Pattern Recognition 156 (2024) 110775

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Application of Tswin-F network based on multi-scale feature fusion in tomato leaf lesion recognition

Yuanbo Ye [a,b], Houkui Zhou [a,b,*], Huimin Yu [c,d], Haoji Hu [c], Guangqun Zhang [a,b], Junguo Hu [a,b], Tao He [a,b]

[a] School of Mathematics and Computer Science, Zhejiang A & F University, Hangzhou 311300, China
[b] Zhejiang Provincial Key Laboratory of Forestry Intelligent Monitoring and Information Technology, Hangzhou 311300, China
[c] College of Information Science & Electronic Engineering, Zhejiang University, Hangzhou 310027, China
[d] State Key Laboratory of CAD & CG, Hangzhou 310027, China

ARTICLE INFO

Keywords:
Plant leaf disease identification
Bilateral attention mechanism
Belf-supervised learning
Feature fuse local attention

ABSTRACT

Tomato leaf lesion identification can greatly help the detection and analysis of plant lesions. This study proposes Tswin-F network, a new network structure based on Transformer, to detect tomato leaf diseases. This Tswin-F network would obtain position information on images by implementing the bilateral local attention module and the self-supervised learning module. Specifically, the bilateral local attention mechanism focuses on the connection with certain continuous tokens, while the self-supervised learning module pays attention to the connection with random token positions. Then the information learned from the above two modules approaches will be combined to create the spatial connection between the final tokens. The combination of the above two modules can enhance the ability to communicate information between the windows of the input images and improve the accuracy of the models. In addition, a Feature Fuse Local Attention (FFLCA) structure is designed to solve the problem that attention distances would increase with the number of layers in the transformer network model. Furthermore, all the feature information is fused through the adaptive fusion strategy and is inputted into the classification network as the final global information of the model. Finally, an accuracy of 99.64% is obtained on 10 types of datasets, reaching the state-of-the-art level of CNN-based methods in terms of accuracy. The accuracy rate of identifying 13 types of tomato leaf lesions reaches 90.81% on average. Code is available at: https://github.com/fightpotato.

1. Introduction

Tomato fruit is one of the important cash crops, rich in nutrition and unique in flavor. Tomatoes can be eaten raw, cooked, processed into tomato paste, juice or canned fruit [1]. During their growth, pests and diseases can cause problems such as leaf spotting, stem wilt, and fruit defects. Accurate and effective identification methods can detect leaf pathologies in time, but this remains a challenging task due to the following reasons. First of all, the highly similar pathogenesis characteristics of different lesion types and the different degrees of pathogenesis of the same lesion can easily lead to inaccurate identification results. Secondly, tomato leaf lesions are more difficult to extract than complex background information. With the rapid development of deep learning, a classification method based on convolutional neural network (CNN) has been proposed to accurately classify the types of lesions in plant images [2]. By adopting efficient image recognition technology, we can improve the image recognition efficiency, reduce the cost, increase the recognition accuracy, and overcome the subjectivity and limitations of traditional manual feature extraction methods [3]. One of the most well-known classification networks is EfficientNet [4]. In addition, many variants have been proposed to improve existing CNNs by introducing attention mechanisms and self-supervised modalities to further improve classification performance [5].

Vision Transformer [6] divides the image into uniformly sized blocks, encodes the blocks and feeds them into the transformer model. This process effectively preserves all the pixel features of the image, resulting in more similarity between features obtained from the shallow layer and those from the deep layer of the transformer model. Compared with CNNs, the transformer model uses a parallel way to calculate the multi-head attention mechanism of the feature map [7], so that the

* Corresponding authors at: School of Information Engineering, ZheJiang A&F University, Hangzhou 311300, China.
E-mail address: zhouhk@zju.edu.cn (H. Zhou).

https://doi.org/10.1016/j.patcog.2024.110775
Received 14 February 2023; Received in revised form 9 July 2024; Accepted 9 July 2024
Available online 14 July 2024

network can capture richer feature information. Therefore, the identification and detection of tomato lesion leaves using the transformer model is the focus of this study.

Vision Transformer is a model that excels in computer vision tasks. However, they often fail to achieve satisfactory results on small datasets for two main reasons: their high capacity and overfitting features, and the phenomenon of discretizing the attention distance of the model as the depth of the network increases. First of all, small datasets usually have a small sample size, which makes Vision Transformer easy to overfit, and its large capacity allows the model to remember the details of training, which is also easy to cause the model to generalize poorly on unseen data. Second, the Vision Transformer's attention mechanism is key, but as the depth of the network increases, so does the attention distance. This increase in attention distance may cause the model to lose the ability to process global context information when processing local details, which affects performance and pan-Chinese capabilities. Therefore, in order to solve the problems of poor performance, easy overfitting and increased attention distance of Vision Transformer on small datasets, the goal of this study is to explore and propose effective solutions to these problems. Through the training strategy, model structure adjustment, and attention mechanism improvement for small datasets, we hope to improve the performance and generalization ability of Vision Transformer on small datasets, and provide more reliable solutions for solving practical computer vision problems.

In this work, we study the effects of bilateral attention mechanism and self-supervised model on transformer, and propose a novel feature fusion method to model the global information of transformer network to improve the accuracy of recognition. Specifically, our contributions are:

- A new network structure for plant leaf disease identification, Tswin-F, is proposed, which is an improved model based on the Vision transformer. The addition of the post-specification module normalizes the output of each residual module in the transformer and merges it with the main branch, which prevents the amplitude of the main branch from increasing, so that the model activation amplitude is more moderate
- In the proposed Tswin-F network structure, the performance of the network is improved by adding plug-and-play self-supervised learning modules. The self-supervised module can calculate the relative position information by randomly extracting image pairs during the training process, deduce the final regression loss by calculating the loss with the simple MLP module prediction results, and introduce the method of self-learning parameters to constrain the regression loss superimposed on the cross-entropy loss function. It can automatically learn the most suitable value during the network training process to constrain the regression loss without losing the spatial relationship between different positions of the image.
- With the increase of the number of network layers, the Transformer network will also increase the self-attention distance, resulting in the loss of underlying information and the problem that global information cannot be used. This study proposes an FFLCA (Feature Fuse Local Attention) structure to better solve this problem. The FFLCA structure integrates the feature information from the bottom to the high level by fusing the feature map output from the low layer to the high level, enhances the modeling ability of the entire network module, and enables the global information to be effectively used.

2. Related works

The traditional ML classification model is to train the datasets by proposing the framework and extracting the corresponding feature information as the basis for judgment, and finally obtaining the recognition result. Yang et al. [8] proposed a novel plant leaf identification method that integrates shape and texture features using a multiscale triangle descriptor (MTD) and local binary pattern histogram Fourier (LBP-HF). This method combines these features through weighted distance measurement, utilizing L1 distance for shape and chi-square distance for texture. The technique shows high retrieval and classification accuracies on the Flavia, Swedish, and MEW2012 leaf datasets, outperforming or matching state-of-the-art methods. However, it performs relatively poorly on the MEW2012 dataset, indicating limitations with certain complex datasets. Le et al. [9] proposes such a framework for the initial treatment of suspicious objects by applying morphological closing and opening to remove unwanted objects. The processed image is filtered by the local binary mode method, namely K-FLBPCM, to obtain the required feature vector, and then the calculated features are used to train the SVM classifier for classification, which achieves 98.63% classification accuracy in the four categories in the "bccr segset" dataset. Xiao et al. [10] proposed a new tomato leaf disease identification framework. By using the binary wavelet transform combined with Retinex (BWTR) to denoise and enhance the image, and then using the artificial bee colony algorithm (ABCK) optimized KSW (The objective function of the segmentation threshold) to separate tomato leaves from the background, the detection accuracy of 89% was finally obtained. However, the calculation process of binary wavelet transforms is relatively complex, especially for large signals or high-dimensional data, which is expensive to compute. This may limit the use of binary wavelet transforms in real-time or large-scale applications. Chen et al. [11] proposed a multimodal fusion encoder leveraging depth and near-infrared data to enhance RGB images for tomato detection in complex agricultural scenarios, achieving significant improvements in detection accuracy and speed with their YOLO-DNA framework. Chen et al. [12] proposed a weakly-supervised learning method utilizing modified MobileNet V2 and atrous convolution with the SPP module. They achieved an average recall rate of 91.99% on a publicly accessible dataset and an average accuracy and specificity of 97.33% and 98.39%, respectively, on a locally collected dataset. While the study demonstrates competitiveness, its limitation lies in its evaluation solely on potato datasets, potentially challenging its scalability to broader disease variations and environmental conditions. Yang [13] proposes a new method for plant leaf recognition that integrates shape and texture features. The proposed multiscale triangle descriptor (MTD) is used to characterize the shape information of a plant leaves, and the local binary pattern histogram Fourier (LBP-HF) is used as the texture feature. Then, the shape and texture features of the leaf image are combined by weighted distance measurement, where L_1 distance acts on the leaf shape and chi-square distance acts on the leaf texture feature. The corresponding classification accuracy obtained by the proposed method on the Flavia, Swedish, and MEW2012 datasets is: 99.1%, 98.4%, 95.6%. Yang et al. [14] proposed a new model LFC—Net. Yang also introduced the idea of self-supervised learning, using a multi-network collaboration strategy in the network training process, and the self-supervised network was synchronized under the iterative guidance of the feedback network, and finally obtained 99.7% accuracy on the tomato dataset. In view of the limitations of traditional data augmentation, Wu et al. [15] proposed a leaf disease identification data enhancement method based on generative adversarial network (Gans) in order to improve the accuracy of tomato leaf lesion recognition. Through this model, the images and original images generated by the deep convolutional generative adversarial network (DCGAN) enhancement are used as the input of GoogleNet, and the average recognition accuracy of 94.33% (concise) is finally obtained on the 5 types of tomato leaf pest dataset. However, the DCGAN network mentioned by the authors has the dilemma of inconsistent generated image quality, large amount of training dataset, and complex parameter tuning. Zhao et al. [16] proposes a convolutional neural network based on Inception and residual structure, and embeds an improved convolutional block attention module (CBAM) aimed at improving the classification of plant leaf diseases. In the experiment, the overall accuracy of the model proposed by Zhao Yun rt al. in identifying three diseases, corn, potato and tomato, was 99.55%. Chen et al. [17]. proposes a Feature Enhanced Convolutional Neural Network

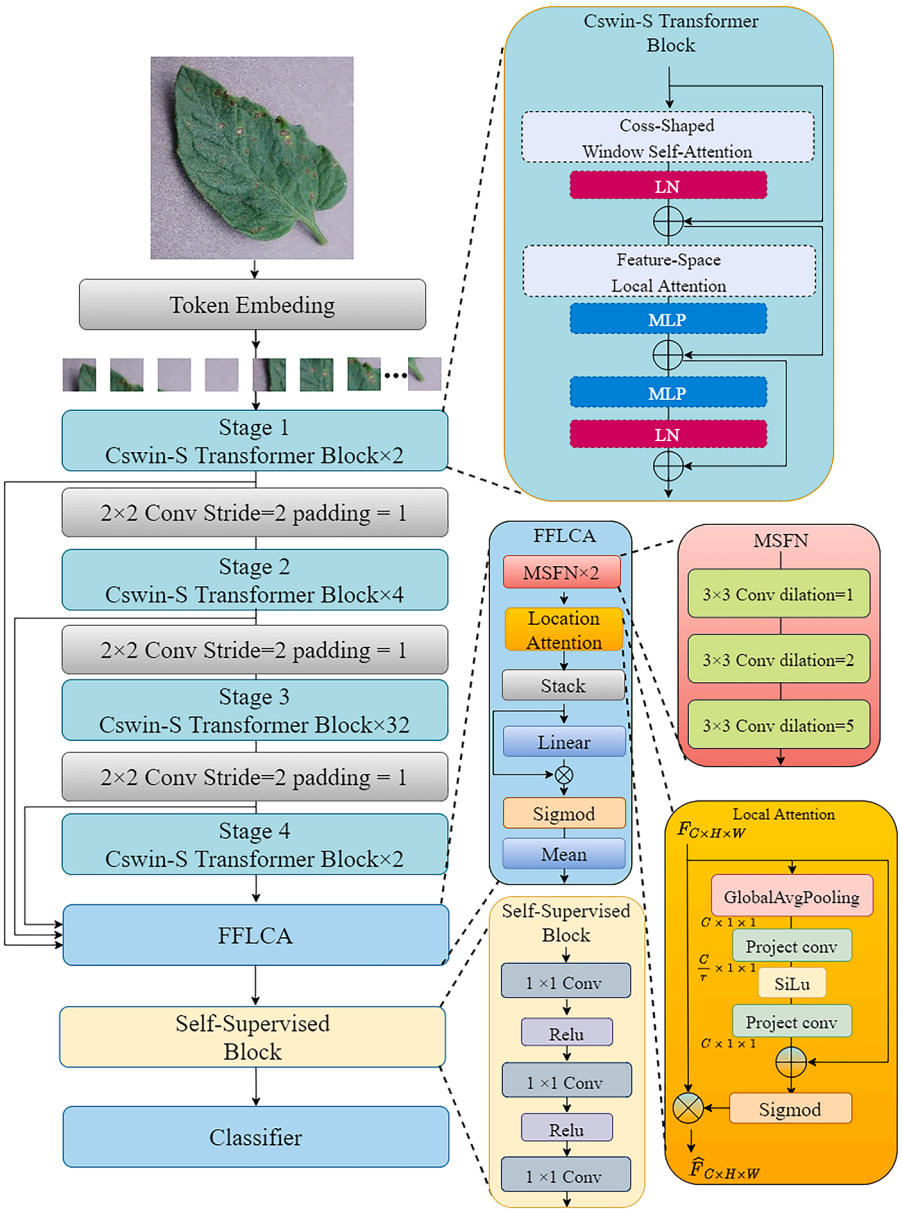

Fig. 1. The architecture of the proposed Tswin-F network.

(FET-FGVC) based on a dynamic Swin Transformer and a GCN branch for fine-grained visual classification tasks. The model incorporates a PFI module to effectively fuse global and local features, achieving an accuracy of 92.24% on the CUB-200–2011 dataset. However, the method still has some limitations, such as the increased complexity of reproduction due to the extensive parameter tuning required. Anandhakrishnan et al. [18] proposed an automatic tomato leaf disease identification system for DCNN with a deep convolutional neural network, which was divided into 6:4 on 18,160 pictures, and finally obtained an accuracy of 98.40% on the DCNN model. Liu et al. [19] first calculates the weights of all segmented patches from each image based on the clustering distribution of these patches to indicate the level of sensitivity of each patch. Then, a weight is assigned to each loss for each patch label pair during weakly supervised training to achieve disease-differentiated partial learning. Finally, 90.01% verification accuracy was obtained in the leaf identification of 271 plant species, and 99.78% accuracy was obtained on plantvillage dataset. Liu et al. [20] built a cervical cancer image recognition framework by mixing CNNs, transformer networks, and MLP perceptrons. The local information was extracted using a CNN, the global information was extracted using Transformer, and then the MLP structure was designed to classify its fusion, finally obtaining 91.72% accuracy on the cervical Pap picture. Wang et al. [21] proposed a network model based on a mix of convolutional neural networks and Transformer networks, in which an average accuracy of 96.30% was achieved in the three sets of images in the three tomato growing areas. Wang et al. [22] proposed a backbone network based on improved SwinT and applied it to data augmentation and identification of practical cucumber leaf diseases. Firstly, SwinT's patch partitioning was improved through gradual small patch embedding to enhance feature extraction without increasing the number of parameters; secondly, the data was enhanced by building a STA-GAN network to enhance the disease dataset; finally, through transfer learning, the proposed backbone network was used to train the cucumber leaf disease recognition model with the enhanced dataset, and an accuracy of 98.97% was finally obtained. Guo et al. [23] proposed a convolutional Swin Transformer (CST) based on Swin Transformer to identify the extent and type of disease, in which a novel convolutional design was adopted and it can achieve high detection accuracy and

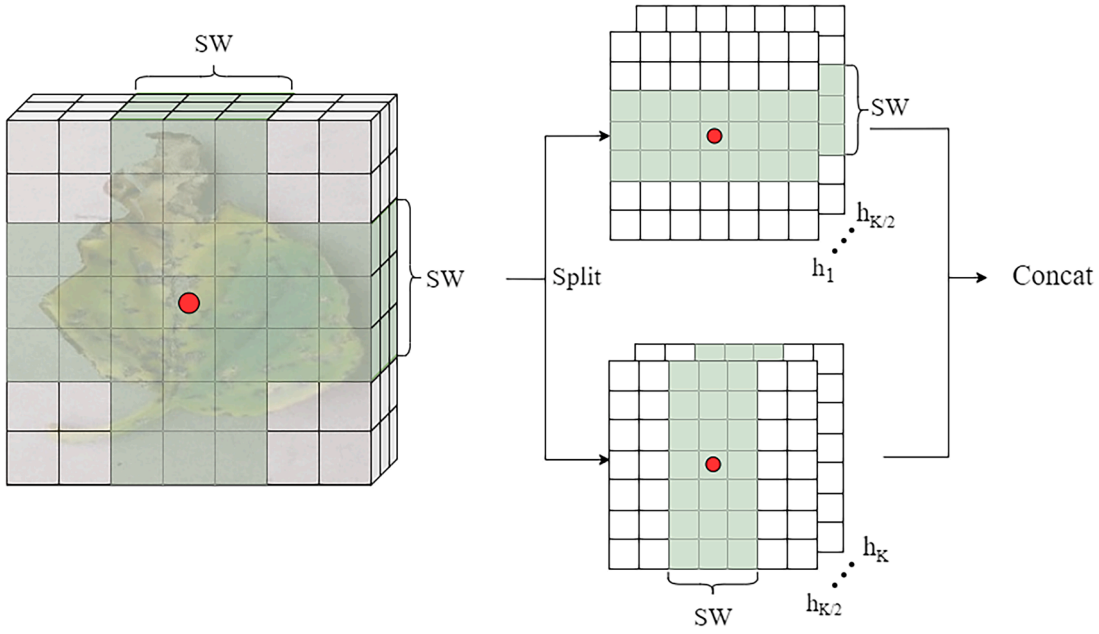

Fig. 2. The calculation process of the cross-shaped window (CSWin) self-attention module.

excellent robustness. In their study, the accuracy of the CST was 90.9% and 92.2% in detecting plant diseases in a natural environment, 97.5% in a controlled environment, and 98.2% in identifying disease categories, respectively. However, the processing of small-scale targets has some drawbacks: Swin-Transform's windowing strategy may have certain challenges when dealing with small-scale targets. Because smaller windows may not capture the complete information or details of the target, it may lead to inaccurate identification and localization of small-scale targets. Compared to the aforementioned methods, our approach achieves superior or comparable performance in leaf disease recognition. Additionally, our method offers several notable advantages. Firstly, it does not require a large amount of training data, yet still achieves impressive results. Secondly, our method effectively integrates both global and local information of the data, enabling the model to exhibit high-dimensional and low-dimensional expressiveness. These characteristics contribute to the outstanding performance of our method in leaf disease recognition tasks.

3. Methodology

In this section, we firstly present the general architecture of the proposed Tswin-F and then describe in detail the main network components. Finally, the training and inference details are presented.

3.1. Network overview

Fig. 1 shows the structural diagram of the Tswin-F model structure diagram proposed in this study. The input images are sliced into blocks by means of the patch embedding layer of the first layer [24]. The layer before each subsequent stage is the Token Embedding layer, which is used to increase the latitude of the input feature map and reduce the size. Each time the token embedding layer is passed, the size of the feature map is reduced to one-half of the original, and the dimension is expanded to twice the output feature layer of the previous layer. Therefore, in the i th stage, the feature map contains $\frac{H}{2^{(i+1)}} \times \frac{W}{2^{(i+1)}}$ tokens and $2^{i-1}C$ channels. The backbone network is composed of Cswin-S. Each stage contains a Cswin-S structure with a different number of layers, as shown in Fig. 1. The input passes through the layer normalization (LN) module and is transferred to the Cross-shaped window (CSWin) self-attention module, which is structured as shown in Fig. 2.

The Cross-Shaped Window Self-attention module is used to form horizontal and vertical stripes of cross-shaped windows for parallel computational attention. After the attention is calculated horizontally and vertically in parallel at one point in each stripe, it is equivalent to the attention calculation for each pixel in the cross-shaped window. The h_k denotes the number of heads in the multi-head attention mechanism. In the cross-shaped attention mechanism, the number of heads is divided into two halves, half for row attention calculation, half for column attention calculation, and finally the row and column attention calculation results are spliced to obtain a local attention calculation result through row and column operations. Because row and column attention are calculated simultaneously, this helps the parallel operation of the entire network structure. SW in the cross window is the size width of the row and column stripes of the input image. This window size can be changed in each layer for different visual tasks, enabling powerful modeling capabilities while saving computational costs. Specifically, for the self-attention of the horizontal bars, the X is evenly divided into non-overlapping horizontal bars:

$$X = \left[X^1, X^2, \ldots, X^M\right] \tag{1}$$

The output of K self-attention heads is defined as follows:

$$Y_K^i = Attention\left(X^i W_k^Q, X^i W_k^K, X^i W_k^V\right) \tag{2}$$

$$H - AttentionAttention_k(X) = \left[Y^1, Y^2, \ldots, Y^M\right] \tag{3}$$

Where $M = H/sw$, and $\left(W_k^Q, W_k^K, W_k^V\right) \epsilon R^{C*d_k}$ is the matrix projection of the input features Q, K, V corresponding to the three variables, d_k is the value of the number of dimensions of the input feature divided by the number of heads. The same can be applied for vertical striped self-attention, where Z/2 self-attention head output is denoted as $V - \mathrm{Attention}(\mathrm{x})$. On average, Z heads are divided into two parallel groups, each group has Z/2 head numbers. The first group performs self-attention calculations for horizontal stripes, the other group performs self-attention calculations for vertical stripes, and finally the outputs of the two parallel groups are merged together, and the formula is:

$$All - Attention(X) = Concat(head_1, \ldots, head_Z)W^O \tag{4}$$

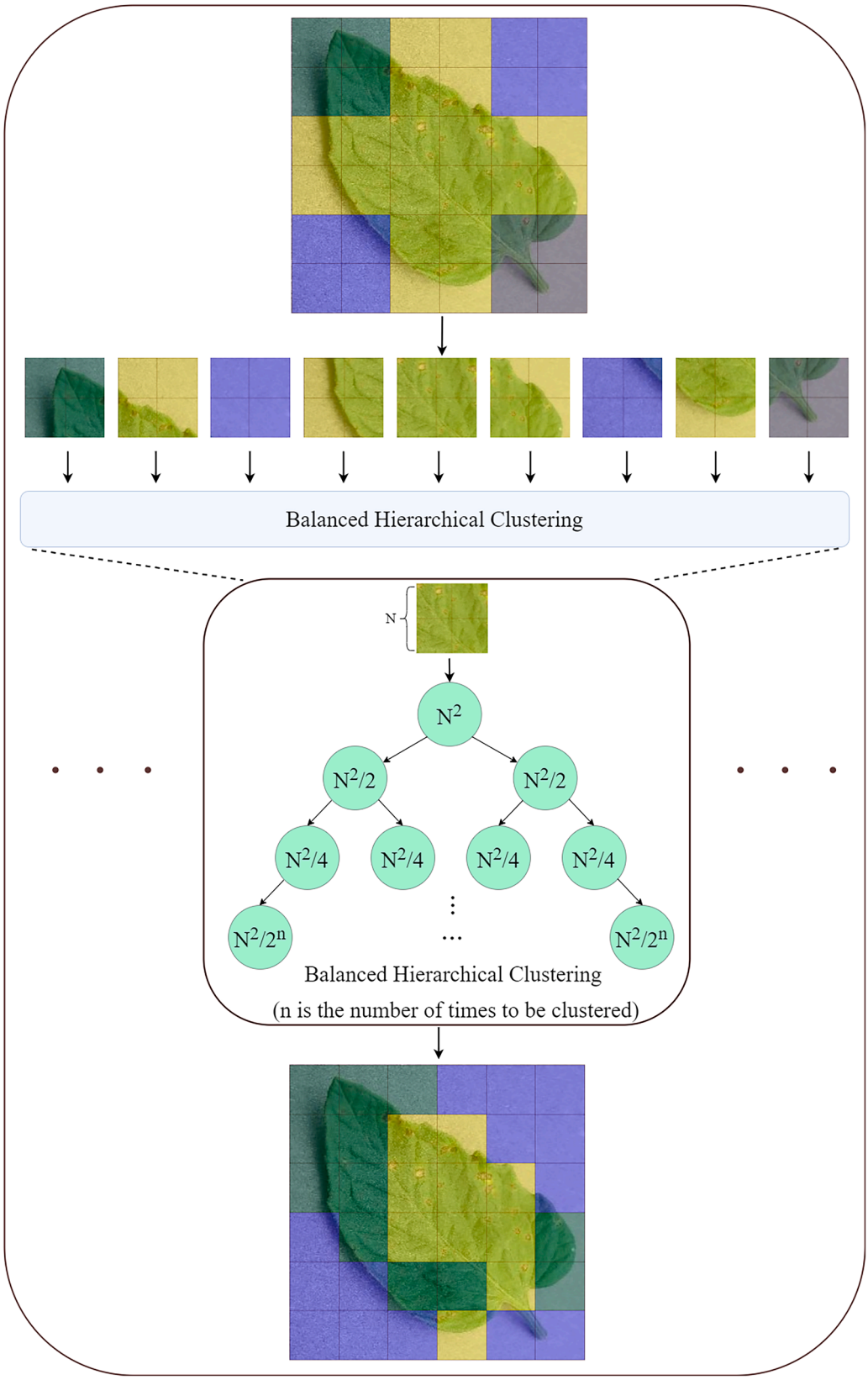

Fig. 3. The operation process of clustering tokens in bilateral local attention mechanism.

$$head_Z = \begin{cases} H - Attention_k(X)\ k = 1, \ldots, Z/2 \\ V - Attention_k(X)\ k = \frac{Z}{2} + 1, \ldots, Z \end{cases} \quad (5)$$

Where $W^O \epsilon R^{C*C}$ is a fixed projection matrix that projects the self-attention result to the target output dimension.

However, in the self-attention module of the cross window, the tokens can only interact with information in the window of the specified

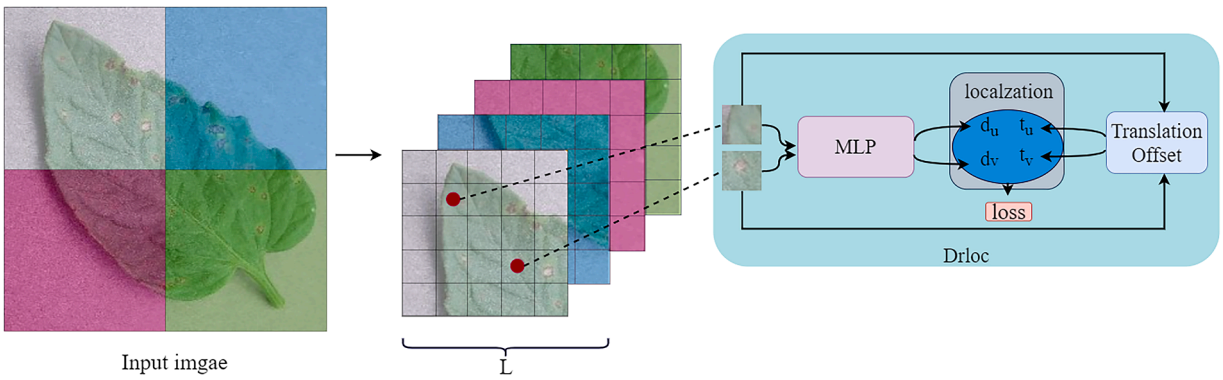

Fig. 4. Distance calculation of different pixel locations in the self-supervised image module.

size, and the tokens in different windows still cannot interact with data information, thus preventing the establishment of long-range associations between notations.

For this case, this study attempts to build a bilateral local attention mechanism, and then add the Feature-Space Local Attention (FSLA) module after the CSWin [25] self-attention module. It differs from image space local attention, which groups tokens according to their spatial position in the image plane, and feature space local attention seeks to classify tokens according to their content (i.e., features). After the cross-window module operation, the FSLA module divides the output tokens from scratch. The FSLA module focuses on clustering token features and calculating self-attention within each token feature. The FSLA module performs balanced hierarchical clustering of tokens at level K. At each level, it conducts a balanced binary clustering, dividing a set of tokens equally into two clusters. The method is shown in Fig. 4. $T=\{t_i\}_{i=1}^{N}$ is the set of input tokens. In the first level, it splits N tokens in T into two subsets with $N/2$ tokens each. By recycling, the tokens that have been divided into categories are evenly divided into new categories. At the K-th level, it splits $N/2^{k-1}$ tokens assigned to the same subset in the upper level into smaller subsets of $N/2^{k-1}$ size. At the end, we obtain 2^k evenly sized subsets in the final level, $\{t_i\}_{i=1}^{2^k}$, and the size of each subset $|T_i|$ is equal to $N/2^k$. This method establishes a long-range association of all tokens in the feature space and divides it into a window in which the self-attention operation is performed. This method reduces the overall computational effort by introducing a feature space attention mechanism instead of a global attention mechanism. The shortcut connection formula for a Cross-Shaped window is:

$$F_{Csw} = F_{in} + CSW(LN(F_{in})) \tag{6}$$

F_{in} is the input tokens collection, CSW is the cross-attention structure, and LN is the layer normalization module. The formula for how the feature space attention layer is connected to the Cross-Shaped window layer is:

$$F_{Fsla} = F_{Csw} + \sigma F_{Fsla}(LN(F_{Csw})) \tag{7}$$

In this module, the *FSLA* calculates the self-attention relationship between close tokens in the entire input feature space, which is complementary to the cross-cross self-attention module. The local attention to the feature space makes the token relationship between the same windows closer. At the same time, this study adds a coefficient σ to the original superposition method. σ acts on the F_{Fsla} layer to prevent the excessive oscillation amplitude of the model parameters due to the difference of different batch data. The σ parameter is an adaptive parameter that is trained with the network model. The final output is:

$$F_{out} = F_{Fsla} + LN(MLP(F_{Fsla})) \tag{8}$$

F_{out} is the output of the entire module, which has been processed by an MLP module and a normalization module to produce the pre-normalization output. We propose to use the residual post normalization approach instead. In this module, F_{Fsla} first goes through the MLP module and then goes through the LN module processing, which helps the model calculates attention and then normalize the output to stabilize the output value.

3.2. Self-supervised learning

Most traditional tasks of classifying plant lesions and leaves adopt supervised learning methods, which requires learning effective classification feature information from a large number of labeled data [25]. Therefore, how to learn and migrate common feature expressions from limited data to the classification network proposed in this project is also one of the key problems to be solved in this study. This study adopts a plug-and-play self-supervised model that encourages the VIT model to learn spatial information about features rather than additional manual annotation [26]. After segmentation by patch embedding, the patches (assumed to be of size $L \times L$) are mapped into the input embedding space to obtain $L \times L$ tokens. Tokens in a patch are tokenized and pairs of samples are randomly selected, and the sample pairs will run a dense relative localization loss. In more details, given the patch ($L \times L$), all tokens are set as $G_x = \{x_{i,j}\}, 1 < i,j < L, x_{i,j} \in R^d$, where d is the dimension of the input image. The random sample pairs taken out are ($x_{i,j}$, $x_{a,b}$). We calculate the 2D normalized target translation offset$(t_u, t_v)^T$, where:

$$t_u = \frac{|i-a|}{L},\ t_v = \frac{|j-h|}{L},\ (t_u, t_v)^T \in [0,1]^2 \tag{9}$$

At the same time, a simple MLP (f) module is established to train the sample pairs of input and predict the relative distance between ($x_{i,j}$,$x_{a,b}$). After passing through the MLP module, it is named $f(x_{i,j}, x_{a,b})$ and let$(d_u, d_v)^T = f(x_{i,j}, x_{a,b})^T$. The loss calculation is then performed on all the data in a batch. The dense relative localization loss:

$$l_{dl} = \sum_{x \in B} \mathbb{E}_{(x_{i,j}, x_{a,b}) \sim G_x} \left[\left| (t_u, t_v)^T \right| - (d_u, d_v)^T \right] \tag{10}$$

where d_u, d_v are the two predicted values obtained by the MLP module's calculation of the input image. The expectation is calculated by sampling uniformly on random sample-pairs ($x_{i,j}$,$x_{a,b}$) in the G_x and the mean $L1$ loss between the corresponding $(t_u, t_v)^T$and $(d_u, d_v)^T$.

To take advantage of the positional association information of the image space learned by self-supervision, we add the l_{dl} as a constraint to the standard cross-entropy loss function (LCE) of the native VIT. The final loss function is:

$$l_t = l_{ce} + \lambda l_{do} \tag{11}$$

For the native formula, we use the trainable parameter λ to constrain the l_{do} loss value, which can prevent the influence of individual extreme

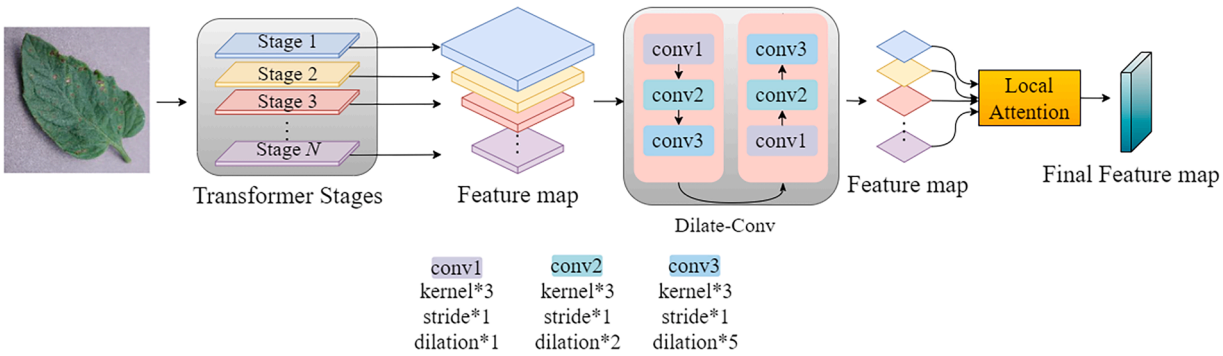

Fig. 5. FFLCA (Feature fuse local attention) fuses the feature maps output of different stage layers to obtain the final output feature maps.

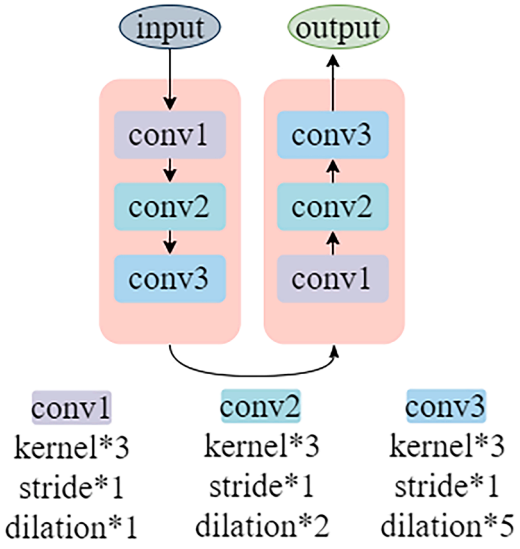

Fig. 6. The combination of dilated convolution during fusion.

exceptions on the overall model and make the entire network output smoother. At the same time, it can reduce the cumbersome steps of parameter tuning and increase the flexibility of experiments. Experiments show that adaptive λ is more friendly to the network model than specific λ values, and the recognition results are more accurate.

The self-supervised module can be inserted into any structure in the network model to calculate the input feature information. In a patch, the spatial connection between pixels can be greatly strengthened, which is complementary to the cross-cross window in this study. Meanwhile, it can be combined with the FSLA module, one focusing on local attention, the other focusing on the spatial connection with random locations, and the combination of the two can integrate the connection of spatial information in the image and express it. The self-supervised module is shown in Fig. 4.

3.3. Feature fuse local attention

During the experiment, we found that in the CNNs, the receiving domain of the feature map increases as the network deepens, and similar observations appear in the VIT network. During the training process of VIT network, its 'attention distance' increases with the increase of network depth. Therefore, we propose the operation of feature map fuse local attention of different layers in the experimental process, which is different from the standard VIT. The FFLCA module takes the feature map generated by each stage of the network structure as input, pays attention to these inputs, fuses the different stage outputs through network learning to achieve attention to the output of different stage layers, and uses the fused output X^{R*H*W} as the basis for final forecast judgment. The FFLCA structure is shown in Fig. 5.

As shown in Fig. 5, each stage layer contains Cswin-S modules with a different number of layers. Therefore, the size of the output feature maps is different for each stage. In order to fuse feature maps of different sizes, the final X (FFLCA) can contain different levels of information with global effects. In this study, we use dilated convolution method instead of pooling operation. The advantage of dilated convolution is that without pooling the loss information, the receptive field is expanded, so that each convolution contains a larger range of information. The benchmark Transformer network tends to focus on the local information in a patch or a window, thus resulting in the global information of the image not being effectively used, so we use dilated convolution in the integration of global information.

The structure of the dilated convolutional module is shown in Fig. 6 All feature layers output feature layers of the same size after dilated convolution, and we need to integrate all feature layers, $X(Dilate)^N = [x_D^1, x_D^2, x_D^3, \ldots, x_D^n]$. N is the number of layers of the feature layers output by the CSwin-S module in each stage. $[x_D^1, x_D^2, x_D^3, \ldots, x_D^n]$ are feature maps for each CSwin-S output and have different scales between x_D. In this regard, the dimension scale of dilated convolution is used, and the

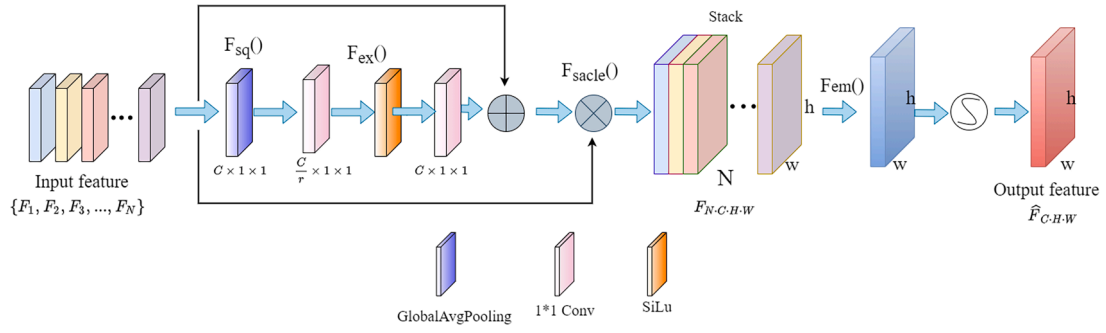

Fig. 7. The fusion process of multi-dimensional feature maps.

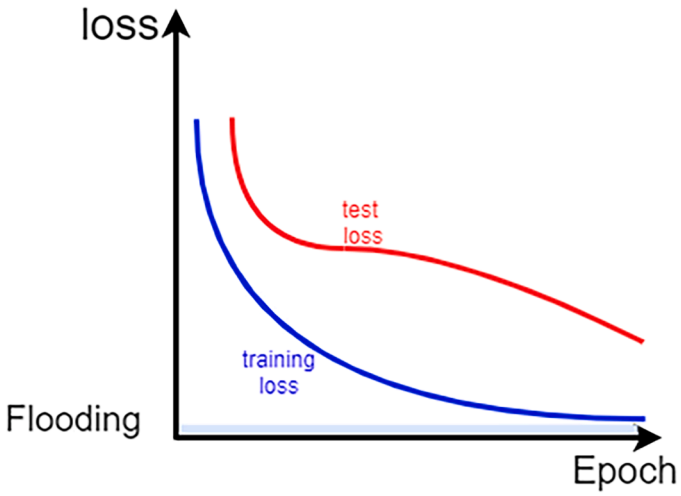

Fig. 8. The effect of the Flood operation on losses.

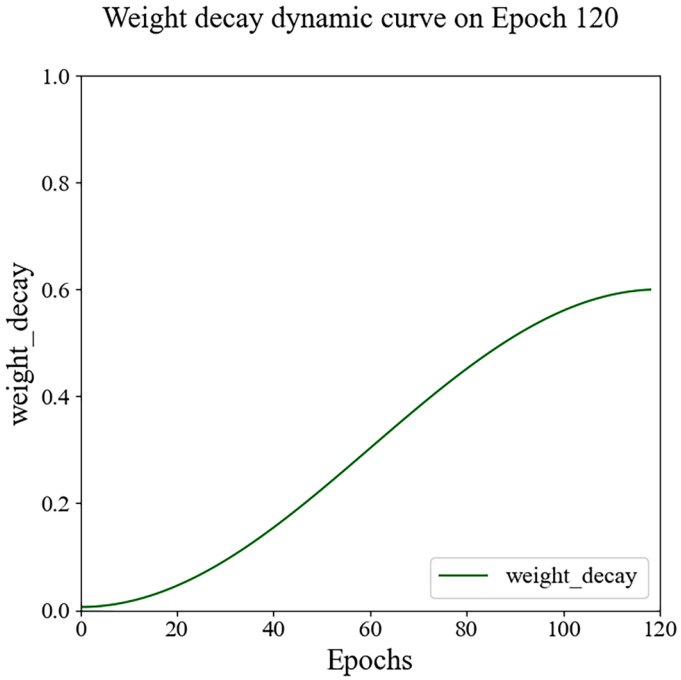

Fig. 9. Curve of weight decay dynamically changing with epoch iteration.

use strategy of dilated convolution is [1,2,5]. This combination method allows the pixels between the layers of convolution to be utilized, which can prevent the feature map from losing a large amount of feature information due to the dilated convolution.

In this study, by performing dilated convolution on different layers, the feature map of each layer is scaled to the same dimension and size, and the feature information can be preserved and the perceptual field can also be expanded. Using the fusion strategy shown in Fig. 7 the feature maps with the same dimension and size of the n-layer are fused according to the weight to size ratio. The specific operation is to compress the multi-dimensional features $F \in R^{H*W*C}$ according to the dimensions, so that each dimension is mapped to a single value, and the entire spatial feature on a channel is encoded as a global feature, which is achieved by averaging pooling operation. The formula is as follows:

$$Z_c = F_{sq}(F_c) = \frac{1}{H*W}\sum_{i=1}^{H}\sum_{j=1}^{W} F_c(i,j),\ Z \in R^C \tag{12}$$

After obtaining the global description characteristics through the Squeeze operation, it is necessary to learn the relationship between different layers. We use the Excitation operation, which learns the nonlinear relationship between the layers to map the different fusion scale values.

$$P_c = F_{ex}(Z_c) = W_2 ReLU(W_1 Z) \tag{13}$$

where $W_1 \in R^{\frac{C}{r}*C}$, $W_2 \in R^{C*\frac{C}{r}}$. In order to reduce the complexity of the model and improve the generalization ability, a bottleneck structure containing two fully connected layers is adopted. The first fully connected layer plays the role of dimensionality reduction, and the hyperparameter r is the dimensionality reduction coefficient. The RELU activation function is used. The fully connected layer of the last layer restores the feature dimension to the dimension from which the original input came in.

The mapping values are updated through the training of the network to learn the relationship between different layers, and then the output mapping values are multiplied back to the original feature dimension. The formula is as follows:

$$U_c = F_{scale}(P_c, Z_c) = Z_c * P_c \tag{14}$$

To satisfy the needs of each layer to be utilized, we use the non-one-hot form and the sigmoid activation function (σ) to quantize the output weight coefficient of the last layer by 0∼1. The formula is as follows:

$$X_{fflca} = F_{em}(U_c) = \sigma\left(\frac{1}{c}\sum_{i=1}^{c} U_c^i\right) \tag{15}$$

Finally, the $X_{fflca}^{L\in B*C*H*W}$ is used as the output of the entire model. In order to prevent drastic changes in values after feature fusion, we add a Sigmod layer to the last layer of the module to normalize it, so that its output tends to be mild. Finally, its output is classified as a prediction by the MLP module.

3.4. Fine tune

In terms of network fine-tuning, we also propose relevant fine-tuning innovation methods. In early experiments, we found that the training loss gradually decreases to zero as the number of training times increases, which can cause the model overfitting. To address this situation, we propose a fine-tuning method for FAWD (Flood and weight decay). Flood [27] refers to flood point setting. According to the ideas proposed in this study, it is clear that simply reducing train loss does not allow the network converge to the optimum and obtain the best experimental results. Conversely, the train loss decreases and approaches zero, leading to overfitting of the model. This problem is caused by the fact that the training set is too small and there is not enough data [28]. In this paper, we strive to achieve the optimal effect of the network structure on small datasets. Therefore, a flooding operation is used. The Flood operation limits the minimum train loss value to a specific value to prevent the train loss from approaching zero due to overfitting during training. This is shown in Fig. 8. In practice, this is achieved by setting a threshold for the final loss, e.g.: setting the flood control point at position ξ, L is the output loss of the model.

$$L = |(L - \xi)| + \xi \tag{16}$$

In addition to setting up the network flood point, we also propose improvements on the weight decay. In the process of network training, weight decay can also improve network model overfitting. However, setting a single weight decay value cannot effectively fit all stages of the entire network training process, so we propose to gradually increase the weight decay operation with the training iteration, and gradually increase the weight decay value during the network model training process, so that the network can be effectively controlled in overfitting. The curve of the weight decay is shown in Fig. 9 below.

Table 1
Basic machine characteristics.

Hardware and software	Characteristics
Memory	12GB
Processor	Intel(R)Core(TM) i7–11700k@3.60GHZ
Graphics	NVIDIA GeForce RTX 3080
Operating system	Window10 Python 3.8 Pytorch 2.1
Training methods	MIXED PRECISION

Table 2
Training and testing images for the proposed Tswin-F of the *dataset*$_1$ (D_1).

Class	No of images used for Training	No of images used for Validation	No of images used for Testing
Bacterial spot	1703	212	212
Early blight	800	100	100
healthy	1272	159	159
Late blight	1527	191	191
Leaf mold	762	95	95
Septoria leaf spot	1417	177	177
Target spot	1124	140	140
Mosaic virus	1097	138	138
Yellow leaf curl virus	4285	536	536
Two-spotted spider mite	1340	168	168
TOTAL	15,327	1916	1916

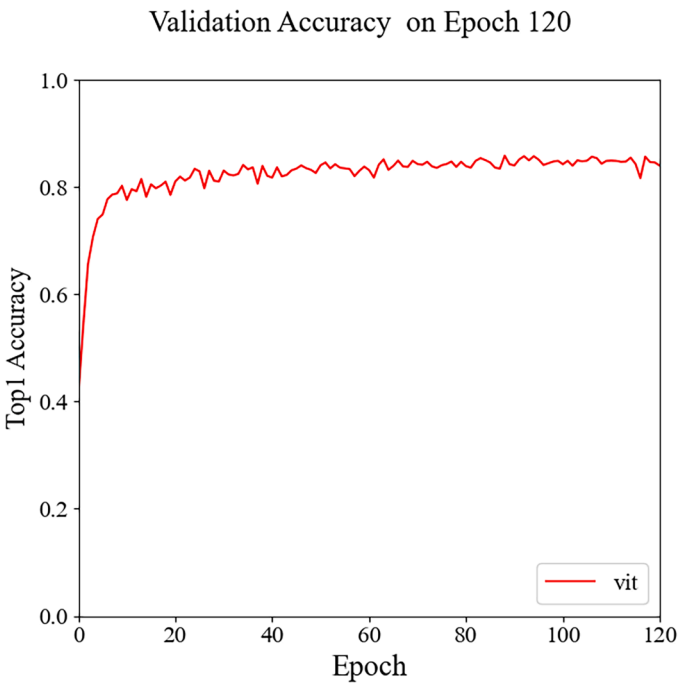

Fig. 10. Validation accuracy curves obtained on VIT model for D_2 identification.

4. Experiments

In this section, we primarily focus on presenting the experimental aspects of the paper, which encompass details about the experimental setup, information about the datasets used, ablation experiments to evaluate module effectiveness, comparative experiments across different network models, and experiments assessing the effectiveness on various datasets.

4.1. Experimental environment

In this experiment, a single-card stand-alone device is used for the tomato leaf disease recognition model throughout the training process. The training of the Transformer network model runs in image processing unit (GPU) mode, and the training method adopts a mixed-precision training strategy. The detailed characteristics of the computer used in this experiment are shown in Table 1.

4.2. Dataset information

In this section, the two datasets and the division ratios of their training set, validation set, and test are introduced. The 10 types of tomato leaf diseases correspond to the *dataset*$_1$ (D_1), and the 13 types of tomato leaf diseases correspond to the *dataset*$_2$ (D_2). The partitioning significance of the dataset with 13 categories involves segregating individual leaf lesion types based on their severity, resulting in a dataset with a higher variety of lesion types but relatively low volume. This setup is particularly effective in testing the model's performance. Additionally, it holds practical significance; identifying varying degrees of lesions allows targeted plant treatment, reducing pesticide usage, mitigating environmental pollution, and improving the efficacy of plant disease treatment.

4.2.1. Dataset of 10 types of tomato leaf diseases (D_1)

Images of tomato leaf disease were obtained from the publicly available dataset PlantVillage [29]. The dataset contains 19,159 images with a total of 10 types of tomato leaf lesions. The image size in the dataset is not uniform, in order to reduce the computational complexity, we scale the image size to 256×256. The division of the dataset was based on the ratio of 8:1:1, with 15,325 pictures for training, 1916 pictures for verification training, and 1916 pictures for testing. Table 2 provides detailed information about the dataset.

4.2.2. Dataset of 13 types of tomato leaf diseases (D_2)

All images of the *dataset*$_2$ (D_2) are divided into 13 categories, including Early Blight Fungus general, Early Blight Fungus serious, Health, Late Blight Wate Mold general, Late Blight Wate Mold serious, Leaf Mold Fungus general, Leaf Mold Fungus serious, Septoria Leaf spot Fungus general, Septoria Leaf spot Fungus serious, Spider MiteDamage general, Spider MiteDamage serious, YLCV virus general, YLCV virus serious. The number of each class is: 251, 442, 1208, 264, 1109, 325, 336, 421, 807, 524, 271, 1414, and 2473. All samples are divided into three parts with a ratio of 6:2:2, and the aggregate of all pictures is 9845 pictures.

4.3. Train from scratch

In the early stage of the experiment, we use the native VIT (15) model to train the data. The validation results obtained on the native VIT model (parameters are not fine-tuned) on D_2 obtained in this study are shown in Fig. 10. As can be seen from Fig. 10, the 86% accuracy of the pure VIT model does not meet the requirements of today's plant pest disease identification. This paper uses the Swin-transformer model based on the VIT structure, which addresses the shortcomings of VIT.

Compared with the VIT network model, Swin-tranformer [30] uses a hierarchical structure method similar to that in convolutional neural networks. Compared with VIT network model that only uses 16x down sampling methods, Swin-transformer can be more intimate with the underlying information, and will not lose the underlying information due to excessive down sampling. Moreover, Swin-transformer will have a stronger grasp of multi-size features and will be more friendly to downstream tasks. Compared with VIT which does global modeling on the entire patch, Swin-transformer can do global self-attention in the corresponding window, thereby reducing the computational complexity. In order to further reduce the sequence length, lower the

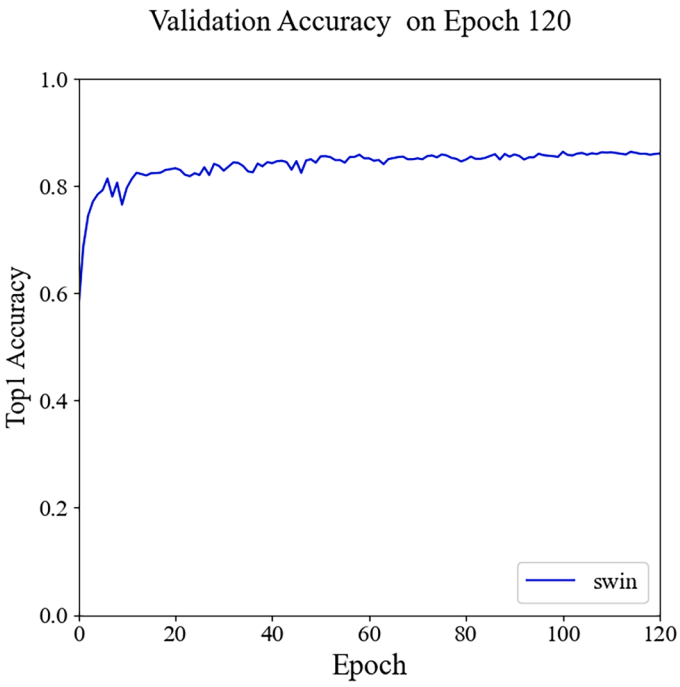

Fig. 11. Validation accuracy curves obtained on Swin model for D_2 identification.

complexity of the calculation, and avoid calculating only the autonomous force within the window, the Swin-transformer uses the design of the sliding window to calculate self-attention. By sliding the window, Swin enhances the information between different windows, so as to achieve the information exchange between windows. The results obtained on the D_2 validation set using the Swin model are shown in Fig. 11.

4.3.1. *Experiment on the effectiveness of FAWD fine-tuning strategies*

Fig. 11 shows that although the verification accuracy of Swin compared to VIT models have been improved, new problems have also occurred, and compared with CNNs, the network model of the Transformer class performs better on large datasets. It can be concluded from the experiments that the dataset with small data volume is prone to overfitting in the network of the Transformer. In this regard, this study uses the structure of flood to suppress the overfitting phenomenon in the model training process. Through comparative experiments, we can find that the structure has a significant effect on reducing overfitting. As shown in Fig. 12, the network model can reduce the validation loss and smooth the accuracy curve of validation after using the Flood operation alone. we have further addressed the issue of overfitting and the potential instability caused by excessive model parameters during the training process. To mitigate these challenges, this study introduces a novel framework called FAWD (Flood And Weight Decay) structure. The FAWD structure not only effectively reduces overfitting and improves accuracy but also helps stabilize the training process by regulating the impact of excessive model parameters on the training loss. As shown in the Fig. 12, this study uses Swin as a network model to train the D_2 and the validation accuracy curve is obtained in the test set on D_2. The left Fig. 12(a) shows the validation accuracy rate and loss value curve without flood operation, and the right Fig. 12(b) demonstrates the accuracy and loss value curve after using flood, and it can be clearly found that flood has a very good suppression effect on the overfitting phenomenon. As shown in Fig. 13, we introduce different coefficient ratios for weight decay at different stages of the training process. This allows us to selectively apply weight decay with varying strengths to different sets of model weights. By doing so, we can effectively control the regularization effect and prevent the model from overemphasizing certain weights while neglecting others.

As depicted in Fig. 14, the curves represent variations in performance concerning two parameters: flood and weight decay. Our network model was evaluated using three approaches: individual utilization of the flood operation and weight decay, as well as their combined application. This analysis was conducted to ascertain the model's optimal performance under different combination scenarios. As shown in Table 3, the Vit, Cswin and Swin network models obtained test accuracy recognition results after using FAWD strategies on D_1 and D_2 datasets. Through the experiments, we demonstrate that the utilization of different coefficient

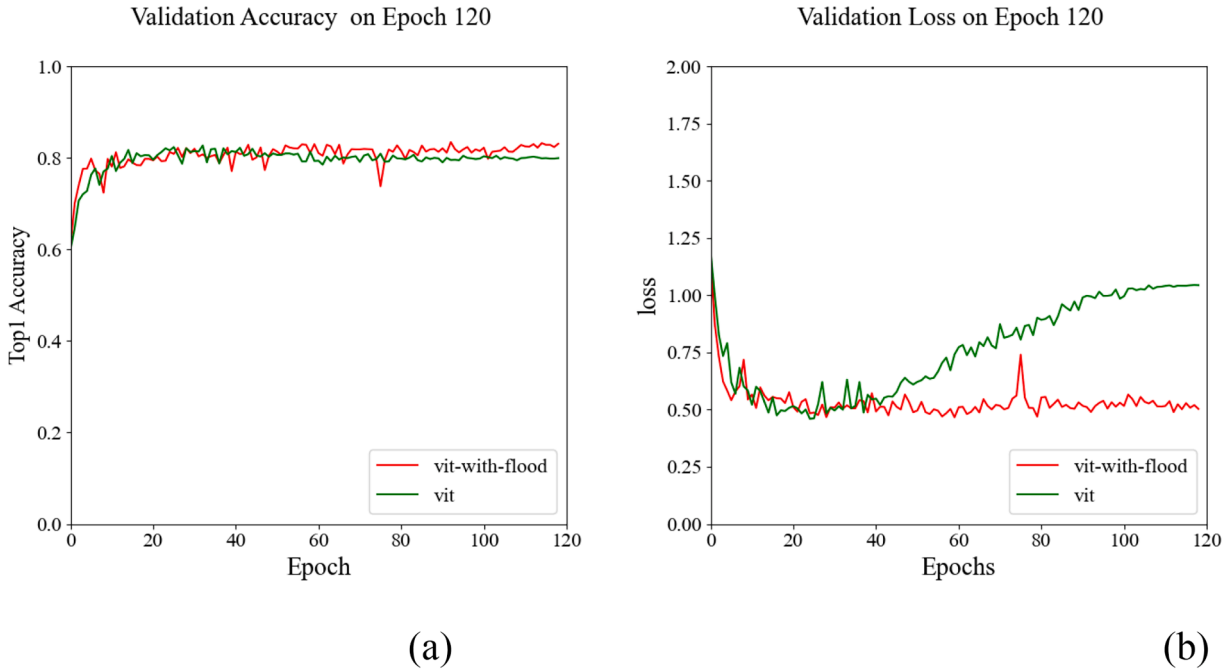

Fig. 12. (a) the accuracy curve obtained by the VIT model with Flood structure and the VIT model without Flood structure on the identification of D_2. (b) the loss curve for both models.

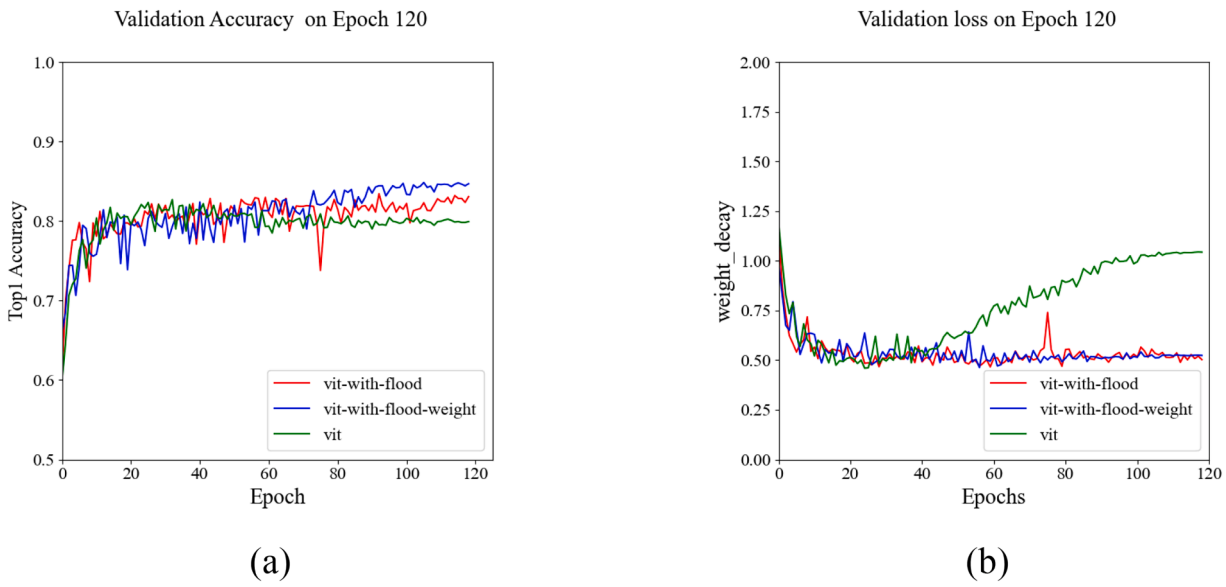

Fig. 13. (a) the comparison curve of the validation accuracy obtained by the three models on D_2. (b) the loss curve for the three models.

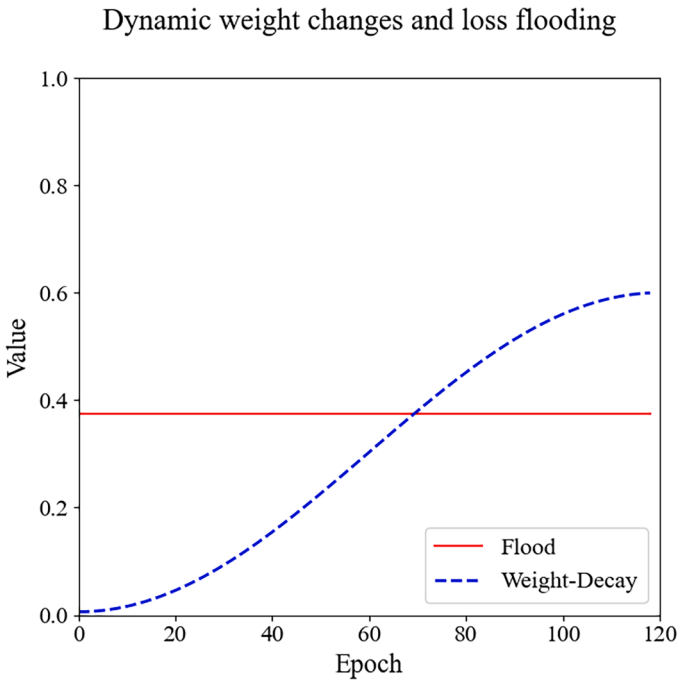

Fig. 14. Flood and weight dynamic decay curves used in the models.

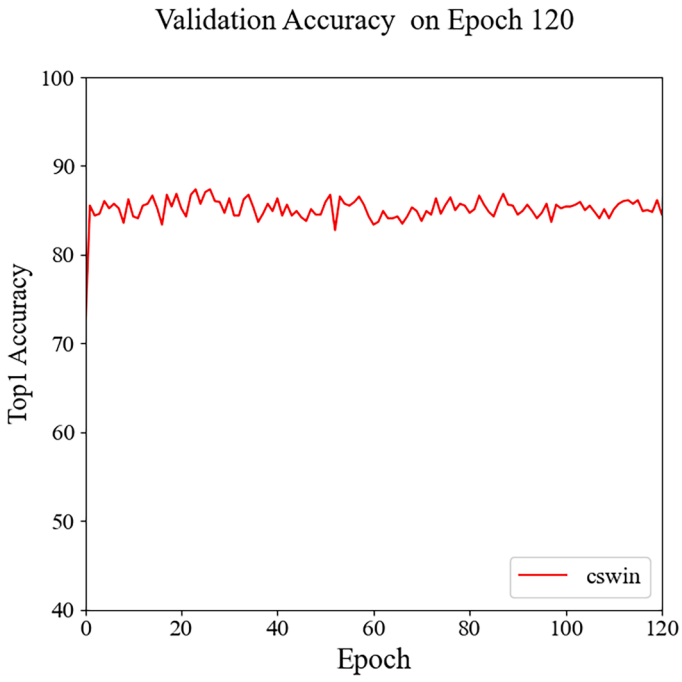

Fig. 15. Validation accuracy curves obtained on Cswin model for D_2 identification.

Table 3
The results of three baseline models within a 95% confidence interval on D_1 and D_2 test sets with the FAWD module.

	Model	ACC(%)	F1 Score	F1 95% CI
D_1	Vit+FAWD	96.13	96.37	(95.44–97.12)
	Swin+FAWD	97.69	97.93	(97.19–98.48)
	Cswin+FAWD	**98.72**	**98.69**	**(98.07–99.11)**
D_2	Vit+FAWD	84.33	84.28	81.87–86.42
	Swin+FAWD	86.47	86.55	84.28–88.54
	Cswin+FAWD	**86.56**	**86.61**	**(84.34–88.6)**

ratios for weight decay enhances the model's generalization ability and helps prevent overfitting. This approach effectively balances the trade-off between model complexity and generalization performance, resulting in improved model performance and robustness.

As seen in Figs. 12 and 13, although Swin-transformer can reduce the complexity of training and enhance the information exchange between windows by moving the window setting, it is undoubtedly that the difference between some data cannot be ignored. Different position information in different images is crucial to the overall model and prediction output. Swin-transformer implements the information exchange between different windows, but in a patch, the design of the sliding window design may not be friendly to the information exchange of relative location distance between windows, so that the network model cannot capture the global information in the patch. As a result, the transformer lacks the position information of the full map in the final output, and the recognition accuracy is poor.

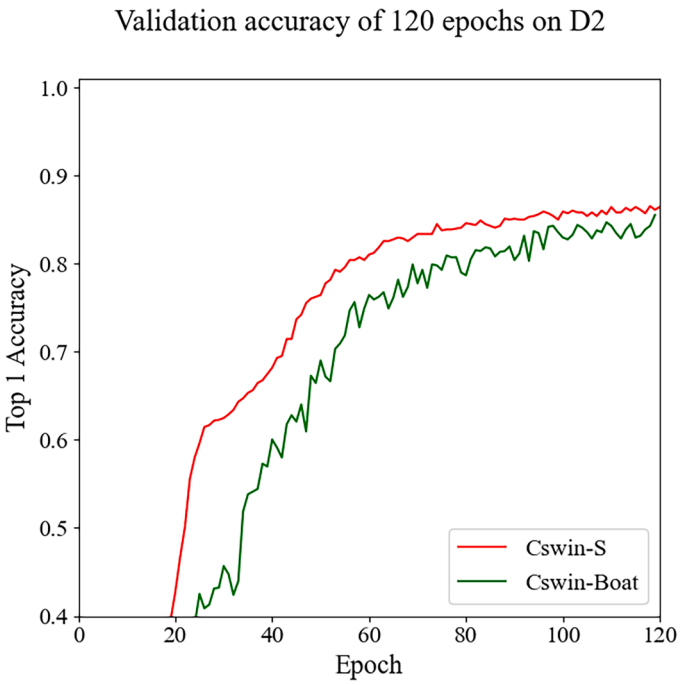

Fig. 16. Validation accuracy curves of Cswin-S and Cswin-Boat on D_2 identification.

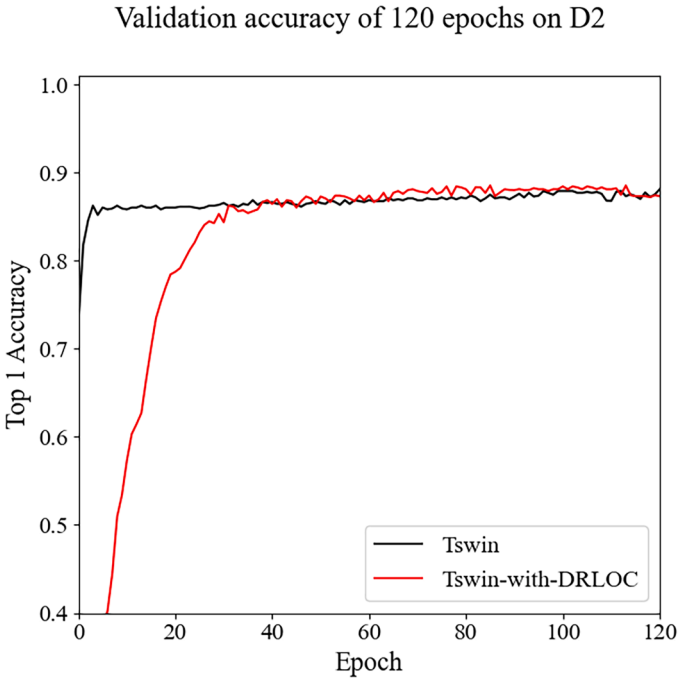

Fig. 17. Validation recognition accuracy curves obtained by tow models on D_2.

In order to keep the accuracy undiminished, this study explores the relevant methods to reduce the computational amount of network models. Finally, by learning from the Cswin [25] network mechanism, the effect of improving the accuracy of the model is improved under the same amount of calculation is achieved. The CSwin network model employs a cross method to compute the self-attention of the feature map, diminishing computation while concurrently computing self-attention in both vertical and horizontal directions. Moreover, as the network stage deepens, the cross's width increases, resulting in a rapid expansion of the receptive field. After training the training set of D_2, the model

Table 4
The results of three baseline models within a 95% confidence interval on test set of D_1 and D_2 with the BLA module.

	Model	Acc(%)	F1 Score	F1 95% CI
D_1	Vit+BLA	98.02	97.86	(97.11–98.42)
	Swin+BLA	98.98	98.81	(98.22–99.21)
	Cswin+BLA	**99.43**	**99.38**	**(98.92–99.65)**
D_2	Vit+BLA	85.43	85.19	(82.84–87.27)
	Swin+BLA	87.24	87.21	(84.98–89.15)
	Cswin+BLA	**87.56**	**87.64**	**(85.44–89.55)**

Table 5
The results of the baseline models within a 95% confidence interval on test set of D_1 and D_2 with the DRLOC module.

	Model	Acc(%)	F1 Score	F1 95% CI
D_1	Vit+drloc	97.32	97.19	(96.35–97.84)
	Swin+drloc	97.89	97.97	(96.35–97.84)
	Cswin+ drloc	99.21	99.30	(98.82–99.59)
	Cswin-*S*+drloc	**99.46**	**99.41**	**(98.95–99.67)**
D_2	Vit+drloc	85.87	85.76	(83.44–87.8)
	Swin+ drloc	86.85	87.11	(84.87–89.06)
	Cswin+ drloc	87.21	87.15	(84.91–89.1)
	Cswin-*S*+ drloc	**88.42**	**88.51**	**(86.37–90.35)**

Table 6
Detailed parameters of the Swin and CSwin models.

Downsp.rate Stage1	Downsp.rate (output size)	Swin	CSwin
Stage1	4 × (56 × 56)	*Concat4* × *4,96-d,LN*	*Concat4* × *4,64-d, LN*
		$\begin{bmatrix} win.size\ 7*7 \\ dim96, head\ 3 \end{bmatrix} \times 2$	$\begin{bmatrix} win.size7*7 \\ sw\ 1, head\ 2 \end{bmatrix} \times 2$
Stage2	8 × (28 × 28)	*Concat4* × *4192-d,LN*	*Concat4* × *4128-d, LN*
		$\begin{bmatrix} win.size\ 7*7 \\ dim192, head\ 6 \end{bmatrix} \times 2$	$\begin{bmatrix} win.size7*7 \\ sw\ 2, head\ 4 \end{bmatrix} \times 4$
Stage3	16 × (14 × 14)	*Concat4* × *4384-d,LN*	*Concat4* × *4256-d, LN*
		$\begin{bmatrix} win.size\ 7*7 \\ dim384, head\ 12 \end{bmatrix} \times 18$	$\begin{bmatrix} win.size7*7 \\ sw\ 7, head\ 8 \end{bmatrix} \times 32$
Stage4	32 × (7 × 7)	*Concat4* × *4768-d,LN*	*Concat4* × *4512-d, LN*
		$\begin{bmatrix} win.size\ 7*7 \\ dim768, head\ 24 \end{bmatrix} \times 2$	$\begin{bmatrix} win.size7*7 \\ sw\ 7, head\ 16 \end{bmatrix} \times 2$

obtains validation accuracy curve on validation set, as shown in Fig. 15.

4.3.2. *Experiment on the effectiveness on BLA and DRLOC modules*

Although Cswin solves the problem of computational complexity, Cswin still has not got rid of the inability to be modeled globally. With the deepening of the network model, the underlying global information will be gradually ignored by the deep network. Therefore, this study introduces the method of bilateral attention on the basis of unilateral attention, draws on the idea of CSwin_boat [31] network, and proposes the network structure of CSwin-S, which is shown in Fig. 1. This structure incorporates the Feature-Space Local Attention (FSLA) mechanism, which involves shuffling all pixels within a patch. Subsequently, these pixels are divided into window-sized tokens through traversal, clustering, and dichotomy methods. This procedure prioritizes processing highly correlated information among each token in advance. It consolidates the correlated pixels within a patch into a single token, thereby reinforcing the internal connections between tokens. A bilateral effect is achieved thereof. The network is also slightly improved in order to enhance the performance of the relationship information between tokens, but it is not expected to dominate, so that the model learns too

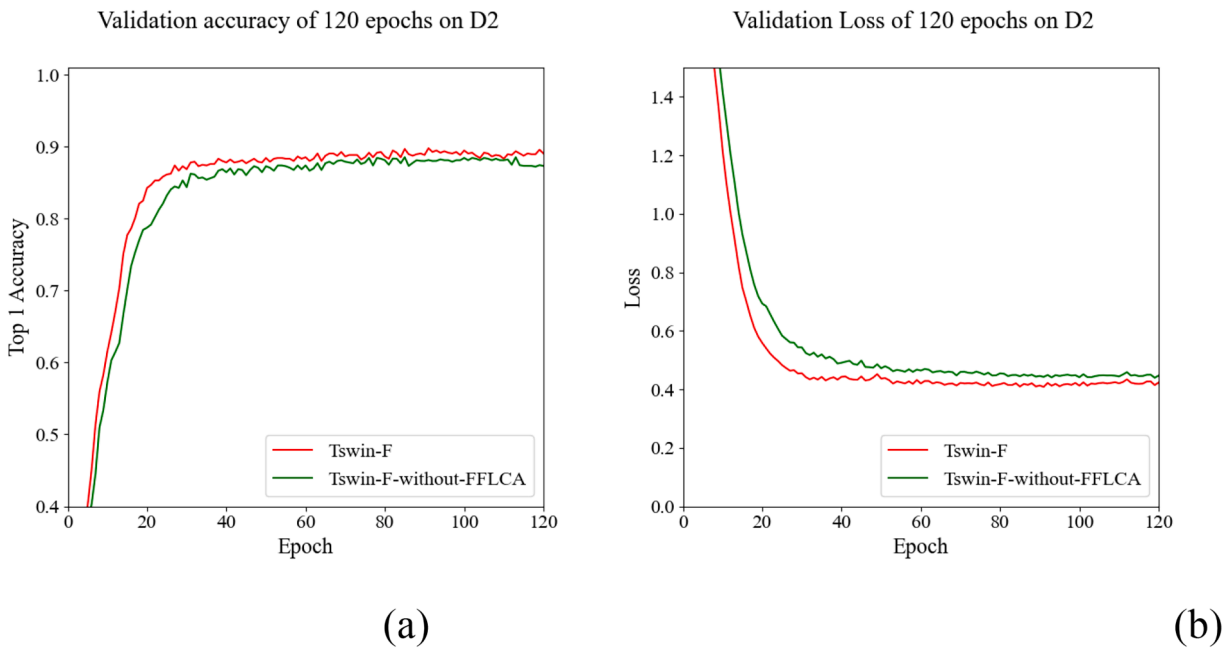

Fig. 18. (a) the accuracy curve of Tswin-F disease identification on D_2 using the FFLCA module and without the FFLCA module. (b) the loss curve for both models.

Table 7
The results of the baseline models within a 95% confidence interval on test set of D_2(The division ratio is 8:1:1).

Model	Acc(%)	F1 Score	F1 95% CI
Tswin	87.05	87.01	(84.77–88.97)
Tswin+Boat	87.36	87.42	(85.2–89.35)
Tswin+Boat+FAWD	88.25	88.31	(86.15–90.17)
Tswin+Boat+FAWD+Drloc	88.83	88.79	(86.67–90.61)
Tswin+Boat+FAWD+Drloc+FFLCA	**89.98**	**89.87**	**(87.83–91.6)**

Table 8
The parameter information generated by different network models on the D_1 dataset and the training time and inference time required to perform 120 epochs.

Model	Flops (G)	Params (M)	Train time (h)	Infer time (h)
EfficientNetV2	2.90	21.5	1.54	0.58
VIT	17.58	87	1.63	0.67
Swin	8.76	49	2.73	0.56
CSwin	6.82	34	5.62	0.86
Swin-boat	10.24	55	5.91	0.93
CSwin-boat	8.06	41.1	6.12	1.05
Densenet121	2.91	7	2.56	0.53
Tswin	11.32	43	9.84	0.73
Tswin-F	28.30	55	11.01	0.73

much about the position information on the image and ignores the overall information. Therefore, we use a coefficient as the constraint of FSLA in the overall self-attention mode, making it the constraint term of the entire self-attention. Hence, the final calculated self-attention contains both local and global information. Meanwhile, after the network model proposes to add the specification module FFN, the output of each residual module in the transformer is normalized and then merged with the main branch to prevent the amplitude of the main branch from increasing and make the model activation amplitude more moderate. We renamed the CSwin-S module as the Tswin module for subsequent research. The output of the Tswin (Cswin-S) network module is shown in Fig. 16.

From Fig. 16, it can be seen that the Tswin proposed in this study outperforms that of the native CSwin-Boat model. In order to improve the performance of the model, this study proposes to introduce a self-supervised learning strategy on the network structure of Tswin, by which a more general expression of the overall dataset is learned and migrated to the network. The self-supervised method of plug-and-play is used, which selects pixel pairs of specified size for the input image, predicts the spatial position of the pixel pairs, calculates the position loss value of the pixel pairs and superimposes them on the cross-entropy loss as a constraint, and updates them during the network training process.

Fig. 17 shows the comparison between the Tswin with the addition of a self-supervised module and the non-self-supervised module. Table 4 shows the highest recognition accuracy results obtained by the three

Table 9
The results of the models within a 95% confidence interval on test set of D_1(8:1:1) and D_2(6:2:2).

Models	Pre (D1)	Rca (D1)	F1 (D1)	Acc(%) (D1)	F1 95%CI	Pre (D2)	Rca (D2)	F1 (D2)	Acc(%) (D2)	F1 95%CI
VGG16	97.3	97.6	97.45	98.2	(96.64–98.06)	86.34	84.65	85.49	84.91	(83.86- 86.97)
Resnet34	97.9	98.6	98.24	98.8	(97.54–98.73)	87.82	85.55	86.67	87.71	(85.09- 88.10)
DenseNet121	99.53	99.41	99.46	99.5	(99.01–99.70)	88.35	87.35	87.85	88.58	(86.33- 89.21)
EfficientNetV2	98.86	97.78	98.32	98	(97.64–98.80)	87.75	86.62	87.18	87.35	(85.63–88.58)
VIT	94.28	94.22	94.25	93.73	(93.11–95.20)	85.92	83.16	84.52	84.15	(82.85–86.05)
Swin	95.48	95.64	95.56	95.41	(94.54–96.39)	87.29	84.62	85.94	85.72	(84.33–87.40)
CSwin	96.54	96.27	96.40	96.34	(95.46–97.14)	87.34	84.65	85.98	85.83	(84.37–87.44)
Swin-boat	97.70	97.07	97.38	97.33	(96.56–98.01)	88.42	84.51	86.89	86.64	(85.32–88.30)
CSwin-boat	98.83	98.28	98.55	98.28	(97.90–98.99)	90.27	86.45	88.32	87.66	(85.43–89.54)
Tswin	99.36	99.14	99.24	99.24	(98.74–99.54)	91.24	86.75	88.94	88.21	(87.47–90.25)
Tswin-F	**99.6**	**99.42**	**99.51**	**99.64**	(99.08–99.73)	**92.80**	**90.86**	**91.82**	**90.81**	(90.53–92.95)

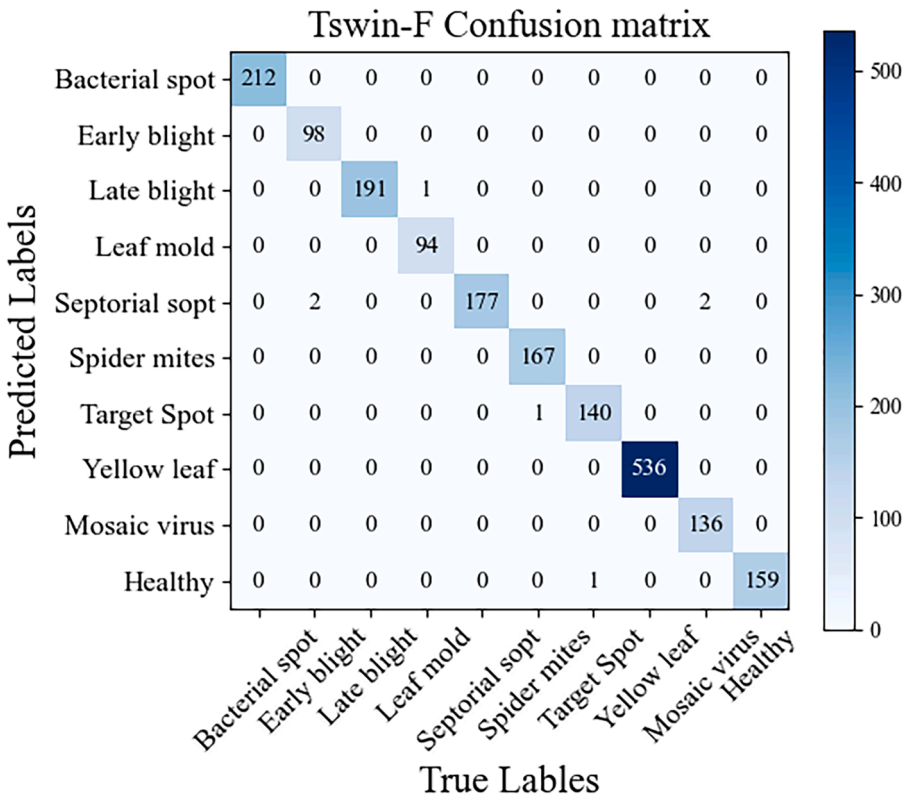

Fig. 19. The confusion matrices of the test using Tswin-F on D_1 test set.

Table 10
Parameters setting for the Proposed Tswin-F.

Hyper-parameter	Settings in D_1	Setting in D_2
Batch Size	32	32
Image Size	224×224	224×224
Drop Path Rate	0.2	0.4
Label Smoothing	0.1	0.1
Base Lr	3e-05	5e-05
Warm Lr	2.5e-06	5e-06
Min Lr	6.25e-08	6.25e-07
WarmUp Epoch	30	30
Weight Decay	0.005	0.00035

models of Vit, Swin, and CSwin on the D_1 and D_2 datasets after using the BLA (Bilateral Local Attention) module. Table 5 is the test accuracy obtained by Vit, Swin, Cswin, and Cswin-S network structures on the D_1 and D_2 datasets after using the DRLOC [26] module. Table 6 is the network parameter details used in the comparison experiment.

4.3.3. *Experiment on the effectiveness on FFLCA module*

In this study, the direction of feature fusion is explored. We found during the experimental training that the transformer model, like the CNN convolutional neural network, faces an increase in the attention distance on the feature map as the network depth increases, so the feature representations learned at different depths different, but the deeper the network model, the more the feature information learned at shallow layers is discarded. Therefore, to fuse the feature information at different levels, we design a classification layer fusion FFLCA module. Different from the native Vit model used before, we use the FFLCA module to fuse the feature information learned in different layers as the final representation of the final L layer, and pay attention to the features of different layers. And by setting the initial value as the fusion ratio of each layer, this practice is to prevent a certain layer of features from dominating the overall feature expression. Also, to prevent the loss surge

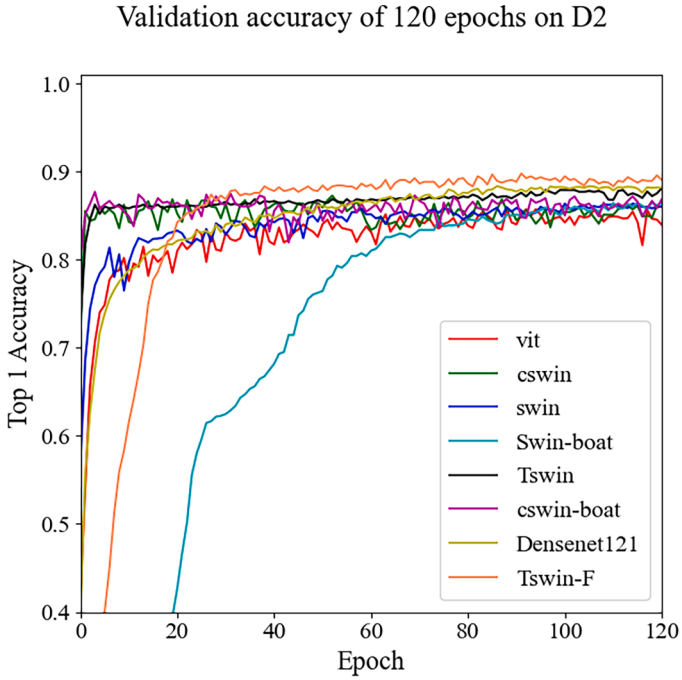

Fig. 20. The validation accuracy obtained by all models on the D_2 validation set.

or sharp drop of data in the reverse update due to feature fusion, this study uses a layer of sigmoid activation function to constrain the entire final output. Finally, our proposed Tswin-F (Tswin+FFLCA) model for the data yields the results shown in Fig. 18. The left Fig. 18(a) shows the validation accuracy curve, and the right Fig. 18(b) shows the validation loss curve.

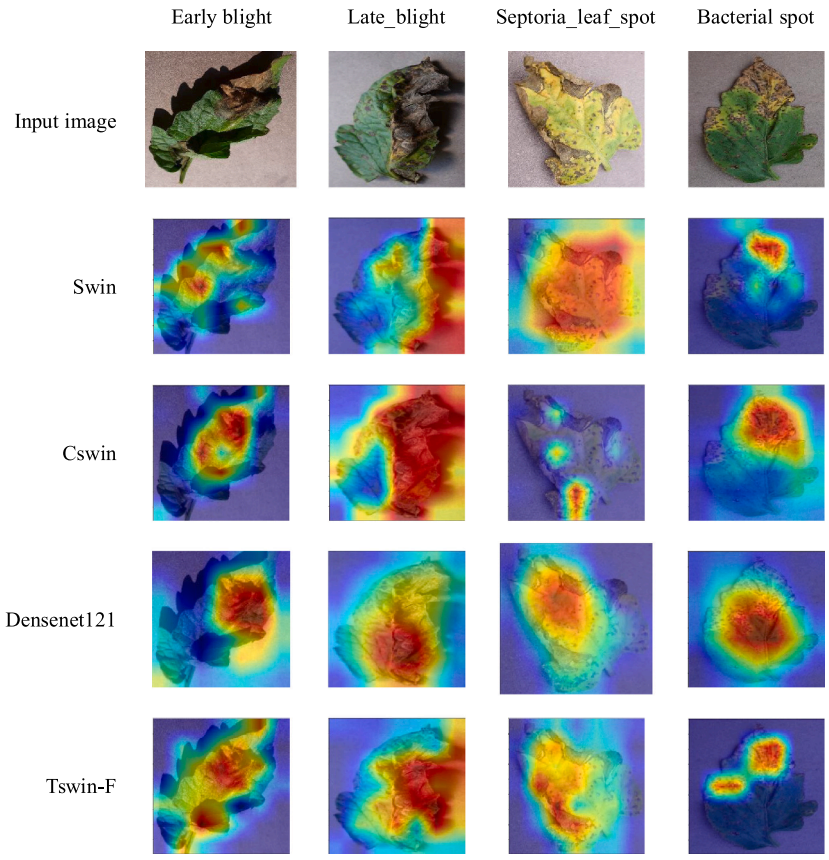

Fig. 21. Network visualization with Grad-CAM.

4.4. Ablation study

To further determine the effect of the proposed module, we conducted ablation studies on the main components of bilateral attention mechanisms, FAWD, DRLOC, and FFLCA. From Table 7, it is evident that the model exhibits significant improvement after the incorporation of the four modules.

4.5. Comparative experiments

Table 8 presents detailed information on the parameter counts of various network models, which serves as an important indicator of model size and complexity. We note that our proposed model has a larger parameter count, resulting in higher computational requirements and longer training times. These are some drawbacks of our model that need to be considered, taking into account the resource and time investment. However, despite these drawbacks, our model exhibits significant advantages in terms of inference speed. Through careful design and optimization strategies, our model achieves faster inference speed while maintaining high performance. This is particularly valuable for real-time or resource-efficient applications.

Table 9 provides the accuracy, precision, recall, and F1-score of various models on datasets D1 and D2. From the results in Table 8, it is evident that our proposed Tswin-F network achieves the highest accuracy and F1-score on both datasets. This indicates the effectiveness of our proposed network in achieving accurate and balanced classification results across different datasets. The superior performance of our model highlights its potential for robust and reliable classification tasks.

As shown in Table 9, when the small dataset is divided for training and testing in an 8:1:1 ratio, the Wilson confidence intervals for various models are relatively close, with this phenomenon becoming more pronounced on smaller datasets. Therefore, we applied a 6:2:2 ratio for partitioning and training/testing on the D_2 dataset, as shown in Table 9. As the dataset size increases, differences between models become more evident, and our proposed Tswin-F achieves optimal performance in D_2.

The confusion matrices of the test results using Tswin- F are shown in Fig. 19. From the confusion matrix in Fig. 19, it can be observed that some individual test samples tend to be misclassified as the Septorial spot class. This phenomenon may be attributed to a slight imbalance in the distribution of classes within the training data.

Table 10 presents the optimal hyperparameters used for training the network models in the study on their respective datasets. It includes key hyperparameters such as batch size, learning rate, weight decay coefficient, and others.

Fig. 20 shows the validation accuracy obtained by training 120 epochs on the dataset D_1, and our proposed Tswin-F network model obtains the highest accuracy, reaching 90.81%. Table 6 details the accuracy, model params, and flops of all models. The size of all input images is 224×224, and the batch size is 32.

Fig. 21 displays the attention visualization of three networks in terms of disease feature recognition. From the figure, it is evident that our proposed network model excels in target localization and small target identification based on the attention regions focused by the network on leaf disease features. These visualizations highlight the effectiveness of our model in accurately identifying and localizing disease-related regions, even for small and subtle features.

5. Conclusion

In this paper, a new effective image classification network Tswin-F is proposed, which focuses on solving the problems of loss of underlying feature information due to lack of data and long attention distance of plant disease spots in the Transformer network. Four core modules are proposed, namely BLA, DRLOC, FFLCA, FAWD. In window partitioning,

we use the BLA module to calculate self-attention within the local window, while introducing a bilateral attention mechanism to capture the relationship between distant but similar patches in the image plane. This paper proposes to use the DRLOC module to perform self-supervised learning on random position information of images, and generate a position attention representation containing sufficient position information. In the final stage of the decoder, the FFLCA module was developed, which further solved the problem of global information loss in the transformer network due to the increase in the number of network layers. The output features of each layer of the network are continuously updated and fused in the network, and finally the network model is represented by features that are refined enough and contain global location information. The classification effect of the leaf data of two published tomato lesions showed the superiority of the proposed method, with an accuracy rate of 90.81% and 99.64%, respectively. In order to solve the phenomenon of overfitting of small datasets on Transformer, we propose a more novel parameter adjustment technique to effectively suppress network overfitting by combining Flood and dynamic weight decay. Furthermore, the Tswin-F network also exhibits some limitations. Firstly, due to the utilization of a multi-head attention mechanism, the computational complexity of the Tswin-F network is higher, resulting in longer training times. Secondly, the model's larger size could pose challenges for training on large datasets. Secondly, because it uses additional attention computation in the feature map extraction stage, it will also further increase the computational overhead. In addition, the void convolution used in the feature fusion stage for normalization of size dimension will also increase the computational overhead of the network model. Future research can be focused on further optimizing the model's parameter count and computational complexity. One potential approach is to explore efficient instance segmentation methods for extracting local features from images. Additionally, adopting multi-head attention mechanisms as a replacement for traditional convolutional operations can help accelerate the model's runtime speed. Additionally, the modular design of this model holds the potential for future transformation into a versatile model encompassing detection, recognition, segmentation, and other functionalities.

CRediT authorship contribution statement

Yuanbo Ye: Writing – original draft. **Houkui Zhou:** Supervision, Conceptualization. **Huimin Yu:** Project administration. **Haoji Hu:** Supervision. **Guangqun Zhang:** Funding acquisition. **Junguo Hu:** Supervision, Funding acquisition. **Tao He:** Investigation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

The authors would like to thank the anonymous reviewers for their constructive comments and suggestions, which significantly contributed to improving the manuscript. This research was supported by Zhejiang Provincial Natural Science Foundation of China under Grant No. LY24F020005.

References

[1] E.J. Collins, C. Bowyer, A. Tsouza, M. Chopra, Tomatoes: an extensive review of the associated health impacts of tomatoes and factors that can affect their cultivation, Biology (Basel) 11 (2) (2022) 239.
[2] Y. Wu, L. Xu, E.D. Goodman, Tomato leaf disease identification and detection based on deep convolutional neural network, Intell. Autom. Soft Comput. 28 (2) (2021) 561–576.
[3] S. Ganguly, P. Bhowal, D. Oliva, R. Sarkar, BLeafNet: a Bonferroni mean operator based fusion of CNN models for plant identification using leaf image classification, Ecol. Inform. 69 (2022) 101585.
[4] P. Zhang, L. Yang, D.L. Li, EfficientNet-B4-Ranger: a novel method for greenhouse cucumber disease recognition under natural complex environment, Comput. Electron. Agric. 176 (2020) 105652.
[5] C.K. Sunil, C.D. Jaidhar, N. Patil, Cardamom plant disease detection approach using efficientNetV2, IEEe Access. 10 (2022) 789–804.
[6] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, N. Houlsby, An Image is Worth 16x16 Words: Transformers for Image Recognition At Scale, arXiv preprint, 2020. arXiv:2010.11929.
[7] K. Han, Y.H. Wang, H.T. Chen, A Survey on Vision Transformer, IEEe Trans. Pattern. Anal. Mach. Intell. 45 (1) (2023) 87–110.
[8] C. Yang, Plant leaf recognition by integrating shape and texture features, Pattern. Recognit. 112 (2021) 107809.
[9] V.N.T. Le, S. Ahderom, B. Apopei, K. Alameh, A novel method for detecting morphologically similar crops and weeds based on the combination of contour masks and filtered Local Binary Pattern operators, Gigascience 9 (3) (2020) giaa017.
[10] X. Chen, G. Zhou, A. Chen, J. Yi, W. Zhang, Y. Hu, Identification of tomato leaf diseases based on combination of ABCK-BWTR and B-ARNet, Comput. Electron. Agric. 178 (2020) 105730.
[11] W. Chen, Y. Rao, F. Wang, Y. Zhang, T. Wang, X. Jin, W. Hou, Z. Jiang, W. Zhang, MLP-based multimodal tomato detection in complex scenarios: insights from task-specific analysis of feature fusion architectures, Comput. Electron. Agric. 221 (2024) 108951.
[12] J. Chen, X. Deng, Y. Wen, W. Chen, A. Zeb, D. Zhang, Weakly-supervised learning method for the recognition of potato leaf diseases, Artif. Intell. Rev. 56 (8) (2023) 7985–8002.
[13] C.Z. Yang, Plant leaf recognition by integrating shape and texture features, Pattern. Recognit. 112 (2021).
[14] G. Yang, G. Chen, Y. He, Z. Yan, Y. Guo, J. Ding, Self-supervised collaborative multi-network for fine-grained visual categorization of tomato diseases, IEEe Access. 8 (2020) 211912–211923.
[15] Q. Wu, Y. Chen, J. Meng, DCGAN-based data augmentation for tomato leaf disease identification, IEEe Access. 8 (2020) 98716–98728.
[16] Y. Zhao, C. Sun, X. Xu, J. Chen, RIC-Net: a plant disease classification model based on the fusion of Inception and residual structure and embedded attention mechanism, Comput. Electron. Agric. 193 (12) (2022) 106644.
[17] H. Chen, H. Zhang, C. Liu, J. An, Z. Gao, J. Qiu, FET-FGVC: feature-enhanced transformer for fine-grained visual classification, Pattern. Recognit. 149 (2024).
[18] T. Anandhakrishnan, S.M. Jaisakthi, Deep Convolutional Neural Networks for image based tomato leaf disease detection, Sustain. Chem. Pharm. 30 (13) (2022) 100793.
[19] X. Liu, W. Min, S. Mei, L. Wang, S. Jiang, Plant disease recognition: a large-scale benchmark dataset and a visual region and loss reweighting approach, IEEe Trans. Image Process. 30 (2021) (2021) 2003–2015.
[20] W. Liu, C. Li, N. Xu, T. Jiang, M.M. Rahaman, H. Sun, X. Wu, W. Hu, H. Chen, C. Sun, CVM-Cervix: a hybrid cervical Pap-smear image classification framework using CNN, visual transformer and multilayer perceptron, Pattern. Recognit. 130 (4) (2022) 108829.
[21] W. Yawei, C. Yifei, W. Dongfeng, Convolution network enlightened transformer for regional crop disease classification, Electronics (Basel) 11 (19) (2022) 3174.
[22] W. FY, R. Yuan, L. Qing, J. Xiu, J. Zhaohui, Z. Wu, L. Shaowen, Practical cucumber leaf disease recognition using improved Swin Transformer and small sample size, Comput. Electron. Agric. 2022 (4) (2022) 107163S.
[23] G. Yifan, L. Yanting, C. Xiaodong, CST: convolutional Swin Transformer for detecting the degree and types of plant diseases, Comput. Electron. Agric. 2022 (2022) 1783445.
[24] H.T. Thai, K.H. Le, N.L.T. Nguyen, FormerLeaf: an efficient vision transformer for Cassava Leaf Disease detection, Comput. Electron. Agric. 204 (21) (2023) 107518.
[25] X. Dong, J. Bao, D. Chen, W. Zhang, N. Yu, L. Yuan, D. Chen, B. Guo, CSWin transformer: a general vision transformer backbone with cross-shaped windows, Comput. Vision Pattern Recognit. 2020 (23) (2022) 12124–12134.
[26] Y. Liu, E. Sangineto, W. Bi, N. Sebe, B. Lepri, M.D. Nadai, Efficient training of visual transformers with small datasets, Comput. Vision Pattern Recognit. (2021).
[27] T. Ishida, I. Yamane, T. Sakai, G. Niu, M. Sugiyama, Do We Need Zero Training Loss After Achieving Zero Training Error? arXiv preprint, 2020. arXpv:2002.08709.
[28] B.-D. Liu, S. Shao, C. Zhao, L. Xing, W. Liu, W. Cao, Y. Zhou, Few-shot image classification via hybrid representation, Pattern. Recognit. (2024) 110640.

[29] D. Hughes, M. Salathé, An Open Access Repository of Images On Plant Health to Enable the Development of Mobile Disease diagnostics[J], arXiv preprint, 2015. arXiv:1511.08060. https://data.mendeley.com/datasets/tywbtsjrjv/1.
[30] X.P. Li, S.Q. Li, Transformer help CNN see better: a lightweight hybrid apple disease identification model based on transformers, Agriculture-Basel 12 (6) (2022) 884.
[31] T. Yu, G. Zhao, P. Li, Y. Yu, BOAT: Bilateral Local Attention Vision Transformer, arXiv preprint, 2022. arXiv:2201.13027.

Yuanbo Ye is a graduate student at the School of Mathematics and Computer Science at Zhejiang A&F University. He is currently working toward M.S. degree in Information Engineering.

Houkui Zhou received the Ph.D. degree in Signal and information processing from Zhejiang University, City, China. His research interests include digital image processing, machine vision, and the theory and application of thematic models.

Huimin Yu is a professor at the School of Information Science and Engineering, Zhejiang University, graduated from the major of signal and information processing, and is a doctoral supervisor.

Haoji Hu graduated from the Department of Electrical and Computer Engineering at the University of Southampton in 2007, and his research direction was the fusion algorithm of face recognition and speaker recognition, and he is currently an associate professor at Zhejiang University.

Guangqun Zhang graduated with a master's degree as an associate professor with a research focus on graphic image processing. Mainly engaged in intelligent image processing, machine learning algorithm application research in agriculture and forestry.

Junguo Hu received the M.S. degree in computer science from Zhejiang University of Technology, mainly engaged in embedded systems, intelligent monitoring and evaluation of forest carbon sink carbon cycle, artificial intelligence, and wireless sensor networks.

Tao He received the PH.D. degree in Forest Management from Beijing Forestry University. His research focuses on forest remote sensing image recognition and forest carbon stock modeling.