



2025 International Conference on Intelligent Computing

July 26-29, Ningbo, China

<https://www.ic-icc.cn/2025/index.php>

## PRMT: Retentive Networks Meet Vision Transformers for Plant Disease Identification

Jialong Guo<sup>1</sup>, Baofang Chang<sup>1</sup> and Guoqiang Li<sup>2</sup>(✉)

<sup>1</sup> School of Computer and Information Engineering, Henan Normal University, Xinxiang  
453007, China

<sup>2</sup> Institute of Agriculture Economics and Information, Henan Academy of Agricultural Sciences,  
Zhengzhou 450002, China  
[gqli@hnagri.org.cn](mailto:gqli@hnagri.org.cn)

**Abstract.** Accurate and rapid classification of plant diseases is crucial for enhancing productivity in contemporary agriculture. Both modern deep-learning models and conventional methods encounter obstacles when it comes to finding plant diseases. For instance, complicated scenarios often increase processing costs and reduce recognition accuracy. This study introduces the PRMT (Retentive Networks Meet Vision Transformers for Plant Disease Identification) framework, utilizing the Retentive Networks Meet Vision Transformers (RMT) architecture. The method utilizes Manhattan distances and spatial prior knowledge to create a spatial attenuation matrix. It improves internal correlations and enables a greater understanding of the relationships among image regions. The design incorporates the Convolutional Block Attention Module (CBAM) to enhance feature representations. Incorporating 2D average pooling in the backbone network diminishes sensitivity to local noise and inhibits an increase in model parameters. We employed datasets on paddy, corn, wheat, and coffee diseases. To enhance the utilization of the datasets, we implemented rotation, scaling, and color modification and conducted three-fold cross-validation. We assessed the PRMT model is performance using recall, specificity, accuracy, and precision metrics and compared it with other models. Studies show that the PRMT model can easily handle big and complicated datasets of agricultural diseases, leading to much better results with only a few extra parameters. Our methodology improves the effectiveness of categorizing intricate plant disease images.

**Keywords:** Plant disease, Manhattan, Networks, Attention, Spatial prior-knowledge.

## 1.1 Introduction

### 1.2 A Subsection Sample

Severe plant losses, estimated at 20 to 40 percent in some countries, have arisen from agricultural illnesses and insect infestations, posing substantial challenges to global food supplies [1]. Traditional pest control methods are also losing their effectiveness, which calls for developing new, creative strategies, such as plant rotation and other agricultural practices to reduce pest infestations [2].

Precise identification of plant diseases ensures plant safeguarding and promotes sustainable farming methods. Traditional methods of classifying agricultural diseases demonstrate numerous shortcomings that impede their precision and efficacy. Conventional techniques for diagnosing agricultural diseases, including visual and tactile evaluations, frequently depend on subjective interpretations and are susceptible to inaccuracies [3]. Furthermore, these technologies require substantial human labor, which is becoming progressively inefficient and time-intensive, especially in managing large agricultural operations [4]. Traditional techniques can fail to detect infections in their first stages, resulting in postponed interventions and increased agricultural losses [5]. The demand for accurate, automated disease detection technologies that provide dependable real-time information is rising [6].

Recent advancements in deep learning, such as Vision Transformers (ViT) [7], Convolutional Neural Networks (CNNs) [8], and Artificial Neural Networks (ANNs) [9], have significantly transformed the detection and classification of agricultural diseases. These algorithms enhance classification accuracy by examining extensive visual data and independently extracting intricate features from plant photos, exceeding conventional methods. CNN and ViT models exhibit enhanced proficiency in feature extraction and acquiring spatial and positional data, improving the identification of subtle illness markers [10]. Deep learning techniques, especially Vision Transformers (ViTs) [11, 12], have significantly improved the handling of intricate datasets with diverse lighting and backdrop circumstances. These algorithms have proven their capacity to adjust to diverse agricultural situations and deliver faster, more accurate results than traditional manual classification methods [13, 14]. Deep learning models are indispensable in the current era because they can surpass previous limitations [15]. Convolutional Neural Networks (CNNs) excel at identifying plant diseases by recognizing patterns and anomalies in leaf surfaces, as they are proficient at discerning spatial hierarchies within images [16]. Conversely, ViTs effectively handle long-range dependencies in images, facilitating the identification of intricate disease patterns across extensive agricultural regions [17]. Advancements in deep learning methodologies facilitate early disease detection and prevention, reducing dependence on manual assessments while providing quicker and more precise results [18]. Implementing these algorithms has facilitated the development of reliable systems for various plants and illnesses, ensuring the applicability of precision agriculture across many scenarios and on a wide scale [19]. Nonetheless, these algorithms possess certain limitations, such as the possibility of substantial computing expenses and time investments, which constrain their scalability for large-scale agricultural applications [20]. Traditional deep-learning algorithms mainly focus on local features, hindering the identification of long-range

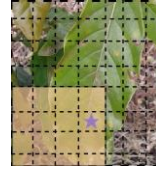
relationships in complex agrarian data. It may also impede the recognition of subtle disease patterns [5]. Researchers have developed sophisticated deep-learning algorithms, such as Vision Transformers (ViT) and Swin Transformers, to tackle these difficulties by integrating attention mechanisms. These developments allow models to focus on the most relevant picture areas, improving feature extraction and overall effectiveness [17]. The sophisticated algorithms enhance accuracy by concentrating on critical areas, reducing the necessity for large datasets, and increasing efficiency and adaptability in agricultural applications [21].

★ : Quarry

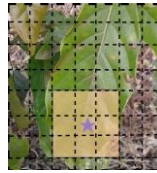
■ : Receptive Field



(a) Vanilla Self-Attention



(b) Window Self-Attention



(c) Neighborhood Self-Attention



(d) Manhattan Self-Attention

**Fig. 1.** A comparison is made between several self-attention methods. The color intensity in the Manhattan Self-Attention (MaSA) mechanism represents the spatial decay rate; lighter parts indicate more considerable spatial decay rates, while darker regions indicate lower spatial decay rates. This distance-related spatial decay rate provides the model with rich spatial prior information.

The attention mechanism and 2D average pooling make the modified RMT (Retentive Network for Vision Transformer) method better for classifying agricultural diseases. This method has many benefits. By adding explicit spatial priors to the Vision Transformer (ViT) [22], RMT overcomes the shortcomings of the conventional self-attention mechanism, which lacks spatial awareness and has a high computational cost. The attention mechanism improves feature selection even more by focusing on the most essential parts of the image. It makes it easier for the model to spot small plant disease patterns [20]. Adding 2D average pooling to the model also makes it easier to use with large datasets and speeds up computation by reducing the number of spatial dimensions

while keeping essential features [6]. This combination improves the model's robustness and increases disease detection accuracy, making it more useful in agricultural settings. In particular, this paper introduces several key innovations:

(1) The PRMT framework, a novel RMT architecture, integrates the Convolutional Block Attention Module (CBAM) with a 2D average pooling layer and an RMT network. It improves disease detection accuracy, reduces computational costs, and enhances generalization and classification performance.

(2) The addition of 2D average pooling minimizes sensitivity to local noise and increases model resilience with only a tiny parameter increase.

(3) A comprehensive analysis of multiple publicly available datasets shows that the proposed model is more effective at classifying plant diseases. It provides a scalable solution with low computational complexity and high detection accuracy.

## **2 Related Work**

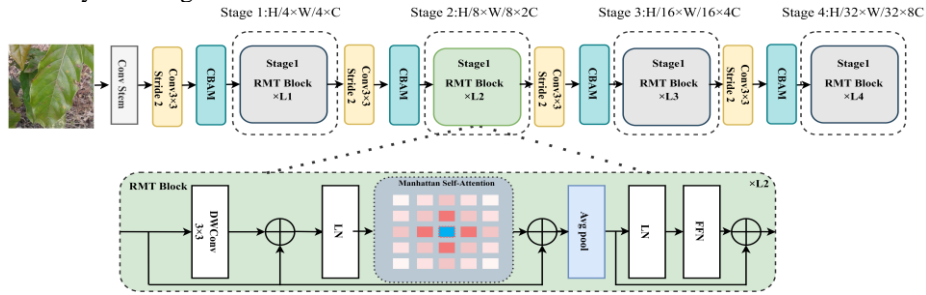
### **2.1 RMT Methods**

In recent years, studies have shown that the central part of the VIT model that deals with self-attention is challenging to compute, doesn't give clear spatial information ahead of time, and becomes a bottleneck when processing substantial volumes of image data. Researchers have conducted multiple investigations to address these restrictions. To solve this problem, the Swin Transformer separates the tokens used in self-attention calculations using spatial prior information, windowing, and relative position encoding. RMT uses the Manhattan Self-Attention mechanism (MaSA) (Figure 1). This network model keeps linear complexity to a minimum while fixing the problems with the VIT network and making up for the lack of prior information. The MaSA decomposition determines the shape of each marker's receptive field. As it matches the shape of the whole MaSA's receptive field, our decomposition method keeps the explicit spatial precedence. RMT creates a two-dimensional, bidirectional spatial decay matrix using the Manhattan distance between tokens. The RETNET framework expands its temporal decay matrix to include the spatial domain. In this spatial decay matrix, the attention score of the target token diminishes as its distance from adjacent tokens increases. It means that tokens at different distances get different amounts of attention, as shown in Figure 4.

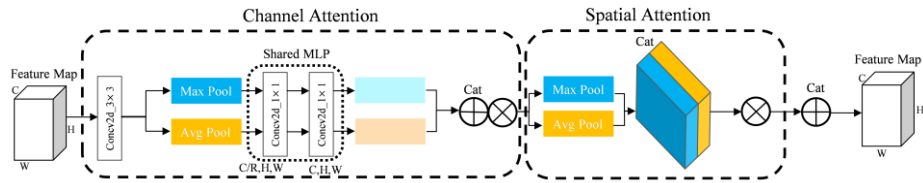
### **2.2 Attention-Based Methods**

Attention mechanisms have become crucial in deep learning models to enhance the model's capacity to concentrate on pertinent information. The channel attention mechanism prioritizes the channel dimension of the feature map and is among the most often employed attention mechanisms. This method evaluates the importance of each channel by analyzing global data from all spatial regions. The spatial attention mechanism is essential since it affects the spatial dimensions of the feature map. The spatial attention mechanism allows the model to concentrate on prominent image areas, including object boundaries or regions with considerable fluctuations. Hybrid attention mechanisms that

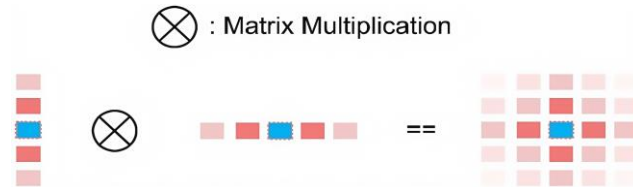
integrate discrete channel and spatial attention processes are also present. These strategies enhance your comprehension of incoming data by elucidating their interconnections in channel and geographical contexts. The Convolutional Block Attention Module (CBAM) is a composite attention mechanism comprising channel and spatial attention modules. Initially, the channel attention module performs average and max pooling operations on the feature map. Next, the channel attention module creates channel attention weights by passing these descriptors through a shared multilayer perceptron (MLP). The spatial attention module unites the averaged and max-pooled feature maps in two dimensions along the channel axis. It finds spatial attention weights. It employs a  $7 \times 7$  convolution kernel. Figure 3 shows that these steps improve the model's ability to tell the difference between things and make more accurate predictions by focusing on relevant information more.



**Fig. 2.** Overview of the PRMT model. In four stages, we implement a Manhattan distance-based Manhattan Self-Attention. Additionally, we integrate the channel and spatial attention modules to analyze plant disease images and improve the feature map effectively. The backbone network is subjected to 2D average pooling, enhancing reliability.



**Fig. 3.** The CBAM block's architecture, which  $\otimes$  represents the original feature map multiplied.



**Fig. 4.** Spatial decay matrix in the decomposed MaSA.

### 2.3 Pooling based methods

In Convolutional Neural Networks (CNNs), pooling is a crucial procedure that reduces the spatial dimensions of feature maps while preserving the most significant information. CNNs perform tasks by applying a 2D average pooling function to a small window of the input feature map, which computes the 2D average. The model gradually lowers the spatial resolution of the feature map by running pooling operations across multiple network layers. It makes the representation of the input data more concise and general. Besides reducing the computational complexity of the model, pooling provides a degree of translation invariance. Combining pooling with attention improves the model's representational capacity, enabling it to consolidate the most pertinent regions for the task preferentially. We specifically favor 2D average pooling for its simplicity and efficiency. It provides a continuous downsampled representation by averaging each two-dimensional window in the feature map. This research enhances the RMT model by including 2D average pooling. By implementing 2D average pooling in the backbone network, the model effectively handles the intricate spatial structure and multi-scale information in agricultural imaging, thereby enhancing plant disease classification tasks.

## 3 Method

### 3.1 Backbone Network

The RMT model's core is the creative enhancement of the self-attention mechanism, which serves as its backbone network. RMT provides the self-attention mechanism with explicit prior knowledge by introducing a spatial attenuation matrix based on the Manhattan distance. RMT enhances the ability to feature expression and efficiently captures the spatial link between pixels. Specifically, RMT extends the temporal attenuation process of RetNet to the spatial domain and produces a two-dimensional bidirectional spatial attenuation matrix. The elements of the spatial decay matrix are defined as follows for any two tokens,  $T_i$  and  $T_j$ :

$$D_{ij}^{2d} = \gamma^{|x_i - x_j| + |y_i - y_j|} \quad (1)$$

In this case,  $(x_i, y_i)$  and  $(x_j, y_j)$  represent the two-dimensional coordinates of  $T_i$  and  $T_j$  in the image, respectively, and  $\gamma$  is the attenuation factor, which controls the decay speed of attention with distance. RMT uses a form of decomposition that computes attention scores along the image's horizontal and vertical directions, followed by applying a 1D bidirectional decay matrix through Hadamard product ( $\odot$ ) element-wise multiplication. Let  $Q_H$ ,  $K_H$ , and  $V$  be the query, key, and value matrices in the horizontal direction, and  $Q_W$  and  $K_W$  be the query and key matrices in the vertical direction. The horizontal and vertical attention scores,  $Attn_H$  and  $Attn_W$ , are computed as follows:

$$Attn_H = Softmax(Q_H K_H^T) \odot D^H \quad (2)$$

$$Attn_W = Softmax(Q_W K_W^T) \odot D^W \quad (3)$$



In this case,  $D_{ij}^H = \gamma^{|y_i - y_j|}$  and  $D_{ij}^W = \gamma^{|x_i - x_j|}$  represent the two-dimensional coordinates of  $D^H$  and  $D^W$  in the image. The attention results of the two directions are multiplied to produce the final self-attention output:  $\text{MaSA}(X) = \text{Attn}_H(\text{Attn}_W V)^T$ . This decomposition technique maintains the linear complexity of the model, efficiently models the global information without destroying the spatial decay matrix, and guarantees that the computational cost increases linearly with the size of the input data. The architecture of the RMT model comprises four stages, each of which processes the input feature map through a series of convolutional layers, normalization layers, and fully connected layers. To offer a strong feature representation for the following tasks, the original MaSA is employed in the final stage after the decomposed MaSA has been used in the first three stages to progressively extract high-level semantic characteristics of the image, as shown in Figure 2.

### 3.2 Attention Blocks

The Channel-based and Spatial Attention Mechanism (CBAM) module, added to the RMT model, improves picture feature expression so that the model can concentrate on the disease marker area. The Channel Attention Module (CAM) and the Spatial Attention Module (SAM), the two components of CBAM, collaborate to improve feature maps from the channel and spatial dimensions, respectively. Finding each channel's significance in the feature representation is the primary objective of the Channel Attention module (CAM). First, the input feature map  $F$  is subjected to the global average pooling (AvgPool) and max pooling (MaxPool) operations, which compress the feature map's spatial dimension to  $1 \times 1$ , producing two  $1 \times 1 \times C$  feature maps, where  $C$  is the number of channels. The formula follows: these two feature maps show the channel's maximum and average features.

$$F_{avg}^c = \text{AvgPool}(F) \quad (4)$$

$$F_{max}^c = \text{MaxPool}(F) \quad (5)$$

The two feature maps were then processed using a shared multilayer perceptron (MLP). The ReLU activation function sits between the two wholly connected layers of the MLP. MLP to reduce the parameter overhead, the first fully connected layer compresses the number of channels by a factor of  $1/r$ , and the second fully connected layer returns the number of channels to the original value. The channel's attention weight  $t$   $M_c(F)$  is computed as follows, assuming that  $W_0$  and  $W_1$  are the weight matrices of the MLP.

$$M_c(F) = \sigma \left( \text{MLP}(\text{Avgpool}(F)) + \text{MLP}(\text{Maxpool}(F)) \right) = \sigma W_1(W_0 F_{max}^c) \quad (6)$$

In this case,  $\sigma$  is the Sigmoid activation function that normalizes the attention weights to the  $[0,1]$  interval. Determining the significance of each spatial location and capturing the spatial dependencies in the feature map are the Spatial Attention Module's (SAM) goals. To derive the input feature  $F'$ :  $F' = M_c(F) \odot F$  of the spatial attention module, the input feature map was first multiplied by the channel attention weight produced by CAM. Then, to create two  $H \times W \times 1$  feature maps for the first-order derivative of the capital letter  $F'$ , we used the 2D average pooling and max pooling procedures along the channel dimension, respectively, where  $H$  stands for the feature map's height and  $W$  for its width. The two feature maps are concatenated in the channel dimension to obtain an  $H \times W \times 2$  feature map, which is then reduced by a  $7 \times 7$  convolutional layer. Finally, the Sigmoid activation function is used to generate the spatial attention weight  $M_s(F)$ ; the formula is as follows:

$$M_s(F) = \sigma(f^{7 \times 7}([Avgpool(F'); Maxpool(F')])) \quad (7)$$

Where the  $7 \times 7$  convolution process is denoted by  $f^{7 \times 7}$ . The final attention feature map is produced by the CBAM module as a whole, combining the channel attention weights and spatial attention weights with element-by-element multiplication:

$$F_{out} = M_s(F) \odot M_c(F) \odot F \quad (8)$$

In this approach, CBAM may adaptively emphasize the critical channels and spatial locations in the input feature map, screen out the key disease information in the plant image, and boost the representation ability of the model for disease features.

### 3.3 2D Average Pooling Module

CBAM can filter essential disease information from the input image and adaptively emphasize important channels and spatial locations in the feature map, enhancing the model's ability to reflect disease characteristics. Using a  $k \times k$  pooling window (usually  $k = 2$ ), the 2D average pooling process slides across the feature map, computes the 2D average of the elements inside each window and produces the output feature map  $F$  pool, which has a size of  $\frac{H}{k} \times \frac{W}{k} \times C$ . The precise method of calculation is as follows:

$$F_{pool(i,j,c)} = \frac{1}{k^2} \sum_{m=0}^{k-1} \sum_{n=0}^{k-1} F(i \times k + m, j \times k + n, c) \quad (9)$$

In this context,  $i$ ,  $j$ , and  $c$  denote the output feature map indices corresponding to height, width, and channel dimensions, respectively, while  $m$  and  $n$  signify the positional indices within the pooling window.

Thus, 2D average pooling effectively reduces the number of model parameters while improving computing efficiency and generalization ability without significantly diminishing model performance. The model's robustness to local fluctuations enables it to effectively identify plant disease traits despite diverse lighting conditions, background interference, and other difficult circumstances.



## 4 Experiments

### 4.1 Datasets

Experiments are conducted on four publicly available datasets to confirm the generality of our model in various settings. The dataset distribution's image information is shown in Table I. Figure 5 lists the number of photos and illness name information utilized for training, validation, and testing datasets. Figure 6 presents some samples of photographs before and after image augmentation for each dataset. We separated all datasets into training, validation, and test sets in an 8:1:1 ratio using random seeds to prevent loss or bias in the data distribution. We preprocess the photos for deep learning training better to capture the distribution and properties of the data and lower the chance of overfitting. Three-fold cross-validation is employed in this work, which performs picture-enhancing procedures by rotating, transforming, scaling, and color-transforming the existing data.

**Paddy:** The paddy [23] is a dataset from a Kaggle competition. Nine rice leaves with particular diseases and one set of healthy rice leaves comprise the dataset's 13876 rice leaf samples. The dimension of each image is  $480 \times 640 \times 3$ . There are 75% labeled and 25% unlabeled data in the dataset. We use labeled data (10,407 samples) for training and testing.

**Corn:** Included in the corn [24] dataset are 3,852 photos of Blight (985 photos), Common Rust (1,192 photos), Gray Leaf Spot (513 photos), and Healthy (1,162 photos). All of the images are taken from the Plant Village dataset. The most widely used and accepted leaf picture dataset for identifying plant diseases is called Plant Village.

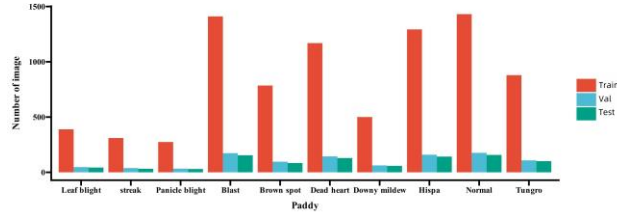
**Wheat:** 4087 photos of various sizes representing seven distinct kinds of wheat [25] illnesses are included in the wheat dataset. These photos show environmental elements like the sky, soil, and weeds that make it challenging to identify wheat crops.

**Coffee:** Three different kinds of coffee [26] leaves are included in the dataset: rust, red spider mite, and healthy. The leaves in each category contain photographs of precise size and resolution. The dataset's photos, which we collected in a natural field environment, show a range of background disturbances. We created a new dataset by selecting a thousand sample features for picture augmentation because certain features weren't significant enough.

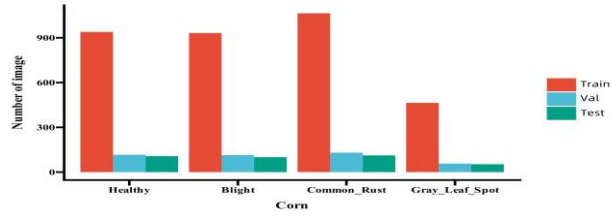
### 4.2 Experiment settings

Table II contains a detailed list of the hyperparameters used in this study. We established the same parameters in each experiment to ensure a fair comparison. For 100 epochs, we trained the model from scratch without using any pre-trained weights. We set the batch size and learning rate to 64 and 0.0001, respectively, using AdamW as the optimizer. In our experience, we minimized overfitting in the model using a weight

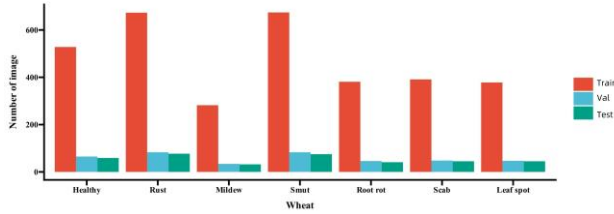
decay value of 0.05. Changing the weight decay rate impacts the training process but does not affect the outcome. All photos are consistently resized to  $224 \times 224$  and then augmented with data to improve the diversity of training samples and supplement the restricted data. Before training the network, We randomly rotate and centrally plant the samples in the training, validation, and test sets. Ultimately, we standardize all photos, utilizing the standard and mean square deviations. Table 2 delineates our hyperparameter configurations for model training.



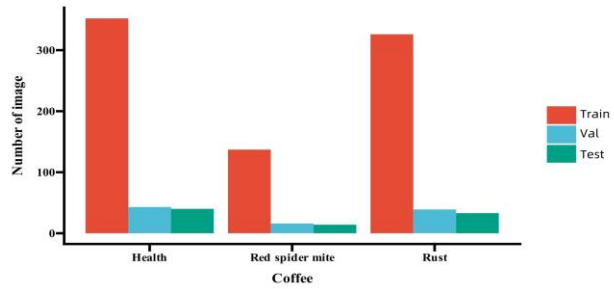
(a) Statistics of Paddy dataset



(b) Statistics of Corn dataset



(c) Statistics of Corn dataset




(d) Statistics of Coffee dataset

**Fig. 5.** Data distributions for the datasets used in our comparative experiments.

**Table 1.** Distributions of datasets.

Name	id	Name of disease	Name	id
Coffee	0	Health	Corn	0
	1	Red spider mite		1
	2	Rust		2
Wheat	3		Paddy	3
	4	Healthy		4
	5	Rust		5
	6	Mildew		6
	7	Smut		7
	8	Root rot		8
	9	Scab		9
		Leaf spot		


  



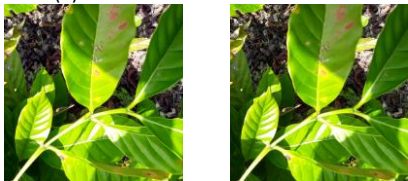
(a) Paddy



(b) Corn



(c) Wheat



(d) Coffee

**Fig. 6.** Examples of the dataset image. The image on the left is before enhancement, and the image on the right is after enhancement.

**Table 2.** Hyperparameter settings for training.

Name	Value	Description
Epochs	100	Number of times the model was trained
Batch Size	64	Number of samples selected for one training
Optimizer	AdamW	Tool used to bootstrap network update parameters
Learning Rate	0.0001	Tune parameters in optimization algorithms
Loss function	Cross entropy	Evaluates the gap between the predicted value and the actual value

### 4.3 Model evaluation

This study used the top-1 accuracy, recall, precision, and specificity to evaluate the best-performing model. True and false positives (TP and FP, respectively) indicate the number of correctly and wrongly predicted positive samples. True and false negatives (TN and FN) reflect the number of correctly and wrongly predicted negative samples. We evaluated the inference speed and model complexity using the number of parameters and the floating point operations per second (FLOPs).

**Table 3.** Detailed ablation result.

Network model	Top-1 Accuracy (%)				Parameters(M)	FLOPs (G)
	Paddy	corn	wheat	Cof-fee		
ResNet-50	96.3	93.8	92.7	77.0	23.51	4.13
SqueezeNet-1.0	89.3	92.1	70.1	79.7	0.74	0.73
ShuffleNetV2-1.0	91.9	92.5	89.6	68.8	1.27	0.15
CVT-Tiny	94.5	91.7	93.6	82.0	19.63	4.08
VGG-16	93.7	91.8	88.7	75.9	134.33	15.47
FasterNet-T0	95.1	92.1	89.6	77.2	2.63	0.34
EfficientNet-B0	97.0	95.1	93.5	81.6	4.01	0.41
PVT-tiny	95.2	95.4	92.2	78.1	23.58	3.69
RepVGG-A0	95.3	95.1	93.5	77.0	9.11	1.53
Swin-tiny	95.2	94.8	92.2	75.8	27.94	4.46
MobileV3	94.3	93.2	88.5	75.8	1.52	0.06
<b>PRMT</b>	<b>97.4</b>	<b>95.9</b>	<b>95.7</b>	<b>85.0</b>	<b>13.39</b>	<b>2.33</b>

$$\text{Top} - 1 \text{ Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (10)$$

$$\text{Recall} = \frac{TP}{TP + FN} \quad (11)$$

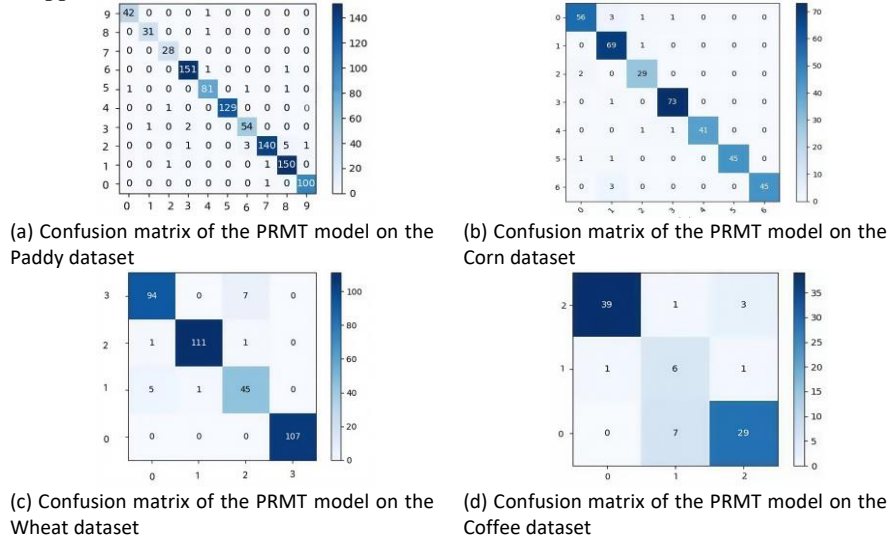
$$\text{Precision} = \frac{TP}{TP + FP} \quad (12)$$

$$\text{Specificity} = \frac{TN}{(TN + FP)} \quad (13)$$

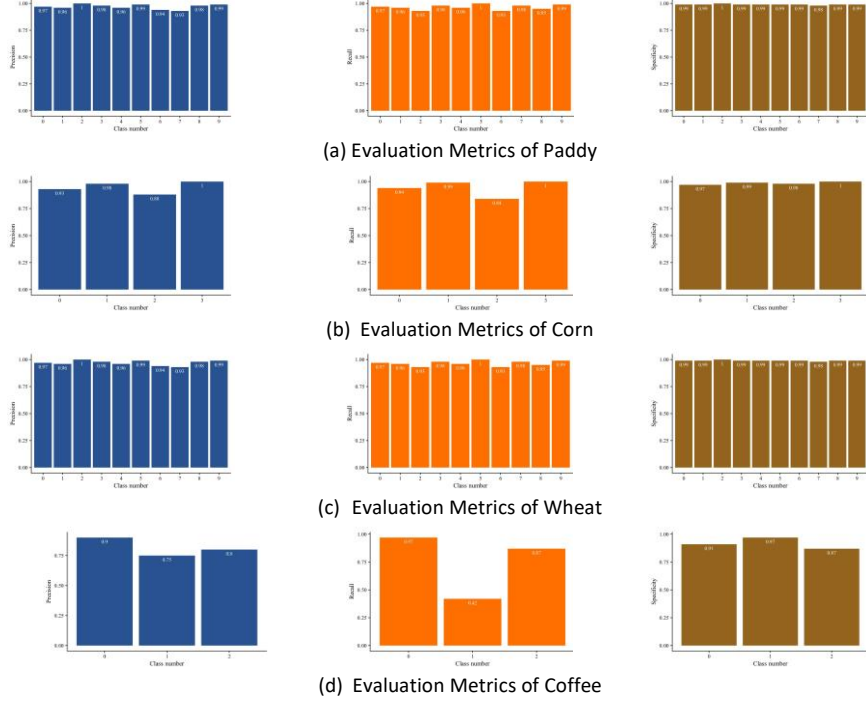
### 4.4 Comparative Experiment

We selected various standard CNN-based and VIT-based networks to compare with our approach. A selection of prominent deep learning models, such as ResNet [27], SqueezeNet [28], ShuffleNet [29], CVT [30], VGG [31], FasterNet [32], EfficientNet

[33], PVT [34], RepVGG [35], Swin [36], and MobileViT [37], was made based on comparative evaluations. The experiment utilizes multiple metrics to evaluate the model's performance thoroughly. These include accuracy, precision, recall, specificity, parameter count, and floating-point operations (FLOPs). The PRMT model has shown high accuracy across multiple data sets in Table III. The PRMT model achieved an accuracy of 97.4% on the rice dataset, while ResNet-50 attained 96.3%, SqueezeNet-1.0 reached 89.3%, and ShuffleNetV2-1.0 recorded 91.9%, among others. The PRMT model demonstrates superior performance relative to most comparative models, achieving an accuracy of 95.9% on the corn dataset. The accuracy rate of 95.7% on the wheat dataset is noteworthy. An accuracy of 85.0% on the coffee dataset exceeds that of many other models. The PRMT model demonstrates high reliability in classification tasks, evidenced by its elevated precision and recall rates, which effectively identify illness samples while minimizing false negatives and false positives. The PRMT model has distinct advantages concerning the number of parameters. The model diminished computational resources for training and deployment, comprising 13.39 million parameters—significantly fewer than larger models such as ResNet-50, which has 23.51 million parameters. The PRMT model executes 2.33 billion floating-point operations (FLOPs). The PRMT model has exceptional accuracy and significant computational efficiency, facilitating swift analysis of picture data. This renders it appropriate for real-time applications in agricultural disease detection.



**Fig. 7.** Confusion matrix of the PRMT model on the datasets



**Fig. 8.** Precision, recall, and specificity of the PRMT model on the datasets.

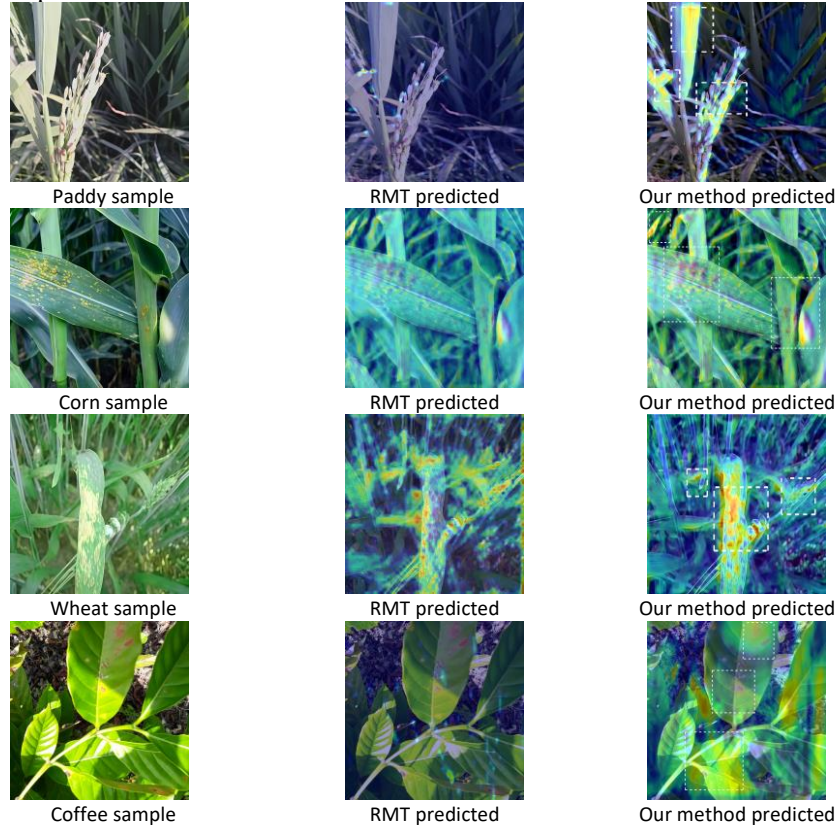
**Table 4.** Ablation experiments investigating each component in the PRMT model (Bold text highlights the best-performing network).

Meth- ods	Top-1accuracy(%)				Params(M)	FLOPs(G)
	Paddy(%)	Corn(%)	Wheat(%)	Coffee(%)		
RMT	94.3	94.8	93.7	77.0	13.2	2.32
+CBAM	96.5(+2.2)	95.6(+0.8)	94.6(+0.9)	82.2(+5.2)	13.81(+0.61)	2.35(+0.03)
+2D Avg pool	97.1(+2.8)	95.0(+0.2)	93.8(+0.1)	79.3(+2.3)	13.19(-0.01)	2.32(+0.00)
<b>PRMT</b>	<b>97.4(+3.1)</b>	<b>95.9(+1.1)</b>	<b>95.7(+2.0)</b>	<b>85.0(+8.0)</b>	<b>13.39(+0.19)</b>	<b>2.33(+0.01)</b>

The PRMT model demonstrates significant stability when utilized on datasets with intricate backdrops and varying lighting conditions. Traditional models like VGG-16 demonstrate less accuracy because they cannot extract features from complex back-grounds.

Using the RMT network to integrate the CBAM module and 2D average pooling layer, the PRMT model can more accurately depict the illness characteristics and successfully handle the interference of complex settings. Figure 8 displays the precision and recall results for each category on this dataset, whereas Figure 7 depicts the confusion matrix performance of the suggested fusion network topology on the four datasets. We display the RMT and PRMT heatmap predictions in Figure 9. The model's projected zones of high and low attention are easily discernible in the heatmap. Our approach

learns more features than the RMT model, as the red area in Figure 9 indicates. This outcome confirms the efficiency of the suggested plant disease identification strategy in this experiment.



**Fig. 9.** Heat maps display predictions from the original RMT model and our proposed method for the Paddy, Corn, Wheat, and Coffee datasets.

#### 4.5 Experiment setup

We list the precise hyperparameters used in this paper in Table II. We preserve the experiment's objectivity by applying the same experimental settings to the results of many models. The Intel(R) Xeon(R) Platinum 8255C CPU, running at 2.50GHz, includes an NVIDIA GeForce RTX 2080Ti GPU, 12 vCPUs, and 43GB of RAM. We make the software compatible with Cuda 11.3, Python 3.8, and PyTorch 1.11.0. We do all of our testing on deep-learning cloud platforms. We are using Ubuntu 20.04 as our operating system.

## 4.6 Ablation Studies

Through comparison experiments, we thoroughly examine the individual and combined effects of the RMT network, CBAM module, and 2D average pooling method on our model's performance to investigate each component's efficacy. The information provided demonstrates how well each element of our model works in Table IV. +Avg pool denotes using a 2D average pooling layer, introduced after the Manhattan matrix in the RMT block, and +CBAM denotes integration with channel attention and spatial attention before the RMT block based on the RMT model. On top of RMT, PRMT is a new backbone network that uses 2D average pooling layers and CBAM modules. Ablation studies demonstrate that by adding spatial prior knowledge, the RMT network enhances the model's comprehension of the link between picture regions. To better focus the disease marker region and improve the model's ability to reflect crucial traits, we used the CBAM module. To further enhance the model's adaptability to local changes while reducing the number of parameters, we included the 2D average pooling layer. PRMT allows the model to continue performing steadily in complex backgrounds, demonstrating the significance of each element of our PRMT method and how they interact.

## 5 Conclusion

In this paper, the PRMT model effectively addresses the issues that conventional and current deep learning models encounter. The feature extraction capability is successfully enhanced by integrating the CBAM module with the RMT network, and robustness to local changes is improved by employing 2D average pooling.

Comprehensive testing across several datasets validates the proposed model's superiority, exceeding the accuracy and generality of existing standard models. This study presents a more practical approach to diagnosing agricultural diseases and facilitating future progress in agricultural image processing.

## References

1. Junaid, M.D., Gokce, A.F.: Global agricultural losses and their causes. *Bull. Biol. Allied Sci. Res.* 2024, 66 (2024)
2. Gullino, M.L., Albajes, R., Al-Jboory, I., Angelotti, F., Chakraborty, S., Garrett, K.A., Hurley, B.P., Juroszek, P., Lopian, R., Makkouk, K., Pan, X., Pugliese, M., Stephenson, T.: Climate change and pathways used by pests as challenges to plant health in agriculture and forestry. *Sustainability* 14, 12421 (2022)
3. Skendžić, S., Zovko, M., Živković, I.P., Lešić, V., Lemić, D.: The impact of climate change on agricultural insect pests. *Insects* 12, 440 (2021)
4. Jafar, A., Bibi, N., Naqvi, R.A., Sadeghi-Niaraki, A., Jeong, D.: Revolutionizing agriculture with artificial intelligence: plant disease detection methods, applications, and their limitations. *Front. Plant Sci.* 15, 1356260 (2024)
5. Ngugi, H.N., Ezugwu, A.E., Akinyelu, A.A., Abualigah, L.: Revolutionizing crop disease detection with computational deep learning: a comprehensive review. *Environ. Monit. Assess.* 196, 302 (2024)



6. Orchi, H., Sadik, M., Khaldoun, M., Sabir, E.: Automation of crop disease detection through conventional machine learning and deep transfer learning approaches. *Agriculture* 13, 2023 (2023)
7. Khan, S., Naseer, M., Hayat, M., Zamir, S.W., Khan, F.S., Shah, M.: Transformers in vision: a survey. *ACM Comput. Surv.* 54(1), 1–41 (2022)
8. Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., Liu, T., Wang, X., Wang, G., Cai, J., Chen, T.: Recent advances in convolutional neural networks. *Pattern Recognit.* 77, 354–377 (2018)
9. Zou, J., Han, Y., So, S.-S.: Overview of artificial neural networks. In: *Artificial Neural Networks: Methods and Applications*, pp. 14–22. Springer (2009)
10. Alzahrani, M.S., Alsaade, F.W.: Transform and deep learning algorithms for the early detection and recognition of tomato leaf disease. *Agronomy* 13, 1184 (2023)
11. Isinkaye, F.O., Olusanya, M.O., Singh, P.K.: Deep learning and content-based filtering techniques for improving plant disease identification and treatment recommendations: a comprehensive review. *Heliyon* 10, e2024 (2024)
12. Gülmez, B.: A comprehensive review of convolutional neural networks based disease detection strategies in potato agriculture. *Potato Res.* (2024)
13. Parez, S., Dilshad, N., Alghamdi, N.S., Alanazi, T.M., Lee, J.W.: Visual intelligence in precision agriculture: exploring plant disease detection via efficient vision transformers. *Sensors* 23, 6949 (2023)
14. Dhanya, V.G., Subeesh, A., Kushwaha, N.L., Vishwakarma, D.K., Kumar, T.N., Ritika, G., Singh, A.N.: Deep learning based computer vision approaches for smart agricultural applications. *Artif. Intell. Agric.* 6, 211–229 (2022)
15. Pacal, I., Kunduracioglu, I., Alma, M.H., Deveci, M., Kadry, S., Nedoma, J., Slany, V., Martinek, R.: A systematic review of deep learning techniques for plant diseases. *Artif. Intell. Rev.* 57, 304 (2024)
16. Ngugi, H.N., Akinyelu, A.A., Ezugwu, A.E.: Machine learning and deep learning for crop disease diagnosis: performance analysis and review. *Agronomy* 14, 3001 (2024)
17. Karthik, R., Ajay, A., Bisht, A.S., Illakiya, T., Suganthi, K.: A deep learning approach for crop disease and pest classification using Swin transformer and dual-attention multi-scale fusion network. *IEEE Access* 12, 152639–152655 (2024)
18. Kunduracioglu, I., Pacal, I.: Advancements in deep learning for accurate classification of grape leaves and diagnosis of grape diseases. *J. Plant Dis. Prot.* 131, 1061–1080 (2024)
19. Kusuma, S., Jothi, K.R.: Early betel leaf disease detection using vision transformer and deep learning algorithms. *Int. J. Inf. Technol.* 16, 169–180 (2024)
20. Shoaib, M., Shah, B., Ei-Sappagh, S., Ali, A., Ullah, A., Alenezi, F., Gechev, T., Hussain, T., Ali, F.: An advanced deep learning models-based plant disease detection: a review of recent research. *Front. Plant Sci.* 14, 1158933 (2023)
21. Wang, D., Cao, W., Zhang, F., Li, Z., Xu, S., Wu, X.: A review of deep learning in multi-scale agricultural sensing. *Remote Sens.* 14, 559 (2022)
22. Fan, Q., Huang, H., Chen, M., Liu, H., He, R.: RMT: Retentive networks meet vision transformers. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5641–5651 (2024)
23. Petchiammal, P.D., et al.: Paddy Doctor: Paddy Disease Classification. Kaggle (2022)
24. Sharma, S.: Five Crop Diseases Dataset. Zenodo (2023)
25. Lian, R.: Wheat disease classification. Mendeley Data (2022)

26. Parraga-Alava, J., Cusme, K., Loor, A., Santander, E.: RoCoLe: A robusta coffee leaf images dataset for evaluation of machine learning based methods in plant diseases recognition. *Data Brief* 25, 104414 (2019)
27. Targ, S., Almeida, D., Lyman, K.: ResNet in ResNet: generalizing residual architectures. *arXiv preprint arXiv:1603.08029* (2016)
28. Koonce, B., Koonce, B.: SqueezeNet. In: *Convolutional Neural Networks with Swift for TensorFlow: Image Recognition and Dataset Categorization*, pp. 73–85. Apress (2021)
29. Ma, N., Zhang, X., Zheng, H.-T., Sun, J.: ShuffleNet v2: practical guidelines for efficient CNN architecture design. In: *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 116–131 (2018)
30. Wu, H., Xiao, B., Codella, N., Liu, M., Dai, X., Yuan, L., Zhang, L.: CVT: introducing convolutions to vision transformers. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 22–31 (2021)
31. Tammina, S.: Transfer learning using VGG-16 with deep convolutional neural network for classifying images. *Int. J. Sci. Res. Publ.* 9(10), 143–150 (2019)
32. Chen, J., Kao, S.-h., He, H., Zhuo, W., Wen, S., Lee, C.-H., Chan, S.-H.G.: Run, don't walk: chasing higher FLOPS for faster neural networks. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12021–12031 (2023)
33. Tan, M., Le, Q.: EfficientNet: rethinking model scaling for convolutional neural networks. In: *International Conference on Machine Learning*, pp. 6105–6114 (2019)
34. Wang, W., Xie, E., Li, X., Fan, D.-P., 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*, pp. 568–578 (2021)
35. Ding, X., Zhang, X., Ma, N., Han, J., Ding, G., Sun, J.: RepVGG: making VGG-style convnets great again. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13733–13742 (2021)
36. 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*, pp. 10012–10022 (2021)
37. Mehta, S., Rastegari, M.: MobileViT: light-weight, general-purpose, and mobile-friendly vision transformer. *arXiv preprint arXiv:2110.02178* (2021)