IIETA International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Traitement du Signal
Vol. 43, No. 1, February, 2026, pp. 385-403
Journal homepage: http://iieta.org/journals/ts

Enhanced Crop Leaf Disease Classification Using Layer Freezing and Convolutional-Pooling Extensions

Anu M. Kiruthika[1*], Angelin Gladston[1], Khanna H. Nehemiah[2]

[1] Department of Computer Science and Engineering, College of Engineering, Guindy Anna University, Chennai 600025, India
[2] Ramanujan Computing Centre, College of Engineering Guindy, Anna University, Chennai 600025, India

Corresponding Author Email: rmtsanudoss@gmail.com

https://doi.org/10.18280/ts.430127

Received: 28 January 2025
Revised: 21 August 2025
Accepted: 16 October 2025
Available online: 28 February 2026

Keywords:
plant diseases, deep learning, selective layer freezing, classification, transfer learning

ABSTRACT

Artificial intelligence is increasingly applied in agriculture, with deep learning emerging as a promising tool to enhance productivity. Leaf diseases pose a major challenge, causing crop damage and economic loss. Timely leaf disease detection is crucial in precision agriculture. CNNs are widely applied in this domain, but small datasets often cause overfitting. Transfer learning resolves this by pretraining on large datasets such as ImageNet and refining the model on smaller agricultural datasets. This paper introduces Selective VGG19-Net, an adaptation of the VGG19 architecture for apple and citrus leaf disease classification. Unlike conventional transfer learning, our approach integrates two novel strategies: (i) selective freezing of Blocks 1 and 2 to retain transferable low-level edge and texture features, and (ii) convolutional-pooling depth extension to strengthen the network's capacity for learning complex, disease-relevant patterns. This dual design mitigates overfitting and improves representational capacity for limited agricultural data. Implemented in TensorFlow and fine-tuned on benchmark datasets, Selective VGG19-Net achieves 99.83% accuracy for apple and 98.03% accuracy for citrus disease classification, outperforming existing baselines. The model demonstrates high efficiency, offering a reliable tool for reducing crop loss through early disease detection.

1. INTRODUCTION

As the world's second-largest producer, India depends greatly on agriculture as an essential part of its economic structure. Farmers raise a diverse range of crops, including vegetables, millets, fruits, and more. Although plants are essential to all living things, they face numerous challenges, including the effects of environmental issues such as water scarcity and climate change on agricultural development. Fruit crops, known for their substantial health benefits, include renowned varieties such as apples [1] and citrus, as highlighted by Khattak et al. [2]. Unfortunately, these crops are sensitive to diseases that cause damage to plant leaves, resulting in production losses. As a result, early diagnosis of plant diseases is critical for maximising crop output [3].

Plant diseases are a significant factor causing damage to crop leaves. While farmers traditionally inspect their fields to evaluate plant growth, this process often leads to inaccuracies and requires considerable time. To address this issue, an autonomous system has been developed, particularly leveraging computer vision technology. This field of artificial intelligence extracts valuable information from extensive collections of images and videos, with broad applications spanning medical imaging, agriculture, transportation, and manufacturing.

In recent years, researchers have been actively investigating new approaches for autonomous disease diagnosis in agriculture, with the goal of overcoming difficulties and improving plant disease detection systems. Machine learning [4] is a prominent domain within computer science and, as noted by Yao et al. [5], is a subset of artificial intelligence that uses data and various algorithms to simulate human behaviour, providing solutions in various practical scenarios. Smart farming, a new agricultural strategy, incorporates many machine learning algorithms to improve performance, combining various technologies and high-performance computing devices to produce a smart farming system that benefits farmers [6].

In species selection, genes with attributes related to disease resistance and climate response are identified, and machine learning models predict the quality of these genes using extensive datasets. Beyond species selection, these models play vital roles in soil management, water management, crop management, and disease detection [7], including weed identification.

Support Vector Machine (SVM), Naive Bayes, Neural Networks, Decision Trees, and K-Nearest Neighbors (KNN) are some of the machine learning techniques used for classification in agriculture [8]. Current literature investigates both supervised and unsupervised machine learning methods for multiple crops. Hessane et al. [9] proposed a framework for classifying palm diseases, focusing on white scale disease. Their framework extracted features from coloured and grayscale images using GLCM and HSV to classify the disease into four separate classes: healthy, low infestation, medium infestation, and high infestation. For classification, they used SVM, KNN, Random Forest (RF), and Light Gradient Boosting Machine (LGBM), with the results validated using two datasets.

Deep learning is an advanced machine learning approach that holds promise in agriculture through image classification [10], speech recognition, and audio classification. This process is based on neural networks and has been successfully applied to solve real-world problems. Barburiceanu et al. [11] presented a deep learning approach to classify healthy leaves among 12 separate groups using data from a PlantVillage dataset. They implemented pre-trained models and machine learning techniques to extract features, and classification was then performed using AlexNet, VGGNet, and ResNet.

Shovon et al. [12] presented PlantDet, a model for classifying leaf diseases based on transfer learning models such as InceptionResNetV2, EfficientNet, and Xception. This model classified rice leaf diseases into five categories and involved several processes, including preprocessing, Global Average Pooling layers, a dropout mechanism, L2 regularization, and the PReLU activation function. The performance of the Xception method was evaluated using Grad-CAM and Score-CAM methods.

These improvements in machine learning and deep learning demonstrate strong potential for AI-centred technologies to transform plant disease detection, providing reliable, effective, and scalable solutions to some of the most critical challenges in agriculture.

1.1 Research contributions

1. Implemented a Selective VGG19-Net framework in which only Blocks 1 and 2 were frozen, preserving fundamental low-level features such as edges, textures, and color patterns from pretrained ImageNet weights, while higher layers were trained to specialize in disease-specific characteristics.
2. To deepen the network and enhance its hierarchical feature representation, a convolutional-pooling extension was introduced, enabling improved recognition of fine-grained variations and intricate disease patterns compared to the baseline VGG19.
3. Applied selective layer freezing and extended feature extraction, which reduced training issues such as overfitting and catastrophic forgetting, thereby improving performance.
4. Implemented and analyzed the Selective VGG19-Net using apple and citrus leaf disease datasets, and compared its performance with existing models, verifying its efficiency through accuracy-based evaluation metrics.

Consequently, our approach leads to improved performance of the VGG19 model. The paper is structured as follows: Section 2 presents a review of prior deep learning approaches for leaf disease detection. Section 3 outlines the classification models adopted in this study. Section 4 explains the system design of the proposed framework. Section 5 discusses the experimental results and performance evaluation. Section 6 compares the proposed Selective VGG19-Net with other existing approaches. Section 7 highlights the ablation study. The discussion of the experimental findings is presented in Section 8, followed by the conclusion of the Selective VGG19-Net work in Section 9.

2. LITERATURE REVIEW

The automation of leaf disease detection has attracted growing attention from the research community, driven by its significance in modern agriculture. Multiple studies have explored the automatic diagnosis of plant diseases, and although challenges remain, researchers continue to employ advanced image processing methods and deep learning architectures to enhance classification performance. Rajpoot et al. [13] have utilized deep learning techniques for the autonomous identification of leaf diseases in multiple crops.

For instance, Ahmad et al. [14] implemented a CNN-based approach for plant disease detection, specifically addressing class frequency issues in imbalanced datasets. Their method involved stepwise transfer learning to minimize the convergence time of the CNN architecture [15], and, to optimize for handheld devices, a MobileNet model without a dense layer was employed. Another notable study by Zhou et al. [16] focused on grape leaf diseases and introduced a data augmentation method surpassing previous Generative Adversarial Networks (GANs). By integrating a faster CNN with a fine-grained GAN, the model accuracy of grape leaf disease classification was improved, particularly in identifying local spot areas through data augmentation.

Liu et al. [17] implemented a lightweight CNN for better fine-grained classification performance. SqueezeNet, used in this context, employed a multiscale convolution kernel and a coordinate technique to identify features. Experiments were performed on the plant disease recognition dataset from the 2018 AI Challenger challenge to assess the proposed approach. Yu et al. [18] proposed a hybrid approach that integrates K-means clustering with deep learning for maize leaf disease prediction. In this method, K-means clustering was employed to segment diseased regions in maize leaves, and the segmented outputs were then fed into a deep learning model for classification. The effectiveness of the approach was evaluated against established models, including ResNet and VGG19.

Bukhari et al. [19] developed datasets for wheat stripe rust and applied various segmentation techniques, such as Watershed, GrabCut, and U2-Net. The influence of segmentation on classification performance was assessed using ResNet-18 as the deep learning backbone. Cap et al. [20] developed LeafGAN, a data augmentation technique employing an attention mechanism to produce distinct diseased images from healthy ones. The LeafGAN model produced images of higher visual quality than the standard CycleGAN, thereby improving the effectiveness of plant disease diagnosis. Wu et al. [21] developed a fine-grained model for identifying diseases in peach and tomato leaves using an attentional deep neural network. During training, a Reconstruction and Generation Model was used to discriminate global features, and the proposed model performed well with minimal memory and computational resource requirements.

Stephen et al. [22] proposed a unique strategy that uses both 3D and 2D Convolutional Neural Networks (CNNs) to extract rich information important for classification tasks. Their methodology included an Improved Backtracking Algorithm (IBS) combined with Generative Adversarial Networks (GANs) to improve classification accuracy. Notably, the settings of the IBS method were carefully chosen to solve typical problems such as instability, vanishing gradients, and model collapse. By combining the power of 3D and 2D CNNs

[23], the approach efficiently gathered varied color and texture information required for disease detection. This optimized GAN framework effectively detected three common rice diseases—bacterial blight, leaf blast, and brown spot—as well as healthy crops, with an impressive accuracy rate of 98.7%.

Elfatimi et al. [24] proposed a network architecture designed to accurately classify angular leaf spots and bean rust in bean leaves. The model demonstrated accurate recognition of diseases in their early stages, utilizing the MobileNet architecture for disease identification on a public dataset. By fine-tuning key parameters, including the optimizer, batch size, and learning rate, the model achieved superior performance over conventional approaches.

The AgriDet framework [25] was developed to tackle issues related to unclear imagery and background interference in plant disease detection based on image processing methods. The INC-VGGN neural network architecture, based on the Inception-Visual Geometry Group Network, played a crucial role. Preprocessing techniques were applied to alleviate issues during image capture, and the GrabCut technique was employed to address occlusion problems. The Kohonen-based deep learning algorithm successfully identified multiscale features in plant leaves. Singh et al. [26] presented a CNN-based approach for identifying several types of diseases in cotton leaves, including fungal, viral, and bacterial infections.

The model involved two processes: identifying diseased leaves and classifying them into their respective disease types. The CNN-based cotton leaf disease prediction model [27] exhibited good performance and computational efficiency. Shafi et al. [28] focused on wheat rust, dividing it into four stages: moderate, resistant, healthy, and susceptible. U-Net was used to eliminate undesirable background features and extract rust disease-related features. The impact of stripe rust disease in wheat was determined using the Xception model and ResNet-50. Intelligent edge detection devices were also employed to monitor rust disease-affected wheat crops. Sunil et al. [29] obtained a cardamom plant leaf disease dataset in Chinnahalli, Sakaleshpur, India, and utilized the U-Net architecture to remove the background for disease detection using EfficientNetV2. The proposed system was tested on grape plant leaves. Liu et al. [30] introduced PiTLid, an apple detection approach based on Inception-v3 transfer learning. This strategy addressed the challenge of insufficient training data. The apple dataset included four classes: black rot, rust, scab, and healthy. PiTLid outperformed other phenotyping models in classification efficiency and demonstrated applicability with limited grape and peach data.

While deep learning has made progress in leaf disease classification, major challenges remain. CNN-based approaches, such as fine-tuned VGG19, ResNet, and MobileNet, frequently overfit small agricultural datasets. Traditional transfer learning algorithms either freeze layers randomly or fine-tune entire networks, potentially resulting in the loss of transferable low-level features. Many architectures also lack focused adjustments to capture complex, disease-specific patterns, resulting in incorrect identification, and some require considerable preprocessing, increasing computational costs and reducing practical efficiency. Moreover, fine-tuning pretrained models necessitates careful selection and adjustment of hyperparameters, such as learning rate, batch size, and optimizer, to optimize performance effectively.

To address these challenges, we propose Selective VGG19-Net, which includes selective layer freezing of Blocks 1 and 2 to preserve crucial low-level features while allowing deeper layers to adapt, as well as a convolutional-pooling depth extension to capture hierarchical, disease-specific patterns.

3. MATERIALS AND METHODS

3.1 Dataset collection

For disease classification, dataset is fundamental and consists of two datasets that are both publicly available, as summarized in Table 1. In the first dataset, the source was a free data source web site “data. mendeley,” which is for citrus. The dataset contains 759 images in total, consisting of one healthy class and four diseased classes, with 730 JPG images depicting black spot, canker, greening, and melanose. All images are 259 × 25 at 72 dots per inch. The second dataset was a free open source "Kaggle" dataset under the title "plant pathology". The dataset contains nearly 125k images in JPEG format, covering ten plant species: tomato, grape, apple, citrus, strawberry, bell pepper, peach, potato, cherry, and corn. Within the apple category, there is one healthy class and three diseased classes, namely apple rust, black rot, and cedar apple rust.

Table 1. Summary of apple and citrus leaf dataset

Type of Leaf	No. of Training Images	No. of Testing Images	No. of Validation Images	Total Images
Apple	4196	468	597	5261
Citrus	609	53	68	730

3.2 Classification models

This paper introduces a Selective VGG19-Net model based on transfer learning for the classification of leaf diseases in citrus and apple leaves.

3.2.1 Convolutional Neural Network

Many techniques based on deep learning are utilised to train neural network algorithms for classification problems. Figure 1 shows a CNN, a deep learning model commonly used for object classification tasks. The CNN [31] accepts images as input and automatically analyses Red, Green, and Blue (RGB) images. The CNN model has several layers, including the input, convolutional, pooling, fully connected, and output layers. The first layer processes the image matrix m, represented as $m \in R^{a*b*c}$, where a is the height, b is the width, and c is the number of RGB channels. In the convolutional layers, let k be the total amount of filters, and the output feature c_o is collected through the convolution operation $c_o = \sigma\left(\sum_i m * k_j\ t_i\right)$, where $m * k_j$ is the convolution with kernel k_j, t_i is the bias, and σ is the activation function. The pooling layers perform pooling operations denoted as p_l, where $p_i c_o$ represents the pooled feature maps connected operation is expressed as $\sigma\left(\sum w_{ij}\ f_c + t_i\right)$, where w_{ij} are the weights, f_c are the inputs t_i is the bias, and σ is the activation function. The output layer is indicated as o_n, where $o_n = \sigma\left(\sum w_{ij} * f_{c_j} + t_i\right)$, demonstrating n-th neuron output.

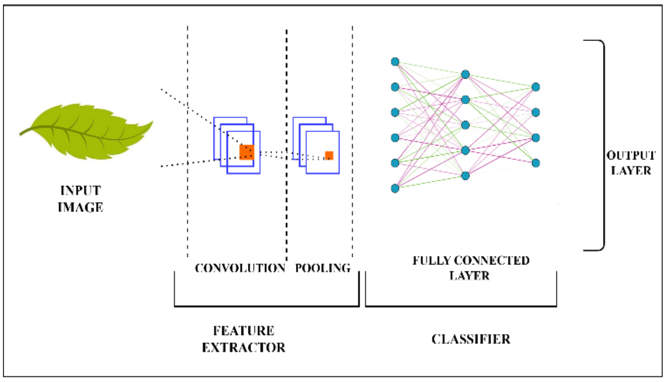


Figure 1. Convolutional neural network

3.2.2 Transfer learning

Figure 2 demonstrates transfer learning, a method frequently applied in computer vision tasks. It involves applying a pre-existing model to tackle a novel problem. This entails leveraging the insights gained from the trained model to address various but closely related issues. The core concept behind transfer learning is to utilize knowledge gathered from a model trained on an extensive dataset to effectively address a different problem with limited data. This eliminates the need to retrain the entire process from the scratch. The advantages of employing this approach encompass reduced training time, enhanced performance, and a diminished need for extensive data.

3.2.3 VGG19

The VGG family of networks, designed by Simonyan and Zisserman in 2014, introduced a deeper convolutional neural network structure for visual recognition tasks, as illustrated in Figure 3. Among its variants, VGG19 is widely used in image classification and detection problems. This model receives 224 × 224 pixel color images and passes them through 16 convolutional layers, with 3 × 3 filters capturing hierarchical feature representations. Each convolutional block is followed by ReLU activation to introduce non-linearity and 2 × 2 max pooling to reduce spatial resolution while retaining significant patterns. After feature extraction, the representations are passed to three fully connected layers, each containing 4096 neurons, which further refine the learned features. At the final stage, the output layer employs a softmax function to provide probability scores for all target classes. The model is trained using backpropagation and gradient descent optimization techniques, which iteratively adjust the network's weights to improve performance.

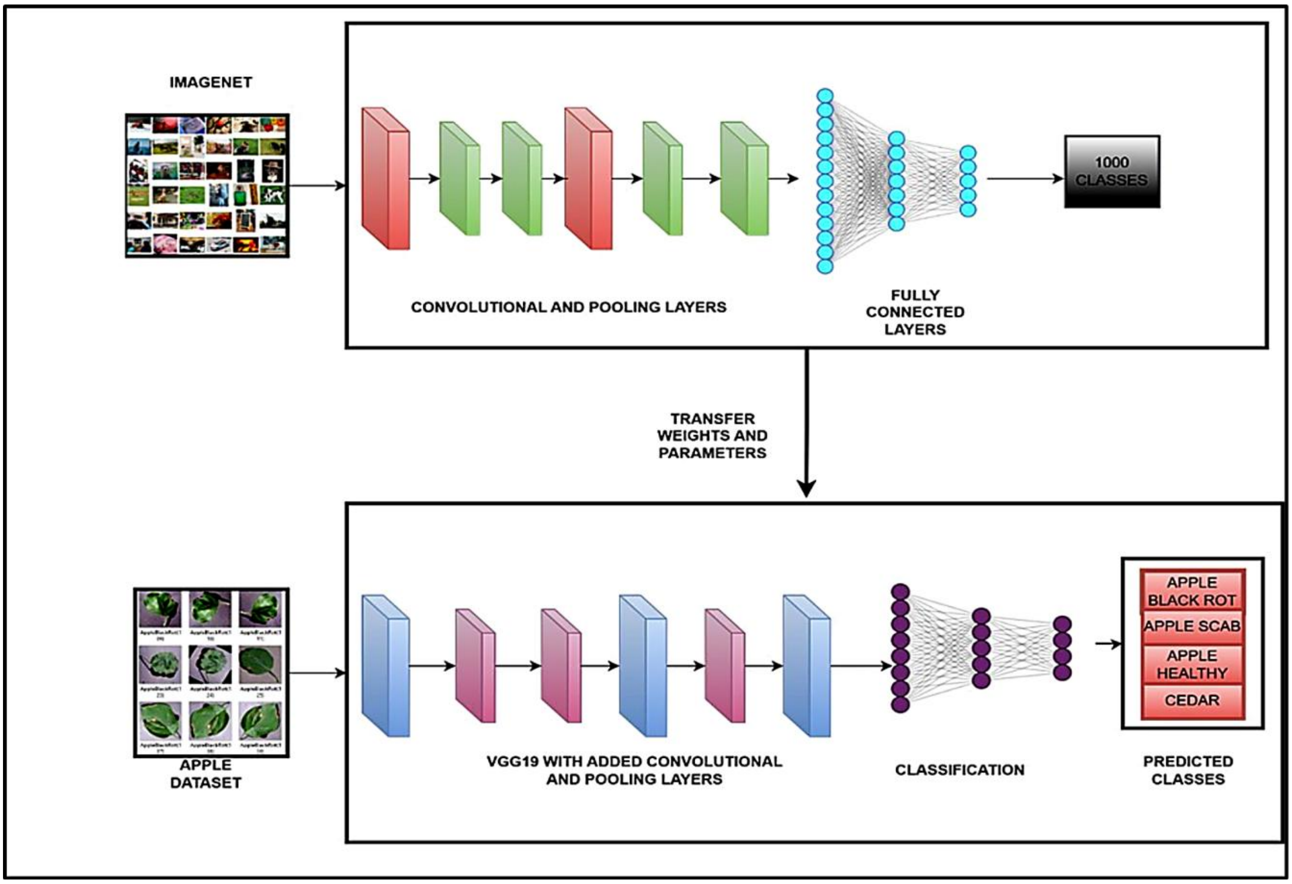


Figure 2. Overview diagram illustrating the transfer learning

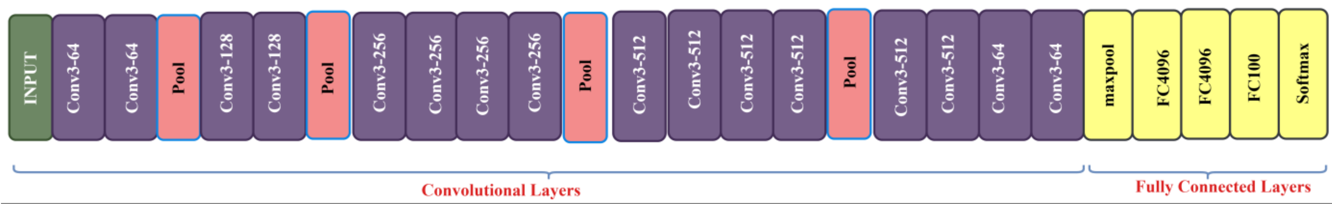


Figure 3. Systematic view of VGG19

4. PROPOSED SELECTIVE VGG19-NET FRAMEWORK

In deep learning, transfer learning is an important technique that applies the knowledge gained by pretrained models to solve different tasks [32]. This approach eliminates the need to develop models from scratch and instead capitalizes on features learned from extensive datasets like ImageNet. Through fine-tuning or feature extraction, transfer learning can attain high accuracy across diverse domains, even when training data is limited. These models are particularly advantageous for scenarios with limited labelled data and computational resources.

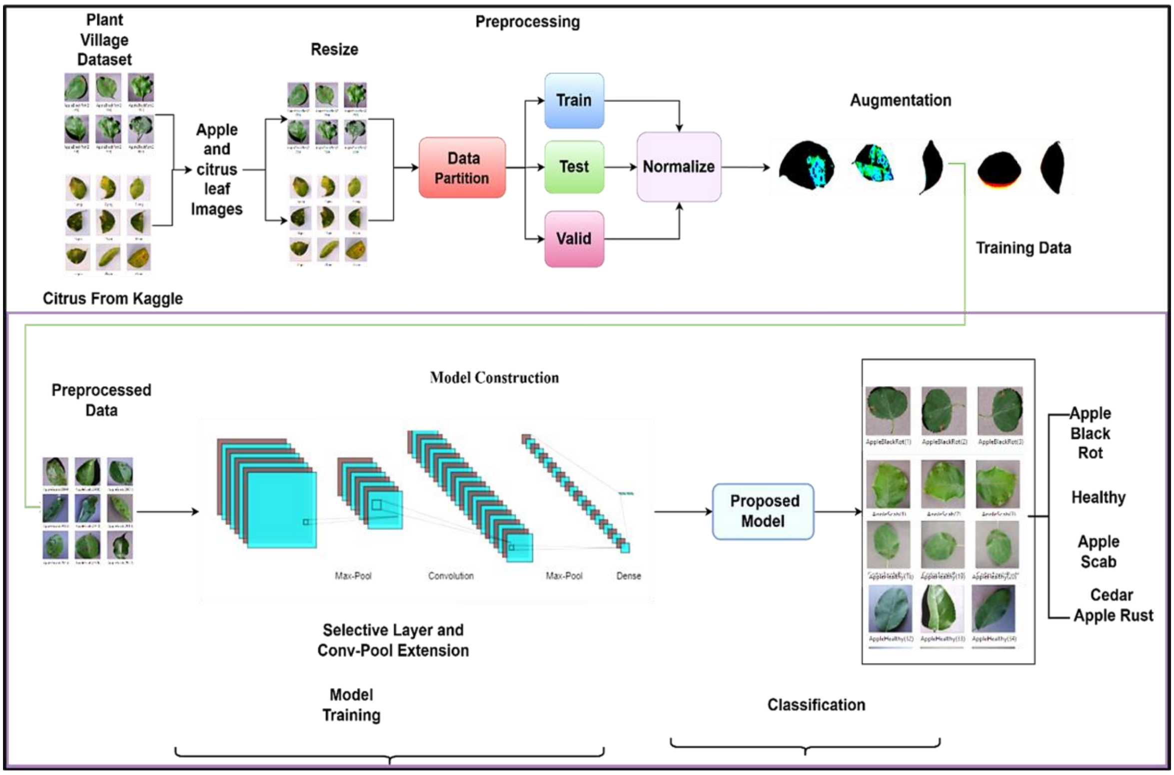


Figure 4. Workflow of proposed architecture for apple and citrus disease classification

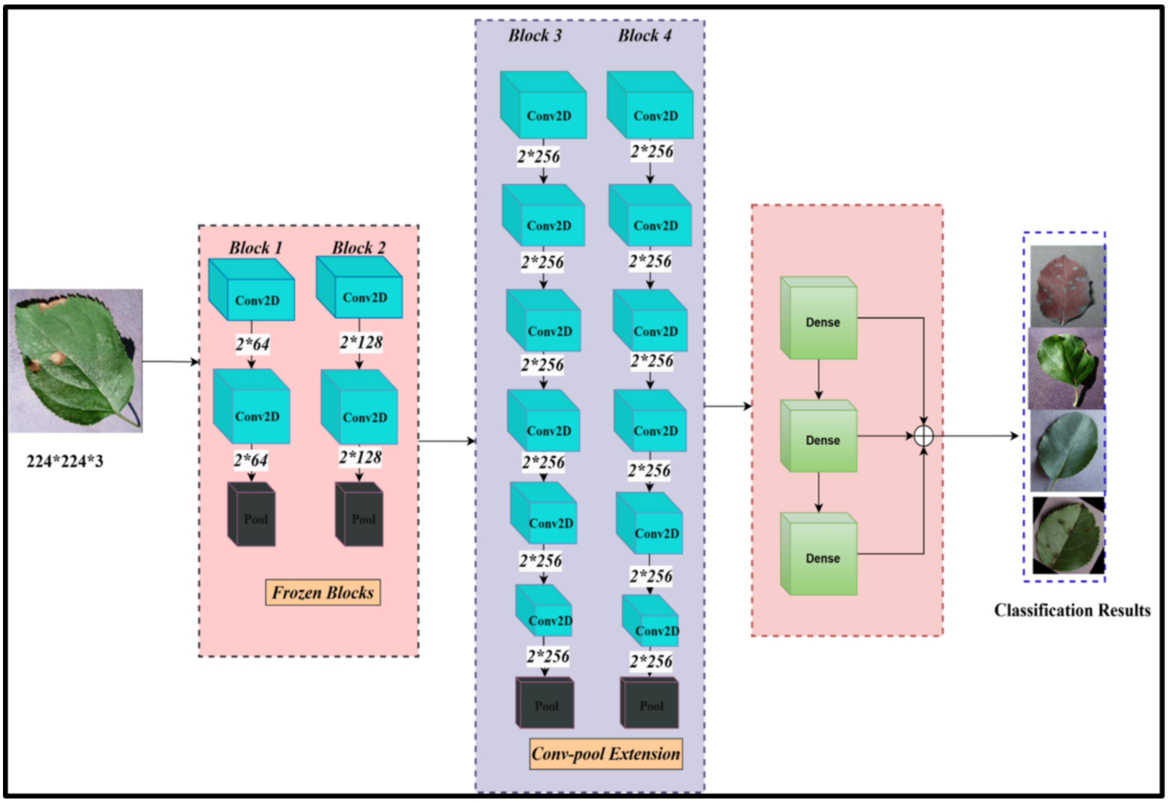


Figure 5. Block structure of Selective VGG19-Net

As illustrated in Figure 4 our study applies transfer learning technique for classifying leaf diseases. we started by collecting a dataset of apple and citrus images from two public repositories and resizing them to 224 × 224 × 3. Divide the dataset into three splits: train, test, and valid set, in the proportions of 70%, 20%, and 10%. Rotation, vertical flip, and horizontal flip are examples of augmentation procedures that improve the dataset. The training data was then merged with some augmented images to develop our proposed Selective VGG19-Net model. Our model involves selectively freezing layers in Block 1 and Block 2 of the VGG19 architecture. By selectively freezing layers, we retain essential low-level features from pretrained weights. Additionally, we enhance the frozen VGG19 model by introducing an extra convolutional and pooling layer as shown in Figure 5, increasing the network's depth to capture intricate patterns and hierarchical representations of leaf diseases. Experimental results demonstrate that our method outperforms other fine-tuning approaches, achieving superior classification accuracy. By combining selective layer freezing and extended feature extraction, the model achieves improved identification of disease-relevant features. To evaluate our Selective VGG19-Net, we compare it to CNN-based pretrained models on datasets containing apple and citrus images.

4.1 Preprocessing

Model training begins with preprocessing, which is the act of preparing a set of input images for further analysis. In this work, both citrus and apple leaf datasets were utilized, and several pixel-level adjustments, such as scaling and normalization, were applied to every image. To further expand the dataset, several augmentation techniques were applied, including image rotation, horizontal and vertical flipping, shifting, and shearing. These methods increased the dataset size, reduced overfitting, and improved the model's robustness to variations in input data. Consequently, the preprocessed images were better suited for efficient model training and evaluation. Figure 6 presents examples of preprocessed images from the apple dataset, while Figure 7 shows preprocessed images from the citrus dataset. Table 2 compares dataset samples before and after augmentation, illustrating how preprocessing enhanced dataset diversity.

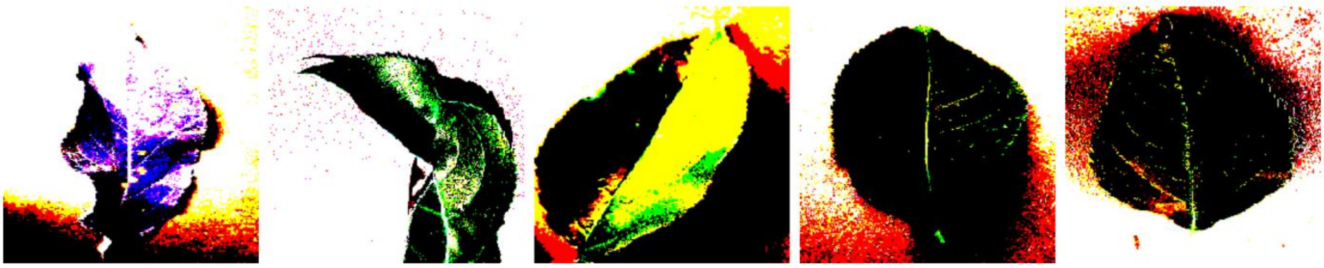


Figure 6. Samples of preprocessed images of apple dataset with augmentation techniques

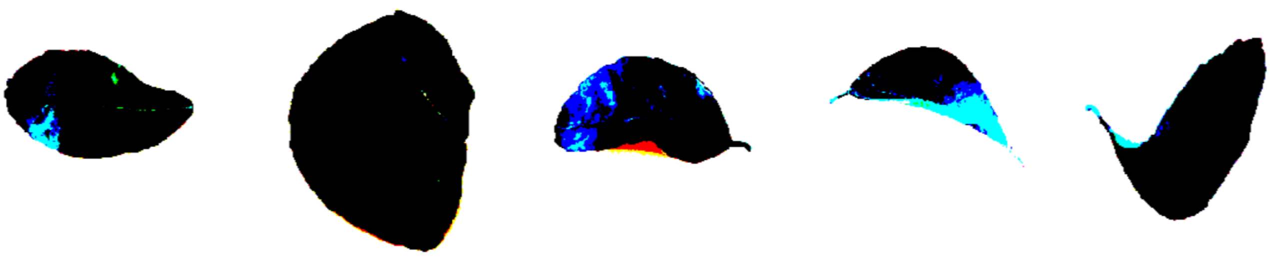


Figure 7. Samples of preprocessed images of citrus dataset with augmentation techniques

Table 2. Augmentation results on apple and citrus leaf images

Type of Leaf	Training Images (Original)	Training Images (After Augmentation)	Testing Images	Validation Images	Total Images (Augmented)
Apple	4196	8380	468	597	9445
Citrus	609	3000	53	68	3121

Algorithm 1: Selective Layer Freezing

Objective:

Preserve critical low-level features from pre-trained weights while allowing specific layers to adapt to the target dataset.

Input:

- Pre-trained model M_{pre} (e.g., VGG19).
- Target layers to unfreeze L_{unfreeze} (e.g., Block 1 and Block 2).

Output:

- Modified model M_{freeze} with selective layer freezing.

Steps:

1. **Load Pre-trained Model:**
Initialize the base model with pre-trained weights:

$$M_{\text{base}} = \text{LoadModel}\left(M_{\text{pre}}, \text{include\_top} = \text{False}\right)$$

2. **Freeze All Layers:**

Set the trainable attribute of all layers to False:

$$\text{Trainable}(L_k) = \text{False}, \quad \forall k \in \{1, \dots, L\}$$

where, L is the total number of layers in M_{base}.

3. **Unfreeze Selected Layers:**
Identify layers in L_{unfreeze} (e.g., Block 1 and Block 2) and set their trainable attribute to True:

$$\text{Trainable}(L_k) = \text{True}, \quad \forall k \in L_{\text{unfreeze}}$$

4. **Compile the Model:**
Prepare the model for fine-tuning:

$$M_{\text{freeze}} = M_{\text{base}}$$

Algorithm 2: Convolutional-Pooling Extension Approach

Objective:
Enhance the model's ability to capture complex patterns by adding additional convolutional and pooling layers.

Input:
- Modified model M_{freeze} from Algorithm 1.
- Additional layers f_{conv} (convolutional), f_{pool} (pooling).
- Dropout rate p.

Output:
- Extended model M_{extended}.

Steps:

1. **Define Additional Layers:**
   - Add a convolutional layer with filter size $K \times K$, stride s, and activation function σ:
   $$f_{\text{conv}} = \text{Conv2D}(K, s, \sigma)$$
   - Add a pooling layer (e.g., MaxPooling):
   $$f_{\text{pool}} = \text{MaxPooling2D}(P)$$

2. **Extend the Base Model:**
Append the new layers to the base model:

$$M_{\text{extended}} = M_{\text{freeze}} + f_{\text{conv}} + f_{\text{pool}}$$

3. **Apply Regularization:**
Add dropout regularization with probability p:

$$f_{\text{dropout}} = \text{Dropout}(f_{\text{fc}}, p)$$

where, f_{fc} represents the fully connected layers.

4. **Add Classification Layers:**
Flatten the output of the extended layers:

$$f_{\text{flatten}} = \text{Flatten}(M_{\text{extended}})$$

Add dense layers and softmax activation for classification:

$$\hat{y}_i = \text{Softmax}(W \cdot f_{\text{flatten}} + b)$$

5. **Compile the Extended Model:**
Prepare the final extended model:

$$M_{extended} = Compile(M_{freeze} + f_{conv} + f_{pool} + f_{dropout})$$

4.2 Selective layer freezing

In our research, we present an approach to fine-tuning a pretrained model, specifically a Selective VGG19-Net tailored for leaf disease classification. Our approach includes strategically freezing certain blocks of layers to optimize the use of pre-existing knowledge while allowing adaptation to the specific task. Additionally, the architecture was extended with extra convolutional-pooling layers to enhance its ability to capture critical features essential for leaf disease classification. A central aspect of the approach involves selectively freezing specific layers, particularly Blocks 1 and 2 during training, as illustrated in Figure 7. By doing so, aim to mitigate overfitting and prevent catastrophic forgetting. Let l_f is the loss function and ∇ be the gradient, the weight update rule during training, w_i is the old weight, w_j is the new weight,

$$w_i \rightarrow w_j - \eta \nabla w_j l_f \quad (1)$$

When Layer frozen then,

$$\nabla_{wf l_f} = 0, w_{f-i} = -w_{f-j} - \eta . 0 = w_{f_j} \quad (2)$$

As expressed in Eqs. (1) and (2), Blocks 1 and 2 are chosen for freezing because they primarily learn generic and low-level features, including edges, textures, and basic shapes. These features prove crucial for effective leaf disease classification.

The selective layer freezing not only facilitates the retention of essential knowledge from the pretrained model but also ensures that the model can adapt specifically to the nuances of leaf disease characteristics.

4.3 Convolutional-pooling extension approach

In section 4.3, we introduce a convolutional-pooling extension strategy to enhance the VGG19 architecture, focusing on specific blocks for improved performance in capturing intricate patterns within leaf images. Blocks 1 and 2 consist of two convolutional layers each, activated by ReLU functions, and are followed by a max pooling layer. Notably, these are made freeze only to these blocks, allowing flexibility in the learning process. In our model, we have made enhancements to Blocks 3 and 4 of the VGG19 architecture. Specifically, added two additional convolutional layers to Block 3 and one additional layer to Block 4. Additionally, we have removed Block 5 entirely from the base architecture and these modifications are illustrated in Figure 7. This extension aims to enhance the network's capacity, resulting in superior performance, particularly in handling complex features in leaf images.

Let x be the input. The output of Block 1 is obtained as shown in Eq. (3):

$$z_{block1} = \sigma(w_{conv1} * x b_{conv1}) \quad (3)$$

The output of Block 2 is then computed as expressed in Eq. (4):

$$z_{block2} = maxpool(\sigma(w_{conv2} * z_{block1} + b_{conv2}) \quad (4)$$

Similarly, the extended Block 3 produces the output feature maps given in Eq. (5):

$$z_{block3} = maxpool(\sigma(w_{conv3} * (\sigma(w_{conv2} * y_{block1} + b_{conv2})) + b_{conv3})) \quad (5)$$

Finally, the output of Block 4 is derived as shown in Eq. (6):

$$z_{block4} = maxpool(\sigma(w_{conv4} * z_{block3} + b_{conv4})) \qquad (6)$$

where, z_{block4} is the output feature of blocks, convolution layer represented by w_{conv1} and b_{conv1}, here w is the convolutional filter and b is the bias.

4.4 Computational complexity

The SelectiveVGG19-Net incorporates multiple convolutional blocks with filter depths progressively scaled to 64, 128, 256, and 512, along with max-pooling layers that compress the feature maps to reduce dimensionality. In the final stage, the network transitions into a fully connected architecture with three dense layers. While the convolutional operations capture hierarchical feature representations, the pooling strategy minimizes spatial resolution and computational overhead in the deeper layers. Although the deep layers add a significant number of parameters, the overall design strikes a compromise between feature richness and computational efficiency. The network includes 74,583,109 trainable parameters, requires 298 MB of weights, and can do inference on a single image in 0.3 to 0.4 seconds on a GPU, making it suitable for agricultural applications. Additional efficiency benefits can be achieved by selective layer freezing, pruning, replacing dense layers with global average pooling.

Algorithm 3: Selective Layer Freezing and layer Extension for Leaf Disease Classification

Input:

- Image dataset $D = \{(x_i, y_i)\}_{i=1}^{N}$, where $x_i \in \mathbb{R}^{224\times224\times3}$ (leaf images) and $y_i \in C$ (class labels), with C being the set of disease classes.
- Pre-trained VGG19 model M_{pre}.
- Learning rate η, number of epochs E, and batch size B.

Output:

- Trained model M_{final} for accurate classification of leaf diseases.

<u>Steps:</u>

1. Data Preparation:

Split the dataset D into:

- Training set D_{train},
- Validation set D_{val},
- Testing set D_{test}.

Resize each image x_i in D to $224 \times 224 \times 3$:

$$x_i' = \text{Resize}(x_i, 224,224,3).$$

2. Data Augmentation:

Apply the following transformations to each x_i':

- **Horizontal Flip**:

$$x_i'' = \begin{cases} \text{Flip}(x_i', \text{horizontal}), & \text{with probability 0.5,} \\ x_i', & \text{otherwise.} \end{cases}$$

- **Vertical Flip**:

$$x_i'' = \text{Flip}(x_i', \text{vertical}).$$

- **Rotation**:

$$x_i'' = \text{Rotate}(x_i', \theta), \quad \theta \sim \text{Uniform}(-45°, 45°).$$

- **Zoom**:

$$x_i'' = \text{Zoom}(x_i', z), \quad z \sim \text{Uniform}(1 - 0.2, 1 + 0.2).$$

- **Translation** (width and height shifts):

$$x_i'' = \text{Translate}(x_i', \Delta w, \Delta h), \quad \Delta w, \Delta h \sim \text{Uniform}(-0.1, 0.1).$$

- **Shear**:

$$x_i'' = \text{Shear}(x_i', \alpha), \quad \alpha \sim \text{Uniform}(-0.2, 0.2).$$

- **Brightness Adjustment**:

$$x_i'' = \text{AdjustBrightness}(x_i', \beta), \quad \beta \sim [0.5, 1.5].$$

3. Batch Generation:

Divide D_{train} into batches of size B:

$$B = \{B_j\}_{j=1}^{\lceil N_{\text{train}}/B \rceil}, \quad B_j = \{(x_i, y_i)\}_{i=1}^{B}.$$

4. Model Initialization:

- Load the pre-trained VGG19 model M_{pre}:

$$M_{\text{base}} = \text{LoadModel}(M_{\text{pre}}, \text{include\_top} = \text{False}).$$

- Freeze all layers:

$$\text{Trainable}(L_k) = \text{False}, \quad \forall k \in \{1, \dots, L\}.$$

5. Selective Layer Freezing:

Unfreeze layers in Block 1 and Block 2:

$$\text{Trainable}(L_k) = \text{True}, \quad \forall k \in \{\text{Block 1, Block 2}\}.$$

6. Model Extension:

1. Add new convolutional (f_{conv}) and pooling (f_{pool}) layers:

$$M_{\text{extended}} = M_{\text{base}} + f_{\text{conv}} + f_{\text{pool}}.$$

2. Apply dropout ($p = 0.5$) to fully connected layers:

$$f_{\text{dropout}} = \text{Dropout}(f_{\text{fc}}, p).$$

7. Model Compilation:

Use softmax activation for multi-class classification:

$$\hat{y}_i = \text{Softmax}(z), \quad z = M_{\text{extended}}(x_i).$$

- Compile the model with categorical cross-entropy loss:

$$L_{\text{CCE}} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{|C|} y_{i,c} \log \hat{y}_{i,c}.$$

8. Training Process:

Train the model for E epochs:

$$M_{\text{final}} = \text{Train}(M_{\text{extended}}, B, \eta, E).$$

9. Evaluation:
Evaluate the model on the test set D_{test}:

$$\text{Metrics} = \{\text{Accuracy},\text{Precision, Recall, F1-Score}\}.$$

10. Result Analysis:
Compute class-wise metrics for each class $c \in C$:

- **Precision**:

$$\text{Precision}_c = \frac{\text{TP}_c}{\text{TP}_c+\text{FP}_c}.$$

- **Recall**:

$$\text{Recall}_c = \frac{\text{TP}_c}{\text{TP}_c+\text{FN}_c}.$$

5. RESULTS AND ANALYSIS

5.1 Dataset

Four apple diseases are employed in the work: apple rust, black rot, cedar apple rust, and healthy, and five citrus leaf categories, including four disease types black spot, canker, greening, and melanoses along with healthy leaves. Figure 8 and Figure 9 show images samples from each dataset. These datasets are the most comprehensive and well-known open-source datasets, kaggle and data mendeley, which encompass various types of plant leaf diseases and have served as benchmark datasets in recent research. The proposed model makes use of 759 images for citrus plant leaves and 5261 images for apple disease. The citrus leaf dimensions 259×25.

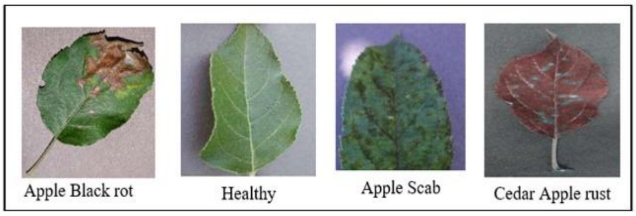


Figure 8. Visual examples of apple leaf images

Figure 9. Sample images depicting citrus leaves

5.2 Experimental setup

Experiments were conducted using Google Colab with Python and an open-source deep learning framework. The system was configured with Windows 11 and powered by an Intel Xeon W-2133 processor. The dataset was divided into 80 percent for training, 10 percent for validation, and 10 percent for testing. All images were resized to 224 by 224 by 3 pixels to meet the VGG19 input requirements. The model was trained using the Adam optimizer with a learning rate of 0.0001, a batch size of 32, and 30 epochs. Categorical cross-entropy was used as the loss function for multi-class classification, with ReLU activation in the hidden layers and softmax activation in the output layer. Model performance was evaluated using accuracy, precision, recall, and F1-score. The hyperparameters used in the experiments are summarized in Table 3.

Table 3. Training parameters utilized in the Selective VGG19 approach

Hyperparameter	Value
Image size	224 × 224 × 3
Learning rate	0.0001
Batch size	32
Number of epochs	30
Loss function	Categorical cross-entropy
Activation	ReLU
Evaluation metrics	Precision, Accuracy, Recall, f1-score
Optimizer	Adam

5.3 Performance evaluation

The Confusion matrix is used to evaluate classification task performance. The categorization model's output takes the form of 0 and 1. A number of 0 implies false, while a value of 1 suggests true. The matrix used to characterise these results is known as a confusion matrix, which is displayed in Figure 10.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{7}$$

$$Precision = \frac{TP}{TP+FP} \tag{8}$$

$$Recall = \frac{TP}{TP+FN} \tag{9}$$

$$F1_{Score} = \frac{2*precision*recall}{precision+recall} \tag{10}$$

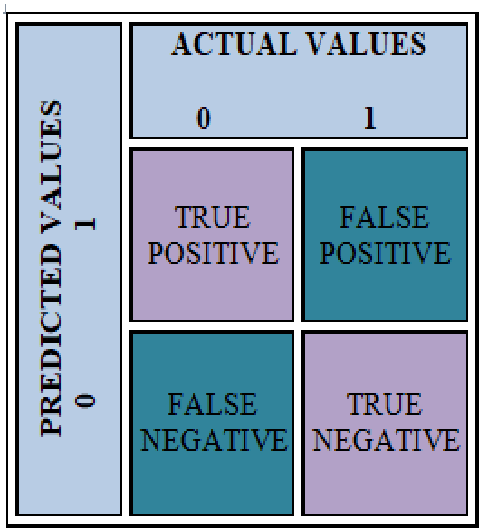


Figure 10. Schematic explanation of the confusion matrix

True Positive (TP) means that both the model's expected outcome and the actual fact are correct. A False Positive means that the model predicted a true, but the actual fact was false. A False Negative indicates that the expected value is false, but the actual value is true. Finally, True Negative (TN) indicates that both the model's predicted and actual outcomes are wrong. The proposed classification model's performance is evaluated using precision, recall, accuracy, the F1 score, and support. These are calculated using the formulas shown in Eqs. (7)-(10).

6. EVALUATION OF PROPOSED SELECTIVE VGG19-NET

The Selective VGG19-Net for the apple leaf has a model accuracy of 0.9983, as shown in Figure 11(a), and a model loss, as shown in Figure 11(b). Figure 12(a) and 12(b) indicate that the SelectiveVGG19-Net trained on the apple dataset divided the disease into four classes, which are evaluated by the confusion matrix and measured by evaluation metrics such as precision, recall, and f1-score.

Figure 13(a) depicts the accuracy of the SelectiveVGG19-Net for citrus leaves, whereas Figure 13(b) showcases the model's loss. Additionally, Figure 14(a) and 14(b) show the results of the SelectiveVGG19-Net based on the citrus dataset, which classified diseases into five distinct class. This evaluation utilizes metrics including precision, recall, and f1-score which are evaluated by confusion matrix. Table 4 shows how our model performed on both the citrus and apple datasets, as measured by a variety of performance measures. These metrics provide insight into how well the model did during both the training and validation phases. During training on the citrus dataset, the model acquired a loss function value of 0.0523, which indicates the level of inaccuracy between predicted and actual citrus leaf classifications. This dataset has a training accuracy of 0.9803, which is the number of correctly classified citrus leaves compared to the total during training.

During validation in citrus, the model achieved an accuracy of 0.9896 with a validation loss of 0.1541, reflecting its performance on an independent validation dataset. For the apple dataset, the model reached a training accuracy of 0.9983, representing the proportion of correctly identified apple leaf diseases during training. The corresponding training loss was 0.0162, indicating the discrepancy between predicted and actual classifications. On the validation set, the model attained an accuracy of 0.9851 with a validation loss of 0.1252. These results provide insights into the model’s performance in terms of classification accuracy and loss for both apple and citrus leaf datasets.

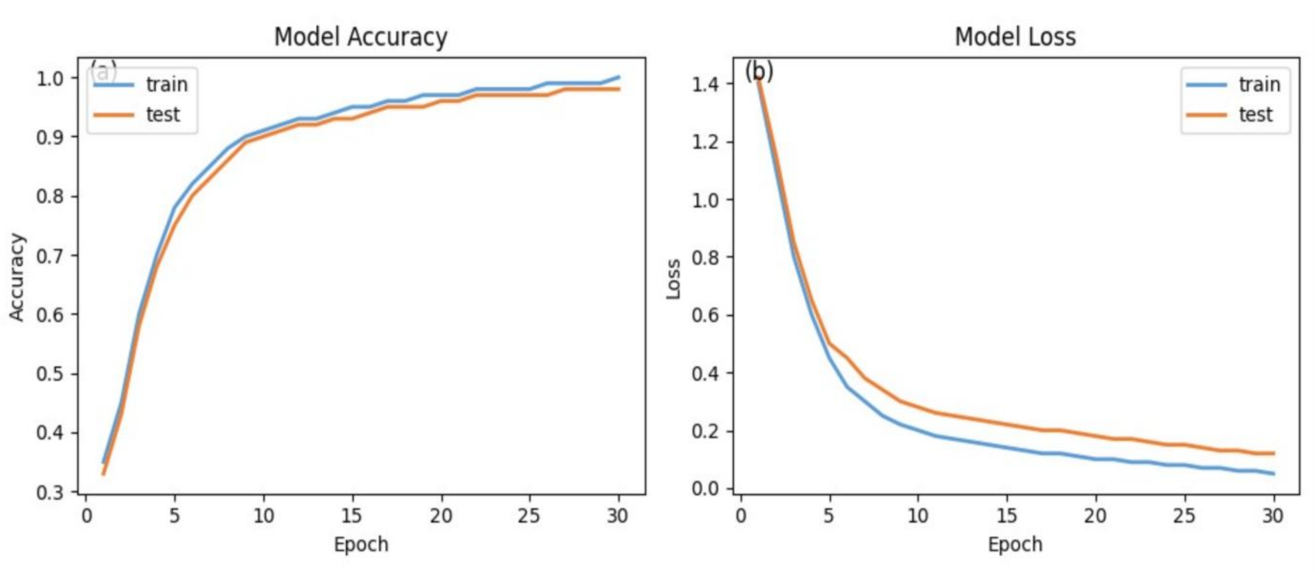


Figure 11. Selective VGG19-Net results for apple leaves: (a) Accuracy and (b) loss

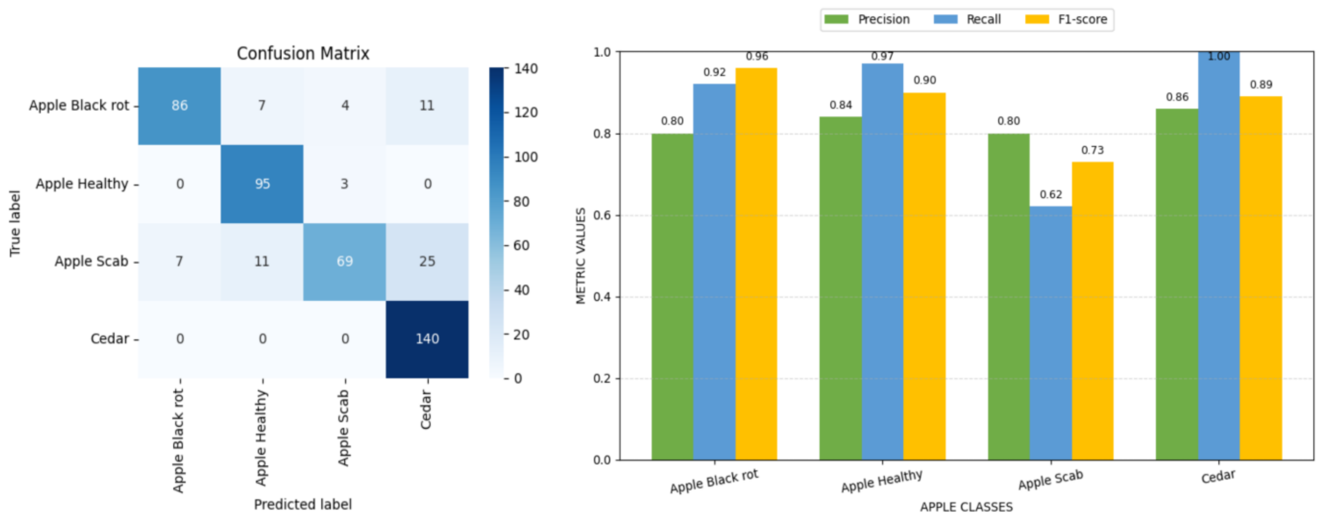


Figure 12. (a) Confusion matrix analysis of selective VGG19-Net for apple (b) performance evaluation of Selective VGG19-Net for apple

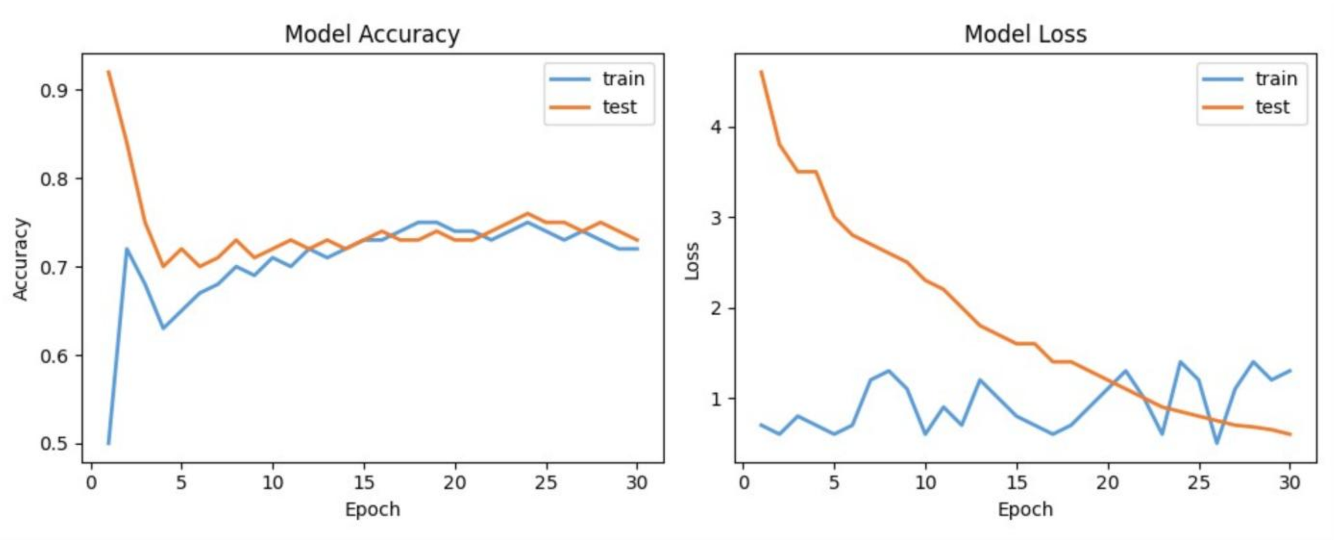


Figure 13. (a) Model accuracy of SelectiveVGG19-Net for citrus (b) Model loss of SelectiveVGG19-Net for citrus

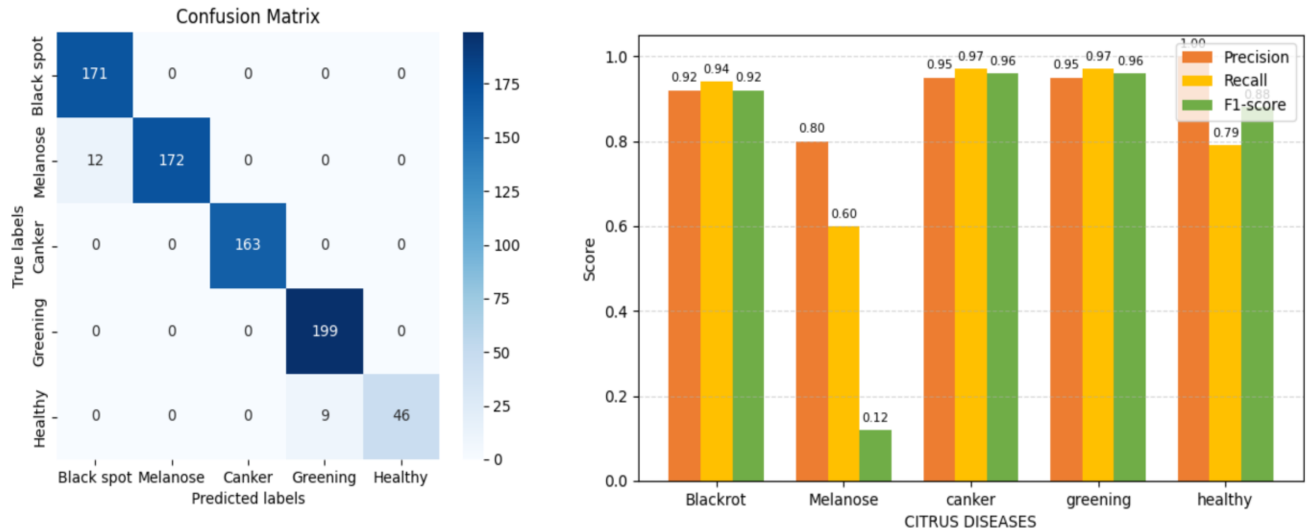


Figure 14. (a) Confusion matrix of Selective VGG19-Net for Citrus (b) Evaluation of Selective VGG19-Net for Citrus

Table 4. Performance of proposed SelectiveVGG19-Net

Type of Leaf	Performance Metrics			
	Loss	**Accuracy**	**Val Acc**	**Val Loss**
Apple	0.0162	0.9983	0.9851	0.1252
Citrus	0.0523	0.9803	0.9896	0.1541

In addition to these metrics, statistical analyses were performed to assess the reliability of the proposed Selective VGG19-Net. For citrus leaves, the mean accuracy was 0.9896, with a 95 percent confidence interval ranging from 0.9811 to 0.9980, a t-statistic of 20.25, and a p-value of 0.0024, indicating strong statistical significance. Similarly, for apple leaves, the mean accuracy was 0.9823, with a 95 percent confidence interval of 0.9599 to 1.0046, a t-statistic of 6.21, and a p-value of 0.0250, demonstrating consistent and reliable performance.

Table 5 shows a sequential overview of the Selective VGG19-Net architecture, beginning with an input image dimension of 224 × 224 × 3 and follows the classical VGG19 block design [33]. The initial layers are made up of two convolutional blocks, each of which applies 64 filters with a kernel size of (3,3) and stride of 1 × 1.

Table 5. Layer summary of SelectiveVGG19-Net

Layer Type	Filters	Feature Map Size	Kernel Size	Stride
Conv2D	64	224×224	3×3	1×1
Conv2D	64	224×224	3×3	1×1
MaxPooling2D	-	112×112	2×2	1×1
Conv2D	128	112×112	3×3	1×1
Conv2D	128	112×112	3×3	1×1
MaxPooling2D	-	56×56	2×2	2×2
Conv2D	256	56×56	3×3	1×1
Conv2D	256	56×56	3×3	1×1
Conv2D	256	56×56	3×3	1×1
Conv2D	256	56×56	3×3	1×1
Conv2D	256	56×56	3×3	1×1
Conv2D	256	56×56	3×3	1×1
MaxPooling2D	-	-	2×2	2×2
Conv2D	512	28×28	3×3	1×1
Conv2D	512	28×28	3×3	1×1
Conv2D	512	28×28	3×3	1×1
Conv2D	512	28×28	3×3	1×1
Conv2D	512	28×28	3×3	1×1
MaxPooling2D	-	14×14	2×2	2×2
Dense	-		-	-
Dense	-		-	-
Dense	4		-	-

Each convolutional block is followed by a MaxPooling2D layer with a pool size of (2, 2), which reduces the feature map dimensions to (112, 112). Subsequent layers follow a similar pattern but with additional convolutional layers in blocks 3 and 4, utilizing our strategy of convolutional and pooling extension while selectively freezing certain layers. These added convolutional layers progressively increase the number of filters, aiding in the extraction of more complex features. Throughout, the stride remains consistent, and the kernel size is unchanged. In the final layers before block 4, the output from the preceding convolutional layers is flattened, followed by a fully connected dense layer with 4096 units to prevent overfitting. Dropout regularization is then applied. The layer that produces the result is made up of a dense layer with softmax activation that calculates probabilities for each class in the classification task. Our Selective VGG19-Net was used to classify five types of citrus and four types of apple leaf diseases, utilising the mentioned layer structure to get accurate classification results.

6.1 Comparative evaluation

This study used several kinds of deep learning techniques to assess and evaluate the proposed SelectiveVGG19-Net. The accuracy of this method was compared with existing techniques, including VGG16 [34], VGG19 [35], MobileNetV3Large [36], DenseNet-121 [37], ResNet [38], and EfficientNet [39].

6.1.1 Various experimental results

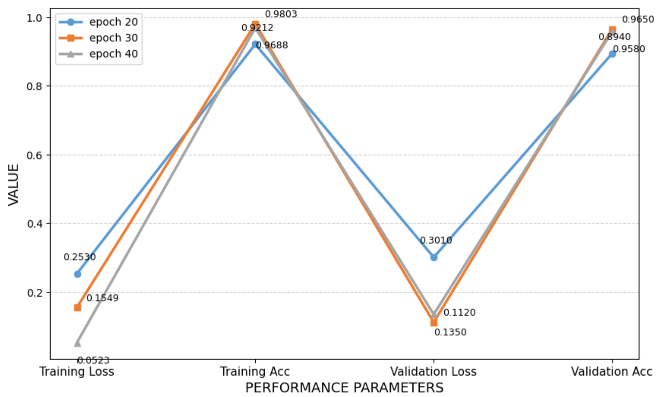


Figure 15. Variation in apple classification performance with changing epochs

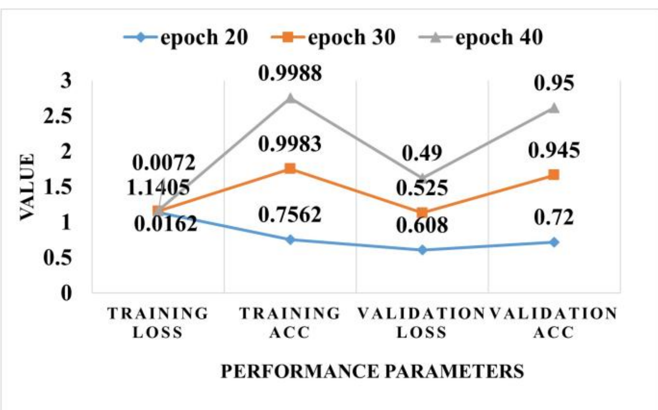


Figure 16. Variation in citrus classification performance with changing epochs

The SelectiveVGG19-Net was trained on apple and citrus datasets at epochs 20, 30, and 40. During training, metrics of performance like as accuracy and loss were calculated for both the training and validation datasets. As the number of epochs increased, the model's performance became easier to analyse and explain. Figure 15 and Figure 16 depicts how the model's performance changed with different training times, revealing insights into its ability to generalise and effectively classify leaf diseases in apples and citrus. In addition, we tested our model with two distinct optimizers, SGD and RMSprop, as shown in Table 6. These optimizers modify the network parameters. For the apple dataset, SGD obtained a training accuracy of 0.96, whereas RMSprop achieved 0.95. For the citrus dataset, SGD produced an accuracy of 0.96, whereas RMSprop achieved 0.94. These changes minimised the loss and increased training accuracy. However, utilising the Adam optimizer enhanced the SelectiveVGG19-Net's accuracy.

Table 6. Comparison of accuracy and loss for different optimizers

Type of Leaf	Optimizer	Tr.loss	Tr.acc	Val.acc	Val.loss
Apple	SGD	0.1130	0.9623	0.9459	0.1421
	RMS Prop	0.1005	0.9564	0.9354	0.1583
Citrus	SGD	0.1012	0.9669	0.9381	0.1627
	RMS Prop	0.8326	0.9438	0.9286	0.1794

Table 7. Evaluation of proposed Selective VGG19-Net against traditional and modern deep learning architectures for leaf disease classification

Author	Technique	Class / Dataset	Accuracy
Chen et al. [40]	MtConvNeXt (Ternary Attention + SVM)	Tomato Leaf Disease (10 classes)	96.15%
Wang et al. [41]	ECA-ConvNeXt (Efficient Channel Attention)	Rice Leaf Disease (6 classes + healthy)	94.82%
Soeb et al. [42]	YOLOv7 (YOLO-T)	Tea leaves	97.3% accuracy, 0.965 F1, 98.2% mAP
Prashanthi et al. [43]	LEViT (Enhanced Vision Transformer)	New Plant Disease Dataset (38 classes)	95.2% (train), 96.2% (val), 92.3% (test)
Hossain et al. [44]	MaxViT, EANet, CCT, PVT (Transformers)	Tomato Leaf Disease Dataset	97%
Yang et al. [45]	DHLC-DETR (Detection Transformer)	IDADP (Rice, 3 diseases)	97.44%
Ours	**SelectiveVGG19-Net**	**Citrus Apple**	**98%(Citrus) 99% (Apple)**

Table 7 summarizes recent deep learning and transformer-based modeling methods for plant leaf disease classification. Chen et al. [40] presented MtConvNeXt with ternary attention mixed with an SVM classifier for tomato leaf disease detection in ten classes, achieving 96.15% accuracy. Wang et al. [41] used Efficient Channel Attention (ECA) and ConvNeXt was employed to detect diseases in rice leaves across six disease classes and a healthy class, achieving an accuracy of 94.82%. Soeb et al. [42] suggested a YOLOv7 light weight detection framework (YOLO-T) for tea leaves, achieving 97.3% classification accuracy with a high F1-score of 0.965 and mean

Average Precision of 98.2%. Hossain et al. [44] investigated a number of transformer models for tomato leaf disease identification, including MaxViT, EANet, CCT, and PVT, but found no accuracies. Yang et al. [45] achieved 97.44% accuracy using DHLC-DETR, a detection-transformer model, using the IDADP dataset containing three rice diseases.

When compared to these cutting-edge approaches, our Selective VGG19 model had a total accuracy for classification of 98.03% with citrus leaf disease datasets and 99.83% with apple leaf disease datasets. This increase in classification accuracy demonstrates not only the efficacy of selective layer freezing, but also the effect of depth extension in improving model understanding for more consistent feature extraction and generalization performance, particularly when working with small and unbalanced agricultural datasets.

Table 8. Performance comparison with existing methods for apple leaf disease detection

Author	Techniques	Class	Evaluation Scores			Accuracy
			Precision	Recall	F1-Score	
Ritharson et al. [34]	VGG16	Apple Black Rot	0.44	0.32	0.40	0.6021
		Apple Healthy	0.63	0.62	0.74	
		Apple Scab	0.87	0.88	0.92	
		Cedar	0.74	0.72	0.73	
Paymode and Malode [35]	VGG19	Apple Black Rot	0.98	1.00	0.99	0.9043
		Apple Healthy	1.00	0.95	0.91	
		Apple Scab	0.83	0.79	0.75	
		Cedar	0.99	0.90	0.97	
Liu et al. [36]	MobileNetV3Large	Apple Black Rot	0.95	0.98	0.97	0.8465
		Apple Healthy	1.00	0.81	0.79	
		Apple Scab	0.88	0.95	0.91	
		Cedar	0.97	0.93	0.95	
Nandhini and Ashokkumar [37]	DenseNet-121	Apple Black Rot	0.52	0.50	0.61	0.7827
		Apple Healthy	0.68	0.78	0.71	
		Apple Scab	0.92	0.90	0.93	
		Cedar	0.75	0.79	0.81	
Yu et al. [38]	ResNet50	Apple Black Rot	0.84	0.83	0.86	0.6925
		Apple Healthy	0.71	0.56	0.65	
		Apple Scab	0.99	0.98	0.95	
		Cedar	0.63	0.54	0.52	
Cai et al. [39]	Efficient Net	Apple Black Rot	0.84	0.74	0.69	0.8006
		Apple Healthy	0.75	0.72	0.71	
		Apple Scab	0.82	0.75	0.74	
		Cedar	0.89	0.75	0.76	
(Ours)	SelectiveVGG19-Net	Apple Black Rot	0.92	0.80	0.96	**0.9983**
		Apple Healthy	0.81	0.97	0.90	
		Apple Scab	0.80	0.62	1.00	
		Cedar	0.86	1.00	0.89	

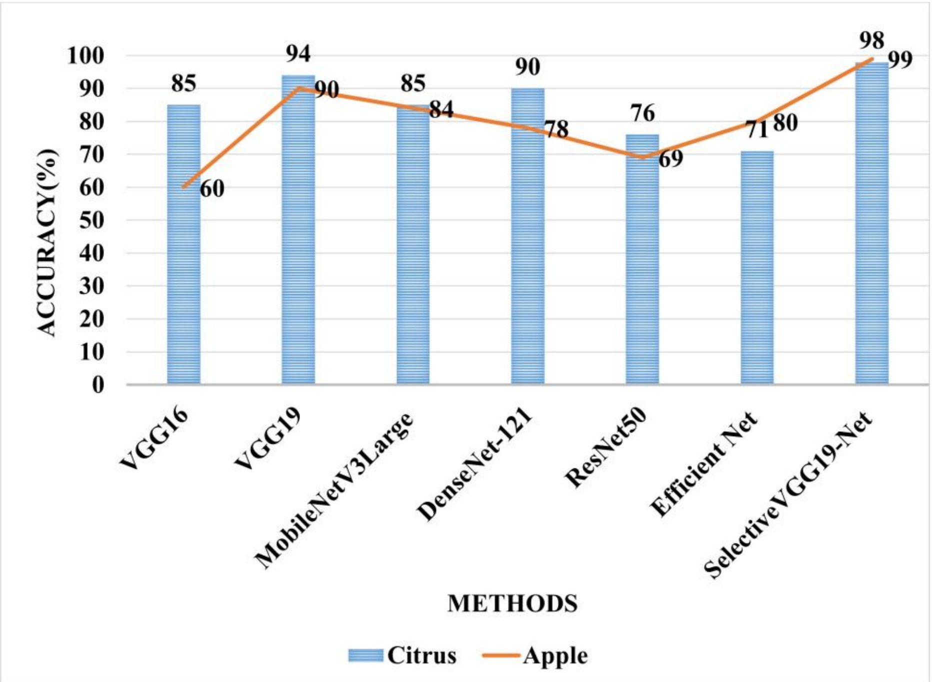


Figure 17. Accuracy performance evaluation of the proposed SelectiveVGG19-Net against other models for apple and citrus leaf disease classification

Table 9. Comparison with state-of-the-art methods for citrus leaf disease detection

Author	Techniques	Class	Evaluation Scores			
			Precision	Recall	F1-Score	Accuracy
Ritharson et al. [34]	VGG16	Black spot	0.32	0.29	0.33	0.8521
		Melanose	1.00	0.07	0.13	
		Canker	1.00	0.99	1.00	
		Greening	0.94	0.98	0.96	
		Healthy	0.98	0.98	0.98	
Paymode and Malode [35]	VGG19	Black spot	0.94	0.92	0.95	0.9442
		Melanose	1.00	0.98	0.91	
		Canker	0.78	0.80	0.72	
		Greening	0.92	0.96	0.94	
		Healthy	0.97	0.70	0.97	
Liu et al. [36]	MobileNetV3Large	Black spot	0.92	0.90	0.87	0.8523
		Melanose	0.78	0.81	0.79	
		Canker	0.95	0.93	0.91	
		Greening	0.91	0.93	0.94	
		Healthy	1.00	0.92	0.93	
Nandhini and Ashokkumar [37]	DenseNet-121	Black spot	0.88	0.82	0.90	0.9032
		Melanose	0.86	0.78	0.71	
		Canker	0.92	0.90	0.93	
		Greening	0.75	0.79	0.81	
		Healthy	0.83	0.90	0.89	
Yu et al. [38]	ResNet50	Black spot	0.75	0.69	0.66	0.7611
		Melanose	0.78	0.85	0.72	
		Canker	0.95	0.93	0.89	
		Greening	0.91	0.85	0.88	
		Healthy	1.00	0.96	0.98	
Cai et al. [39]	Efficient Net	Black spot	0.85	0.86	0.89	0.7141
		Melanose	0.82	0.78	0.74	
		Canker	0.95	0.89	0.92	
		Greening	0.96	0.89	0.92	
		Healthy	0.90	0.92	0.94	
(Ours)	SelectiveVGG19-Net	Black spot	0.92	0.91	0.92	**0.9803**
		Melanose	0.80	0.06	0.12	
		Canker	1.00	1.00	1.00	
		Greening	0.95	0.97	0.96	
		Healthy	1.00	0.79	0.88	

Table 8 summarizes the evaluations of the four apple dataset classes: Apple Black Rot, Apple Healthy, Apple Scab, and Cedar. We tested our model against the models given above and obtained the results. Our proposed model was tested for precision, recall, and f1-score in contrast to existing cutting-edge techniques. The SelectiveVGG19-Net performed well in terms of accuracy, with a score of 0.9983 for apple disease. Table 9 compares our Selective VGG19-Net to other conventional deep neural network models.

While these models were originally trained on their dataset and performed well on it, they exhibited lower training accuracy when applied to our dataset, specifically the citrus dataset. Among these models, VGG19 outperformed VGG16 with an accuracy of 0.9442 compared to 0.85. Additionally, MobileNetV3Large, DenseNet, ResNet, and EfficientNet achieved accuracies of 0.8523, 0.9032, 0.7611, and 0.7141, respectively. We evaluated the performance of each model on different classes of citrus leaf diseases, including Black Spot, Melanose, Canker, Greening, and Healthy, using metrics such as precision, recall, and F1-score. Our Selective VGG19-Net demonstrates superior performance compared to these state-of-the-art methods.

To compare Selective VGG19-Net model with other existing models, we implemented it on two datasets and achieved impressive results is 98% accuracy for citrus classification and 99% accuracy for apple classification. Figure 17 visually represents the performance of our model alongside other existing CNN-based models.

7. ABLATION STUDY

To assess the effects of select layer freezing, depth extension, and dropout regularization on the model performance for leaf disease classification, an ablation study was conducted. This study has been described as the following four configurations: Freeze = False, Extend = True, Dropout = False; Freeze = True, Extend = True, Dropout = True. For comparison, a baseline experiment with the VGG19 architecture without alterations was also included. Across 14 epochs, the trends in training and validation accuracy and loss are reported, with the main outcomes summarized in Figures 18(a, b) and 19(a, b).

Baseline VGG19 Performance

At epoch 10, the baseline VGG19 performance was 86.04% for training and 83.02% for validation, with a training loss of 0.4163 and a validation loss of 0.5061. While these numbers are within normal ranges, this baseline had a higher validation loss and lower validation accuracy when compared to our selective configurations, which performed better. There is an opportunity for improvement in the baseline, leading to enhanced performance.

Performance of Combined Configuration

The Freeze = True, Extend = True, Dropout = True configuration produced the best results, with a peak validation accuracy of 87.50% at epoch 10 and the lowest validation loss of 0.3898. This configuration achieved a compromise between generalization and strength, as seen by its epoch-level stability.

Unlike other settings, the graph of validation accuracy versus epochs for this setup was stable, indicating consistent performance. This combination emphasizes the complementary nature of layer freezing, depth extension, and dropout regularization.

Role of Depth Extension

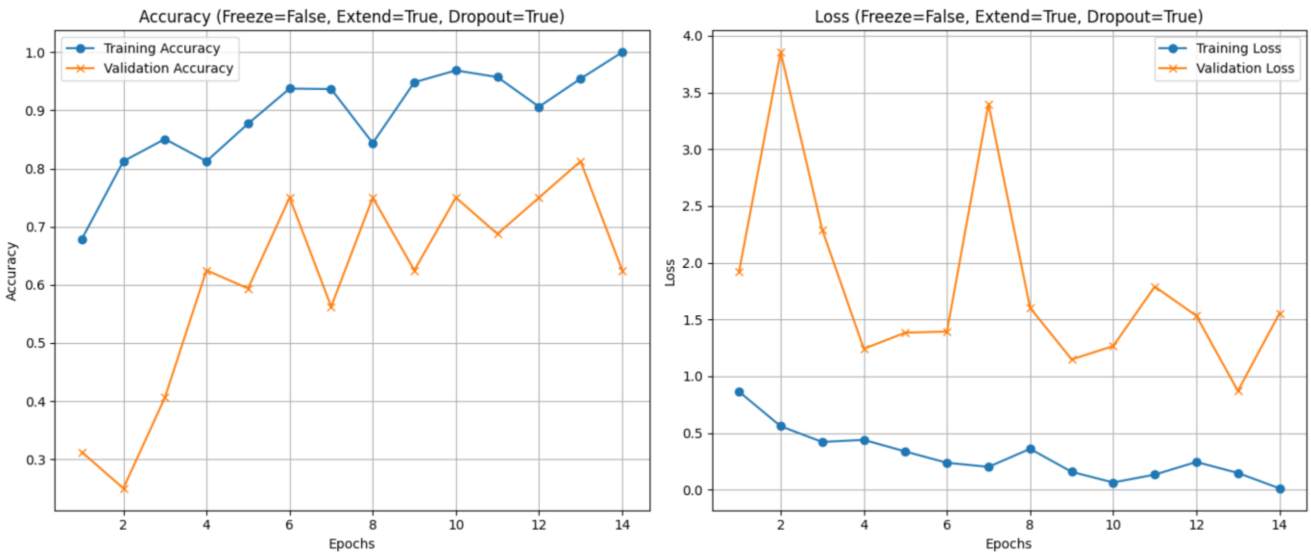
Depth extension emerged as an important factor for accurate classification. Configurations with depth extension (Extend = True) regularly outperformed those without (Extend = False). For example, with the configuration Freeze = True, Extend = True, Dropout = True, validation accuracy reached 87.50%, whereas Freeze = True, Extend = False, Dropout = True achieved just 68.75%. Similarly, validation loss decreased dramatically from 1.7518 to 0.3898, as shown in the graphs of validation loss. This finding demonstrates that increasing the depth allows the model to acquire complicated features that are required to identify between small variations in leaf diseases.

The Impact of Dropout Regularization

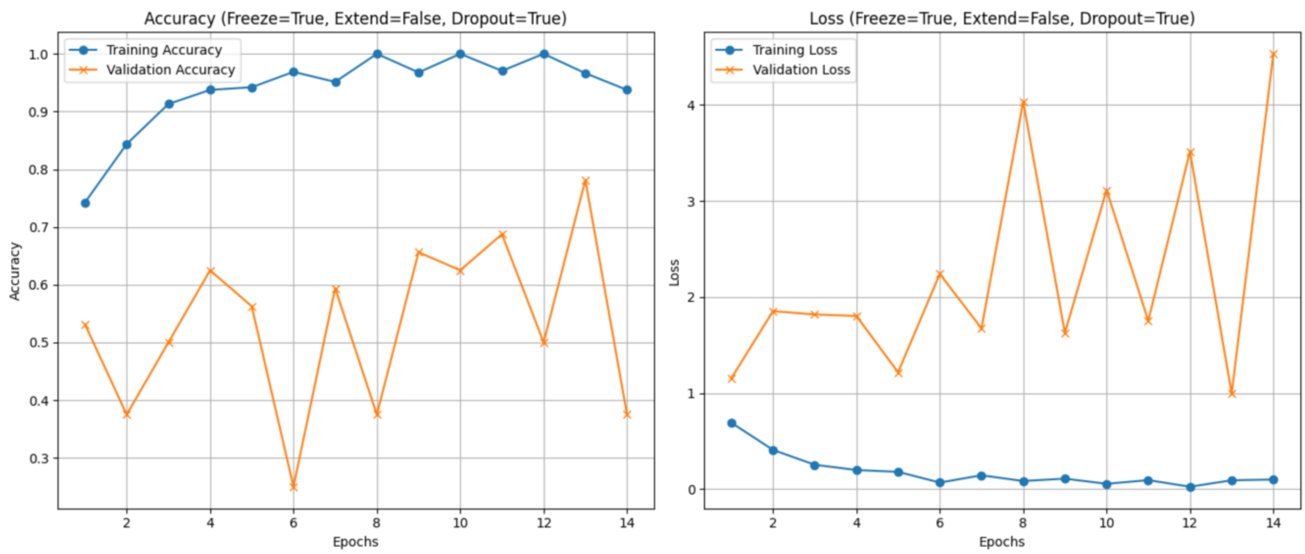
Dropout regularization proved useful in reducing overfitting, particularly for configurations with a large depth. In Freeze = True, Extend = True, Dropout = False, although attaining reasonable training accuracy, validation accuracy declined immediately to 50.00% at epoch 8, followed by a larger validation loss of 4.8004 (epoch 12). This result demonstrates that the model is capable of overfitting without dropout. By using dropout (Freeze = True, Extend = True, Dropout = True), the model maintained consistent performance, decreasing overfitting and attaining decreased validation losses across epochs.

Effect of Layer Freezing

When combined with depth extension, selective layer freezing helped to increase generalization. When Freeze = False, Extend = True, Dropout = True was compared to Freeze = True, Extend = True, Dropout = True, the latter had a higher peak validation accuracy of 87.50% at epoch 10 than 81.25% at epoch 13. This implies that freezing prior layers preserved important low-level feature representations, allowing the newly added layers to concentrate on learning high-level features relevant to the classification.

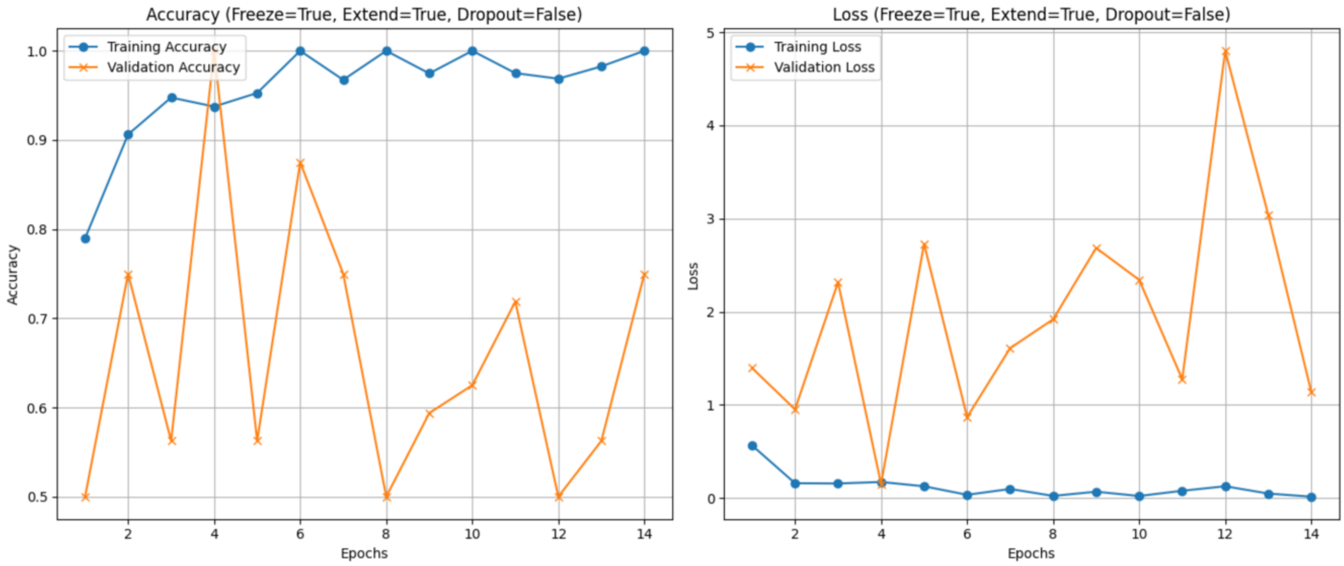


(a)

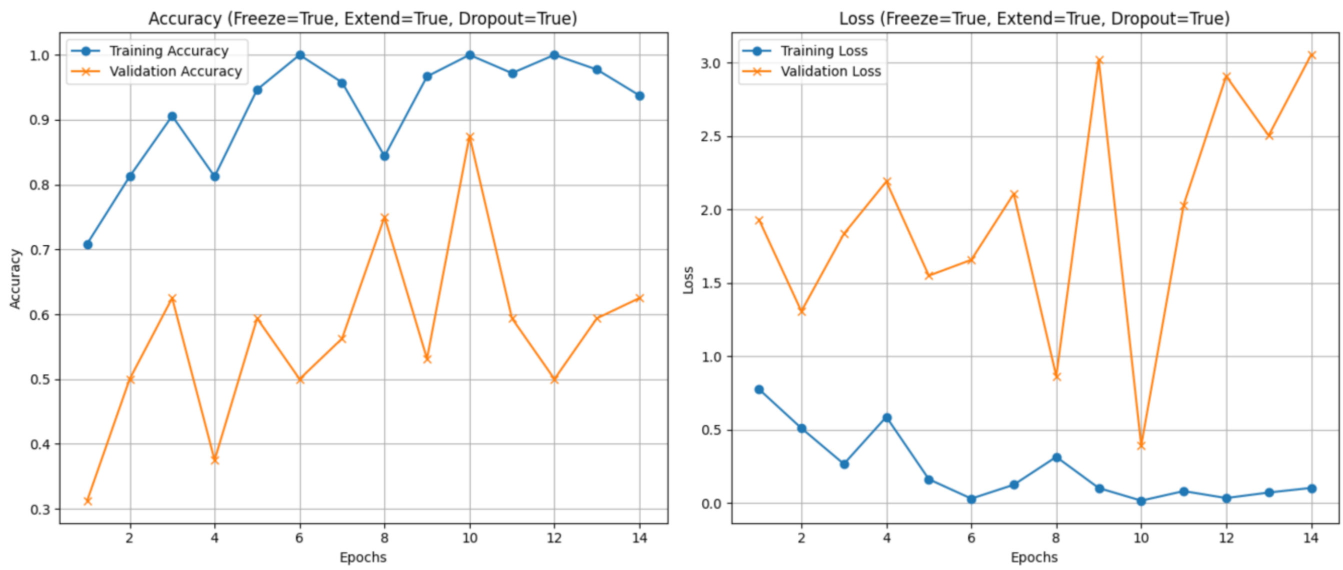


(b)

Figure 18. (a) Freeze = False, Extend = True, Dropout = True: Performance analysis (b) Freeze = True, Extend = False, Dropout = True: Performance analysis



(a)



(b)

Figure 19. (a) Freeze = True, Extend = True, Dropout = False: Performance analysis (b) Freeze = True, Extend = True, Dropout = True: Achieving optimal performance

Comparative Analysis of Configurations

The validation accuracy and loss curves over the training epochs provide a clear comparison of the performance achieved by different configurations. The configuration Freeze = True, Extend = True, Dropout = True resulted in the smoothest and most consistent increase in validation metrics, whereas configurations without dropout (Dropout = False) or depth extension (Extend = False) showed dramatic variations and larger ultimate losses. This comparison analysis emphasizes the need of combining all three tactics to get optimal model performance. The ablation investigation shows that the combination of selective layer freezing, depth extension, and dropout regularization improves the lightweight model's accuracy and robustness. The configuration Freeze = True, Extend = True, Dropout = True consistently produced superior results, with a cutting-edge validation accuracy of 87.50% and the lowest validation loss of 0.3898. Thus, the ablation study shows that selective layer freezing, depth extension, and dropout regularization combined result in a significant improvement over both the baseline VGG19 and the partial combinations it comprises, demonstrating higher accuracy and robustness when classifying for leaf disease classification.

8. DISCUSSION

The proposed Selective VGG19-Net outperformed both classic convolutional neural networks and recent transformer-based architectures in classifying apple and citrus leaf illnesses, with 99.83% accuracy for apple and 98.03% for citrus. The framework's selective freezing method, which preserves Blocks 1 and 2, allows it to maintain low-level structural feature such as edges, contours, and textures from pretrained ImageNet weights. This approach assures strong feature extraction in the early stages, while the additional convolutional-pooling depth improves hierarchical learning and allows the network to locate tiny, disease-specific patterns that are often difficult to detect in agricultural datasets.

This selective adaptation not only prevents overfitting and catastrophic forgetting, which are common problems with very small and imbalanced datasets, but it also stabilizes convergence during training. By leveraging a deeper hierarchical structure, the model demonstrates superior discrimination of closely resembling disease categories when compared with conventional CNNs like VGG16, ResNet50, and MobileNet, and even with more sophisticated transformer-driven models such as Vision Transformers and ConvNeXt. These findings demonstrate the efficiency of combining selective layer freezing with deeper architecture extensions as a reliable and generalizable technique to plant disease detection.

Beyond predicting performance, there are broader implications for real-world deployment to consider. Ethical considerations may develop as a result of dataset biases, as most images are from specific geographic regions and environmental conditions, thereby limiting generalizability across climates and crop kinds. In practice, the additional layers increase processing complexity, potentially limiting deployment on low-power or edge devices. Optimization solutions including pruning, quantization, and edge deployment can minimize memory usage and inference delay while maintaining accuracy, allowing for scalability in real-world agricultural environments.

To summarize, Selective VGG19-Net not only offers state-of-the-art performance in detecting apple and citrus leaf diseases, but it also provides a technically robust, flexible, and scalable precision agriculture framework. Its adaptation to various crops, disease kinds, and pretrained backbones demonstrates its promise as a dependable and generalizable tool for promoting sustainable farming methods and contributing to global food security.

9. CONCLUSION

Plant diseases are the most damaging to the development of agriculture around the world, and they also have an enormous effect on food crop production. It is possible that the plants will not be harvested completely. As a result, automatic diagnosis of plant diseases is increasingly required. These challenges are primarily addressed by deep learning techniques. Deep learning models play an essential role in image processing. In this study, we used transfer learning-based models to classify apple and citrus diseases. We collected two datasets from open sources. The popular transfer learning architecture VGG19 have been used in this work to overcome the sparse dataset. A novel approach, termed selective layer freezing and convolutional pooling layer extension, was introduced to enhance leaf disease prediction accuracy. This method involves fine-tuning the model by selectively freezing specific blocks, namely Block 1 and Block 2, while incorporating additional convolutional and pooling layers to the base architecture. Special attention was given to mitigating challenges such as overfitting and catastrophic forgetting during the fine-tuning process. As a result, performance has improved, further we analyzed our findings and compared them to the previously mentioned models. The Selective VGG19-Net demonstrated a classification accuracy of 0.9983 for apple diseases, and when tested on citrus leaf diseases, it achieved an accuracy of 0.9803. While this study focused on apple and citrus datasets, the selective layer freezing technique and its convolutional-pooling extension are model independent and can be applied to other crops with minimal changes. The technique, which improves extraction speed while reducing overfitting, could be used for larger, more complicated, and complex agricultural datasets that include different crops and diseases. In future work, we plan to classify plant disease with more classes of disease and also use other pretrained models for classification.

REFERENCES

[1] Zhu, X., Li, J., Jia, R., Liu, B., Yao, Z., Yuan, A., Zhang, H. (2022). LAD-Net: A novel light weight model for early apple leaf pests and diseases classification. IEEE/ACM Transactions on Computational Biology and Bioinformatics, 20(2): 1156-1169. https://doi.org/10.1109/TCBB.2022.3191854

[2] Khattak, A., Asghar, M.U., Batool, U., Asghar, M.Z., Ullah, H., Al-Rakhami, M., Gumaei, A. (2021). Automatic detection of citrus fruit and leaves diseases using deep neural network model. IEEE Access, 9: 112942-112954. https://doi.org/10.1109/ACCESS.2021.3096895

[3] Shewale, M.V., Daruwala, R.D. (2023). High performance deep learning architecture for early detection and classification of plant leaf disease. Journal of Agriculture and Food Research, 14: 100675. https://doi.org/10.1016/j.jafr.2023.100675

[4] Rauf, H.T., Saleem, B.A., Lali, M.I.U., Khan, M.A., Sharif, M., Bukhari, S. (2019). A citrus fruits and leaves dataset for detection and classification of citrus diseases through machine learning. Data in Brief, 26: 104340. https://doi.org/10.1016/j.dib.2019.104340

[5] Yao, J., Tran, S. N., Sawyer, S., Garg, S. (2023). Machine learning for leaf disease classification: Data, techniques and applications. Artificial Intelligence Review, 56(Suppl 3): 3571-3616. https://doi.org/10.1007/s10462-023-10610-4

[6] Mohamed, E.S., Belal, A.A., Abd-Elmabod, S.K., El-Shirbeny, M.A., Gad, A., Zahran, M.B. (2021). Smart farming for improving agricultural management. The Egyptian Journal of Remote Sensing and Space Science, 24(3): 971-981. https://doi.org/10.1016/j.ejrs.2021.08.007

[7] Chakraborty, K.K., Mukherjee, R., Chakroborty, C., Bora, K. (2022). Automated recognition of optical image based potato leaf blight diseases using deep learning. Physiological and Molecular Plant Pathology, 117: 101781. https://doi.org/10.1016/j.pmpp.2021.101781

[8] Sunil, C.K., Jaidhar, C.D., Patil, N. (2023). Systematic study on deep learning-based plant disease detection or classification. Artificial Intelligence Review, 56: 14955-15025. https://doi.org/10.1007/s10462-023-10517-0

[9] Hessane, A., El Youssefi, A., Farhaoui, Y., Aghoutane, B., Amounas, F. (2023). A machine learning based framework for a stage-wise classification of date palm white scale disease. Big Data Mining and Analytics, 6(3): 263-272. https://doi.org/10.26599/BDMA.2022.9020022

[10] Perveen, K., Debnath, S., Pandey, B., Chand, S.P., Bukhari, N.A., Bhowmick, P., Alshaikh, N.A., Arzoo, S., Batool, S. (2023). Deep learning-based multiscale CNN-based U network model for leaf disease diagnosis and segmentation of lesions in tomato. Physiological and Molecular Plant Pathology, 128: 102148.

https://doi.org/10.1016/j.pmpp.2023.102148
[11] Barburiceanu, S., Meza, S., Orza, B., Malutan, R., Terebes, R. (2021). Convolutional neural networks for texture feature extraction. Applications to leaf disease classification in precision agriculture. IEEE Access, 9: 160085-160103. https://doi.org/10.1109/ACCESS.2021.3131002
[12] Shovon, M.S.H., Mozumder, S.J., Pal, O.K., Mridha, M.F., Asai, N., Shin, J. (2023). PlantDet: A robust multi-model ensemble method based on deep learning for plant disease detection. IEEE Access, 11: 34846-34859. https://doi.org/10.1109/ACCESS.2023.3264835
[13] Rajpoot, V., Tiwari, A., Jalal, A.S. (2023). Automatic early detection of rice leaf diseases using hybrid deep learning and machine learning methods. Multimedia Tools and Applications, 82(23): 36091-36117. https://doi.org/10.1007/s11042-023-14969-y
[14] Ahmad, M., Abdullah, M., Moon, H., Han, D. (2021). Plant disease detection in imbalanced datasets using efficient convolutional neural networks with stepwise transfer learning. IEEE Access, 9: 140565-140580. https://doi.org/10.1109/ACCESS.2021.3119655
[15] Nagaraju, M., Chawla, P., Upadhyay, S., Tiwari, R. (2022). Convolution network model based leaf disease detection using augmentation techniques. Expert Systems, 39(4): e12885. https://doi.org/10.1111/exsy.12885
[16] Zhou, C., Zhang, Z., Zhou, S., Xing, J., Wu, Q., Song, J. (2021). Grape leaf spot identification under limited samples by fine grained-GAN, IEEE Access, 9: 100480-100489. https://doi.org/10.1109/ACCESS.2021.3097050
[17] Liu, Y., Gao, G., Zhang, Z. (2022). Crop disease recognition based on modified light-weight CNN with attention mechanism. IEEE Access, 10: 112066-112075. https://doi.org/10.1109/ACCESS.2022.3216285
[18] Yu, H., Liu, J., Chen, C., Heidari, A.A., Zhang, Q., Chen, H., Turabieh, H. (2021). Corn leaf diseases diagnosis based on K-means clustering and deep learning. IEEE Access, 9: 143824-143835. https://doi.org/10.1109/ACCESS.2021.3120379
[19] Bukhari, H.R., Mumtaz, R., Inayat, S., Shafi, U., Haq, I.U., Zaidi, S.M.H., Hafeez, M. (2021). Assessing the impact of segmentation on wheat stripe rust disease classification using computer vision and deep learning. IEEE Access, 9: 164986-165004. https://doi.org/10.1109/ACCESS.2021.3134196
[20] Cap, Q.H., Uga, H., Kagiwada, S., Iyatomi, H. (2020). LeafGAN: An effective data augmentation method for practical plant disease diagnosis. IEEE Transactions on Automation Science and Engineering, 19(2): 1258-1267. https://doi.org/10.48550/arXiv.2002.10100
[21] Wu, Y., Feng, X., Chen, G. (2022). Plant leaf diseases fine-grained categorization using convolutional neural networks. IEEE Access, 10: 41087-41096. https://doi.org/10.1109/ACCESS.2022.3167513
[22] Stephen, A., Punitha, A., Chandrasekar, A. (2023). Optimal deep generative adversarial network and convolutional neural network for rice leaf disease prediction. The Visual Computer, 40: 919-936. https://doi.org/10.1007/s00371-023-02823-z
[23] Hassan, S.M., Maji, A.K., Jasiński, M., Leonowicz, Z., Jasińska, E. (2021). Identification of plant-leaf diseases using CNN and transfer-learning approach. Electronics, 10(12): 1388. https://doi.org/10.3390/electronics10121388
[24] Elfatimi, E., Eryigit, R., Elfatimi, L. (2022). Beans leaf diseases classification using MobileNet models. IEEE Access, 10: 9471-9482. https://doi.org/10.1109/ACCESS.2022.3142817
[25] Pal, A., Kumar, V. (2023). AgriDet: Plant leaf disease severity classification using agriculture detection framework. Engineering Applications of Artificial Intelligence, 119: 105754. https://doi.org/10.1016/j.engappai.2022.105754
[26] Singh, P., Farooq, U., Khurana, S.S., Verma, J.K., Kumar, M. (2023). CottonLeafNet: Cotton plant leaf disease detection using deep neural networks. Multimedia Tools and Applications, 82: 37151-37176. https://doi.org/10.1007/s11042-023-14954-5
[27] Nazeer, R., Ali, S., Hu, Z., Ansari, G.J., Al-Razgan, M., Awwad, E.M., Ghadi, Y.Y. (2024). Detection of cotton leaf curl disease's susceptibility scale level based on deep learning. Journal of Cloud Computing, 13(1): 50. https://doi.org/10.1186/s13677-023-00582-9
[28] Shafi, U., Mumtaz, R., Qureshi, M.D.M., Mahmood, Z., Tanveer, S.K., Haq, I.U., Zaidi, S.M.H. (2023). Embedded AI for wheat yellow rust infection type classification. IEEE Access, 11: 23726-23738. https://doi.org/10.1109/ACCESS.2023.3254430
[29] Sunil, C.K., Jaidhar, C.D., Patil, N. (2021). Cardamom plant disease detection approach using EfficientNetV2. IEEE Access, 10: 789-804. https://doi.org/10.1109/ACCESS.2021.3138920
[30] Liu, K., Zhang, X. (2022). PiTLiD: Identification of plant disease from leaf images based on convolutional neural network. IEEE/ACM Transactions on Computational Biology and Bioinformatics, 20(2): 1278-1288. https://doi.org/10.1109/TCBB.2022.3195291
[31] Joshi, B.M., Bhavsar, H. (2023). A nightshade crop leaf disease detection using enhance-nightshade-CNN for ground truth data. The Visual Computer, 40: 5639-5658. https://doi.org/10.1007/s00371-023-03127-y
[32] Yu, F., Xiu, X., Li, Y. (2022). A survey on deep transfer learning and beyond. Mathematics, 10(19): 3619. https://doi.org/10.3390/math10193619
[33] Simonyan, K., Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556. https://doi.org/10.48550/arXiv.1409.1556
[34] Ritharson, P.I., Raimond, K., Mary, X.A., Robert, J.E., Andrew, J. (2024). DeepRice: A deep learning and deep feature based classification of Rice leaf disease subtypes. Artificial Intelligence in Agriculture, 11: 34-49. https://doi.org/10.1016/j.aiia.2023.11.001
[35] Paymode, A.S., Malode, V.B. (2022). Transfer learning for multi-crop leaf disease image classification using convolutional neural network VGG. Artificial Intelligence in Agriculture, 6: 23-33. https://doi.org/10.1016/j.aiia.2021.12.002
[36] Liu, Y., Wang, Z., Wang, R., Chen, J., Gao, H. (2023). Flooding-based MobileNet to identify cucumber diseases from leaf images in natural scenes. Computers and Electronics in Agriculture, 213: 108166. https://doi.org/10.1016/j.compag.2023.108166
[37] Nandhini, S., Ashokkumar. (2022). An automatic plant leaf disease identification using DenseNet-121 architecture with a mutation-based henry gas solubility

optimization algorithm. Neural Computing and Applications, 34(7): 5513-5534. https://doi.org/10.1007/s00521-021-06714-z

[38] Yu, M., Ma, X., Guan, H. (2023). Recognition method of soybean leaf diseases using residual neural network based on transfer learning. Ecological Informatics, 76: 102096. https://doi.org/10.1016/j.ecoinf.2023.102096

[39] Cai, J., Pan, R., Lin, J., Chen, X., Zhang, X. (2023). Improved EfficientNet for corn disease identification. Frontiers in Plant Science, 14: 1224385. https://doi.org/10.3389/fpls.2023.1224385

[40] Chen, Y., Zhang, N., Wang, A., Liu, Z., Yue, J. (2024). Automatic classification of tomato leaf diseases based on MtConvNeXt model. In 2024 8th International Conference on Electrical, Mechanical and Computer Engineering (ICEMCE), Xi'an, China, pp. 2021-2025. https://doi.org/10.1109/ICEMCE64157.2024.10862217

[41] Wang, X., Wang, Y., Zhao, J., Niu, J. (2023). ECA-ConvNeXt: A rice leaf disease identification model based on ConvNeXt. In 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Vancouver, Canada, pp. 6235-6243. https://doi.org/10.1109/CVPRW59228.2023.00663

[42] Soeb, M., Jubayer, M., Tarin, T., Mamun, M., Ruhad, F., Parven, A., Mubarak, N., Karri, S., Meftaul, I. (2023). Tea leaf disease detection and identification based on YOLOv7 (YOLO-T). Scientific Reports, 13: 6078. https://doi.org/10.1038/s41598-023-33270-4

[43] Prashanthi, B., Krishna, A., Rao, C. (2024). LEViT- Leaf Disease identification and classification using an enhanced Vision transformers (ViT) model. Multimedia Tools and Applications, 84: 23313-23344. https://doi.org/10.1007/s11042-024-19866-6

[44] Hossain, S., Reza, M., Chakrabarty, A., Jung, Y. (2023). Aggregating different scales of attention on feature variants for tomato leaf disease diagnosis from image data: A transformer driven study. Sensors, 23(7): 3751. https://doi.org/10.3390/s23073751

[45] Yang, H., Deng, X., Shen, H., Lei, Q., Zhang, S., Liu, N. (2023). Disease detection and identification of rice leaf based on improved detection transformer. Agriculture, 13(7): 1361. https://doi.org/10.3390/agriculture13071361