



# Detection and Classification of Plant Stress Using Hybrid Deep Convolution Neural Networks: A Multi-Scale Vision Transformer Approach



Bhargavi Thokala<sup>\*</sup>, Sumathi Doraikannan<sup>1</sup>

School of Computer Science and Engineering, VIT-AP University, Amaravati 522237, India

Corresponding Author Email: [sumathi.d@vitap.ac.in](mailto:sumathi.d@vitap.ac.in)

Copyright: ©2023 IETA. This article is published by IETA and is licensed under the CC BY 4.0 license (<http://creativecommons.org/licenses/by/4.0/>).

<https://doi.org/10.18280/ts.400625>

## ABSTRACT

**Received:** 4 March 2023

**Revised:** 14 July 2023

**Accepted:** 6 August 2023

**Available online:** 30 December 2023

### Keywords:

*plant stress, multi-scale vision transformer, cross-attention, deep convolutional neural network*

Agriculture forms the bedrock of India's economy, contributing significantly to the nation's development and sustaining the majority of its population. However, plant stress, specifically biotic stress, poses a significant threat to agricultural sustainability, leading to a substantial decline in crop production. Biotic stress, caused by living organisms such as bacteria, fungi, and viruses, damages plant tissues and weakens their overall health. As such, the control of biotic stress is pivotal to the enhancement of agricultural sustainability. In this study, a novel approach to early detection of plant biotic stress is proposed, utilizing advancements in deep learning techniques. A Hybrid Deep Convolution Neural Network (DCNN), termed "DCNN-MCViT", has been developed, employing a multi-scale vision transformer with cross-attention for efficient detection and classification of plant illnesses. This approach diverges from traditional Convolution Neural Networks (CNNs) and leverages the emerging capabilities of Vision Transformers, a recent development in the field of computer vision that has demonstrated superior performance in image classification tasks. Evaluation results have indicated that the DCNN-MCViT model significantly outperforms other state-of-the-art techniques, achieving an average accuracy of 99.51% in stress classification and a remarkable 99.78% accuracy on the comprehensive PlantVillage dataset. Moreover, the model demonstrated a high accuracy of 99.82% in estimating the degree of severity and classifying various forms of plant biotic stress. The findings of this study underscore the potential of the DCNN-MCViT model in improving agricultural sustainability through the early detection and intervention of plant biotic stress. This research represents a significant step forward in the application of deep learning techniques to agricultural challenges and holds promise for future applications in plant health monitoring and disease management.

## 1. INTRODUCTION

India, a stronghold of agriculture, contributed 19.9% to the nation's economy, as reported in the GDP statistics for 2020-21. The burgeoning population, projected to reach 9.2 billion by 2050, necessitates a substantial increase in crop production for sustainable growth [1]. In this context, Phytopathology, the study of plant diseases, plays a pivotal role. It assists in quantifying and diagnosing plant stress, often gauged through visual assessment of plant tissues [2]. Importantly, pinpointing the causative agent behind plant growth abnormalities remains a challenging yet crucial endeavor [3].

Recently, the Convolutional Neural Network (CNN), a novel technological approach, has shown promise in identifying the severity of symptoms in apple leaves affected by black rot, with accuracy exceeding 90% [4]. Another innovative method, PD2SE-Net, a multi-tasking CAD system, was able to categorize plant disease and estimate severity on the PlantVillage dataset with remarkable accuracy - 91% for severity estimation and 98% for disease classification [5]. These advancements underscore the importance of accurate and rapid plant disease identification, as diseases significantly

impact food production and grain yield [6]. The main focus of the research is to identify the correct disease that affects the crops and induce appropriate measures to reduce the spread of the infection from one plant to another [7].

With the rapid development of Artificial Intelligence, many fields opted to solve their issues by incorporating time and manpower, and accurate measures to take immediate action don't contribute to these AI-based technologies among them, object identification and classification captured the cause of the abnormality but also enhancing the yields and thereby increasing the economy of the country which in turn relies on Agriculture [8]. Usually, the pathological differences that occur in a plant are reflected in its leaves, stem, root, shoot, and flowers and among them, leaves are more identical in showing the difference [9]. One of the major concerns is to identify plant diseases at the early stage as plants are prone to various illnesses at every stage of their life cycle [10]. The traditional methods for detecting plant illness over large geographical area requires more manual methods leading to huge loss and it is important to implement timely the limelight in which different AI methods are effectively used in achieving the best results in the field of agriculture and the

work here focused on extracting the useful features from tomato plant leaves using Gabor and color based filters and radial basis function was used to detect the disease at the early stage with an accuracy of 90.37% [11]. With new advancements in Computer vision, Deep Learning methods in particular Convolution Neural Networks proved to project good work in many areas like face recognition, image formation, face detection, object tracking, and image classification, and with this improvement, many works used CNN for detection as well as classification of plant diseases [12]. The improvements in deep learning algorithms were remarked with the introduction of a CNN algorithm for its distinguishing capabilities to differentiate between a spatial and temporal relationship which is used for extracting useful features from the image which stores meaningful information [13]. Image feature extraction plays an important role. The CNNs use a specific operation called convolution to extract the local features from a raw image which minimizes the need for human intervention. Traditional Machine learning algorithms need a manual process to extract features like shape, texture, and colour and finally using AI-based algorithms like KNN, and SVM the features are reduced as in the case of cucumber and many other species [14]. Several CNN (Convolution neural networks) models are used to identify plant diseases for instance Mask-RCNN was used to detect fusarium head blight disease in wheat which attained an average accuracy of 92.01% [15]. Similarly, Resnet152, MobileNet, and Inception V3, models have attained an accuracy of 77.65%, 73.50%, and 75.59% [16].

Apart from plant diseases, plant stress both Biotic and Abiotic also has significant importance. A pertained VGG-16 model is used to identify Biotic and Abiotic stress from Paddy from the field images belonging to 12 different stresses which obtained an average accuracy of 92.89% [17]. Though CNNs are good at exploring the hidden features of the image they fail to project the positional information of relative features and this is because the interpretation of deeper layers is not done appropriately by the new layers and this can be resolved by increasing the number of filters, this creates a great increase in the computational cost. Despite having many advantages, the CNN models have a major drawback concerning the size of the kernel which leads to the loss of focus on the global information of an image [18]. Several architectural changes are suggested but the experts which gave a path for the introduction of the attention mechanism by replacing it with the convolution layer have shown commendable results [19]. Transformers are a standard paradigm of Natural Language Processing that uses an attention mechanism in the field of computer vision. Vision Transformers (ViT) is the first designed Transformer model to handle 2D images where the self-attention mechanism is used to collect the information from an image which is further broken down into non overlapping patches with size  $16 \times 16$  with 49M parameters which focuses on the global information by resolving long-range dependencies without compromising the computational efficiency. Furthermore, the newly formed 2D token are flattened into 1D tokens using linear Projection and are fed onto the Encoder layer [20]. But, the main drawback of ViT is that it could not work with small datasets as it focuses on extraction of long-range dependency features than on local features and the Transformer model requires large amount of data for training [21]. Though ViT outperforms existing CNN models, just by increasing the layers doesn't give better performance and requires more memory to deal with high

resolution images [22]. Several modifications were applied to the Transformer models finally concluding that to extract both local and global information it is better to combine ViT with CNN where convolution operations are replaced with attention mechanism [23].

An attempt to reduce the overhead of computational cost of the transformers, a pyramid structure was introduced to reduce the feature dimensions. But the pyramid structure works well with dense prediction tasks whereas while dealing with Image classification a new horizon should be implemented [24]. Analysing multi scale features on vision transformers proved to have achieved satisfying results for video and image classification [25]. A new technique called cross attention with Vision Transformers (cross-ViT) has gained more attention in recent times [26]. In order to obtain best results, we developed a hybrid model that uses Deep Convolution Neural Network with Multi scale Vision Transformers along with cross Attention (MCViT) to attain perfect balance between local and global spectral features and for identification of plant illness and classification of diseases and stress. Due to the wide range of managerial implications in the agricultural sector, several advancements are done in accurately identifying the biotic stress in plants. Mainly for early detection, transformers models are used to identify the severity of stress at the earliest by proactively identifying the onset of the disease. By applying the real time monitoring mechanism, this approach helps the farmers in identifying the loss on point to protect the crop from further damage. By introducing transformer models, the scope for detecting the plant stress at a large scale becomes more flexible. Even in Precision farming to improve the sustainability of the production plant stress detection gives a detailed insight into the pattern and analysis of yield loss and it greatly helps the farmers to concentrate on the particular target for the prevention of wide spread of diseases using transformer models, farmers can mitigate the use of pesticides while combatting with the environmental abnormalities effectively. In the area of Research, plant stress detection and classification using transformer models generates data related to frequency of occurrence of diseases, pattern analysis of stress which contributes to the future research for disease control strategies by fostering innovations in agricultural sector. From the above stated managerial implications, we conclude that our research focuses mainly on how transformer models can work effectively in identifying the biotic stress plants. The analysis of leaf diseases relies heavily on feature extraction. We can identify the root of plant stress more precisely. Even though CNN does feature extraction on its own, it is limited in what it can do because it only extracts a small number of crucial features. As a result, we added CNN functionalities to our study on vision transformers. With the aid of deep convolutional neural networks (DCNN) and multi-scale vision transformers, our work attempts to extract both local and global features from plant leaves. This model focuses on attaining correlation between different scales with dual branches. The contributions of our work include:

1. Inclusion of Multi Scale Vision Transformer architecture where a multi scale dual branch is used to fuse features obtained from various scales.
2. A hybrid model that combines the DCNN with proposed MCViT to extract absolute features.
3. To classify the disease and estimate its severity.

This paper focuses on the summary of the related works in section 2. Section 3 focuses on the dataset used for our work and the methodology to carry out our research followed by

results and discussion in section 4 and the work was summarized with conclusion in section 5.

## 2. RELATED WORKS

In the field of plant disease identification and classification, numerous studies have made significant strides. For instance, Khan et al. [26] achieved a commendable accuracy of 98.60% by developing a fusion method for detection, classification, and feature extraction of plant diseases on the CASC-IFW database. Kundu et al. [27] 's Custom-Net model, designed to detect pearl millet diseases utilizing Raspberry Pi, achieved a slightly higher average accuracy of 98.78%.

In a different approach, Chen et al. [28] proposed a hybrid model to detect plant diseases in maize using three different datasets. By integrating two blocks of pre-trained VGG16 and two Inception v3 blocks, accuracies of 84.25%, 92%, and 80.38% were achieved. Similarly, Anami et al. [29] utilized traditional conventional classifiers, such as the Support Vector Machine (SVM) and Back Propagation Neural Network (BPNN), to identify Biotic and Abiotic crop stress in Paddy, achieving an accuracy of 89.12%.

The integration of the residual CNN and self-attention mechanism, as proposed by Zeng and Li [30], was applied to the AES CD9214 and MK-D2 datasets. The results were promising, with accuracies of 95.33% and 98%, respectively. Lu et al. [31] designed a Ghost enlightened Transformer (GeT) architecture, which was applied to a grape's dataset comprising 12,615 vine images across 11 classes, and achieved an accuracy of 98.14%.

The potential of Vision Transformers (ViT) in the agricultural sector was highlighted by Reedha et al. [32]. They emphasized the importance of classification without modifying the basic versions with 16 and 32 attention blocks (ViT-B16 and ViT-B32). In a similar vein, Thakur et al. [33] designed a hybrid model called PlantXViT, which combines two convolutional blocks of VGG16, one inception block, and ViT architecture. This model, designed to capture the local features of the image, demonstrated remarkable performance compared to other state-of-art techniques.

Graham et al. [34] proposed a hybrid transformer-based architecture, LeViT, which combined the advantages of CNN and transformer for vision tasks. Meanwhile, Wang et al. [35] suggested incorporating attention into the EfficientNet network on self-built datasets for apple leaf disease and achieved 98.92% recognition accuracy, albeit with large FLOPs and Parameters. Bi et al. [36] attempted to redefine the model by using fewer FLOPs and Parameters for Mobile-Net to identify apple diseases, achieving a 73.5% recognition accuracy.

Rangarajan et al. [37, 38] applied AlexNet and VGG16 architectures for identifying plant disease in tomato leaves from segmented images, where the background pixel is set to zero, and achieved a 97.49% accuracy. Sumari et al. [39] proposed a novel technique to identify the disease in Manggis fruit using hybrid deep learning architectures in comparison with traditional state of art techniques to classify the fruits that are free from diseases as this helps the farmers to take effective measures to prevent further spread of the disease. The model achieved an accuracy of 94.99%. Vasavi et al. [40] used various Machine learning algorithms to choose the best algorithm for identifying the diseases in chili crops by examining the images taken from the field and achieved an

accuracy of 94% and 96% for Gradient boost and random forest algorithms. A ViT-based multi-branch architecture, i.e., Cross ViT, was used to fuse the features obtained from branches belonging to different scales. This architecture processes the image patch obtained from the dual branch of the transformer operated at different scales. Each branch extracts the local information of the image patch of respective scales, and an efficient model fuses the features obtained from two branches. To incorporate positional information in each token of the corresponding branches, a learnable positional embedding is introduced at the head of each token, akin to the Vision Transformer (ViT) architecture. A cross-attention model was proposed to fuse information obtained by multi-scaling of patch tokens from both branches. Inspired by this, a dual-branch vision transformer is proposed for identifying plant diseases.

The aforementioned studies mainly focus on using various CNN models to identify plant diseases, emphasizing the efficiency of various neural networks in classifying biotic stress in plants using images. The introduction of transformer models to biotic stress classification is due to their ability to capture both global and local features, whereas CNN captures only important features for the purpose of extracting important features. Transformer models have shown remarkable results in exploring various features of an image to analyze the root cause of biotic stress.

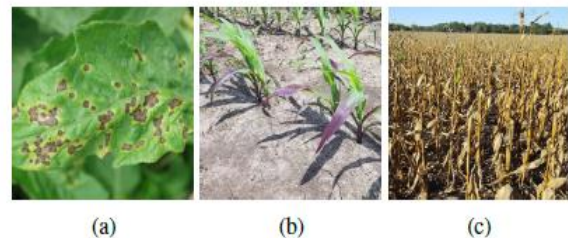
## 3. MATERIALS AND METHODS

### 3.1 Dataset overview

In the Proposed work we have compared the results with 3 datasets-PlantVillage, Plant Village Paddy, DiaMOS dataset (peer leaf).

#### 3.1.1 PlantVillage dataset

The PlantVillage dataset is mainly used for multiclass classification of images and it contains 55,448 background-only images (61,486 in the enhanced version) that are divided into 39 classes categorizing the images into healthy and diseased. This dataset contains variety of images belonging to various species like cherry, apple, cornrasphberry, blueberry, grape, orange, potato, bell pepper, peach, soyabean, squash, strawberry, tomato altogether 14 species. In addition to this this dataset also contains 17 fungal disease images, diseases caused by bacteria, mold, virus and mites. The example images of Plant Village dataset are shown in Figure 1.



**Figure 1.** Plant stress images. (a) bacterial spot of tomato, (b) phosphorus deficiency in corn plants, and (c) corn fields having stalk rot

The tomato dataset is made up of RGB images with size 256×256 collected in a controlled setting, with each leaf

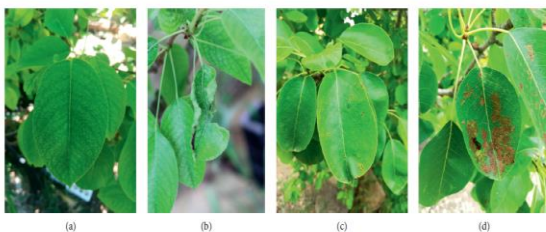
centered on a continuous background (tabletop or black). It has a total of 15,892 images in 10 categories (normal, stress), under 10 different labels like training: 11,204 images, Validation: 3,092 images and Testing: 1,596 images. The corn dataset includes 18MP ground photographs captured with an 18MP camera, as well as RGB images of corn leaves in their natural habitat and pictures of noisy cornfields (3456×5184 or 5184×3456 image size generated). The region of interest (plant stress, marked with the occurrence in each case) is situated in the exact middle of the image. There are 8,911 images total in the dataset among 11 categories (10 stress and normal) (divides as training-6,232 images, Testing-886 and validation-1,793 images). From the 6 categories, five labeled as stress images and one category as normal. Total of 6,635 images altogether make up the dataset (divided as training-4,642 images, Testing-660 and Validation-1,335 images). Because the plant leaf center plays a crucial role the corn and soybean datasets, each image center is cropped (made square) to 345×345 (1/10 resolution). No cropping or size changes are made to the tomato image.

### 3.1.2 Paddy crop

One of the wide publicly available data sources is PlantVillage which constitutes a total of 54,306 images belonging to 26 classes of 14 species with 3355 specifically focusing on paddy leaves. Among them paddy leaves are represented by 3355 images which includes images of various categories like Brown spot (523 images), Hispa (565 images), Leaf blast (779 images) and remaining images in a total of 1488. The dataset is enriched using classical image augmentation techniques, resulting in 167,750 images. To ensure consistency and computational efficiency, the images are resized into a resolution of 256×256 pixels. The dataset is then divided into three groups as training, validation, and testing. This partitioning enables effective model training, parameter tuning, and unbiased evaluation of model performance.

### 3.1.3 DiaMOS plant

It is a real time dataset created, to identify and track plant ailments. Images of pear fruit with three different biotic stresses-primarily from leaves-are included in the DiaMOS plant dataset. In total, 3057 images were gathered, comprising both healthy and sick leaves that had experienced one or more biotic stressors, such as leaf spot, leaf desiccation, and slug damage as shown in Figure 2.



**Figure 2.** Sample of pear leaves stress images in DiaMOS plant dataset

## 3.2 Methodology

### 3.2.1 Data augmentation

Another crucial stage in computer vision is resizing the images before processing. By changing the size (width and height) of images it becomes easy for the model to train on

newly preprocessed or scaled images. When compared to images, that are twice or three times larger than machine learning models can learn more quickly on tiny images since they don't require the network to learn on as many pixels. When feeding the images into the Keras-based deep learning architecture pipeline, we used a technique to resize the images to 150×150 pixels and 224×224 pixels. There are two input sizes (150×150×3) and (224×224×3), where 150 and 224 are width and height and 3-way is a colour channel. Since samples of healthy plant leaves are significantly less than those of the other two types, we made a label-preserving alterations to plant leaves to fictitiously enhance the data/sample size and lessen model overfitting.

We followed the same with the images having same range with smaller pixel values to cut down the computing costs. This is done using a scale transformation. The parameter value (1/255) indicates that all pixel values fall between 0 and 1. To rotate an image by a specified angle, a transformation called rotation is applied. To randomly offset the image to the right or left, use the Width Offset Range transform and a value of 0.1 for the Width Offset parameter. By setting the Height Shift Range parameter to 0.1 can vertically shift the training images. The shearing angle is the angle at which the other axis is scaled up in a shearing transformation while the first axis is fixed to the image. In order to avoid this, a 0.2 shear angle is used. To conduct a random scaling transformation, the scaling range parameter is applied. Zooming in>1.0 on a picture requires a zoom factor of 0.2, whereas zooming out on an image requires a factor of<1.0. The image is turned horizontally with the flip apply function. A straightforward image processing technique generates 32,073 enhanced images, balancing the size of each class with images from the dataset of 2,000 images.

### 3.2.2 Elimination of natural background

In Pre-processing, segmentation plays a crucial role and is often considered as difficult operation. Before evaluating individual leaves, plant leaves must be segmented. The inclusion of depth information improves the accuracy of leaf segmentation process. Based on depth characteristics, backdrop plant leaves are first eliminated. However, the depth of the leaves is obviously different from that of the items in the backdrop. Since depth noise cannot be utilised to segment leaves, depth pictures of plant leaves are softer and cannot be represented. The depth of the leaf in mentioned in Figure 3.

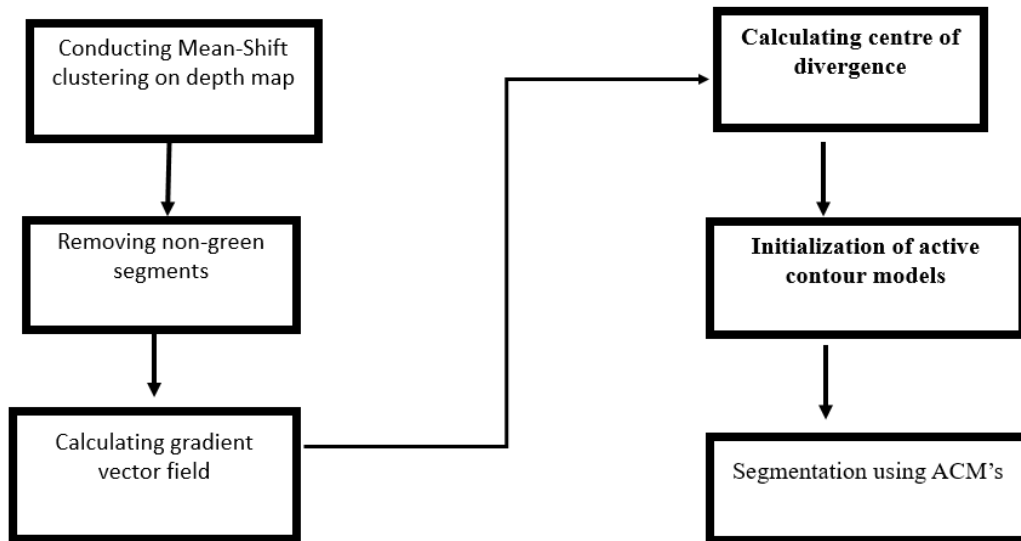


**Figure 3.** RGB image of leaf depth

### 3.2.3 Enhanced mean-shift segmentation algorithm

Enhanced Mean-shift Segmentation techniques are used to estimate the local density gradients in correlation with image pixels as shown Figure 4. The process of gradient estimation is iteratively performed to identify similar pixels in corresponding images. It employs an enhanced mean-shift technique and segments images of object depth.





**Figure 4.** The proposed leaf segmentation method

The d-dimensional point set  $x_i=1$  is the average motion vector at position  $x_i$  determined by  $n$  is given in Eq. (1):

$$M(x) = \frac{\sum_{i=1}^n x_i K_w(x_i - x)}{\sum_{i=1}^n K_w(x_i - x)} - x \quad (1)$$

where, the kernel function is represented in Eq. (2):

$$K_w(x) = |W|^{-\frac{1}{2}} K(W^{-\frac{1}{2}}x) \quad (2)$$

In practice, a diagonal matrix termed W-add bandwidth matrix with positive symmetry-is utilized, with  $x$  designating the kernel's center.  $x$  represents the position at which the vector in motion is calculated. The average moving vector  $M(x)$  computes the local density gradient and the most significant direction of density increase for that pixel in improved average moving clustering.  $x_i$  represents the  $i^{\text{th}}$  position in the d-dimensional point set and  $K_w(x_i - x)$  represents the difference between  $x_i$  and  $x$  applied to the kernel function  $K$ . Summation  $\Sigma$  is applied to values from 1 to  $n$ . From Eq. (2) the kernel function with parameter  $w$  is applied to  $x$  denoted by  $K_w(x)$  and  $|W|$  the determinant of matrix  $W$  is calculated.  $W^{-\frac{1}{2}}x$  represents the square root of inverse of  $W$  and kernel function  $K$  is applied to it. In order to find local density peaks, apply the formula given in Eq. (3):

$$y_{j+1} = \frac{\sum_{i=1}^n x_i K_w(y_i - x_i)}{\sum_{i=1}^n K_w(y_i - x_i)} \quad (3)$$

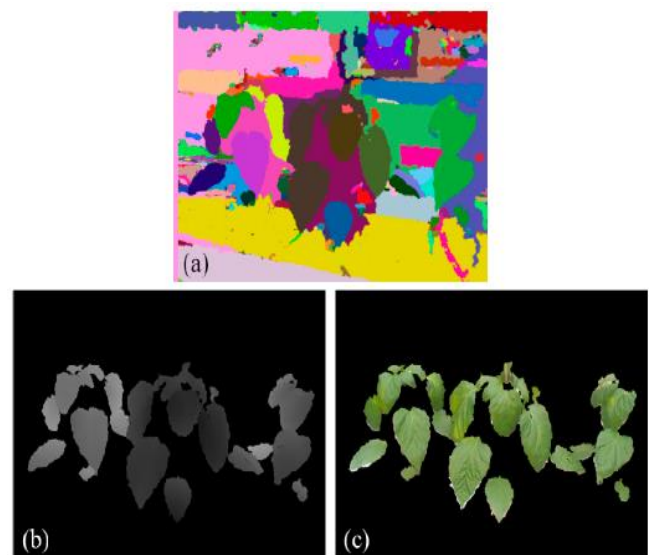
All points pointing to the apex that are plotted on the same peak are regarded as belonging to the same segment. This results in a general improvement in Mean Moving Segmentation.

Grayscale depth data are processed using a modified mean-shifting clustering approach in this work to produce leaf areas and background. Let  $x_i = 1, 2, \dots, n$  represent the 2D input in the grayscale image space domain. The local density peak is represented by  $y_{i,j}$  at pixel  $x_i$  at iteration  $j$ . A more effective approach for shifting depth data segmentation is presented in the next section.

1. Creates the initial state of a window for coverage and a window for navigation in the radius  $ps$  and  $rs$

domains, respectively. A local density peak should be established at  $y_i$ ,  $I=x_i$ , iteration step  $j=I$ .

2. Depth difference for each pixel is calculated in the space window, distance window, and the centre pixel, then work out the motion vector of the navigation window.
3. Determine and set the average depth value  $y_i$  and average depth  $j+I$  for each pixel in the spatial frame.
4. In the spatial window, clusters with cpp in range  $[1 \dots m]$  are formed by pixels whose depth difference is less than the radius of the range window  $rs$ . Repeat the preceding methods until the shift values are modest enough to suggest convergence to determine whether the enhanced mean shifting clustering has reached convergence.



**Figure 5.** Enhanced mean shift segmentation of leaves, (a) mean-shift segmentation outcome, (b) segmentation results of the image depth and (c) RGB image segmentation view

Therefore, the depth picture is separated into sub-regions when mean-shift convergence takes place. These pieces are shown in different hues in Figure 5. By contrasting the RGB

tones of green plant images with leaf images, Figure 5 demonstrates how the segmentation results are generated. In this stage, background items that are not green are eliminated. The plant picture is divided into several leaf images, which are then extracted from the segmented leaf depth and colour images.

In order to calculate a, use the GVF vector's four surrounding pixels,  $p(i, j)$ ,  $p(i+1, j)$ ,  $p(i, j+1)$ , and  $p(i+1, j+1)$ .  $I$  and  $j$  are the pixel coordinates of the image such that  $v(i, j) = x(i, j)$  and  $y(i, j)$ . for computing the pixel  $p(i, j)$  GVF vector for a given pixel. "v" includes a function to indicate direction as in Eq. (4):

$$\text{sign}(x) = \begin{cases} 1, & x > 0 \\ 0, & x = 0 \\ -1, & x < 0 \end{cases} \quad (4)$$

Therefore, the potential scattering point set  $sP$  is as follows mentioned in Eqs. (5)-(7):

$$P_{sx} = \{p(i, j) | x(i, j) \leq x(i + 1, j) \text{ and } \text{abs}(\text{sign}(x(i, j)) + \text{sign}(x(i + 1, j))) \leq 1\} \quad (5)$$

$$P_{sy} = \{p(i, j) | y(i, j) \leq y(i + 1, j) \text{ and } \text{abs}(\text{sign}(y(i, j)) + \text{sign}(y(i + 1, j))) \leq 1\} \quad (6)$$

$$P_s = P_{sx} \cap P_{sy} \quad (7)$$

where,  $P_{sx}$ ,  $P_{sy}$  represent the potential scattering points in the x and y direction. According to the center of divergence, the initialization of the contour model distinguishes lobes from occlusions. Each lobe's borders are shown by yellow lines, while the initialized model is shown by green circles in Figure 6.



Figure 6. Simulation results of plant stress segmentation

### 3.3 Proposed multi-scale vision transformer with DCNN

The input image is trimmed into patches of fixed size and constant scale for conventional Vision Transformers. Various cropping ratios are used for this process. Multi-scale feature representations of images are created for image recognition by dividing them into large and tiny chunks. Both tiny and large blocks can be used to represent little or huge patches,

respectively, using small or large branches. This study employs two FCs to incorporate two branch outputs in turn, enabling the application of inputs obtained from two branches into the Vision Transformer. Figure 7 displays the class token and has an output dimension of 768. Dimension is utilized by each quadrant.

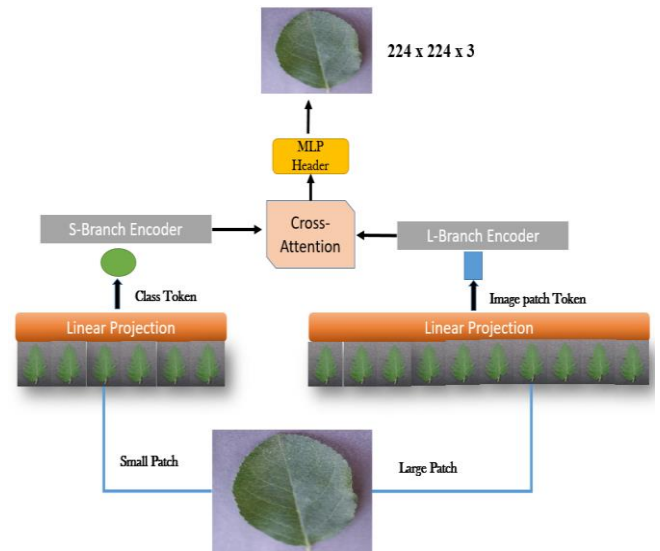


Figure 7. Architecture of proposed transformer with cross attention layer (CrossViT)

Normally, the Vision Transformer inputs the image tokens into multi-layer perceptron and the multi-head iterates them and the output is added to the token itself, and then again, the input is fed into the MLP, and multi-head layers iterate, and adds the output before that. Then, in the same way, repeat this process additional L times before sending the result to the next transformer encoder. By simply repeating the embedding, this method does a poor job of shifting focus from the current input to the following layer. This cycle multiplies the attention of the next layer with the attention of the preceding layer to obtain new active attention. This technique allows for the effective utilization of the attention of every branch of Vision Transformer. The results of the following categorization efforts are improved by providing a piece of caution information. In Cross-Attention Fusion's most recent two quarters, the class tokens are represented by the two circles above.

Our model is used to process large and small image patch tokens in two independent branches having different computational difficulties, and these image patch tokens are recurrently fused to complete one another. This project's major objective is to create a classification system that works for visual translators. Using a cross-alert module that is effective, each transformer branch creates non-patch tokens to act as intermediaries for communicating information with other branches through attention. By merging CNN and ViT to add multi-scale capabilities rather than using a model that is only based on ViT, as in earlier work, our suggested method differs significantly from that of the prior research.

As an addition to the ResNet family of designs, the Non Local Network (NLN) is proposed. In this architecture, the last block includes a non-local aggregation procedure. Multi Head self-attention is connected to the ResNet bottleneck block using a Bottleneck Transformer introduced by the NLN design. These two techniques significantly enhance multi-vision work.

Similar to how adding a CNN with non-local operations enhances image classification, we construct a DCNN and present it as CrossViT (DCNN-MCViT), which is a DCNN with ViT added, and use the DETR model for object recognition and picture activity. Contrary to earlier non-local block applications, we implement multi-scale interactions directly using our suggested DCNN-MCViT architecture by gradually adding CNNs.

Representations are typically transferred at low resolution levels in CNN-based image categorization networks. ResNet, for instance, has five stages with a final feature map that is 1/32×1/32 in each dimension. Each level reduces the resolution by half. ViT, on the other side, starts off with tokens that are 16×16 in size, which lowers the resolution in this dimension. At that resolution, the final layer yet persists. As a result, ViT has a higher likelihood of maintaining position data than ResNet. Since location information is not necessary for classification judgements in image classification tasks, we cannot claim that ViT has an advantage over ResNet since it keeps track of locations.

Due to the associations being explicitly embedded in the DCNN architecture, CrossViT typically needs a lot of training data to learn them. CrossFit can nonetheless discover intra-image correlations in the absence of these guided deflections. For example, capturing spatial rather than local semantics is impossible with DCNNs. To merge these two architectures, DCNN-MCViT uses CrossViT to compute non-local spatial semantics obtained from CNN to encode local data. The markers  $T_p$  from the obtained final feature map before the customary global pooling step are extracted to create the CNN feature map with kernel size  $P=1$ . Unlike the CrossFit model, which extracts labels directly from the input image  $X$ , this is not the case. We put forth a multi-scale hybrid visual transformer based on earlier jobs that join non-local activities with already-existing CNNs like CrossViT. Our suggestion is to apply CrossViT to the backbone CNN at various scales, as opposed to the original HViT which simply extended the backbone CNN with CrossViT. To translate the non-local spatial semantics of images to other scales, we additionally introduce cross-scale links between ViTs.

### 3.3.1 Cross-attention

Class tokens will be cautious while interacting with block tokens from larger branches of modules' small branches in order to finally produce new class tokens. According to the architecture, in order to obtain a new class token for the large branch. To facilitate attention in the small branch, we perform a trade between the corresponding class token and the block token. This exchange ensures that the attention mechanism in the small branch focuses on the relevant information. Instead of directly using the class token, we substitute it with the block token, allowing the attention mechanism to consider the local context captured by the block token. This enables the small branch to concentrate on important features and enhance its attention-based operations.

Fusion should only be used on the minor branch. Then, as demonstrated in formula, project the minor branch's class tokens via FC and compare the results to the major branch's class token given in Eq. (8):

$$x^{ls} = [f^s(x_{cl}^s) || x_{patch}^l] \quad (8)$$

where, the sub-branch class label is represented by  $x_{cl}^s$  and  $f^s$  is the FC used to project it to this dimension. Between class

markers in the minor branch and block markers in the main branch, cross-annotation is performed throughout the entire module. The following formula can be used to numerically express cross attention following Eq. (9) and Eq. (10):

$$q = x_{cls}^{ls} W_q, k \quad (9)$$

$$A = softmax\left(\frac{qk^T}{\sqrt{\frac{C}{h}}}\right) \quad (10)$$

in which,  $C/h$  is the learnable parameter,  $C$  is the embedding dimension, and  $h$  is the total number of attention heads. The attention map produced by Cross-computational attentions and memory complexity is linear in our study because we only employ CLS tokens.

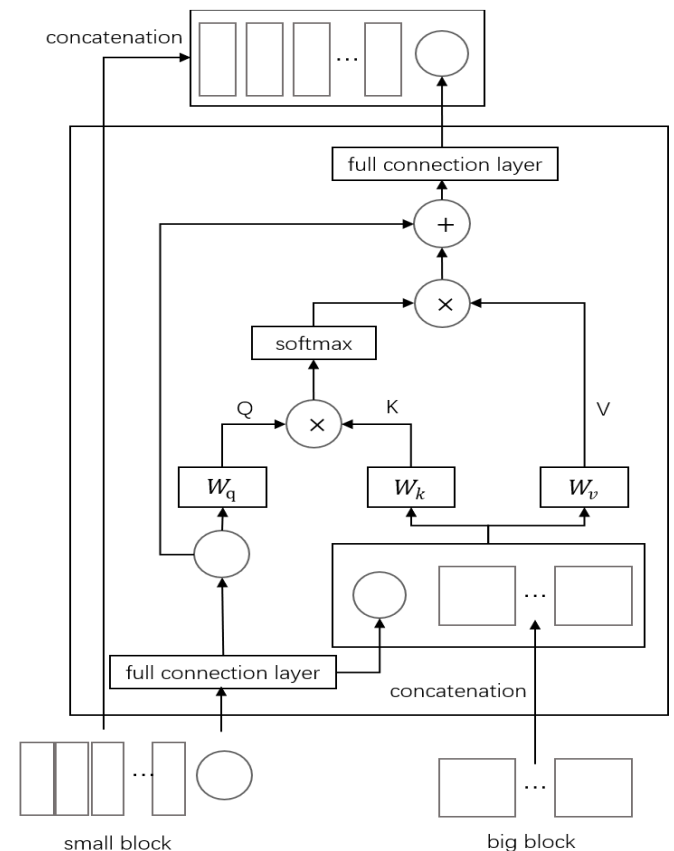
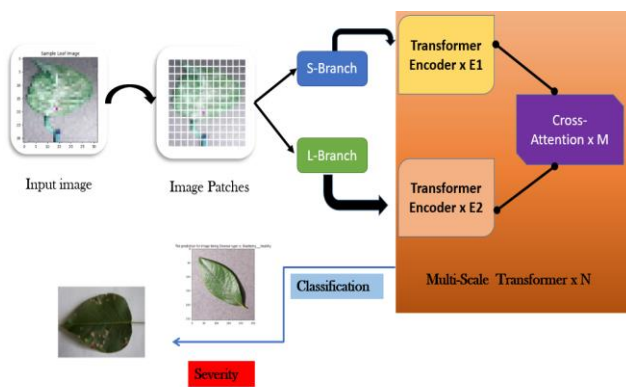


Figure 8. Cross-attention fusion sketch

Figure 8 illustrates the fundamental concept of our suggested intersectionality, wherein one quarter uses patch tokens and the other use Fusion CLS tokens. For example, to communicate information between patch tokens of various branches more quickly and efficiently, we first utilise the CLS tokens of each branch as a proxy before converting it into our own Backproject to the branch. The interaction between patch tokens in different forks helps contain information at various scales because the CLS token has learned abstract knowledge across all patch tokens in its own fork. The CLS token interacts with its own patch token in the following converter encoder after being fused with other branch tokens to amplify the information of each patch. The primary branch's cross-attention module is described in the section that follows, and the operation is carried out by simply switching the secondary branch's indexes  $l$  and  $s$ . The process of creating Attention maps in Cross-attention is more effective than All-Attention

since we only employ CLS in the query, which reduces the computational and memory cost from quadratic to linear.

An input layer, one or more hidden layers that are switched sequentially from the input layer, and an output layer make up a multilayer neural network structure. A multilayer neural network contains many connections from the first layer to the network and vice versa. At this time, concentrate on the process's closing steps for the greatest outcomes. In order to do this, a classifier with a softmax hierarchy is fed with a pooled set of the features that were determined in the previous stage. You may then decide which response is the most accurate using this. Unlike the activation function used in hidden layer, the output layer activation function is special. Each layer has a distinct function and execution strategy. After a classification procedure, the final hierarchy can produce class probabilities for incoming data as shown in Figure 9.



**Figure 9.** Proposed DCNN with CrossViT (DCNN-MCViT) with N multi scale transformer encoders having 2 inbuilt transformer encoders E1, E2 with a stack of M cross attention layers

## 4. RESULTS AND DISCUSSION

### 4.1 Classification results

We carried out tests utilising the Deep Learning Toolbox available in the MATLAB 2021 b programming environment to categorise biotic stress by Importing a pretrained CNN model into Deep Network designer; change the required hierarchy characteristics to get it ready for transition learning.

A recent collection of images depicting life stress is matched to categories by substituting the output layer for the last learnable layer of all models (also known as the classification layer). On a single Windows 10 workstation with an NVIDIA GPU, an Intel Core i7-11700 CPU, 16GB of RAM, and 12GB of RAM, we tested and trained the CNN model.

By dividing the pixel values of the original and supplemented datasets by 255, we normalized each image in this study. Then scale it down to match the model's basic measurements. To account for these variations, the Inception V3 model is set to 224×224 pixels, while the AlexNet, ResNet50, and VGG16 pictures are set to 227×227 pixels. All models in the EfficientNet architecture have to be scaled to match the input picture resolutions of our experimental research due to hardware constraints.

A minimal amount of data alignment is used to update weights and biases during backpropagation. It is often advised to choose a value divided by sum of samples in the dataset when selecting a size that is less than the complete dataset. Because it strikes a compromise between rapid network convergence and precise predictions, this value is helpful for training. The maximum allowable mini-batch size in this study is 16, which takes into account the hardware resources of all models. Table 1 provides a list of the fundamental parameters utilised in each experiment.

**Table 1.** Simulation parameters setting

Model	Input Size	Parameters and Optimization Technique	Model Learning Rate
AlexNet	227×227	A (B 1=0.9, B 2=0.999, D=0.0)	0.001
VGG16	224×224	SGD (M=0.0, D=0.0)	0.01
ResNet50	224×224	A (B 1=0.9, B 2=0.999, D=0.0)	0.001
Inception V3	229×229	A (B 1=0.9, B 2=0.999, D=0.0)	0.001
DCNN-MCViT	132×132	A (B 1=0.9, B 2=0.999, D=0.0)	0.001

\*M-Momentum, \*A-Adam, \*B-Beta, \*D-Decay

#### 4.1.1 Training dataset

The PlantVillage dataset and DiaMOSPlant datasets are utilized for training and testing data in this work. Then 60%, 30%, and 10% of the total datasets used as training, testing and validation is tabulated in Table 2.

**Table 2.** Division of images into train, test and validation sets

Total	Training (60%)	Validation (30%)	Testing (10%)	Training (70%)	Validation (20%)	Testing (10%)
<b>PlantVillage image dataset</b>						
<b>30920 (Original)</b>	18552	9276	3092	21644	6184	3092
<b>15,46,000 (Augmented)</b>	927600	463800	154600	1082200	309200	154600
<b>PlantVillage paddy crop stress image dataset</b>						
<b>3355 (Original)</b>	2013	1007	335	2349	671	335
<b>167750 (Augmented)</b>	100650	50325	16775	117425	33550	16775
<b>DiaMOSPlant pear leaves stress image dataset</b>						
<b>3057 (Original)</b>	1835	917	306	2140	612	306
<b>152850 (Augmented)</b>	91710	45855	15285	106995	30570	15285



#### 4.2 Performance measures

Using binary or multiclass outputs, evaluate the effectiveness of classifying plant diseases. A confusion matrix tracks actual and anticipated performance. Additionally, specificity as the actual rate indicated in Eq. (12) sensitivity as the genuine positive rate in Eq. (13), and the capacity to properly differentiate between healthy and sick leaves. The number of positive outcomes accurately categorised by the total of all positive results in Equation is the recovery rate, also known as the chance of detection in Eq. (11). Calculate the Mathews Correlation Coefficient (MCC) in Eq. (15) to take class imbalance into consideration. The F-measure in Eq. (14), which is the harmonic mean of precision and recall, may be used to assess the equation’s recall and precision.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (11)$$

$$Recall/Sensitivity = \frac{TP}{TP + FN} \quad (12)$$

$$Specificity = \frac{TN}{N} \quad (13)$$

$$F\ measure = \frac{2 * Precision * Recall}{Precision + Recall} \quad (14)$$

$$\begin{aligned} & \text{Mathews correlation coefficient (MCC)} \\ & = \frac{(TN * TP) - (FN * FP)}{((FP + TP) (FN + TP) (FP + TN) (FN + TN))^{0.5}} \end{aligned} \quad (15)$$

True positive outcomes are those that come in line with the prediction (TP). A very poor outcome is a true negative (TN). A false positive (FP) result is one that seems to be positive.

False Voice is the outcome of false Negative (FN). To evaluate the performance of the model, classification accuracy, precision, recall and F1-score metrics are included. We

compare the proposed DCNN-MCViT model against cutting-edge models.

By utilizing 9-fold cross-validation, the data were divided by 10%:90%, 20%:80%, 30%:70%, 40%:60%, 50%:50%, and 60 to evaluate the performance of the model. did. %: 40%, 70%: 30%, 80%: 20%, 90%: 10% (for testing and training) in Table 3.

Comparing the traditional methods like ResNet-50, AlexNet, VGG16, Inception-v3 based on the performance metrics like accuracy, precision, recall, and F1-score, suggested DCNN-MCViT performs better. Our complexity study shows that the proposed DCNN-VxT model can learn fewer parameters in terms of model size and number than conventional transform learning methods. The model prediction procedure is less complicated with a smaller set of learnable parameters and a lower model size. In contrast, the suggested DCNN-MCViT performs better in classifying plant leaf diseases compared to VGG16, ResNet 50, Inception v3 and AlexNet for PlantVillage and DiaMOS dataset is shown in the above table.

We further examined the overall crop stress discrimination and classification performance for individual and joint colour feature categories using AlexNet, VGG16, ResNet50, Inception V3, and the proposed DCNN-MCViT classifier. The below Table 4 shows the average classification accuracy for the DCNN-MCViT, Inception V3, ResNet50, VGG16, and AlexNet classifiers have shown the AVERAGE Accuracy values as mentioned in the below table for PlntVillage, Paddy crop and DiaMOS datasets.

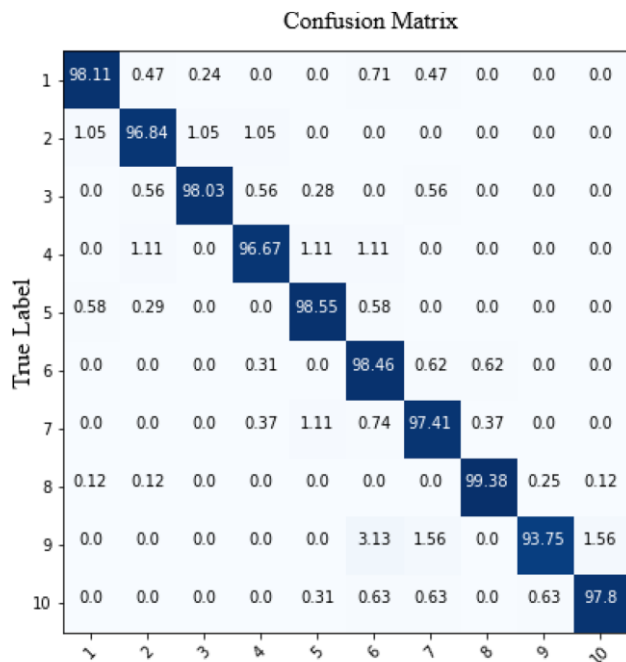
In order to regulate model learning, crucial learning parameters such as initial learning rate, validation frequency, number of epochs, and mini-batch size were set at 0.0001, 10, 30, and 32, respectively. The “ReLU” function allows all hidden layers while the “Softmax” function just allows the output layer. The network is tuned using a stochastic gradient descent method and a categorical cross-entropy algebraic loss function. The augmented picture dataset is used to train and evaluate the customized CNN model, as shown in Table 1.

**Table 3.** 9 fold cross validation of PlantVillage and DiaMOS dataset

Validation	Accuracy (%)		Sensitivity (%)		Specificity (%)		F1 (%)	
	PlantVillage	DiaMOS	PlantVillage	DiaMOS	PlantVillage	DiaMOS	PlantVillage	DiaMOS
10:90	98.13	98.12	96.78	99.82	99.24	99.82	98.49	98.39
20:80	98.46	99.05	99.45	99.07	99.43	99.07	98.58	99.08
30:70	97.13	97.17	99.46	97.34	99.34	97.34	97.27	97.25
40:60	97.08	97.14	98.42	97.17	98.17	97.17	97.23	97.20
50:50	96.84	97.28	96.38	98.28	96.40	98.28	96.84	97.83
60:40	98.38	98.90	98.35	98.10	98.31	98.10	98.35	98.49
70:30	98.06	97.92	99.02	97.14	97.15	97.14	98.08	98.12
80:20	98.09	99.30	97.56	98.56	98.57	98.56	98.05	99.19
90:10	96.13	99.91	93.72	98.64	98.66	98.64	96.15	99.85

**Table 4.** Average classification accuracy (%) across the PlantVillage, paddy, DiaMOS datasets

Model name	Avg Accuracy (%)			Avg Sensitivity (%)			Avg Specificity (%)			Total Training Time (s)		
	Plant Village	Plant VillagePaddy	DiaMOS	Plant Village	Plant VillagePaddy	DiaMOS	Plant Village	Palnt VillagePaddy	DiaMOS	PlantVillage	Plant VillagePaddy	DiaMOS
AlexNet	94.12	95.14	94.46	94.20	94.25	92.34	94.82	95.51	90.15	332.54	334.76	324.60
VGG16	96.91	96.36	96.32	95.43	96.21	94.78	94.83	95.69	93.67	328.42	330.24	329.42
ResNet50	97.51	97.85	97.16	97.35	95.74	97.40	95.78	93.24	95.83	325.21	326.76	327.67
Inception v3	98.63	98.36	98.36	98.24	97.67	98.12	97.48	95.90	96.36	324.90	323.76	324.18
DCNN-MCViT	99.51	99.78	99.82	98.87	98.92	99.26	98.56	99.12	98.76	321.06	320.76	321.79



**Figure 10.** Confusion matrix for plant disease detection-PlantVillage dataset

The success of the suggested model for the PlantVillage dataset is demonstrated by the confusion matrix in Figure 10 of the report. This makes it feasible to assess the performance of the model visually. Actual class output is indicated by the rows and columns. In contrast to non-diagonal cells, which reflect inaccurate observations, diagonal cells represent accurate observations.






### 4.3 Severity results

The results are intriguing since actual occupations take into account more forms of stress than those that are reported. There are several options for plant pathologists to measure crop stress. The quantification approach, which measures each stress level on a scale from 0 to 100% depending on how severe the stress is, is an extension of the categorization method. The next step in future research is thought to be the measurement of stress. Given the great degree of unpredictability in the external environment, this task is difficult and complex. The suggested approach can withstand fluctuations in outside illumination pretty well.






For healthy, and stressed classes, a severity level is assigned, where each level is set based on the percentage of the leaf area that is affected due. The stress severity was then categorized into five classes as no risk (0%), very low (1-20%), low (20-40%), medium (40-50%) and highest is (>50%).

The following are the experimental conditions: the task was initially done by evaluating the models to get the classification results, and the same was done using our suggested DCNN-MCViT model over training and validation data. Their respective best models were then preserved. Second, we used the DCNN-MCViT optimum model to test on the PlantVillage Stress Dataset and the DiaMOS Plant Dataset. The severity detection of the sample photos from the PlantVillage Stress Dataset and the DiaMOS Plant Dataset is shown in Tables 5 and 6.

**Table 5.** Biotic stress severity level of the PlantVillage stress dataset

Sample Image	Severity (% Range)
	40.0+ to 50.0
	70.0+ to 80.0
	70.0+ to 80.0
	40.0+ to 50.0
	60.0+ to 70.0

**Table 6.** Biotic stress severity level of the DiaMOS stress dataset

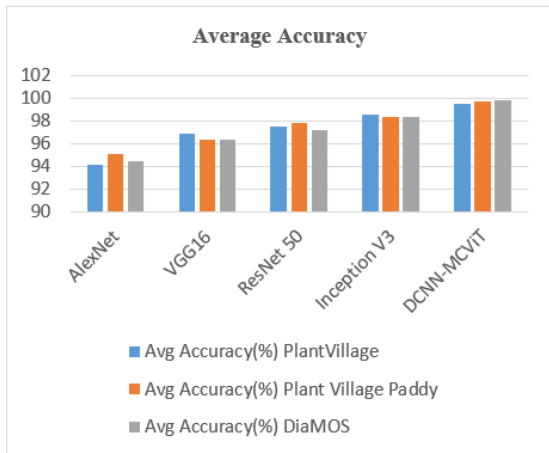
Sample Image	Severity (% Range)
	80.0+ to 90.0
	30.0+ to 40.0
	50.0+ to 60.0
	30.0+ to 40.0
	40.0+ to 50.0

**Table 7.** Comparison of the training time of PlantVillage dataset and DiaMOS dataset

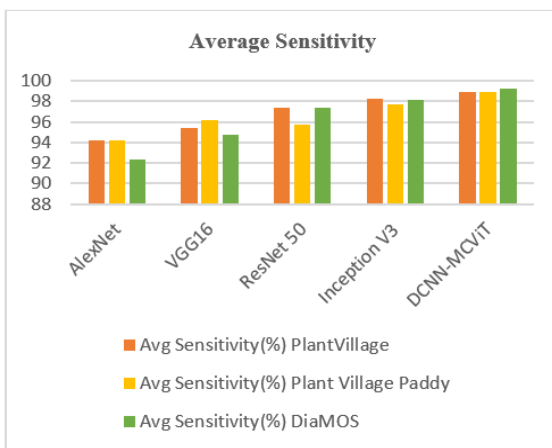
Model Name	Average Training Time (s)	
	PlantVillage Dataset	DiaMOS Dataset
AlexNet	321.10	245.28
VGG16	320.76	232.14
ResNet50	320.54	227.96
Inception V3	323.57	215.49
DCNN-MCViT	316.02	190.63

Further evidence for a model’s ranking comes from its computational efficiency or training time. Shorter time periods are required, which is predicted given the characteristics of networks built to fully use the resources of ResNet50 and DCNN-MCViT, followed by VGG-16, InceptionV3, and AlexNet. The average training time of the proposed method is about 316.02s whereas the conventional VGG16 training time took around 320.76s in the PlantVillage Dataset. In DiaMOS Dataset, the average training time is very less about 190.63s when compared to the other exiting methods in Table 7.

**4.4 Discussion**



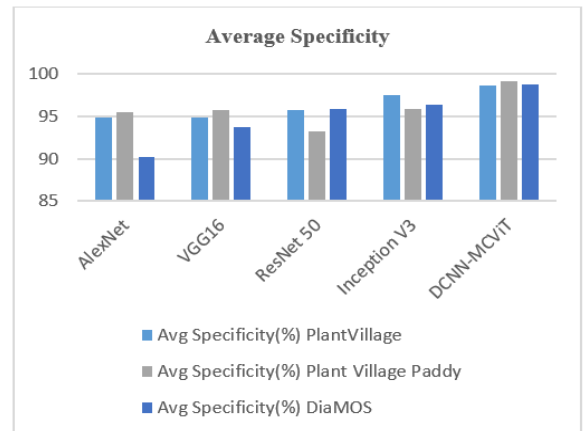
**Figure 11.** Average accuracy results



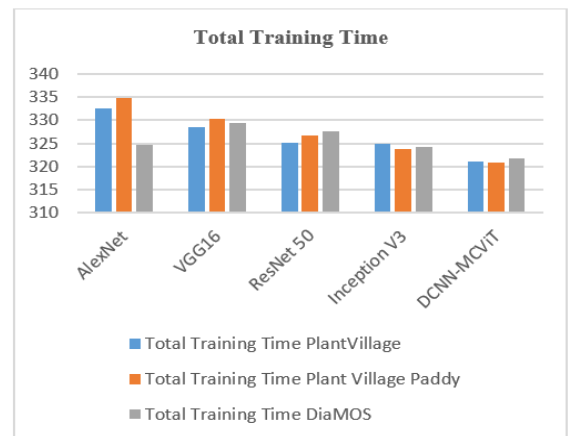
**Figure 12.** Average sensitivity results

In this work, we collaborated CNN with ViT to recognize and detect plant illnesses. The model combines the benefits of sensors and CNNs to enhance feature extraction performance and classification accuracy and the same has been shown in

Figures 11-14. The main effects of DL are in activation processes. This serves as a statistical “gate” between the output of the layer-skipping neuron and the input of the neuron that is now feeding it. A step function that, in response to a threshold or rule, switches neuron output on or off. It is crucial for the fusion of arbitrary linear models. When developing solutions to difficult issues, a variety of activation functions might enhance performance. Invisible layers between input and output are used in deep learning (DL), which makes it simpler to learn elaborate and complex patterns. Since there are more parameters to estimate, it needs more training data than other DL learning algorithms. The suggested methodology boosts productivity and unearths intricate patterns from multiple datasets on plant stress. The traditional architectures used for comparisons in this work are contemporary versions with attention mechanisms and lightweight constructions. Because each model has a unique method for identifying illness, comparing the findings is helpful. The model’s parameters are also used in the CrossViT model. ViT is more successful at managing attention than CNNs’ internal controls.



**Figure 13.** Average specificity results



**Figure 14.** Total training time

**5. CONCLUSION**

In our proposed work, we introduced a novel approach for biotic stress categorization and disease severity assessment using a multi-output image-based convolutional neural network. Our proposed procedure encompasses the entire workflow, starting from image acquisition to deep network

training and assessment. We incorporate several innovative techniques to improve the efficiency of the model.

A crucial aspect of our research is the integration of mean shifting clustering and active contour models for accurate plant leaf segmentation in natural environments. This segmentation technique enables the visualization of complex leaf structures and enhances the understanding of defected patterns. We evaluate the efficiency of segmentation through the automatic initialization of contour models and gradient flow calculations.

To automatically extract discriminative characteristics from damaged leaves, we utilize a shared architecture based on the multi-task learning paradigm. Our approach combines the strengths of Multi-scale Vision Transformers and traditional Deep Neural Networks, resulting in an architecture called MCViT. This architecture efficiently extracts both local and global features, enabling accurate stress analysis from the input images.

We evaluate our model's performance against various CNN models, and the EfficientNetB0 network and InceptionV3 network achieve the most promising outcomes. For the PlantVillage dataset, the hybrid DCNN-CrossViT (DCNN-MCViT) model achieves an accuracy of 99.51%, outperforming conventional methods such as InceptionV3, ResNet50, VGG16, and AlexNet classifiers. The average classification accuracy for the DCNN-CrossViT (DCNN-MCViT) model is 99.82% for the DiaMOS Plant Dataset.

To further enhance our model and improve its representation, we anticipate expanding the current dataset in future studies. This expansion will contribute to the model's correctness and robustness. Additionally, extending the interoperability of our research to include more factors responsible for biotic stress will contribute to a more comprehensive assessment of plant health.

In conclusion, our research provides a key analysis of stress detection and severity estimation by establishing a bridge between plant pathology and precision agriculture. By introducing novel techniques and architectures, we enhance the accuracy and efficiency of biotic stress categorization. We anticipate that our work will serve as a foundation for future studies and applications in the field of plant health assessment, aiding in the development of sustainable and efficient agricultural practices.

## REFERENCES

- [1] Friedrich, T. (2015). A new paradigm for feeding the world in 2050 the sustainable intensification of crop production. *Resource Magazine*, 22(2): 18.
- [2] Bock, C.H., Poole, G.H., Parker, P.E., Gottwald, T.R. (2010). Plant disease severity estimated visually, by digital photography and image analysis, and by hyperspectral imaging. *Critical Reviews in Plant Sciences*, 29(2): 59-107. <https://doi.org/10.1080/07352681003617285>
- [3] Barbedo, J.G.A., Koenigkan, L.V., Santos, T.T. (2016). Identifying multiple plant diseases using digital image processing. *Biosystems Engineering*, 147: 104-116. <https://doi.org/10.1016/j.biosystemseng.2016.03.012>
- [4] Wang, G., Sun, Y., Wang, J.X. (2017). Automatic image-based plant disease severity estimation using deep learning. *Computational Intelligence and Neuroscience*, 2017. <https://doi.org/10.1155/2017/2917536>
- [5] Liang, Q.K., Xiang, S., Hu, Y.C., Coppola, G., Zhang, D., Sun, W. (2019). PD2SE-Net: Computer-assisted plant disease diagnosis and severity estimation network. *Computers and Electronics in Agriculture*, 157: 518-529. <https://doi.org/10.1016/j.compag.2019.01.034>
- [6] Yu, S., Xie, L., Huang, Q.L. (2023). Inception convolutional vision transformers for plant disease identification. *Internet of Things*, 21: 100650. <https://doi.org/10.1016/j.iot.2022.100650>
- [7] Mohanty, S.P., Hughes, D.P., Salathé, M. (2016). Using deep learning for image-based plant disease detection. *Frontiers in Plant Science*, 7: 1419. <https://doi.org/10.3389/fpls.2016.01419>
- [8] Jiang, Z.C., Dong, Z.X., Jiang, W.P., Yang, Y.Z. (2021). Recognition of rice leaf diseases and wheat leaf diseases based on multi-task deep transfer learning. *Computers and Electronics in Agriculture*, 186: 106184. <https://doi.org/10.1016/j.compag.2021.106184>
- [9] Liu, Z., Lin, Y.T., Cao, Y., Hu, H., Wei, Y.X., Zhang, Z., Lin, S., Guo, B.N. (2021). Swin transformer: Hierarchical vision transformer using shifted windows. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, Montreal, QC, Canada, pp. 10012-10022. <https://doi.org/10.1109/ICCV48922.2021.00986>
- [10] Wiharto, Nashrullah, F.H., Suryani, E., Salamah, U., Prakisyia, N.P.T., Setyawan, S. (2021). Texture-based feature extraction using Gabor filters to detect diseases of tomato leaves. *Revue d'Intelligence Artificielle*, 35(4): 331-339. <https://doi.org/10.18280/ria.350408>
- [11] Saeed, F., Khan, M.A., Sharif, M., Mittal, M., Goyal, L.M., Roy, S. (2021). Deep neural network features fusion and selection based on PLS regression with an application for crops diseases classification. *Applied Soft Computing*, 103: 107164. <https://doi.org/10.1016/j.asoc.2021.107164>
- [12] Yang, D., Martinez, C., Visuña, L., Khandhar, H., Bhatt, C., Carretero, J. (2021). Detection and analysis of COVID-19 in medical images using deep learning techniques. *Scientific Reports*, 11(1): 19638. <https://doi.org/10.1038/s41598-021-99015-3>
- [13] Zhang, S.W., Zhang, S.B., Zhang, C.L., Wang, X.F., Shi, Y. (2019). Cucumber leaf disease identification with global pooling dilated convolutional neural network. *Computers and Electronics in Agriculture*, 162: 422-430. <https://doi.org/10.1016/j.compag.2019.03.012>
- [14] Thai, H.T., Le, K.H., Nguyen, N.L.T. (2023). FormerLeaf: an efficient vision transformer for cassava leaf disease detection. *Computers and Electronics in Agriculture*, 204: 107518. <https://doi.org/10.1016/j.compag.2022.107518>
- [15] Bi, C., Wang, J.M., Duan, Y.L., Fu, B.F., Kang, J.R., Shi, Y. (2022). MobileNet based apple leaf diseases identification. *Mobile Networks and Applications*, 1-9. <https://doi.org/10.1007/s11036-020-01640-1>
- [16] Anami, B.S., Malvade, N.N., Palaiah, S. (2020). Deep learning approach for recognition and classification of yield affecting paddy crop stresses using field images. *Artificial Intelligence in Agriculture*, 4: 12-20. <https://doi.org/10.1016/j.aiia.2020.03.001>
- [17] Khan, A., Sohail, A., Zahoor, U., Qureshi, A.S. (2020). A survey of the recent architectures of deep convolutional neural networks. *Artificial Intelligence Review*, 53: 5455-5516. <https://doi.org/10.1007/s10462->



- 020-09825-6
- [18] Luong, M.T., Pham, H., Manning, C.D. (2015). Effective approaches to attention-based neural machine translation. arXiv Preprint arXiv: 1508.04025. <https://doi.org/10.48550/arXiv.1508.04025>
- [19] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X.H., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Hounsby, N. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. arXiv Preprint arXiv: 2010.11929. <https://doi.org/10.48550/arXiv.2010.11929>
- [20] Neyshabur, B. (2020). Towards learning convolutions from scratch. *Advances in Neural Information Processing Systems*, 33: 8078-8088.
- [21] Gong, C.Y., Wang, D.L., Li, M., Chandra, V., Liu, Q. (2021). Vision transformers with patch diversification. arXiv Preprint arXiv: 2104.12753. <https://doi.org/10.48550/arXiv.2104.12753>
- [22] Zhao, Y., Sun, C., Xu, X., Chen, J.G. (2022). RIC-Net: A plant disease classification model based on the fusion of inception and residual structure and embedded attention mechanism. *Computers and Electronics in Agriculture*, 193: 106644. <https://doi.org/10.1016/j.compag.2021.106644>
- [23] Yuan, L., Chen, Y.P., Wang, T., Yu, W.H., Shi, Y.J., Jiang, Z.H., Tay, F.E.H., Feng, J.S., Yan, S.C. (2021). Tokens-to-token vit: Training vision transformers from scratch on imagenet. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, Montreal, QC, Canada, pp. 558-567. <https://doi.org/10.1109/ICCV48922.2021.00060>
- [24] Shen, Y., Zhu, S.J., Chen, C., Du, Q., Xiao, L., Chen, J.Y., Pan, D. (2020). Efficient deep learning of nonlocal features for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 59(7): 6029-6043. <https://doi.org/10.1109/TGRS.2020.3014286>
- [25] Chen, C.F.R., Fan, Q.F., Panda, R. (2021). Crossvit: Cross-attention multi-scale vision transformer for image classification. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, Montreal, QC, Canada, pp. 357-366. <https://doi.org/10.1109/ICCV48922.2021.00041>
- [26] Khan, M.A., Akram, T., Sharif, M., Awais, M., Javed, K., Ali, H., Saba, T. (2018). CCDF: Automatic system for segmentation and recognition of fruit crops diseases based on correlation coefficient and deep CNN features. *Computers and Electronics in Agriculture*, 155: 220-236. <https://doi.org/10.1016/j.compag.2018.10.013>
- [27] Kundu, N., Rani, G., Dhaka, V.S., Gupta, K., Nayak, S.C., Verma, S., Ijaz, M.F., Woźniak, M. (2021). IoT and interpretable machine learning based framework for disease prediction in pearl millet. *Sensors*, 21(16): 5386. <https://doi.org/10.3390/s21165386>
- [28] Chen, J.D., Chen, J.X., Zhang, D.F., Sun, Y.D., Nanekaran, Y.A. (2020). Using deep transfer learning for image-based plant disease identification. *Computers and Electronics in Agriculture*, 173: 105393. <https://doi.org/10.1016/j.compag.2020.105393>
- [29] Anami, B.S., Malvade, N.N., Palaiah, S. (2020). Classification of yield affecting biotic and abiotic paddy crop stresses using field images. *Information Processing in Agriculture*, 7(2): 272-285. <https://doi.org/10.1016/j.inpa.2019.08.005>
- [30] Zeng, W.H., Li, M. (2020). Crop leaf disease recognition based on self-attention convolutional neural network. *Computers and Electronics in Agriculture*, 172: 105341. <https://doi.org/10.1016/j.compag.2020.105341>
- [31] Lu, X.Y., Yang, R., Zhou, J., Jiao, J., Liu, F., Liu, Y.F., Su, B.F., Gu, P.W. (2022). A hybrid model of ghost-convolution enlightened transformer for effective diagnosis of grape leaf disease and pest. *Journal of King Saud University-Computer and Information Sciences*, 34(5): 1755-1767. <https://doi.org/10.1016/j.jksuci.2022.03.006>
- [32] Reedha, R., Dericquebourg, E., Canals, R., Hafiane, A. (2022). Transformer neural network for weed and crop classification of high resolution UAV images. *Remote Sensing*, 14(3): 592. <https://doi.org/10.3390/rs14030592>
- [33] Thakur, P.S., Khanna, P., Sheorey, T., Ojha, A. (2022). Explainable vision transformer enabled convolutional neural network for plant disease identification: PlantXViT. arXiv Preprint arXiv: 2207.07919. <https://doi.org/10.48550/arXiv.2207.07919>
- [34] Graham, B., El-Nouby, A., Touvron, H., Stock, P., Joulin, A., Jégou, H., Douze, M. (2021). LeViT: A vision transformer in convnet's clothing for faster inference. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, Montreal, QC, Canada, pp. 12259-12269. <https://doi.org/10.1109/ICCV48922.2021.01204>
- [35] Wang, P., Niu, T., Mao, Y.R., Zhang, Z., Liu, B., He, D.J. (2021). Identification of apple leaf diseases by improved deep convolutional neural networks with an attention mechanism. *Frontiers in Plant Science*, 12: 723294. <https://doi.org/10.3389/fpls.2021.723294>
- [36] Bi, C.K., Wang, J.M., Duan, Y.L., Fu, B.F., Kang, J.R., Shi, Y. (2022). MobileNet based apple leaf diseases identification. *Mobile Networks and Applications*, 27: 172-180. <https://doi.org/10.1007/s11036-020-01640-1>
- [37] Rangarajan, A.K., Purushothaman, R., Ramesh, A. (2018). Tomato crop disease classification using pre-trained deep learning algorithm. *Procedia Computer Science*, 133: 1040-1047. <https://doi.org/10.1016/j.procs.2018.07.070>
- [38] Fan, H.Q., Xiong, B., Mangalam, K., Li, Y.H., Yan, Z.C., Malik, J., Feichtenhofer, C. (2021). Multiscale vision transformers. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, Montreal, QC, Canada, pp. 6824-6835. <https://doi.org/10.1109/ICCV48922.2021.00675>
- [39] Sumari, P., Ahmad, W.M.A.W., Hadi, F., Mazlan, M., Liyana, N.A., Bello, R.W., Mohamed, A.S.A.M., Talib, A.Z. (2021). A precision agricultural application: Manggis fruit classification using hybrid deep learning. *Revue d'Intelligence Artificielle*, 35(5): 375-381. <https://doi.org/10.18280/ria.350503>
- [40] Vasavi, P., Punitha, A., Venkat Narayana Rao, T. (2023). Chili crop disease prediction using machine learning algorithms. *Revue d'Intelligence Artificielle*, 37(3): 727-732. <https://doi.org/10.18280/ria.37032>