agriculture

MDPI

Article

A Dual-Branch Model Integrating CNN and Swin Transformer for Efficient Apple Leaf Disease Classification

Haiping Si, Mingchun Li, Weixia Li, Guipei Zhang, Ming Wang, Feitao Li and Yanling Li *

College of Information and Management Science, Henan Agricultural University, Zhengzhou 450046, China; haiping@henau.edu.cn (H.S.); lmc1224307033@stu.henau.edu.cn (M.L.); lwx@stu.henau.edu.cn (W.L.); zdsqpei@stu.henau.edu.cn (G.Z.); wangming@henau.edu.cn (M.W.); lft045006@henau.edu.cn (F.L.)
* Correspondence: lyl_lingling@163.com

Abstract: Apples, as the fourth-largest globally produced fruit, play a crucial role in modern agriculture. However, accurately identifying apple diseases remains a significant challenge as failure in this regard leads to economic losses and poses threats to food safety. With the rapid development of artificial intelligence, advanced deep learning methods such as convolutional neural networks (CNNs) and Transformer-based technologies have made notable achievements in the agricultural field. In this study, we propose a dual-branch model named DBCoST, integrating CNN and Swin Transformer. CNNs focus on extracting local information, while Transformers are known for their ability to capture global information. The model aims to fully leverage the advantages of both in extracting local and global information. Additionally, we introduce the feature fusion module (FFM), which comprises a residual module and an enhanced Squeeze-and-Excitation (SE) attention mechanism, for more effective fusion and retention of both local and global information. In the natural environment, there are various sources of noise, such as the overlapping of apple branches and leaves, as well as the presence of fruits, which increase the complexity of accurately identifying diseases on apple leaves. This unique challenge provides a robust experimental foundation for validating the performance of our model. We comprehensively evaluate our model by conducting comparative experiments with other classification models under identical conditions. The experimental results demonstrate that our model outperforms other models across various metrics, including accuracy, recall, precision, and F1 score, achieving values of 97.32%, 97.33%, 97.40%, and 97.36%, respectively. Furthermore, detailed comparisons of our model's accuracy across different diseases reveal accuracy rates exceeding 96% for each disease. In summary, our model performs better overall, achieving balanced accuracy across different apple leaf diseases.

Keywords: dual branch; feature fusion; image classification; CNN; Transformer

Citation: Si, H.; Li, M.; Li, W.; Zhang, G.; Wang, M.; Li, F.; Li, Y. A Dual-Branch Model Integrating CNN and Swin Transformer for Efficient Apple Leaf Disease Classification. *Agriculture* **2024**, *14*, 142. https://doi.org/10.3390/agriculture14010142

Academic Editor: Jiehao Li

Received: 20 November 2023
Revised: 8 January 2024
Accepted: 12 January 2024
Published: 18 January 2024

1. Introduction

China, a major global producer and importer of apples, faces substantial impacts on agricultural yields due to diseases affecting apple crops [1]. Failure to accurately detect these diseases may result in considerable economic losses to the apple industry. Traditional methods for disease identification, reliant on manual field observations, incur substantial human and resource costs in large-scale cultivation and may miss optimal intervention windows. Therefore, there is an urgent need for a convenient and accurate methodology for apple disease classification.

The development of machine learning has provided effective solutions to the problem of disease classification, including Support Vector Machine (SVM) [2], K-Nearest Neighbors (K-NN) [3], Random Forest [4], Genetic Algorithms [5], and Principal Component Analysis (PCA) [6], which have been employed for disease classification in the past few years.

With the advancement of deep learning, notably the outstanding performance of convolutional neural networks (CNNs) and Transformers in computer vision, scholars have progressively applied these methodologies to tasks related to leaf diseases.

1.1. CNN-Based Methods

Several notable CNN-based models [7–12] have been developed for leaf disease recognition. Yan et al. [13] proposed an enhanced VGG16 model for classifying apple leaf diseases. This model reduces the number of parameters by replacing the fully connected layer with a global average pooling layer and incorporates a batch normalization layer to enhance convergence speed. The experimental results demonstrate the model's ability to classify apple leaf diseases with an accuracy of 99.01%. Helong Yu et al. [14] introduced MSOResNet, a novel apple leaf disease recognition model, built upon the ResNet50 residual neural network foundation. The experimental results illustrate that the proposed model achieves an average precision, recall, and F1 score of 95.7%, 95.8%, and 95.7%, respectively. Yuanqiu Luo et al. [15] proposed an apple disease classification model based on multi-scale feature fusion. They enhanced the model's effectiveness by optimizing the information flow within the ResNet architecture and substituting standard convolutions with pyramid and dilated convolutions. The model achieved an impressive classification accuracy of 94.24% on a dataset comprising apple disease leaf images. Lili Fu et al. [16] introduced a convolutional neural network based on the AlexNet architecture. This network incorporates an attention mechanism to adapt to channel features and mitigate the influence of complex backgrounds. Additionally, it reduces the number of parameters by replacing two fully connected layers with global pooling. The model achieves a commendable recognition accuracy of 97.36%. Helong Yu et al. [17] proposed a lightweight ResNet model for the recognition of apple diseases. Building upon the deep residual network, they constructed a multi-scale feature extraction layer using group convolution. Additionally, an efficient channel attention module was incorporated to suppress noise originating from complex backgrounds. The experimental results revealed that LW-ResNet achieved an impressive average precision, recall, and F1 score of 97.80%, 97.92%, and 97.85%, respectively, on the test dataset.

While the preceding research has achieved success, it is not without limitations. Due to the presence of receptive fields, shared parameters, and inductive biases, CNNs excel at extracting local features but lack the ability to capture global information and establish long-range dependencies. Some studies have proposed effective solutions, such as enlarging the receptive field or employing deeper network architectures. However, these approaches come with the drawback of requiring increased computational resources and hardware.

1.2. Transformer-Based Methods

Transformer [18], originating in natural language processing, introduced the self-attention mechanism, significantly enhancing its ability to handle long-range dependencies. Subsequently, this concept has been successfully applied in computer vision, leading to the development of a series of Transformer models [18–21]. Alexey et al. [19] proposed the first Transformer model for computer vision, named Vision Transformer (ViT). ViT partitions images into fixed-size non-overlapping patches, flattening them into one-dimensional vectors while incorporating positional encoding information. Leveraging the self-attention mechanism, ViT effectively captures global information from images. However, ViT also has its limitations, including a quadratic relationship between computational complexity and input feature size, demanding substantial data and computational resources for training. Self-attention mechanisms generally require significant computational resources, potentially resulting in longer training and inference times and a higher hardware resource burden. In response to these challenges, Ze Liu et al. [21] proposed Swin Transformer, a hierarchical Transformer based on a shift window design. It introduces a window-based self-attention mechanism and a shift window self-attention mechanism. The first attention mechanism aims to reduce the computational complexity of self-attention, while the second one addresses the limitation of non-interaction between information from different windows imposed by the window-based self-attention mechanism. These two self-attention mechanisms help the model to better integrate local and global information, contributing to the enhancement of model performance.

Based on the preceding discussion, it is evident that CNNs and Transformers emphasize different aspects. CNNs inherently excel in processing local features through operations such as convolution and pooling layers, yet they face challenges in capturing global information due to limited receptive fields. In contrast, Transformers excel in capturing global information and establishing long-range dependencies through the self-attention mechanism, but they lack the ability to effectively extract local features.

1.3. The Contributions of This Study

To address their individual deficiencies, we propose a dual-branch model, denoted as DBCoST, which integrates CNN and Transformer architectures for the purpose of apple disease recognition. In this study, the CNN branch, designed on the foundation of ResNet-18, is tasked with extracting localized features and contextual information. Concurrently, the Transformer branch, built upon Swin Transformer Tiny, is specifically tailored to capture global information and establish long-range dependencies.

Furthermore, we introduce a feature fusion module with the objective of amalgamating the distinctive characteristics of both branches. Through the amalgamation of features and subsequent refinement via residual layers, a more enriched set of features is obtained. Following the acquisition of feature representations from the individual branches, a channel attention mechanism is introduced to autonomously learn and adjust the importance weights among feature channels. This mechanism enhances the fusion of information within the model, thereby significantly augmenting its performance and generalization capabilities. The primary contributions of this paper are delineated as follows:

(1) We propose DBCoST, a dual-branch model integrating CNN and Swin Transformer. The CNN branch is designed based on ResNet18, and the Transformer branch adopts the structure of Swin Transformer Tiny. This model harnesses the capabilities of both CNN and Transformer to capture local features and global information with long-range dependencies. It demonstrates noteworthy results, particularly in challenging environments.
(2) An effective feature fusion mechanism is introduced, comprising a residual structure and a channel attention mechanism. By integrating features from the CNN and Transformer branches and further refining them through a residual structure, channel attention mechanisms autonomously learn and adjust importance weights between feature channels. This facilitates improved integration of information from both branches, leading to enhanced performance and generalization.
(3) Under consistent conditions, we conducted a comparative evaluation against selected traditional state-of-the-art (SOTA) models. The experimental results clearly demonstrate the commendable performance of our model.

2. Materials and Methods

2.1. Dataset

Our study employed the publicly available dataset "Plant Pathology 2021 - FGVC8" [22] sourced from Kaggle, which comprises 18,632 high-quality images with dimensions of 4000×2672 pixels. To ensure the precision of our research, we concentrated on five specifically defined disease types, excluding any unspecified disease types present in the dataset. Figure 1 illustrates the selected disease types, and Table 1 provides detailed information on these disease types and their respective quantities used in our study.

(**a**) (**b**) (**c**) (**d**) (**e**)

Figure 1. The selected apple disease category. (**a**) Frog eye spot; (**b**) powdery mildew; (**c**) scab; (**d**) rust; (**e**) healthy.

Table 1. The distribution of apple leaf disease name and quantity.

Categories	Original	After Process
frog eye spot	3181	3181
powdery mildew	4624	4394
rust	1184	4003
scab	1860	4004
healthy	4826	4121

2.2. Image Preprocessing

The collected data have a fixed size of 4000 × 2672, which does not meet the requirements of our model. Therefore, we used the OpenCV method to resize the images to 224 × 224. Additionally, the dataset exhibits an imbalance in quantity, and some images suffer from overexposure, which poses learning challenges for the model. Consequently, we removed overexposed images of diseases. To address the imbalance, the following operations were applied to expand the dataset: (1) random rotation images; (2) mirroring images; (3) adding Gaussian noise; (4) applying color jitter to modify image saturation, brightness, and contrast. The results of these operations are illustrated in Figure 2. After expansion and trimming, the dataset's total size is 19,703 images, including 3181 images of frog eye leaf spot, 4393 healthy images, 4003 images of powdery mildew, 4004 images of rust, and 4121 images of scab. The data quantity for each category is shown in Table 1.

Random Rotation Mirror Gaussian noise Color jitter

Figure 2. Image after data enhancement.

Furthermore, in our experiments, we applied data augmentation techniques to the dataset using torchvision's provided methods. Specifically, we employed operations including horizontal flipping, vertical flipping, random rotation, color jitter, and normalization. The objective of these data augmentation techniques is to mitigate the risk of overfitting and enhance the model's generalization capability.

2.3. Methods

2.3.1. Model Design

We acknowledge that different apple leaf diseases exhibit significant variations in their characteristics. For example, frog eye spot typically presents as punctate lesions, indicating a concentrated distribution of disease features, while scab manifests in a strip-like pattern, indicating a more dispersed distribution of disease features. Furthermore, images obtained from natural environments often contain background noise, which may influence the

accuracy of classification. This necessitates a model capable of integrating local features and global information while addressing noise.

In this section, the paper proposes a network that leverages the advantages of both CNN and Swin Transformer structures to enhance recognition accuracy. The overall structure, depicted in Figure 3, is divided into four layers. Each layer comprises multiple CNN and Swin Transformer blocks, as well as a feature fusion module (FFM). The CNN branch is employed to extract local features from the images, while Transformer is utilized to capture global information. The FFM module further refines and extracts information from both branches, incorporating attention mechanisms to highlight crucial information and suppress irrelevant details.

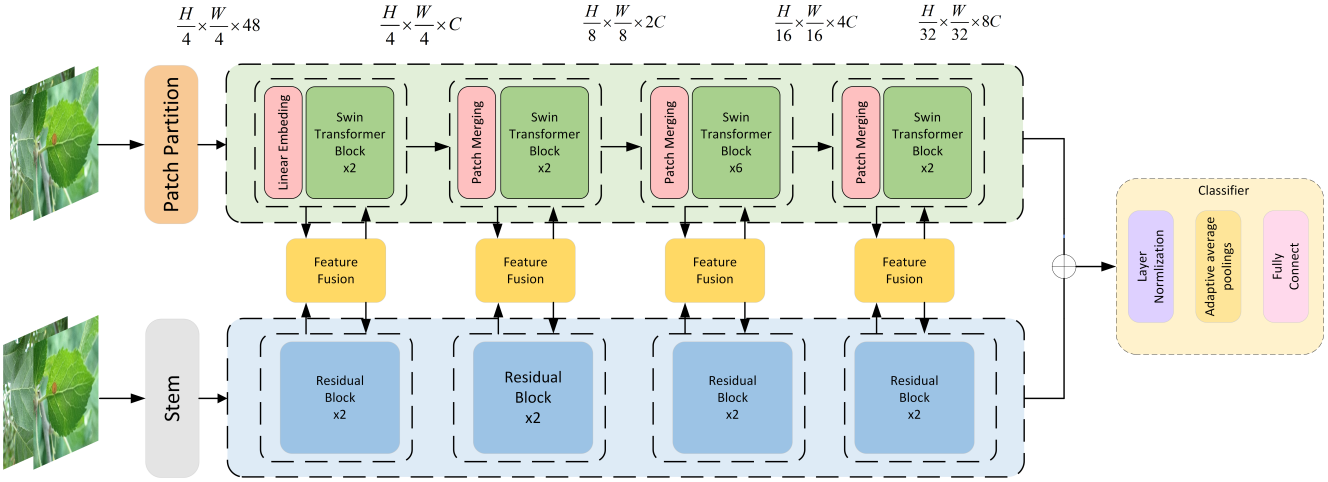


Figure 3. The architecture of DBCoST model.

In summary, our proposed model integrates the strengths of CNN and Swin Transformer, utilizes the FFM module to combine local and global information, and addresses noise.

The initial input image has dimensions of $H \times W \times C$, where H, W, and C denote the image's height, width, and channels, respectively. After passing through the patch embed and stem modules, the input is split into two branches. Within each layer, images from the Transformer and CNN branches undergo fusion processing before re-entering their respective branches for additional learning. The fusion process combines features from both branches, facilitating the extraction of distinctive features.

2.3.2. CNN Branch

We designed the CNN branch based on the structure of ResNet18, aiming to extract local features and contextual information from images. With the progression of model depth, the resolution gradually decreases while the number of channels increases. The overall structure is divided into four layers, each containing 2 ResBlocks. The structure of each ResBlock is illustrated in Figure 4a. Each ResBlock consists of three convolutional blocks: a 1×1 downsampling convolution, a 3×3 spatial convolution, and a 1×1 upsampling convolution. Additionally, there is a residual module connecting the input and output. The distinctive feature of Swin Transformer is its approach of dividing the input image into fixed-size patches for processing. Each patch contains only a portion of local information, potentially leading to the loss of some local details. Traditional CNNs capture local features by sliding convolutional kernels over the image, allowing them to better capture local information. This supplements the Transformer branch with finer local details.

2.3.3. Transformer Branch

In this branch, we adopted the original Swin Transformer structure as the backbone for the Transformer branch, responsible for capturing global information and long-range dependencies. The number of Swin Transformers at each stage is set as 2, 2, 6, 2. The struc-

ture is illustrated in Figure 4b. The Swin Transformer module comprises two sub-modules. The first sub-module includes layer normalization (LN) and a multi-layer perceptron (MLP) along with a window-based multi-head attention mechanism (W-MSA). The second sub-module is similar but utilizes a shifted window-based multi-head attention mechanism (SW-SWA) instead of the window-based multi-head attention mechanism. A Patch Merging module is employed for downsampling.

Assuming the input image is of size $H \times W \times 3$, in the first layer, the model utilizes the Patch Partition module, inspired by ViT, to divide the image into non-overlapping patches of size 4×4. These patches are then flattened along the channel direction. Each patch has a feature dimension of 16. After flattening across the R, G, and B channels, the feature dimension becomes 48. Consequently, the image size transforms to $H/4 \times W/4 \times 48$. Finally, a Linear Embedding layer is applied to perform a linear transformation on the channel data of each pixel. This final step changes the shape of the image from $H/4 \times W/4 \times 48$ to $H/4 \times W/4 \times C$.

From the second to fourth layers, the Patch Merging module is used for downsampling. Assuming the input feature map is 4×4, the Patch Merging partitions adjacent 2×2 image blocks in the feature map into multiple patches, which are then concatenated along the channel dimension and subjected to dimensionality reduction through a linear layer. In summary, as we move from the second to the fourth layers, the resolution of the feature map is halved progressively, while the number of channels increases to C, 2C, 4C, and 8C, respectively.

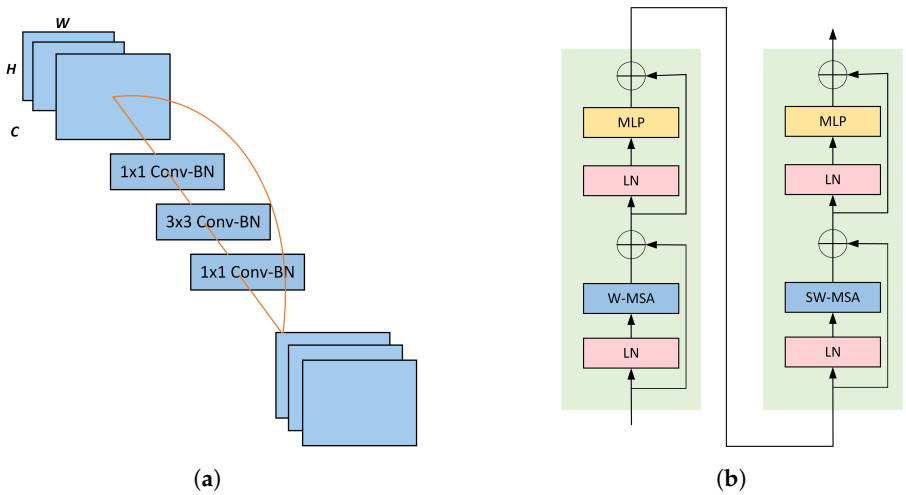


Figure 4. The details of residual block and Swin Transformer block. (**a**) Residual block; (**b**) Swin Transformer block.

2.3.4. Window-Based Multi-Head Self-Attention

The self-attention [18] mechanism originated in the field of NLP, where it treats text data as a sequence of individual words, each considered as a unit. This approach effectively captures contextual relationships in computations. In computer vision, a similar strategy is applied, where images are divided into fixed-size patches. These patches are then flattened and mapped into a one-dimensional vector S, generating three learnable matrices: Queries (W^Q), Keys (W^K), and Values (W^V). For the generated sequence, these matrices are multiplied to obtain the Q, K, and V parameters. The attention mechanism computes the dot product of Q and K, normalizes the result using SoftMax to produce an attention weight matrix, and finally multiplies it with V to yield the updated weighted matrix. The computation formula is shown below:

$$\text{Attention} = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right) \times V \quad (1)$$

Traditional self-attention mechanisms require calculating relationships for all elements, resulting in high computational complexity. However, the Swin Transformer introduces a window-based multi-head self-attention mechanism shown in Figure 5a. It first divides the image into fixed-size patches and then applies attention individually to each patch, significantly reducing computational complexity.

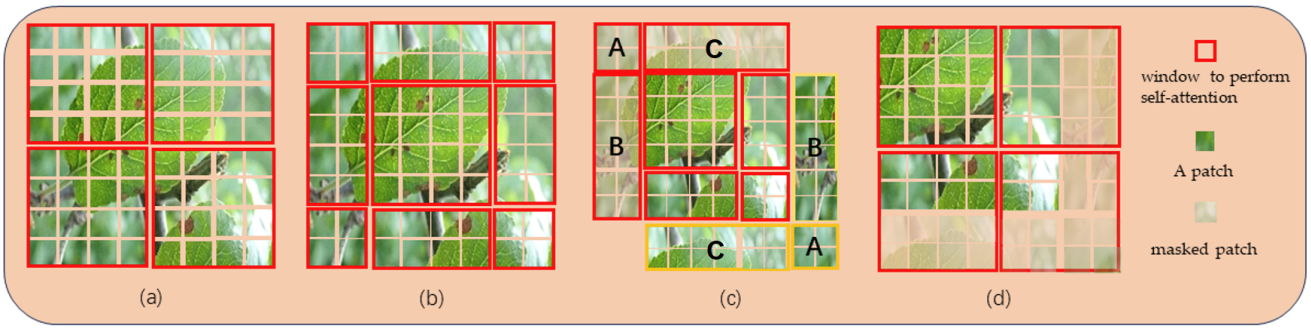


Figure 5. Illustration of the operation of the window-based attention mechanism and the shift window-based attention mechanism. (**a**) The window partition process based on W-MSA; (**b**) The shift window process based on SW-MSA; (**c**) Window cyclic shift process based on SW-MSA; (**d**) The results with masks

2.3.5. Shifted Window-Based Multi-Head Self-Attention

When employing the Window Self-Attention mechanism (W-MSA), despite the reduction in computational complexity achieved through the split operation, attention computation remains confined within individual windows, which hinders the information interaction between different windows. To address this limitation, the Swin Transformer introduces the Shifted Window Attention Mechanism, as depicted in Figure 5b. Through window shifting, the image is initially divided into nine parts. Figure 5c illustrates the cyclic shift mechanism, aimed at reorganizing the image into four parts similar to those in Figure 5a. The regions labeled as A, B, and C correspond to the top-left corner. Subsequently, these parts undergo cyclic shifts to align with the respective yellow areas. The primary objective of SW-SMA is to facilitate information exchange between adjacent windows. However, the split process leads to an increased number of windows, requiring additional computational resources and potentially introducing ambiguity when computing attention mechanisms for non-adjacent regions. To address these issues, the authors introduced a masking mechanism, as depicted in Figure 5d. This mechanism is designed to mask patches from different regions, only computing attention for patches within the same window.The described measures not only address the issue of the inability for interaction between different windows but also substantially reduce computational workload, thereby mitigating the demand for computational resources.

2.3.6. FFM

Different types of diseases may manifest multi-level abstract features in images, spanning from subtle texture differences to the overall structure of the leaves. Therefore, during the training process, the importance of both local details and global contextual information is acknowledged. Due to the distinct focuses of information extraction between the CNN and Swin Transformer branches, the fusion of these two types of information becomes imperative. The absence of such fusion may impact the accuracy of disease recognition.

To effectively integrate the local features extracted by the CNN branch and the global features captured by the Transformer branch, we propose a feature fusion module, as illustrated in Figure 6.

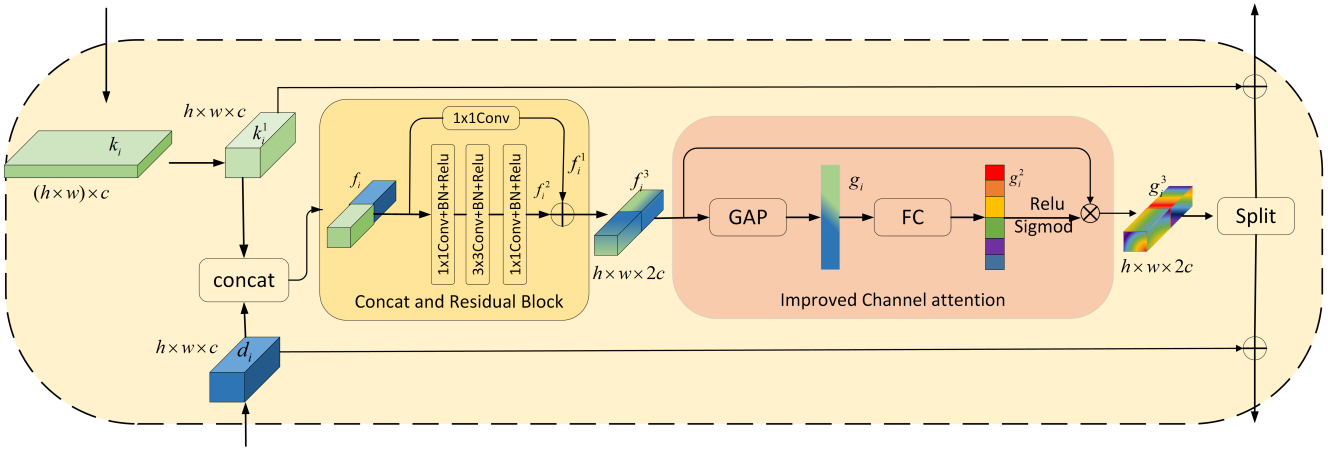


Figure 6. The architecture of FFM.

The FFM module operates as follows. Firstly, we establish bidirectional connections for features extracted from independent branches. Specifically, features extracted from the CNN branch and Transformer branch at the ith block are denoted as d_i and k_i, respectively. We use Fusion Module M_{fuse} to aggregate these two feature maps as

$$f_i = M^i_{fuse}(\text{reshape}(k_i)||d_i) \quad 1 \leq i \leq N \tag{2}$$

where $f_i \in R^{2c\times h\times w}$ denotes the fused feature, reshape represents reshaping the characteristics of the feature map from the Transformer branch, and $||$ denotes concatenation. We construct our Feature Fusion using two 1×1 convolutions and one 3×3 convolution for channel-wise fusion. Then, a channel attention mechanism is applied to suppress noise and highlight important features. Finally, the fused features are split into two along the channel dimension $k_i \in R^{c\times h\times w}$ and $d_i \in R^{c\times h\times w}$. Each fused feature is then directed back to its respective branch and added to the initial inputs d_i and k_i.

Our feature fusion module consists of two parts, each serving a distinct function. The first part is the Concatenation and Residual Block (CRB). CRB is designed to fuse and extract features from both branches. Specifically, k_i and d_i represent the feature maps from the Swin Transformer branch and the CNN branch, respectively. Assume that $d_i \in R^{h\times w\times c}$, while the 2D (two-dimensional) Swin branch feature $k_i \in R^{(h\times w)\times c}$, where $h \times w$ denotes the patch size, and c represents the number of channels in the feature map.

In this module, we first use a reshape operation to convert k_i of size $(h \times w) \times c$ to $h \times w \times c$, then concatenate d_i and k_i along the channel direction to obtain the fused feature. Subsequently, this fused feature is input into a residual structure for learning. The processing steps of CRB are defined by the following equations:

$$f_i = \text{cat}(d_i, k_i) \tag{3}$$

$$f_i^1 = \text{conv}(f_i) \tag{4}$$

$$f_i^2 = \text{residual}(f_i) \tag{5}$$

$$f_i^3 = f_i^2 + f_i^1 \tag{6}$$

Here, cat represents the concatenation operation, conv represents the convolution operation, residual represents the residual block, and the final summation denotes the residual connection. Equation (3) is used to concatenate features from both branches, while Equations (4) and (5) further extract features from the concatenated features. Equation (6) forms the residual connection, alleviating the vanishing gradient problem associated with increasing depth.

The second part is the improved channel attention (ICA) module. The different channels of the feature map contain various information, but not all of it is necessary.

We need to highlight important regions and ignore environmental factors. Therefore, we propose an improved channel attention mechanism. The experiments in ECA [23] revealed that the use of two fully connected layers by SE [24] has certain drawbacks. While this strategy reduces the model's complexity, it disrupts the direct correspondence between channels and their weights. Consequently, we chose to adopt a single fully connected layer to mitigate the impact. We input the feature map processed through CRB into the ICA module, which combines local information and global information. Firstly, we use the global average pooling (GAP) operation to transform it into a one-dimensional vector. Subsequently, it undergoes a mapping through the fully connected layer to generate attention weights. Finally, the weighted matrix is obtained by performing a dot product operation with the input features. The processing steps of ICA are defined by the following equations:

$$g_i = \mathrm{GAP}(f_i^3) \tag{7}$$

$$g_i^2 = \mathrm{fc}(g_i) \tag{8}$$

$$g_i^3 = f_i^3 \times \mathrm{sigmoid}(\mathrm{relu}(g_i^2)) \tag{9}$$

Finally, we partition the feature map along the channel dimension and then add it to the feature map from the original branch to continue training. It is worth noting that reshaping is necessary when incorporating features from the Transformer branch.

In summary, our FFM module effectively integrates channel attention and self-attention mechanisms, incorporating information from both branches. This enables the model to better capture correlations between features.

2.4. Experimental Enviroment and Parameter Settings

Equipment

The devices and parameters used in the experiment are as follows, as shown in Table 2. The base operating system is Ubuntu 20.04.4, with an NVIDIA GeForce RTX 3090 graphics card. The CPU utilized is the Intel(R) Xeon(R) Silver 4214R CPU @ 2.40GHz. The Python 3.9.16 compiler was employed for training, with the PyTorch 2.0.0 framework for deep learning.

Table 2. The parameters of training environments.

Experimental Tool	Specific Model
CPU	Intel(R) Xeon(R) Gold 6330
GPU	NVIDIA GeForce RTX 3090
Operating System	Ubuntu 20.04.4
Programing Language	Python 3.9.16
Deep Learning Framework	Pytorch 2.0.0

Table 3 presents the hyperparameters used in the experiments. In this experiment, we split the dataset into a training set and a test set in an 8:2 ratio. The training set consists of 15,811 samples, while the test set contains 3892 samples. Following the requirements for training and testing the network, we preprocessed the images by resizing them to 224 × 224. To enhance the training speed and stability of the model, we performed standardization on the images in the dataset.

During training, we utilized the cross-entropy loss function. The AdamW optimizer was employed with a batch size of 128, an initial learning rate of 1×10^{-4}, and a cosine annealing algorithm to control the learning rate decay. The weight decay was set to 0.05, and the minimum learning rate was set to 1×10^{-6}. The total number of training epochs was 100.

Table 3. Training environment parameter configuration.

Parameter	Value
Epochs	100
Batch Size	128
Optimizer	AdamW
Weight deacy	0.05
Initial learning rate	0.0001
Minimum learning rate	0.000001
Scheduler	CosineAnnealingLR
Loss function	The cross-entropy loss function

3. Experiment and Results

3.1. Experimental Evaluation Indices

In this experimental study, Accuracy serves as the primary evaluation metric for apple disease classification. To further analyze the model's performance, we introduced F1 score, Precision, Recall, and model size as metrics for evaluating the apple disease identification model. The calculation methods for these metrics are shown in Equations (10)–(13)·

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} \tag{10}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{11}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{12}$$

$$\text{F1-Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{13}$$

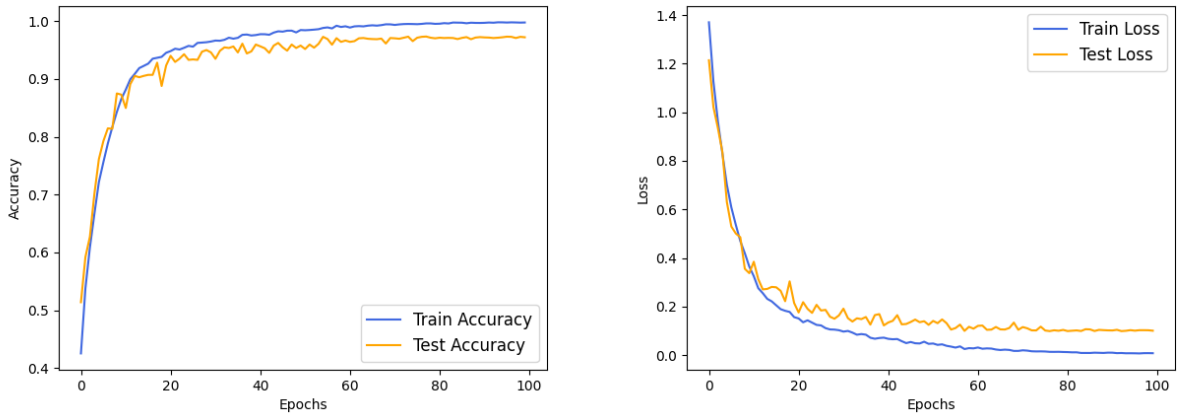
When using these metrics, the following parameters are employed: true positives (TP), which represents the number of instances where the model correctly classified leaves as healthy. True negatives (TN) represents the number of instances where the model correctly classified leaves as diseased. False positives (FP) represents the number of instances where the model incorrectly classified healthy leaves as diseased. False negatives (FN) represents the number of instances where the model incorrectly classified diseased leaves as healthy.

3.2. Experiments on the Effectiveness of DBCoST

To assess the effectiveness of our proposed model, we conducted an analysis of the accuracy and loss of DBCoST. The dataset was divided with an 8:2 ratio for training and validation. As depicted in Figure 7, the results after 100 training epochs indicate that our model initiated convergence around the 20th epoch and stabilized by the 100th epoch, achieving a final accuracy of 97.32%.

Additionally, we conducted a comparative analysis between our model and the baseline, which comprises ResNet18 and Swin Tiny models. Table 4 illustrates the disparities in various evaluation metrics between our model and the baseline. For ResNet18, the baseline achieved accuracy, precision, recall, and F1 score values of 93.21%, 93.54%, 93.21%, and 93.33%, respectively. On the other hand, Swin Tiny attained values of 94.78%, 95.08%, 94.75%, and 94.84%. In comparison, our model exhibited superior performance, reaching values of 97.32%, 97.33%, 97.40%, and 97.36%, indicating a significant improvement.

Table 5 presents a comprehensive performance analysis for each model across various disease types. ResNet18 achieved accuracies ranging from 88.15% to 98.10%, while Swin Tiny exhibited a range of 91.58% to 98.75%. In contrast, our model achieved a minimum accuracy of 96.26% and a maximum of 99.25%, with a fluctuation range of 3%, indicating that our model provides more stable recognition performance for specific disease types.



(**a**) (**b**)

Figure 7. The accuracy and loss curves of the DBCoST model. (**a**) Accuracy curve; (**b**) loss curve.

Table 4. Comparision with the baseline model.

Model	Accuracy	Precision	Recall	F1 Score	Parameters
ResNet18	93.75%	94%	93.74%	93.84%	19.9 M
Swin Tiny	95.91%	96.08%	95.92%	95.98%	28 M
Ours	97.32%	97.33%	97.40%	97.36%	45 M

In summary, our model is better suited for the detection of apple diseases in complex environments.

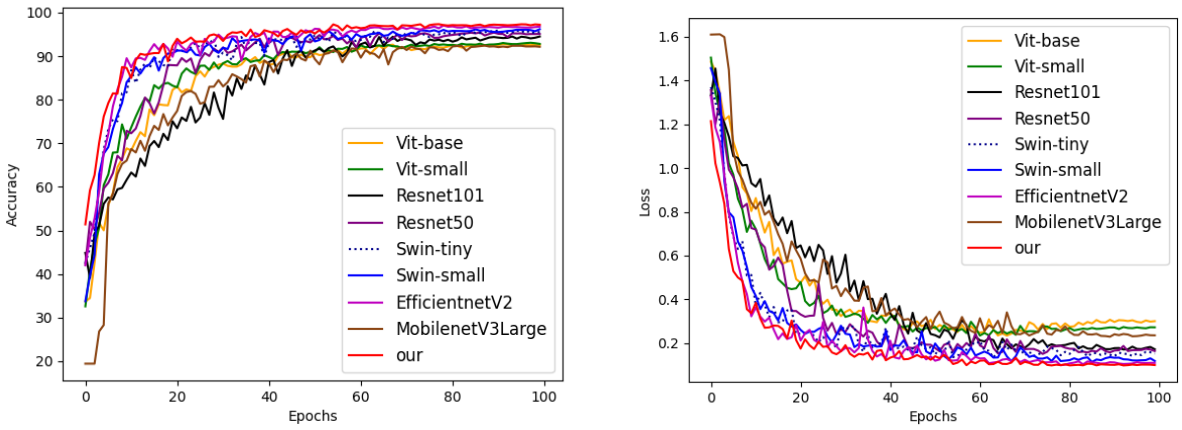
Table 5. Comparison with the baseline model across different disease types.

Model	Frog Eye Spot	Powdery Mildew	Healthy	Scab	Rust
ResNet18	95.21%	95.69%	92.83%	88.15%	98.10%
Swin Tiny	97.71%	97.01%	95.34%	91.58%	98.75%
Ours	96.74%	97.77%	96.26%	96.65%	99.25%

3.3. Comparison with State-of-the-Art Methods

To validate the effectiveness of our proposed model for apple disease classification, we conducted a comparative analysis with various CNN and Transformer models, including EfficientNet V2S, ResNet, MobileNetV3L, Vision Transformer, and Swin Transformer. Figure 8 illustrates the accuracy and loss values obtained by these models in our experiments. Our model achieved the highest metrics for accuracy, precision, recall, and F1 score, scoring 97.32%, 97.33%, 97.40%, and 97.37%, respectively. These results indicate that our model outperforms other models in recognizing apple disease leaf patterns, especially in complex environments.

As shown in the Table 6, EfficientNetV2-Small reached an accuracy of 97.04%, ranking second only to our proposed model. MobileNetV3L exhibited the lowest accuracy, with a value of 92.42%. Among the Transformer-based models, Swin Transformer outperformed Vit Transformer in terms of accuracy. In the CNN-based models, ResNet101 demonstrated lower accuracy compared to ResNet50. With similar parameter counts, our model achieved a higher accuracy compared to Swin Small and ResNet101 by 1.14% and 2.82%, respectively.



(**a**) (**b**)

Figure 8. The comparison of accuracy and loss curves among different models. (**a**) Accuracy curve; (**b**) Loss curve.

Table 6. Comparision with other state-of-the-art methods.

Model	Accuracy	Precision	Recall	F1 Score	Parameters
Vit Small	93.06%	93.07%	93.01%	93.03%	22 M
Vit Base	92.75%	92.72%	92.76%	92.71%	86 M
Swin Tiny	95.91%	96.08%	95.92%	95.98%	28 M
Swin Small	96.17%	96.23%	96.19%	96.20%	48 M
Resnet50	95.42%	95.69%	95.42%	95.51%	25.6 M
Resnet101	94.50%	94.65%	94.62%	94.62%	44.5 M
MobilenetV3L	92.42%	92.88%	92.49%	92.64%	5.5 M
EfficientnetV2S	97.04%	97.09%	97.06%	97.07%	21 M
Ours	97.32%	97.33%	97.40%	97.37%	45 M

3.4. Model Performance on Different Disease Categories

To evaluate the model's performance in recognizing specific disease types, we conducted experiments, and the results are outlined in Table 7. Our proposed model achieved precision rates of 96.74%, 97.77%, 96.26%, 96.65%, and 99.25% for each disease type, with particularly notable values for healthy and scab. Notably, the model also attained a 99.25% accuracy for rust, marginally below Swin Small's 99.36%.

ResNet, a classical CNN model, utilized residual connections to mitigate the gradient vanishing problem inherent in deep models. While ResNet50 achieved precision rates of 97.25%, 98.07%, 95.46%, 89.31%, and 98.37%, the deeper variant, ResNet101, did not surpass these metrics, yielding precision rates of 94.26%, 97.96%, 93.25%, 89.04%, and 98.48%.

MobileNetV3L, characterized by depth-wise separable convolutions and designed as a lightweight model with fewer parameters, exhibited relatively lower accuracy, with rates of 94.75%, 97.57%, 88.75%, 85.62%, and 97.72%.

EfficientNetV2S, incorporating max-pooling and dropout for feature map random dropout, achieved competitive accuracies of 96.89%, 98.39%, 96.25%, 94.83%, and 99.12%, notably demonstrating 98.39% accuracy for powdery mildew.

ViT, leveraging self-attention mechanisms to process image features, performed well on powdery mildew and rust but exhibited lower accuracy for other types. Specifically, ViT Small achieved precision rates of 89.43%, 97.36%, 89.94%, 89.87%, and 98.73%, while ViT Base achieved rates of 86.21%, 97.77%, 89.13%, 91.65%, and 98.84%.

Swin Transformer, employing window-based attention mechanisms for global information capture, demonstrated robust accuracy across various disease types. Swin Tiny

achieved precision rates of 97.71%, 97.01%, 95.34%, 91.85%, and 98.75%, while Swin Small achieved rates of 96.03%, 96.99%, 95.83%, 92.92%, and 99.36%.

Table 7. Comparison with other state-of-the-art methods on different disease types.

Model	Frog Eye Spot	Powdery Mildew	Healthy	Scab	Rust
Vit Small	89.43%	97.36%	89.94%	89.87%	98.73%
Vit Base	86.21%	97.77%	89.13%	91.65%	98.84%
Swin Tiny	97.71%	97.01%	95.34 %	91.85%	98.75%
Swin Small	96.03%	96.99%	95.83%	92.92%	99.36%
Resnet50	97.25%	98.07%	95.46%	89.31%	98.37%
Resnet101	94.26%	97.96%	93.25%	89.04%	98.48%
MobilenetV3L	94.75%	97.57%	88.75%	85.62%	97.72%
EfficientnetV2S	96.89%	98.39%	96.25%	94.83%	99.12%
Ours	96.74%	97.77%	96.26%	96.65%	99.25%

It is crucial to note that the images in our dataset were captured in natural environments, inevitably containing some noise. Additionally, different diseases exhibit distinct characteristics, and the sensitivity of various models to lesion areas may differ. Consequently, variations in the accuracy of recognizing different diseases could arise. In the dataset we employed, lesions caused by frog eye spot are relatively small and scattered on the leaf surface. The frog-eye-like texture makes models such as CNN, which excel at handling local features, more sensitive to these characteristics. On the other hand, rust disease lesions have a larger area, and the vibrant color of the infection results in both CNN and Transformer models achieving high accuracy.

Powdery mildew typically covers the leaf surface, presenting a powdery appearance. This distinctive feature enables both CNN and Transformer models to achieve good results in distinguishing infected leaves from healthy ones. Regarding scab, its early symptoms are not prominent and are distributed radially or in band-like patterns on the leaf surface. Transformer models, adept at capturing global information, generally outperform CNN-based models in this scenario. In summary, in complex environments, these models have not proven to be the optimal choice. Our proposed model outperforms others across all five diseases, indicating its effectiveness in recognizing apple disease leaves. The visualization of the model output results was conducted using ScoreCam [25], as depicted in Figure 9. The portion inside the red area in the figure represents the location of the disease. The figure illustrates the attention levels of different models, highlighting that other models overly focus on background images. In contrast, our model prioritizes most of the disease areas, disregarding irrelevant complex backgrounds, leading to higher recognition accuracy.

3.5. Ablation Experiment

In this paper, we introduce the dual-branch model and FFM module. To assess their effectiveness, we implemented several modifications to the model. We employed accuracy, precision, recall, and F1 score as evaluation metrics, and the dataset was partitioned into a training set and a testing set in an 8:2 ratio. Table 8 provides the performance metrics for our five schemes. Schemes 1 and 2 utilized the original ResNet18 and Swin Tiny architectures, while Schemes 3–4 incorporated the dual-branch structure with the exclusion of ICA and CBR modules. Scheme 5 represents our comprehensive model. Figure 10 depicts the accuracy and loss values of different models on the testing and training sets.

In Schemes 1 and 2, utilizing only ResNet18 and Swin Tiny, Scheme 1 achieved accuracy, precision, recall, and F1 score values of 93.75%, 94%, 93.74%, and 93.84%, respectively. Scheme 2 achieved accuracy, precision, recall, and F1 score of 95.91%, 96.08%, 95.92%, and 95.98%, respectively.

Scheme 3 introduced the ICA attention mechanism and removed the CRB module on top of the dual-branch model, exhibiting significant improvements over Scheme 2 in

accuracy, precision, recall, and F1 score by 1.65%, 1.59%, 1.63%, and 1.63%, respectively. This indicates a notable enhancement in performance due to the introduced attention mechanism.

In Scheme 4, the dual-branch model was combined with the CRB module, and the ICA attention module was removed, resulting in further performance improvements. Compared to Scheme 3, accuracy, precision, recall, and F1 score increased by 0.79%, 0.7%, 0.88%, and 0.79%, respectively.

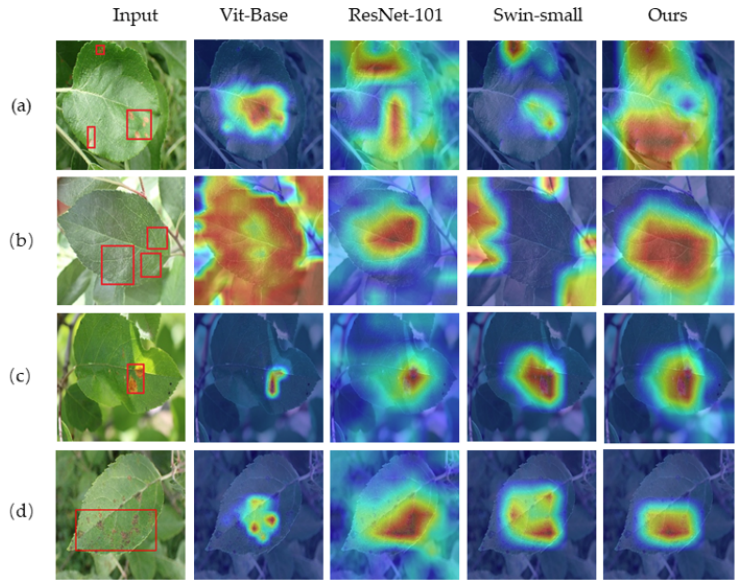
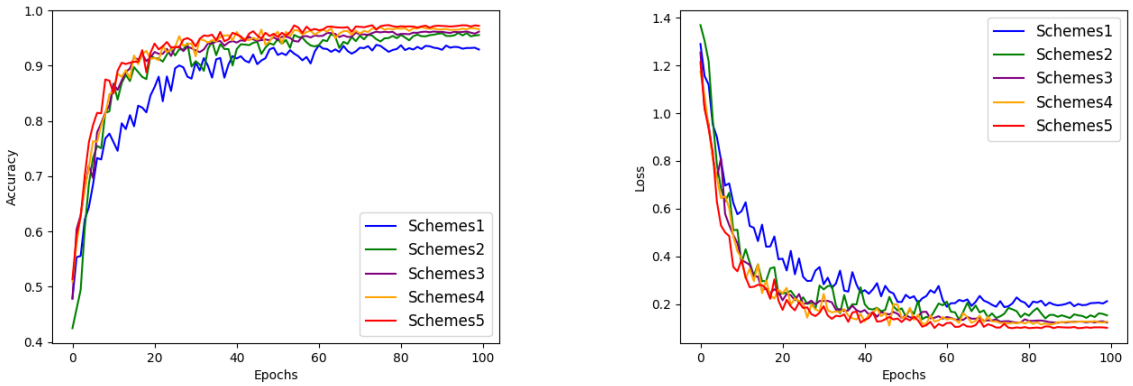


Figure 9. Visualized results of different models in recognizing apple leaf diseases. The red box indicates the presence of the diseased area (**a**) Frog eye spot; (**b**) powdery mildew; (**c**) rust; (**d**) scab.

Table 8. Results of ablation experiment.

Schemes	FFM		ResNet	Transformer	Accuracy	Precision	Recall	F1 Score
	Attention	ResBlock						
Scheme 1	-	-	✓	-	93.75%	94%	93.74%	93.84%
Scheme 2	-	-	-	✓	95.91%	96.08%	95.92%	95.98%
Scheme 3	ICA	-	✓	✓	96.17%	96.27%	96.16%	96.21%
Scheme 4	-	CBR	✓	✓	96.96%	96.97%	97.04%	97%
Scheme 5	ICA	CBR	✓	✓	97.32%	97.33%	97.40%	97.36%

- indicates removal of this module; ✓ indicates addition of this module.



(**a**) (**b**)

Figure 10. The comparison of accuracy and loss curves of different schemes. (**a**) Accuracy curve; (**b**) Loss curve.

Finally, in Scheme 6, incorporating the dual-branch structure and FFM module, the model achieved the maximum values across all metrics: 97.32% accuracy, 97.33% precision, 97.40% recall, and 97.36% F1 score. This indicates the effectiveness of the proposed FFM module in enhancing the overall model performance.

3.6. Experiments on the Effectiveness of Each FFM

In this paper, we introduced the feature fusion module (FFM) to integrate local and global information. While the preceding ablation experiments verified the impact of the two branches of DBCoST and various components of FFM on the model, a comprehensive validation of the influence of each FFM module on the model is still lacking. In this section, we adopted the following method to assess the effectiveness of the FFM, as illustrated in Figure 3. From left to right, there are four modules. Initially, all the modules were removed. Subsequently, each time, one module was retained, while the other three were excluded, allowing us to assess their respective impacts on the model's performance. The results are presented in Table 9.

The experiments indicate a notable improvement in model accuracy compared to the removal of all modules. Specifically, retaining the first FFM module resulted in accuracy, precision, recall, and F1 score of 96.12%, 96.08%, 96.16%, and 96.10%, respectively, representing improvements of 1.87%, 1.4%, 1.63%, and 1.52%. Retaining the third module led to peak performance, with accuracy, precision, recall, and F1 score reaching 97.04%, 97.11%, 97.02%, and 97.06%, respectively, showcasing improvements of 2.79%, 2.43%, 2.49%, and 2.48%. However, retaining the second and fourth modules resulted in comparable accuracies.

The primary function of the FFM module is to fuse features from the CNN and Transformer branches, followed by channel-wise splitting and addition to the original branch for further training. Therefore, we conjecture that the relatively lower performance when retaining the second module may stem from insufficient information. Conversely, retaining the fourth module could result in information redundancy as it lacks the preceding modules' further feature extraction. This leads to similar accuracy between the second and fourth modules.

Table 9. FFM module ablation study.

Strategy	Accuracy	Precision	Recall	F1 Score
Remove all	94.25%	94.68%	94.53%	94.58%
Module1	96.12%	96.08%	96.16%	96.10%
Module2	96.76%	96.87%	96.72%	96.79%
Module3	97.04%	97.11%	97.02%	97.06%
Module4	96.78%	96.78%	96.85%	96.81%

3.7. Generalization Performance of DBCoST

The results presented in Table 6 indicate that EfficientNetV2S performs similarly to our model. To further validate the generalization performance of DBCoST, we conducted extended experiments on another dataset named AppleLeaf9, obtained from Ref. [26]. This dataset comprises 417 images of Alternaria leaf spot, 411 images of brown spot, 339 images of gray spot, 371 images of mosaic, 3181 images of frog eye spot, 2757 images of rust, 516 images of health, 5410 images of scab, and 1184 images of powdery mildew, covering a total of nine disease types. Due to partial overlap with the previously used dataset, we selected four disease types, namely Alternaria leaf spot, brown spot, gray spot, and mosaic. Additionally, we introduced 120 images of mixed disease (rust and frog eye spot) and 636 images of mixed disease (scab and frog eye spot) from the Plant Pathology 2021 dataset [22] to enhance the complexity of our dataset. Similarly, to prevent overfitting, we employed data augmentation techniques such as Gaussian noise, salt-and-pepper noise, mirroring, and rotation, thereby expanding the dataset size. The experimental results are presented in Tables 10 and 11.

Table 10. Comparision using AppleLeaf9.

Model	Accuracy	Precision	Recall	F1 Score	Parameters
Vit Small	95.34%	94.75%	92.97%	93.65%	22 M
Vit Base	94.09%	92.87%	91.5%	92.04%	86 M
Swin Tiny	95.98%	95.8%	93.56%	94.38%	28M
Swin Small	95.89%	95.4%	93.53%	94.24%	48 M
Resnet50	94.18%	93.70%	91.41%	92.22%	25.6 M
Resnet101	92.15%	91.03%	88.72%	89.44%	44.5 M
MobilenetV3L	89.02%	88.69%	83.87%	84.49%	5.5 M
EfficientnetV2S	96.07%	95.54%	93.93%	94.57%	21 M
Ours	98.06%	97.67%	97.56%	97.61%	45 M

In summary, compared to the second-ranked EfficientnetV2S model, our model demonstrates improvements of 1.99%, 2.13%, 3.63%, and 3.04% in accuracy, precision, recall, and F1 score, respectively. Notably, our model significantly outperforms others in the recall metric, indicating increased stability in disease classification. Across various disease types, our model attains the highest accuracy in Alternaria leaf spot, mosaic, rust frog eye spot, and scab frog eye spot. Notably, in the mixed disease types of rust frog eye spot and scab frog eye spot, our model exhibits significantly higher accuracy than other models. We conjecture that this difference may stem from the necessity to integrate both local and global information in identifying mixed disease types. Models based on a singular CNN or Transformer structure excel in handling either local or global information. In contrast, our model integrates both global information and local features, resulting in higher precision.

Table 11. Comparing the accuracy of different disease types on AppleLeaf9.

Model	Alternaria Leaf Spot	Brown Spot	Gray Spot	Mosaic	Rust Frog Eye Spot	Scab Frog Eye Spot
Vit Small	97.15%	99.5%	95.8%	99.25%	88.81%	87.98%
Vit Base	94.01%	99.75%	96.22%	98.99%	81.88%	86.37%
Swin Tiny	97.95%	99.97%	97.49 %	98.52%	92.97%	87.89%
Swin Small	97.45%	99.98%	97.26%	99.26%	90.23%	88.18%
Resnet50	94.44%	99.5%	94.28%	99%	88.71%	86.32%
Resnet101	93.09%	98.28%	92.68%	97.29%	81.2%	83.68%
MobilenetV3L	87.12%	99%	87.13%	94.66%	84.51%	79.71%
EfficientnetV2S	98.47%	99.97%	97.74%	99.01%	89.86%	88.15%
Ours	98.96%	99.75%	97.3%	99.5%	93.98%	96.55%

4. Discussion

This paper introduces a dual-branch model that integrates CNN and Swin Transformer for the classification of apple leaf diseases in complex environments. The CNN and Swin Transformer branches in the model are employed to extract local features and capture global information, respectively. Feature fusion modules are incorporated to fuse these two types of information, enabling the model to capture more comprehensive information. In terms of experimental results, compared to other models, our approach demonstrates notable performance in complex environments, showing significant accuracy and robustness in disease identification.

Currently, some studies on plant disease classification commonly utilize either a single CNN or Transformer architecture. Due to the distinct focus of these models, certain crucial features may be overlooked when dealing with different types of diseases. To address this issue, Ref. [27] investigated the complementarity between CNN and Transformer in image classification, introducing the convolutional Swin Transformer (CST) model for detecting different plant diseases. It achieved an accuracy of over 92.2% on various plant disease datasets. However, as the selected datasets were mainly from laboratory environments, the accuracy showed some fluctuation. Ref. [28] introduced ConViT, a lightweight model for recognizing apple leaf diseases based on Vision Transformer. By combining the strengths of

both convolutional and Transformer structures, the model achieved an impressive accuracy of 96.85% in identifying apple diseases in complex environments. Due to its lightweight architecture, there was significant variability in accuracy among different disease types. These research findings indicate that integrating CNN and Transformer structures resulted in varying degrees of improvement in accuracy.

In our study, we also explored the complementarity between CNN and Transformer models, introducing a dual-branch structure composed of CNN and Transformer branches along with multiple feature fusion modules. Ablation experiments were conducted to assess their impact on the foundational model, as depicted in the table. The results indicate that, compared to using only ResNet or Swin Transformer single-branch structures and a dual-branch structure without the added feature fusion module, our model achieved respective improvements of 3.57%, 1.41%, and 3.07% in accuracy. Furthermore, Table 9 provides additional validation of the influence of integrating CNN and Transformer information on the model. Fusing information from any layer significantly enhanced the model's performance compared to strategies without fusion.

Some studies have incorporated attention mechanisms, proven effective in enhancing model performance. In Ref. [29], researchers combined attention mechanisms such as CBAM [30] and SE with a lightweight CNN model for tomato disease recognition, achieving an exceptional accuracy of 99.69%. In Ref. [31], Coordinate Attention was incorporated into the EfficientNet-B4 network, integrating spatial and channel information to facilitate learning important features from both types of information. In this study, the accuracy rate for identifying apple leaf diseases reached 98.92%. These research findings indicate that incorporating attention mechanisms can result in different levels of improvement in model accuracy. In our research, we observed an increase in redundancy due to the growing complexity of images and the deepening of channels. Therefore, in our feature fusion module, we introduced the ICA module to enable the model to focus on key features and reduce the impact of redundancy. Ablation experiments demonstrated significant improvements in all metrics (accuracy, precision, recall, and F1 score) compared to scenarios without the ICA module, with increases of 1.65%, 1.59%, 1.63%, and 1.63%, respectively. These experimental results clearly demonstrate that our model has a distinct advantage in classifying apple leaf diseases.

5. Conclusions

The DBCoST model proposed in this paper demonstrates better performance in recognizing apple leaf diseases under natural environments, with stable accuracy across all disease types.

Our model utilizes CNN for local feature extraction and Swin Transformer for capturing global information and establishing long-range dependencies. During the input phase, local features are extracted by the CNN branch, while global information and long-range dependencies are extracted by the Transformer branch. These features are then further extracted through the feature fusion module, combined with the results of the original branch, and subsequently input back into the original branch for further learning.

First, we validated the effectiveness of the model proposed in this paper. We initially trained the model on the apple leaf disease dataset for 100 epochs. The results showed that our model began to converge after approximately 20 epochs and stabilized around 100 epochs. We then conducted a comparative analysis with two baseline models, ResNet18 and Swin Transformer Tiny, which constitute the two branches of our model. The results indicated that our model outperformed the selected baseline models across four key metrics: accuracy, precision, recall, and F1 score.

Secondly, we compared the performance of the model proposed in this paper with select CNN- and Transformer-based models. Our experiments showed that our proposed model attained the highest values in accuracy, precision, F1 score, and recall, indicating its strong performance compared to select CNN- and Transformer-based network models.

Thirdly, we assessed the accuracy of our model and other models across various diseases and conducted generalization tests on additional datasets. The results revealed that Swin Tiny achieved the highest accuracy of 97.71% for frog eye spot disease. For powdery mildew disease, EfficientnetV2S achieved the highest accuracy at 98.39%. Our model demonstrated the best accuracies on healthy and scab diseases, reaching 96.26% and 96.65%, respectively. Compared to the second-highest EfficientnetV2S, our model exhibited a significant improvement of 1.82% on scab disease, which is particularly crucial as it impacts both leaves and fruits. Regarding rust disease, Swin Small outperformed with a high accuracy of 99.36%, while our model slightly lagged behind at 99.25%. On the second dataset, our model achieved optimal performance and excelled in handling mixed diseases. Overall, our model consistently maintained accuracy levels between 93% and 99% across all diseases, underscoring the stability of our model in identifying apple leaf diseases.

Finally, we conducted both ablation experiments and efficacy experiments on the FFM module using the DBCoST network. The results demonstrated that each component within our proposed methodology contributed to the model's performance, with a particular emphasis on FFM. Following the integration of feature maps from both the CNN and Transformer branches, the model exhibited significant improvements in accuracy, precision, recall, and F1 score across these four evaluation metrics.

In summary, the combination of CNN and Transformer has effectively enhanced the model's performance, increasing its robustness and stability. We propose an approach for diagnosing apple leaf diseases based on a dual-branch model of CNN and Transformer. Our experimental results indicate that, compared to traditional convolutional neural networks and Transformer-based models, our model exhibits more stable recognition across various diseases. However, challenges arise from the high computational complexity of the Transformer branch and the significant computational load of the feature fusion module, posing obstacles to portability to mobile devices. In the future, we plan to employ lightweight techniques, considering the use of depthwise separable convolutions or adjusting the model's depth to reduce model parameters. These strategies aim to facilitate deployment on mobile platforms while maintaining effective disease diagnosis.

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

Funding: This research is funded by the Henan Province Key Research and Development Special Project, with the project titled 'Research on the common key technology of creating new germplasm resources based on artificial intelligence'. The grant Nos. are 231111211300 and 231111110100. The Henan Provincial Programs for Science and Technology Development also provide support for the project titled 'Research on the organization methodology of germplasm resources mass data based on multiple attributes', with grant No. 232102520006.

Institutional Review Board Statement: Not applicable.

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

Data Availability Statement: Data are contained within the article.

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

References

1. Patriarca, A. Fungi and mycotoxin problems in the apple industry. *Curr. Opin. Food Sci.* **2019**, *29*, 42–47. [Crossref] [CrossRef]
2. Akshay, S.; Shetty, D. Categorization of fruit images using artificial bee colony algorithm based on glcm features. In Proceedings of the 2022 International Conference on Electronic Systems and Intelligent Computing (ICESIC), Chennai, India, 22–23 April 2022; pp. 46–51. [Crossref]
3. Alqethami, S.; Almtanni, B.; Alzhrani, W.; Alghamdi, M. Disease detection in apple leaves using image processing techniques. *Eng. Technol. Appl. Sci. Res.* **2022**, *12*, 8335–8341. [Crossref] [CrossRef]

4. Huang, Y.; Zhang, J.; Zhang, J.; Yuan, L.; Zhou, X.; Xu, X.; Yang, G. Forecasting alternaria leaf spot in apple with spatial-temporal meteorological and mobile internet-based disease survey data. *Agronomy* **2022**, *12*, 679. [Crossref] [CrossRef]
5. Hasan, S.; Jahan, S.; Islam, M.I. Disease detection of apple leaf with combination of color segmentation and modified dwt. *J. King Saud Univ.-Comput. Inf. Sci.* **2022**, *34*, 7212–7224. [Crossref] [CrossRef]
6. Jose, V.D.; Santhi, K. Early detection and classification of apple leaf diseases by utilizing ifpa genetic algorithm with mc-svm, svi and deep learning methods. *Indian J. Sci. Technol.* **2022**, *15*, 1440–1450. [Crossref] [CrossRef]
7. Xing, B.; Wang, D.; Yin, T. The evaluation of the grade of leaf disease in apple trees based on pca-logistic regression analysis. *Forests* **2023**, *14*, 1290. [Crossref] [CrossRef]
8. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. In Proceedings of the Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012, Lake Tahoe, NV, USA, 3–6 December 2012. [Crossref]
9. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556. [Crossref]
10. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778. [Crossref]
11. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv* **2017**, arXiv:1704.04861. [Crossref]
12. Huang, G.; Liu, Z.; Maaten, L.V.D.; Weinberger, K.Q. Densely connected convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4700–4708. [Crossref]
13. Yan, Q.; Yang, B.; Wang, W.; Wang, B.; Chen, P.; Zhang, J. Apple leaf diseases recognition based on an improved convolutional neural network. *Sensors* **2020**, *20*, 3535. [Crossref] [CrossRef] [PubMed]
14. Yu, H.; Cheng, X.; Chen, C.; Heidari, A.A.; Liu, J.; Cai, Z.; Chen, H. Apple leaf disease recognition method with improved residual network. *Multimed. Tools Appl.* **2022**, *81*, 7759–7782. [Crossref] [CrossRef]
15. Luo, Y.; Sun, J.; Shen, J.; Wu, X.; Wang, L.; Zhu, W. Apple leaf disease recognition and sub-class categorization based on improved multi-scale feature fusion network. *IEEE Access* **2021**, *9*, 95517–95527. [Crossref] [CrossRef]
16. Fu, L.; Li, S.; Sun, Y.; Mu, Y.; Hu, T.; Gong, H. Lightweight-convolutional neural network for apple leaf disease identification. *Front. Plant Sci.* **2022**, *13*, 831219. [Crossref] [CrossRef] [PubMed]
17. Yu, H.; Cheng, X.; Li, Z.; Cai, Q.; Bi, C. Disease recognition of apple leaf using lightweight multi-scale network with ecanet. *CMES-Comput. Model. Eng. Sci.* **2022**, *132*, 711–738. [CrossRef]
18. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł. ; Polosukhin, I. Attention is all you need. In Proceedings of the Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, Long Beach, CA, USA, 4–9 December 2017. [Crossref]
19. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An image is worth 16 $\times$ 16 words: Transformers for image recognition at scale. *arXiv* **2020**, arXiv:2010.11929. [Crossref]
20. Wang, W.; Xie, E.; Li, X.; Fan, D.; Song, K.; Liang, D.; Lu, T.; Luo, P.; Shao, L. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 568–578. [Crossref]
21. Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; Guo, B. Swin transformer: Hierarchical vision transformer using shifted windows. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 10012–10022. [Crossref]
22. Thapa, R.; Zhang, K.; Snavely, N.; Belongie, S.; Khan, A. The plant pathology challenge 2020 data set to classify foliar disease of apples. *Appl. Plant Sci.* **2020**, *8*, e11390. [Crossref] [CrossRef] [PubMed]
23. Wang, Q.; Wu, B.; Zhu, P.; Li, P.; Zuo, W.; Hu, Q. Eca-net: Efficient channel attention for deep convolutional neural networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 11534–11542. [Crossref]
24. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7132–7141. [Crossref]
25. Wang, H.; Wang, Z.; Du, M.; Yang, F.; Zhang, Z.; Ding, S.; Mardziel, P.; Hu, X. Score-cam: Score-weighted visual explanations for convolutional neural networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Seattle, WA, USA, 14–19 June 2020; pp. 24–25. [Crossref]
26. Yang, Q.; Duan, S.; Wang, L. Efficient identification of apple leaf diseases in the wild using convolutional neural networks. *Agronomy* **2022**, *12*, 2784. [Crossref] [CrossRef]
27. Li, X.; Li, S. Transformer help cnn see better: A lightweight hybrid apple disease identification model based on transformers. *Agriculture* **2022**, *12*, 884. [Crossref] [CrossRef]
28. Guo, Y.; Lan, Y.; Chen, X. Cst: Convolutional swin transformer for detecting the degree and types of plant diseases. *Comput. Electron. Agric.* **2022**, *202*, 107407. [Crossref] [CrossRef]
29. Bhujel, A.; Kim, N.; Arulmozhi, E.; Basak, J.K.; Kim, H. A lightweight attention-based convolutional neural networks for tomato leaf disease classification. *Agriculture* **2022**, *12*, 228. [Crossref] [CrossRef]

30. Woo, S.; Park, J.; Lee, J.; Kweon, I.S. Cbam: Convolutional block attention module. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 3–19. [Crossref]
31. Wang, P.; Niu, T.; Mao, Y.; Zhang, Z.; Liu, B.; He, D. Identification of apple leaf diseases by improved deep convolutional neural networks with an attention mechanism. *Front. Plant Sci.* **2021**, *12*, 723294. [Crossref] [CrossRef] [PubMed]