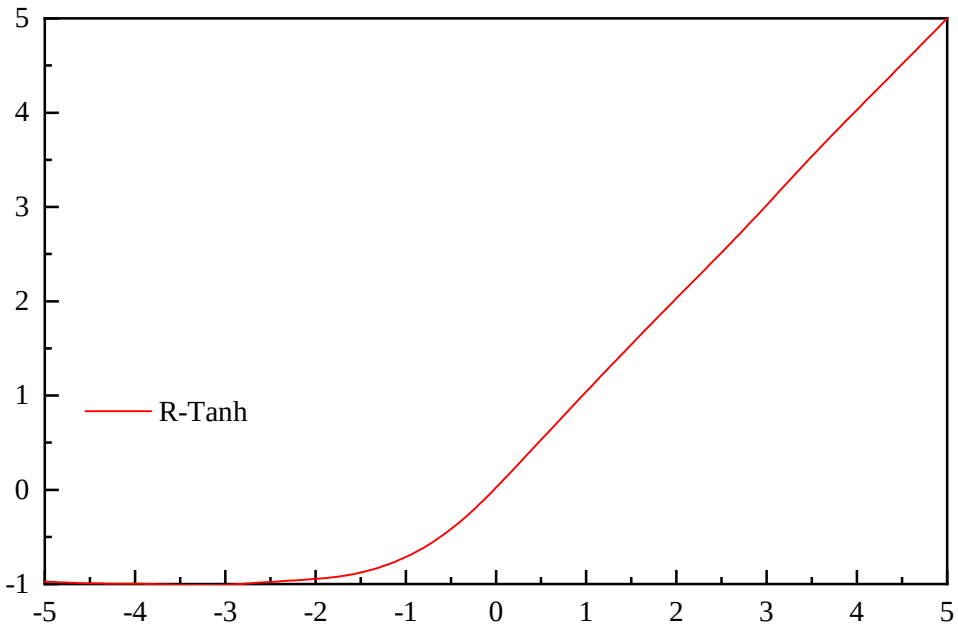

Figure 5. TReLU activation function

used and trained by the later network structure. The R-Tanh activation function has a soft saturation in the negative semiaxis interval, which provides a non-zero input for the next layer of neuron computation, and therefore the output image features are robust to data noise, thus improving the network's ability to recognise images.

4.2. **Validation of the activation function.** In the experiment to validate the classification effect of the improved activation function, the improved activation function is validated using the FWDB dataset with a two-layer convolutional neural network with four experimental control groups.

The classification accuracy can be obtained by training the convolutional neural network on using different activation functions as shown in Table 6.

Table 6. Performance Comparison of Activation Functions

Activation function	**Accuracy/%**
R-Tanh	81.9
ReLU	80.3
Tanh	78.6
Sigmoid	77.7

Sigmoid accuracy is lower than all other types of activation functions, which indicates that the saturated nature of the Sigmoid function affects its effectiveness. Tanh results are better than Sigmoid because it maintains neuron activity in the negative intervals, which makes the backward weighted features trained more accurately. The experimental accuracy of the R-Tanh function proposed in this paper is 81.9%, which is higher than the other three sets of activation functions. The main reason for the higher performance of the R-Tanh function is the negative semi-axis which does not lose features like ReLU (retains some negative features), and at the same time utilises the linear mapping non-compression property of the positive interval of the ReLU function, thus improving the recognition ability of the network.

4.3. **Construction of CNN-SVM model.** CNNs learn abstract features with a hierarchical structure in an image, and SVMs can benefit from the learning of these high-level features. By using the features extracted by the CNN as input to the SVM, the key information in the image can be better captured and the performance of image recognition can be improved.

The features generated by CNN are usually high-dimensional, and SVM is relatively good at handling high-dimensional data. Using CNN extracted feature vectors as input to SVM can reduce the dimensionality of the data, making SVM more effective in processing complex lawn plant image data.SVM is robust to noise to a certain extent. By feeding the CNN extracted features into SVM, the robustness of the model to image noise can be improved to some extent. Therefore, in this work, the multidimensional feature vectors extracted from the convolutional layer are used as the input data for SVM, so as to achieve the combination of CNN model and SVM model, as shown in Figure 6.

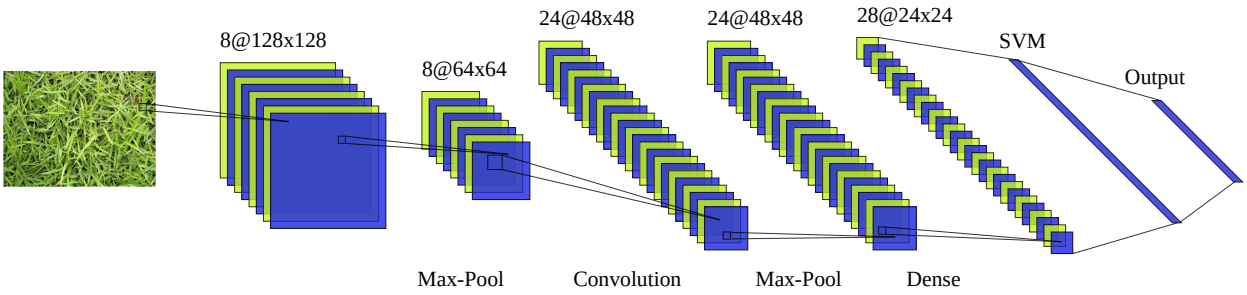

Figure 6. CNN-SVM Models

Let n samples of lawn plant images containing m-dimensional features be $X = (x_1, x_2, \ldots, x_n)$. The jth node of the lth hidden layer is $x_{l,j}$, which is represented as:

$$x_{l,j} = f\left(\sum_{j \in M_j} x_{l-1} * k_{l,j} + b_{l,j}\right) \tag{9}$$

where $k_{l,j}$ and $b_{l,j}$ are the weight and bias of node j in layer l, $*$ is the convolution, and $f()$ is the R-Tanh function proposed in this paper.

Let the convolution size be $h \times w$, then the pooling is done as shown below:

$$g(x) = \frac{\sum_{k=1}^{h \times w} x_k}{h \times w} \tag{10}$$

Let $M = n/h \times w$, then the sample $X = (x_1, x_2, \ldots, x_n)$ becomes $X' = (x_1, x_2, \ldots, x_M)$ by convolution and pooling.

$$x'_j = f\left(\sum_{i=1}^{M} a_{ij}(x_i^{l-1} * k_{l,j}) + b_{l,j}\right), \sum a_{ij} = 1, 0 \le a_{ij} \le 1 \tag{11}$$

In using CNN as classification, assume that the CNN output of input sample x_k is y_k, and the error between y_k and its actual category d_k is δ_k.

$$\delta_k = (d_k - y_k) y_k (1 - y_k) \tag{12}$$

The error sum E for all samples is:

$$E = \frac{1}{2} \sum_{k=1}^{M} (d_k - y_k)^2 \tag{13}$$

Taking the minimum value of E as the objective function, the deep CNN layer-by-layer weights and thresholds are continuously solved to finally obtain a stable deep CNN model.

4.4. **Parameter optimisation of CNN.** As mentioned in the previous section, by using the features extracted by CNN as inputs to SVM, the key information in the image can be better captured and the performance of image recognition can be improved. However, combining CNN and SVM may increase the computational complexity of the model, especially on large-scale datasets. This may lead to an increase in training and inference time.

The main reason for the increase in training and inference time is that hyperparameter tuning is more difficult. Combining two different models requires careful tuning of the hyperparameters to ensure that they work well together. This may require more tuning work.

In order to solve this problem, the Atomic Search Optimisation (ASO) algorithm [31] is used to optimise the CNN network parameters to further improve the accuracy and efficiency of image recognition. The ASO algorithm is essentially a search process that determines the optimal solution based on the motion state of the atoms, the state of motion of the atoms themselves, and the combined gravitational and repulsive relationships between the atoms to determine their motion state.

In the ASO process, the motion of atoms is always a spatial traversal search around atoms with large mass, and the mass of atoms is closely related to the fitness of atoms. $Fit_{min}(t)$ and $Fit_{max}(t)$ denote the minimum and maximum values of fitness after t iterations, respectively. Assume that the atom size is N, the mass of the ith atom is $m_i(t)$, and the estimated mass is $M_i(t)$.

$$m_i(t) = \frac{M_i(t)}{\sum_{j=1}^{N} M_j(t)} \tag{14}$$

$$M_i(t) = e^{\left(-\frac{Fit_i(t)-Fit_{min}(t)}{Fit_{max}(t)-Fit_{min}(t)}\right)} \tag{15}$$

where $Fit_i(t)$ is the fitness value of individual i after the tth iteration.

At the t-th iteration, the force of atom j on atom i in one dimension is $F_{ij}(t)$, which is expressed as:

$$F_{ij}(t) = -\eta(t)\left[2(h_{ij}(t))^{13} - (h_{ij}(t))^{7}\right] \tag{16}$$

$$\eta(t) = \alpha\left(1 - \frac{t-1}{T}\right)^{3} e^{\left(-\frac{20t}{T}\right)} \tag{17}$$

where T is the total number of iterations and α is a constant.

$$h_{ij}(t) = \begin{cases} h_{\min}, & \frac{r_{ij}(t)}{\sigma(t)} < h_{\min} \\ \frac{r_{ij}(t)}{\sigma(t)}, h_{\min} \le \frac{r_{ij}(t)}{\sigma(t)} \le h_{\max} \\ h_{\max}, & \frac{r_{ij}(t)}{\sigma(t)} > h_{\max} \end{cases} \tag{18}$$

$$h_{min} = g_0 + 0.1 \times \sin\left(\frac{\pi t}{2T}\right) \tag{19}$$

where $r_{ij}(t)$ denotes the Euclidean distance between atom j and atom i, and $\sigma(t)$ denotes the sum of the radii of the two atoms.

Let the total dimension of the space of atoms involved in the operation in ASO be K. The combined force generated in atom j on atom i pair in all dimensions is:

$$F_i^d(t) = \sum_{d \in K} F_{ij}^d(t) \tag{20}$$

Let the coordinates of atom i in the dth dimension be $x_i^d(t)$:

$$G_i^d(t) = \beta e^{\left(-\frac{20t}{T}\right)} \left(x_{\text{best}}^d(t) - x_i^d(t)\right) \tag{21}$$

where β is a constant and $x_{\text{best}}^d(t)$ are the coordinates of the optimal atom in the d-th dimension.

Then the acceleration of atom i in the d-th dimension is:

$$a_i^d(t) = \frac{F_i^d(t) + G_i^d(t)}{m_i^d(t)} \tag{22}$$

where $m_i^d(t)$ denotes the mass of an atom in the d-th dimension.

In the iterative process, the velocity and position are updated by the methods respectively:

$$v_i^d(t+1) = v_i^d(t) + a_i^d(t) \tag{23}$$

$$x_i^d(t+1) = x_i^d(t) + v_i^d(t+1) \tag{24}$$

Continuously iterating, updating atom velocities and positions, and updating the fitness values of the atoms, preserving the individual atoms with the best fitness.

5. **Experimental results and analyses.**

5.1. **Experimental environment.** The experiments for this work were carried out in a Lenovo laptop environment with the following configurations: Windows 10 operating system, 8 GB memory capacity, and 2 T hard disc capacity. CPU frequency is 2.30 GHz and GPU model is Ge Force GTX 960M. programming software is Python version 3.10.0.

5.2. **Training of ASO-CNN-SVM model.** The ASO-CNN-SVM model, CNN-SVM model are trained with the LPDB_train training dataset respectively and the training results are compared.

Orthogonal tests were chosen to be used to determine the values of the empirical parameters learning-rate, sample size, and number of training rounds in order to obtain the best learning results. The final choice of learning-rate is set to 0.0004, epochs is set to 40, and batch-size is set to 32. 4500 training samples from LPDB_train are used to train the ASO-CNN-SVM model constructed, and the same dataset is used to train the CNN-SVM, to obtain the overall training image recognition accuracy is shown in Figure 7 and the loss function values are shown in Figure 8.

The training results show that the ASO-CNN-SVM model converges around 50 times when training the LPDB_train dataset, which is faster than the CNN-SVM model and has higher training efficiency; the recognition accuracy of the ASO-CNN-SVM is higher than the CNN-SVM model throughout the training process, indicating that the ASO-CNN-SVM model has a better training effect than the original model has better training effect.

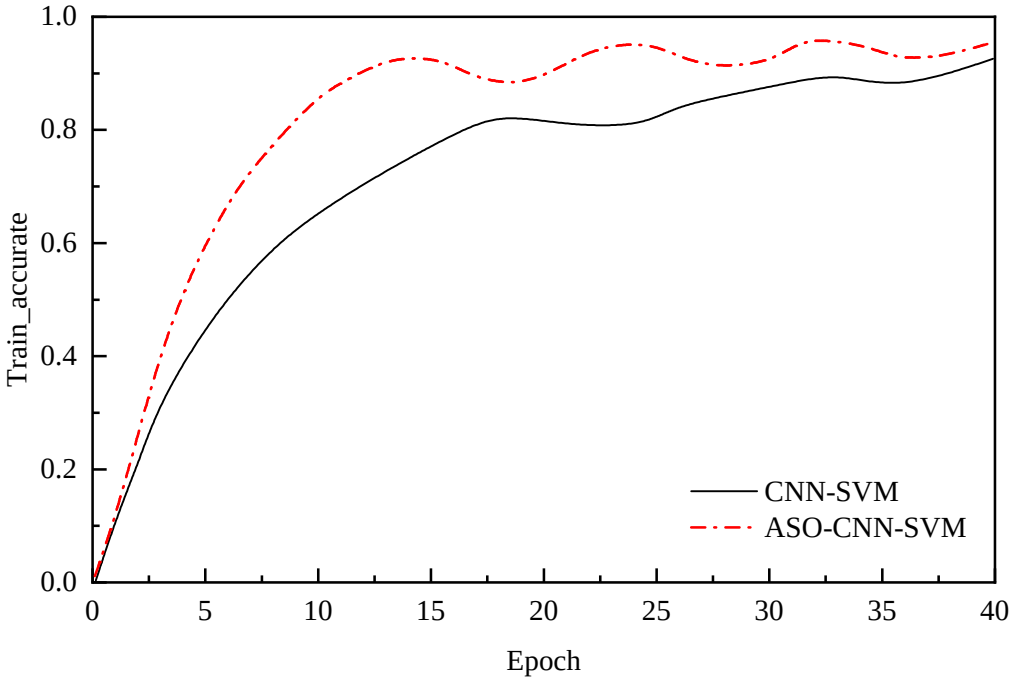

Figure 7. Comparison of Accuracy of Model Training

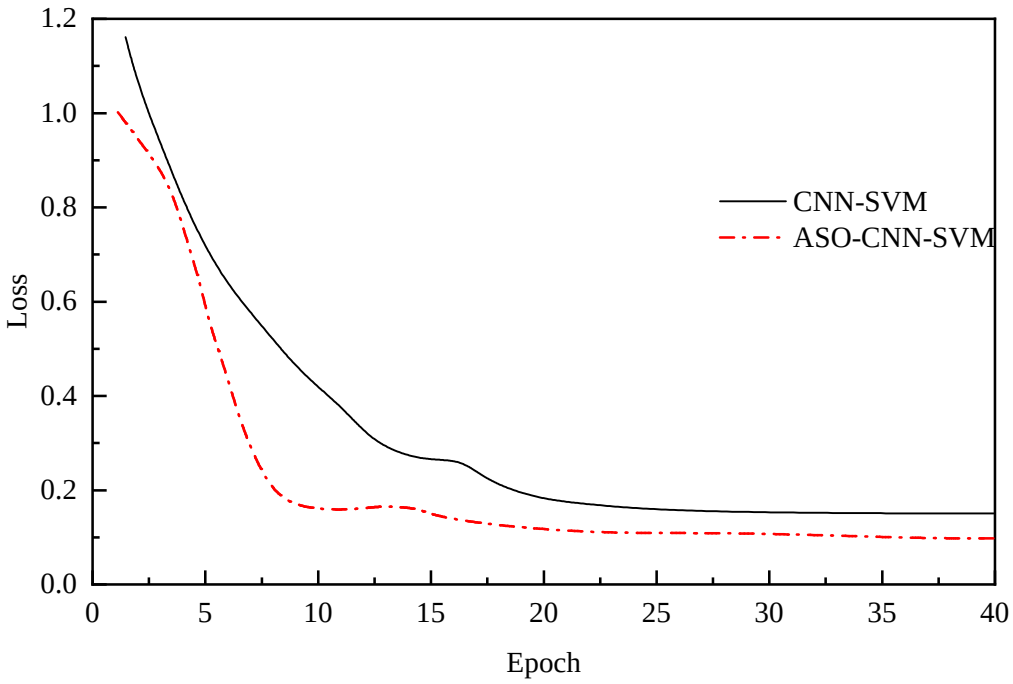

Figure 8. Comparison of loss values for model training

5.3. **Testing of ASO-CNN-SVM model.** Each class of lawn plants is now tested separately. The trained ASO-CNN-SVM model and CNN-SVM model are tested separately using 500 test sample images from LPDB_test.

The recognition results of ASO-CNN-SVM model and CNN-SVM model on the test set are now shown in Table 7 respectively.

It can be seen that during the recognition test, the highest recognition accuracy was achieved for Golf green and lower for Bermudagrass for both the ASO-CNN-SVM model

		ASO-CNN-SVM		CNN-SVM	
	Number of pictures	Number of correct identifications	Recognition accuracy	Number of correct identifications	Recognition accuracy
Zoysia	100	94	0.9400	92	0.9200
Axonopus compressus	150	143	0.9533	141	0.9400
Golf green	150	146	0.9733	142	0.9467
Bermudagrass	100	89	0.8900	88	0.8800
Average value			0.9392		0.9217

and the CNN-SVM model. Compared to the CNN-SVM model, the ASO-CNN-SVM model recognition method improved on every lawn plant species tested, with an average increase in correctness of 1.75%.

6. **Conclusion.** In order to improve the accuracy and recognition efficiency of lawn plant classification, this work implements the extraction of plant image features by CNN and feeds the features into SVM to achieve classification and recognition of lawn plants. The advantages of different convolutional neural networks such as LeNet, AlexNet, and VGG are compared, and the AlexNet network model is selected to extract image features. A new activation function is proposed and the parameters of the CNN network are optimised using the ASO algorithm to further improve the accuracy and efficiency of image recognition. The ASO-CNN-SVM model was trained and tested, and comparison experiments were done with the original CNN-SVM model. The results showed that the ASO-CNN-SVM model recognition method improved on every lawn plant species tested with an average correctness increase of 1.75% compared to the CNN-SVM model. However, end-to-end training that combines CNN and SVM can be challenging because the training processes for both are typically separate. Effective end-to-end training strategies may require additional work. Further research will follow to address this issue.

REFERENCES

[1] G. L. Grinblat, L. C. Uzal, M. G. Larese, and P. M. Granitto, "Deep learning for plant identification using vein morphological patterns," *Computers and Electronics In Agriculture*, vol. 127, pp. 418–424, 2016.

[2] Y. Sun, Y. Liu, G. Wang, and H. Zhang, "Deep learning for plant identification in natural environment," *Computational Intelligence and Neuroscience*, vol. 2017, 2017.

[3] O. M. Bruno, R. de Oliveira Plotze, M. Falvo, and M. de Castro, "Fractal dimension applied to plant identification," *Information Sciences*, vol. 178, no. 12, pp. 2722–2733, 2008.

[4] M. M. Ghazi, B. Yanikoglu, and E. Aptoula, "Plant identification using deep neural networks via optimization of transfer learning parameters," *Neurocomputing*, vol. 235, pp. 228–235, 2017.

[5] J. S. Cope, D. Corney, J. Y. Clark, P. Remagnino, and P. Wilkin, "Plant species identification using digital morphometrics: a review," *Expert Systems with Applications*, vol. 39, no. 8, pp. 7562–7573, 2012.

[6] Y. Ma, Y. Peng, and T.-Y. Wu, "Transfer learning model for false positive reduction in lymph node detection via sparse coding and deep learning," *Journal of Intelligent & Fuzzy Systems*, vol. 43, no. 2, pp. 2121–2133, 2022.

[7] T.-Y. Wu, A. Shao, and J.-S. Pan, "CTOA: Toward a Chaotic-Based Tumbleweed Optimization Algorithm," *Mathematics*, vol. 11, no. 10, 2339, 2023.

[8] T.-Y. Wu, H. Li, and S.-C. Chu, "CPPE: An Improved Phasmatodea Population Evolution Algorithm with Chaotic Maps," *Mathematics*, vol. 11, no. 9, 1977, 2023.

[9] K. Wang, Z. Chen, X. Dang, X. Fan, X. Han, C.-M. Chen, W. Ding, S.-M. Yiu, and J. Weng, "Uncovering Hidden Vulnerabilities in Convolutional Neural Networks through Graph-based Adversarial Robustness Evaluation," *Pattern Recognition*, vol. 143, 109745, 2023.

[10] M.-E. Wu, J.-H. Syu, and C.-M. Chen, "Kelly-Based Options Trading Strategies on Settlement Date via Supervised Learning Algorithms," *Computational Economics*, vol. 59, no. 4, pp. 1627–1644, 2022.

[11] Y. Yang, "Medical Multimedia Big Data Analysis Modeling Based on DBN Algorithm," *IEEE Access*, vol. 8, pp. 16350–16361, 2020.

[12] L. Ma, L. Zhao, L. Cao, D. Li, G. Chen, and Y. Han, "Inversion of Soil Organic Matter Content Based on Improved Convolutional Neural Network," *Sensors*, vol. 22, no. 20, 7777, 2022.
[13] A. Zaibi, A. Ladgham, and A. Sakly, "A lightweight model for traffic sign classification based on enhanced LeNet-5 network," *Journal of Sensors*, vol. 2021, pp. 1–13, 2021.
[14] J. H. Lee, Y. J. Kim, Y. W. Kim, S. Park, Y.-i. Choi, Y. J. Kim, D. K. Park, K. G. Kim, and J.-W. Chung, "Spotting malignancies from gastric endoscopic images using deep learning," *Surgical Endoscopy*, vol. 33, pp. 3790–3797, 2019.
[15] T. Kaur, and T. K. Gandhi, "Deep convolutional neural networks with transfer learning for automated brain image classification," *Machine Vision and Applications*, vol. 31, no. 3, 20, 2020.
[16] M. Ye, L. Ji, L. Tianye, L. Sihan, Z. Tong, F. Ruilong, H. Tianli, G. He, G. Ying, and S. Yu, "A Lightweight Model of VGG-U-Net for Remote Sensing Image Classification," *Computers, Materials & Continua*, vol. 73, no. 3, pp. 6195–6205, 2022.
[17] B. Saju, and R. Rajesh, "Eye-Vision Net: Cataract Detection and Classification in Retinal and Slit Lamp Images using Deep Network," *International Journal of Advanced Computer Science and Applications*, vol. 13, no. 12, pp. 117–131, 2022.
[18] L. Abualigah, "Group search optimizer: a nature-inspired meta-heuristic optimisation algorithm with its results, variants, and applications," *Neural Computing and Applications*, vol. 33, no. 7, pp. 2949–2972, 2021.
[19] A. Kaveh, and A. Dadras, "A novel meta-heuristic optimisation algorithm: thermal exchange optimization," *Advances in Engineering Software*, vol. 110, pp. 69–84, 2017.
[20] L. Abualigah, D. Yousri, M. Abd Elaziz, A. A. Ewees, M. A. Al-Qaness, and A. H. Gandomi, "Aquila optimizer: a novel meta-heuristic optimisation algorithm," *Computers & Industrial Engineering*, vol. 157, pp. 107250, 2021.
[21] M. Jain, V. Saihjpal, N. Singh, and S. B. Singh, "An overview of variants and advancements of PSO algorithm," *Applied Sciences*, vol. 12, no. 17, pp. 8392, 2022.
[22] S. Kashef, and H. Nezamabadi-pour, "An advanced ACO algorithm for feature subset selection," *Neurocomputing*, vol. 147, pp. 271–279, 2015.
[23] D. Karaboga, and B. Basturk, "A powerful and efficient algorithm for numerical function optimisation: artificial bee colony (ABC) algorithm," *Journal of Global Optimization*, vol. 39, pp. 459–471, 2007.
[24] Y. Sun, B. Xue, M. Zhang, G. G. Yen, and J. Lv, "Automatically designing CNN architectures using the genetic algorithm for image classification," *IEEE Transactions on Cybernetics*, vol. 50, no. 9, pp. 3840–3854, 2020.
[25] M. M. Wheeler, C. Neill, P. M. Groffman, M. Avolio, N. Bettez, J. Cavender-Bares, R. R. Chowdhury, L. Darling, J. M. Grove, and S. J. Hall, "Continental-scale homogenisation of residential lawn plant communities," *Landscape and Urban Planning*, vol. 165, pp. 54–63, 2017.
[26] G. Wei, G. Li, J. Zhao, and A. He, "Development of a LeNet-5 gas identification CNN structure for electronic noses," *Sensors*, vol. 19, no. 1, 217, 2019.
[27] S. Lu, Z. Lu, and Y.-D. Zhang, "Pathological brain detection based on AlexNet and transfer learning," *Journal of Computational Science*, vol. 30, pp. 41–47, 2019.
[28] S. Tammina, "Transfer learning using vgg-16 with deep convolutional neural network for classifying images," *International Journal of Scientific and Research Publications (IJSRP)*, vol. 9, no. 10, pp. 143–150, 2019.
[29] I. Daubechies, R. DeVore, S. Foucart, B. Hanin, and G. Petrova, "Nonlinear approximation and (deep) ReLU networks," *Constructive Approximation*, vol. 55, no. 1, pp. 127–172, 2022.
[30] F. Godin, J. Degrave, J. Dambre, and W. De Neve, "Dual rectified linear units (DReLUs): a replacement for tanh activation functions in quasi-recurrent neural networks," *Pattern Recognition Letters*, vol. 116, pp. 8–14, 2018.
[31] W. Zhao, L. Wang, and Z. Zhang, "Atom search optimisation and its application to solve a hydrogeologic parameter estimation problem," *Knowledge-Based Systems*, vol. 163, pp. 283–304, 2019.