International Journal of Mathematical, Engineering and Management Sciences
Vol. 9, No. 4, 943-964, 2024
https://doi.org/10.33889/IJMEMS.2024.9.4.050

Ram Arti Publishers

Enhancing Apple Leaf Disease Detection: A CNN-based Model Integrated with Image Segmentation Techniques for Precision Agriculture

Nidhi Parashar
School of Computing Science and Engineering,
Galgotias University, Greater Noida, Uttar Pradesh, India.
Corresponding author: nidhi.csit@gmail.com

Prashant Johri
School of Computing Science and Engineering,
Galgotias University, Greater Noida, Uttar Pradesh, India.
E-mail: johri.prashant@gmail.com

(Received on November 13, 2023; Revised on February 7, 2024 & March 26, 2024 & April 8, 2024; Accepted on April 29, 2024)

Abstract
The agriculture industry has an enormous influence on a nation's economy. Loss of yield due to plant diseases remains a reason, reducing crop quantity and quality. Incorrect diagnosis of crop diseases can result in improper application of chemical pesticides, which promotes immune microbial strains, raises expenses, and triggers fresh outbreaks that are harmful to the economy and the ecosystem. Despite the potential of Machine Learning (ML) and Deep Learning (DL) approaches in plant disease detection, their limited effectiveness results in poor or late disease detection. Resolving this issue is critical, requiring the development of more accurate disease detection methods. This research introduces an innovative approach for the detection of apple leaf diseases utilizing the CNN-based Inception-v3 model. The dataset comprises images taken on location without having any control over the image-capturing settings may provide better relevance to real-world scenarios. The proposed method integrates canny edge detection and watershed transformation to achieve accurate image segmentation, thereby enhancing the identification of disease regions. Additionally, exploratory data analysis was performed, and channel distributions were visualized to understand the dataset's characteristics. To ensure robust evaluation, the model's performance underwent stratified 5-fold cross-validation. The model classified plant images with 84.60% precision, 87.40% recall, 85.00% F1-score, and 94.76% accuracy. Experimental results substantiate the efficacy of the proposed approach, surpassing existing methods in disease classification.

Keywords- Convolutional neural network, Disease detection, Deep learning, Inception, Apple disease.

1. Introduction

Among the primary mechanisms for a sustainable and nutritious food system is the development of agriculture. The causes of food scarcity are agricultural losses caused by multiple illnesses, extreme weather, and a shortage of assets. The principal causes of leaf disease during their life cycle include a variety of fungi, bacteria, viruses, and other ecologically transmissible organisms. Considering the health benefits of apples, it is critical to reduce crop losses and maximize their benefits. To preserve its production and supply, emphasis must be on adopting techniques to reduce diseases, pests, adverse environmental factors, and post-harvest crop losses. The most common diseases noticed on apple leaves include rust, scab, powdery mildew, and many more. Apple scab, resulting from a fungi infection, is a commercially significant fungal disease of apples worldwide (Barbedo, 2019). Apple scab characteristics include apparent fungi growths on the leaf's exterior. Rust disease also results in crop losses when surroundings are suitable for disease progression. Appropriate identification of plant illnesses benefits agriculture, lowering crop loss, and improving the quality and quantity of crops. In the beginning, researchers used methods from the fields of molecular biology and immunological processes to identify crop diseases at the onset (Dinh et al., 2020; Sankaran et al., 2010). However, these methods demanded human specialists, significant resources, and money. Most crop fields are operated by those with low incomes in developing nations. Therefore, the researchers must provide farmers with low-cost and practical agricultural techniques. As there are many transformations in colour, size, shape, and place of leave, it is challenging to localize and diagnose crop diseases effectively and efficiently. Furthermore, the process of disease diagnosis becomes difficult because of brightness changes that occur during the acquisition of images of leaves. Incorrect plant disease diagnosis can result in an excessive application of chemical-based pesticides, encouraging immune strains of bacteria and increasing costs. This cycle promotes new outbreaks, putting a significant burden on both the agricultural economy and environmental sustainability. Incorrect diagnosis wastes money and encourages a destructive cycle of pesticide dependency, compromising agricultural profitability and ecological balance. A subsequent preferred option is to use ML approaches to plant leaves, which have been considered in various kinds of research studies. ML is a standard technique that supports machines to behave as humans. Various approaches, including random forest (RF), K-Nearest Neighbour (kNN), decision tree (DT), and support vector machine (SVM), contribute to training machines employing identical data. A variant of the KNN method merged with fuzzy logic in (Jakjoud et al., 2019) delivered notable outcomes with a precision of 98.38%. An image-processing methodology for detecting tomato leaf diseases using an SVM classifier was presented in the work by (Rahman et al., 2023). Their method gave excellent, with an accuracy of 100%, 95%, 90%, and 85% for healthy leaves, early blight, later blight, and septoria leaf spot respectively based on experiments. Numerous traditional ML models have been developed in previous research to categorize and identify plant diseases. Such ML models demand a challenging feature retrieval technique, which directly impacts the speed of the classification process. Due to its built-in feature extraction, DL is preferred over ML by researchers. The key contributions of this research study are as follows:

- This research shows in-depth exploratory data analyses including visualizations of class distribution, alluvial diagram to check for the presence of overlapped classes, channel distributions of red, green, and blue channels in all classes of leaf images to identify many diseases accurately, even recognizing more than one disease on the same leaf, and to examine the depth insight, incorporating orientation, light, shading, and leaf physiology.
- The features extracted by canny edge detection and the segmentation provided by the watershed transformation assist the model in recognizing and differentiating between healthy and diseased areas. The variations in textures, edges, and structural differences in the leaves due to diseases are emphasized through these techniques, potentially aiding in the model training.
- The authors trained an Inception-v3 model, to distinguish among apple leaves suffering from scab, rust, many diseases, and healthy leaves. This study uses stratified 5-fold cross-validation to solve the class imbalance problem in the dataset and validate the model performance.
- The disease detection model was performed with a precision of 84.6%, recall of 87.4%, F1 score of 85%, and overall accuracy of 94.76%.
- To the best of our knowledge, no research study in the existing literature on plant disease detection models integrated canny edge detection, watershed transformation, and inception v3 for the classification of plant leaves. The apple dataset used in this research study consists of four classes of leaves. Focusing on fewer classes can result in a more practical, and detailed analysis of these specific classes, including identifying their features, causes, and suitable mitigation actions. In some agricultural conditions, focusing on the detection of fewer and more prevalent diseases may be more convenient to the farmer than a wider classification approach that covers many different kinds of diseases.

The subsequent segments of this article are arranged in the order that follows: Section 2 reviews the related prior research papers based on plant disease detection. Section 3 presents the dataset details, problem formulation, and a comprehensive explanation of the proposed method. Performance evaluation, including the evaluation metrics and implementation particulars, are presented in Section 4. Section 5 presents the experimental results. The discussion of the findings, comparison with previously used disease detection models, and limitations of the proposed model are in Section 6. The research concludes in Section 7 with some final observations and recommendations for improved classification performance.

2. Literature Review

Specialists have provided many techniques for identifying plant diseases integrating image processing, ML, and DL approaches. This section shows a thorough examination of the literature-based research to identify plant diseases at their initial growing stages. This study considers methods based on ML and DL for classifying plant diseases. A light CNN model was proposed by (Thakur et al., 2023)to identify crop diseases from visuals of plant leaves. The approach exceeded many previous deep-learning methods for crop disease detection and offered an accuracy of 99.16% on the plant village data. In another study (Hassan & Maji, 2022), the authors used a deep CNN model for acquiring the classification features, which were classified using logistic regression (LR), SVM, KNN, RF, and Naive Bayes. The result showed that SVM and LR outperformed some other DL models in accuracy, precision, and recall. In their study (Zeng & Li, 2020) suggested a self-attention-based CNN that retrieves relevant attributes from crop disease lesions to recognize a specific disease. The model consisted of a basic network to obtain the image's general features and a self-attention structure for collecting the primary features of the damaged area. This configuration improves the disease identification accuracy by 2.9%. Two optimized DL models were proposed for maize leaf disease detection (Zhang et al., 2018). The Cifar10 model obtained a mean accuracy of 98.8%, while the GoogLeNet model obtained a mean accuracy of 98.9%. The focus of (Too et al., 2019) was on refining and evaluating a VGG 16, Inception-v4, ResNet, and DenseNets for image-based diagnosis of plant diseases. DenseNet outperformed all other models with an accuracy of 99.75%. They used Keras with Theano implementation to train the models. Another study (Atole & Park, 2018) created a deep CNN using AlexNet architecture to classify rice plant images into three categories such as normal, sick, or infected with golden apple snails. The model classified the images with 91.23% accuracy.

In (Ferentinos, 2018), the authors developed CNN models to recognize and identify plant diseases using imagery of plant leaves. They used a public dataset of 25 distinct plants grouped into 58 classes for model training. The most successful model architecture identified the corresponding plant, a disease pair, or a healthy plant with an accuracy of 99.53%. To diagnose and categorize grapes diseases using RGB plant leaf images, (Math & Dharwadkar, 2022) built a Deep CNN model. The generated model's special feature is that it used transfer learning to create a CNN model and offered accuracy superior to some pre-trained models. The model attained a 99.34% accuracy rate. The authors (Hari & Singh, 2023) developed a CNN model with the notion of reused features at three distinct levels to detect diseased leaves in bananas, guavas, and mango crops. According to the results, the model classified the leaves with a 99.14% success rate. Also, it performed superior to 15 distinct modern pre-trained DL models. In another study, (Rajpoot et al., 2023) examined three different types of diseases that damage rice. They extracted the features using a VGG-16 model with an R-CNN structure and then categorized the features using the RF method. The technique had an average class prediction precision of 97.3% for rice disease detection. The motivations for developing a new plant disease detection model are as follows:

- Disease detection DL models need a huge amount of categorized information for training the classification model. However, it is difficult to acquire and label excellent quality plant disease datasets. Models might be biased due to unbalanced datasets or inadequate representation of some diseases.
- Models developed for a particular species of plants or diseases did not interpret well to other plant types or plant diseases. Limited adaptability prevented them from being deployed extensively.
- Existing models suffer when dealing with images captured under different types of light, camera positions, and external conditions as PlantVillage (Mohanty et al., 2016) provides data sets for detecting

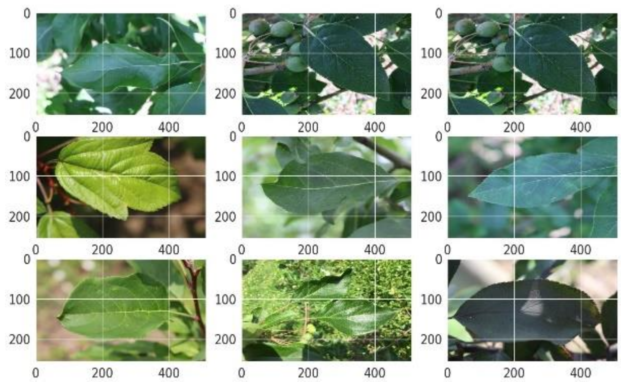
diseases using computer vision. However, its images with uniform backdrop settings limit practical relevance, resulting in lower prediction accuracy for different conditions. Individual leaf images from laboratory experiments lack realistic settings to realize farmers' field realities. The dataset comprising images taken on location without having any control over the image-capturing settings may provide better relevance to real-world scenarios.
- It is difficult to incorporate computationally expensive DL-based disease detection models into current farm practices and decision-making procedures. Farmers require resources that are easy to use and readily available.

The study aims to enhance the accuracy of apple leaf disease diagnosis under practical situations by using an innovative CNN-based approach that incorporates powerful image segmentation techniques in agricultural applications.

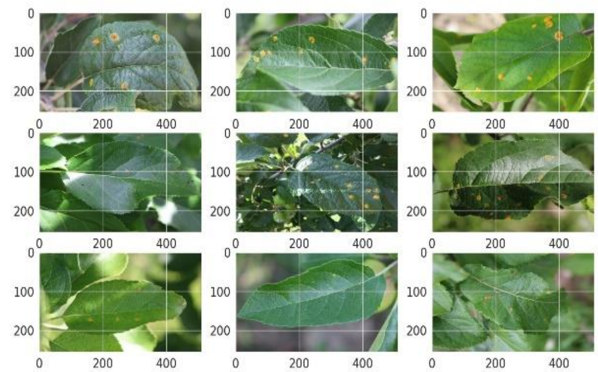
3. Materials and Methods
This section describes the apple leaf disease dataset, all the preprocessing steps, image segmentation techniques, and the Inception-v3 model architecture for apple leaf disease detection. Section 3.1 covers the dataset, its preprocessing, and exploratory data analysis. Subsection 3.2 presents the segmentation techniques used on the images for feature extraction. Section 3.3 demonstrates the problem formulation for the Apple multi-class classification, subsection 3.4 explains the Inception v3 model, and subsection 3.5 explains the methodology performed in this research.

3.1 Dataset Details, Pre-processing and Exploratory Data Analysis
This study uses the openly available apple dataset provided in a crop pathology contest 2020- FGVC7 on the Kaggle platform for detecting apple diseases (Thapa et al., 2020). This contest was funded by the Cornell Initiative for Digital Agriculture. **Figure 1** shows the visualization of apple leaf images belonging to each category in the dataset. The entire set contains 1,821 labeled images of apple leaves in landscape mode (1,368 × 2,048) and portrait mode (2048 × 1368). The images are present in the four categories of "Healthy," "Rust," "Scab", and "Multiple Diseases". The images have been taken on location without having any control over the image-capturing settings.

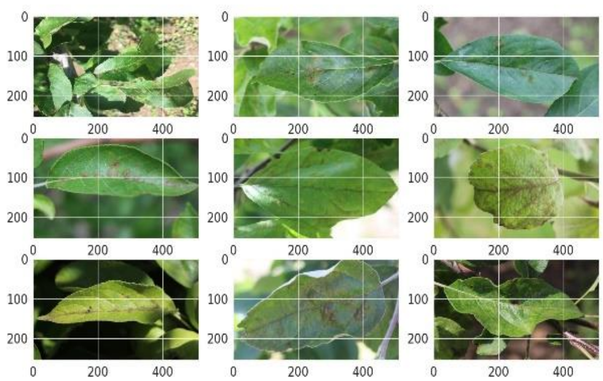
To gain insights into the dataset's characteristics, exploratory data analysis involves visualizing the distribution of images across different disease categories and identifying any class imbalances. Additionally, visualizations of channel distributions helped to analyze the colour variations between healthy and diseased leaves. As shown in **Figure 2**, the dataset consists of 592 images of apple leaves infected with scab, 622 images of leaves infected with rust, 516 healthy leaf images, and 91 images of leaves having multiple diseases. Compared to multiple diseases, the dataset contains a disproportionately high number of images of leaves infected by a single disease. In the alluvial plot (**Figure 2**), ‘0’ represents the negative instances, and ‘1’ represents the positive instances of a particular class. In all four classes, there are no overlapped 1’s. Hence, the alluvial diagram depicts no overlapping of classes. The finding encourages data augmentation to address class imbalances and ensure a more proportionate representation of various disease classes during model training. The absence of overlapped cases in the alluvial plot influenced model design decisions and emphasized the importance of effective feature extraction that can accurately distinguish between different disease classes.



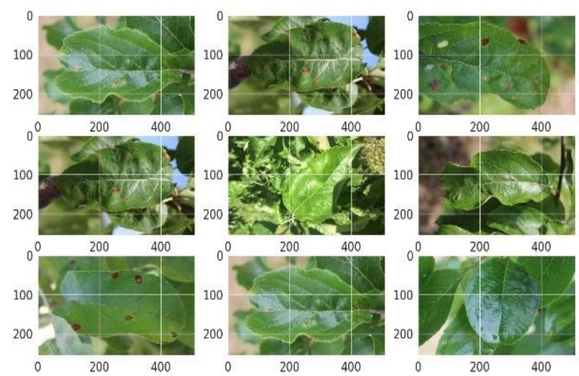
(Healthy leaf images)



(Rust infected images)

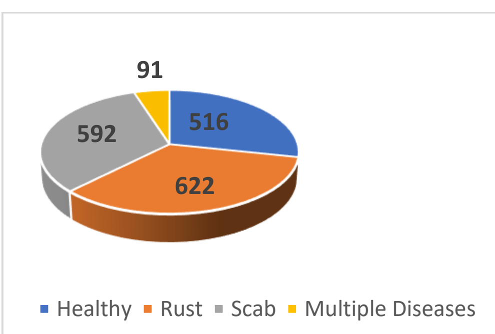


(Scab infected images)



(Multiple diseases infected images)

Figure 1. Apple leaf images of all four classes in the dataset.



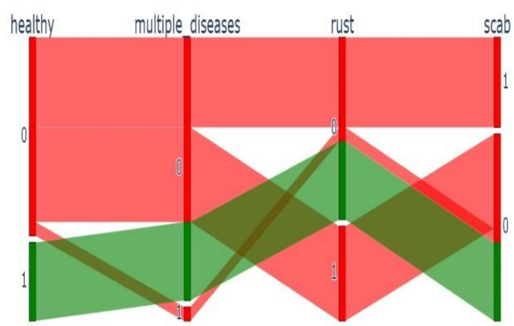
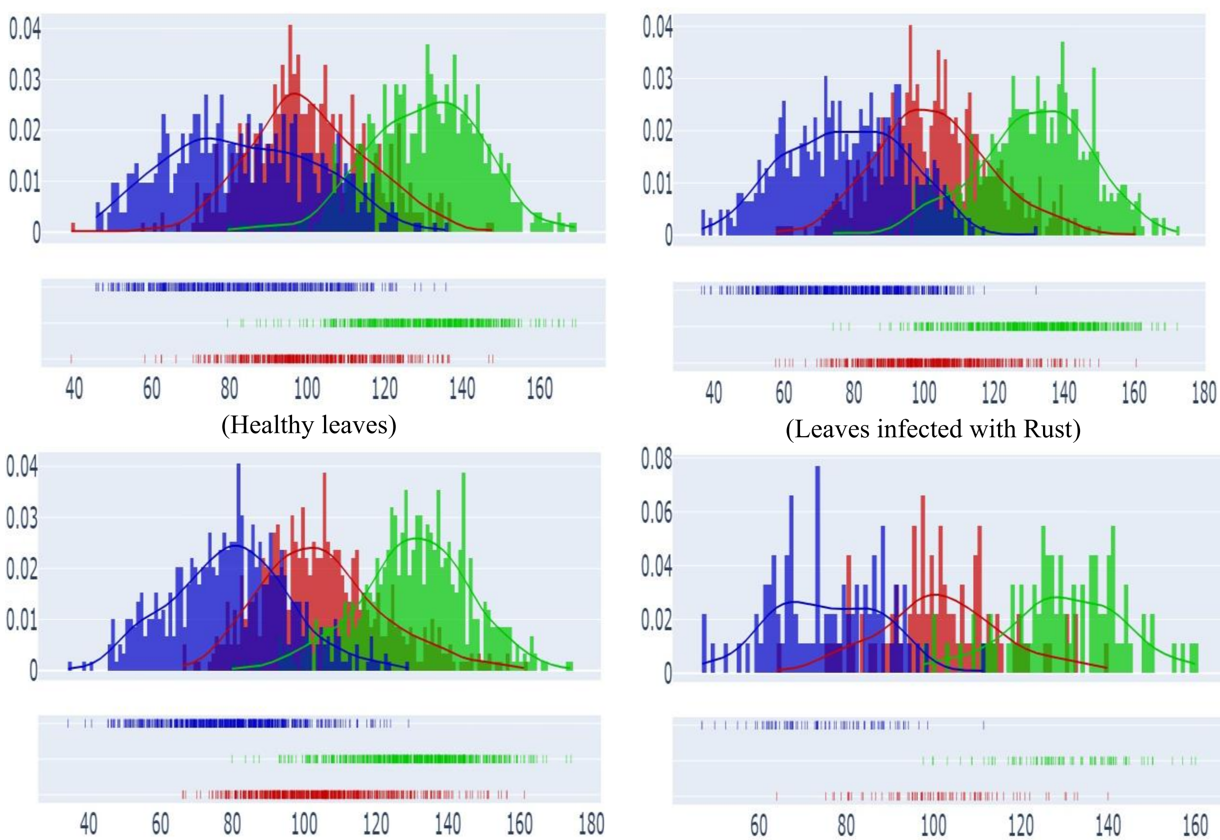


Figure 2. Class distribution in dataset and alluvial diagram.

The visualizations of the channel distribution to highlight image features (**Figure 3**, **Figure 4**) present a channel-wise comparison of leaves with different disease classes. In red channel distribution, densities for different leaf classes vary, but the mean value remains in the range of 90 - 100. Red-channel distribution is nearly the same for scab and rust classes so it could not be helpful to differentiate between these diseases.

Green channel distributions are also roughly equal, but the distribution on the blue channel varies greatly for all leaf classes. This inference concludes that the blue-channel holds some vital information regarding the class of the leaves. This finding influenced the decision to emphasize features obtained from the blue channel in preprocessing, which could enhance the ability of the model to distinguish between disease categories. To improve the image quality before submitting them to the model for training and to enhance the model's performance, the authors apply image pre-processing techniques. To ensure identical dimensions, we resize the images to a standardized resolution of 256 × 256 while preserving the aspect ratio to lower the computation power. It involves adjustments to color, dimensions, and orientation to fit the input size specified by the model design. Normalizing pixel values of the images on a scale of 0 to 1 maintains uniformity in pixel values and facilitates efficient model training. To achieve this, every pixel's value is divided by 255, which restricts values within the required range. The image augmentation technique strives to artificially increase the training dataset size producing transformed imitations of the dataset's original images (Chlap et al., 2021). Image augmentation processes such as zooming, rotation at 20-degree, shear range of 0.2, brightness range between 0.5 to 1.5, vertical flipping, and horizontal flipping are applied during the training of the model. Image augmentation may also address the class imbalances in the dataset resulting in a more generalized model performance. Keras Image data generator is used in this study to generate augmented images while training a model so that it is not required to store these extended sets of images.



(Healthy leaves) (Leaves infected with Rust)

(Leaves infected with Scab) (Leaves infected with multiple diseases)

Figure 3. The channel distribution of healthy leaves, leaves infected with rust, leaves infected with scab, and leaves infected with multiple diseases.

3.2 Image Segmentation

This section covers the image segmentation steps performed on the apple leaf images. For background elimination, and to avoid bias in retrieved features, we transform input RGB images using canny edge detection to detect prominent edges in the images highlighting areas of significant intensity change. Canny edge detection can effectively recognize edges and borders in images and reduce noise. It provides better edge localization and noise robustness than other edge detection methods (Sobel or Prewitt operator). So, it is appropriate for exact edge identification in complex apple leaf images. The canny operator first smooths the image using a Gaussian convolution and then applies 2-D first derivation to highlight the smoothed image areas with elevated first positional derivatives. It uses a Gaussian kernel and linear filtration to minimize the background noise and evaluates edge direction and intensity for each pixel in the smoothened image. The pixels that endure the thinning process are recognized as candidate edge pixels. Each edge pixel in this process has its edge intensity set to zero provided, it is not higher than the edge intensities of the two neighbouring pixels in a gradient direction. After that, hysteresis employs a pair of thresholds for edge intensity on the image having the thinner edge strength. All edge pixels under the minimal threshold are classified as non-edges. Following edge detection, the watershed transformation is applied to segment the image into regions based on the edges detected. This segmentation isolates distinct areas of the plant leaves, potentially separating healthy regions from affected areas. Watershed transformation can segment images based on the gradient details and nearby intensity variations, splitting them into regions of interest. It uses an automated and data-driven process for segmentation, unlike threshold-based segmentation approaches, which may struggle with complicated image structures due to manual threshold selection. Watershed transformation is effective for segmenting regions with inconsistent shapes or overlapped structures, which are usual characteristics of damaged apple leaves. **Figure 5** presents the visualizations of canny edge detection on the apple leaf images and **Figure 6** presents the visualization of watershed transformation on the edge-emphasized apple leaf images.

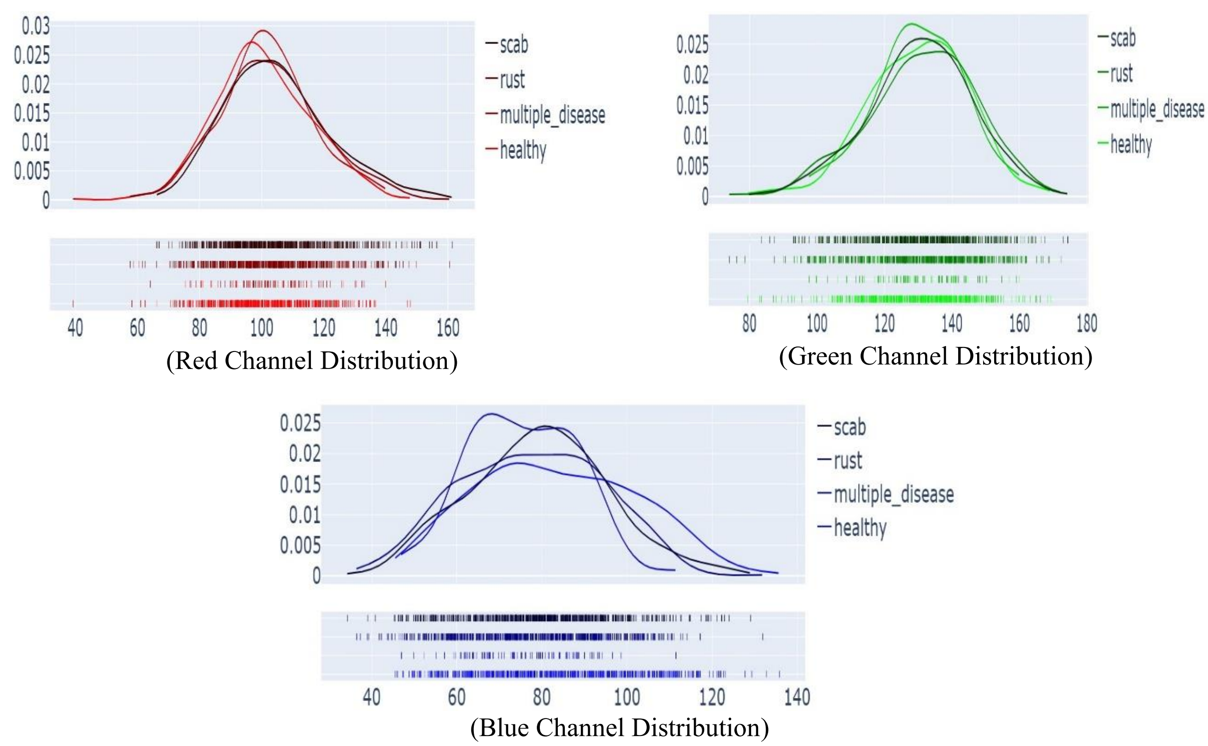


Figure 4. Red, green, and blue channel distributions of all classes.

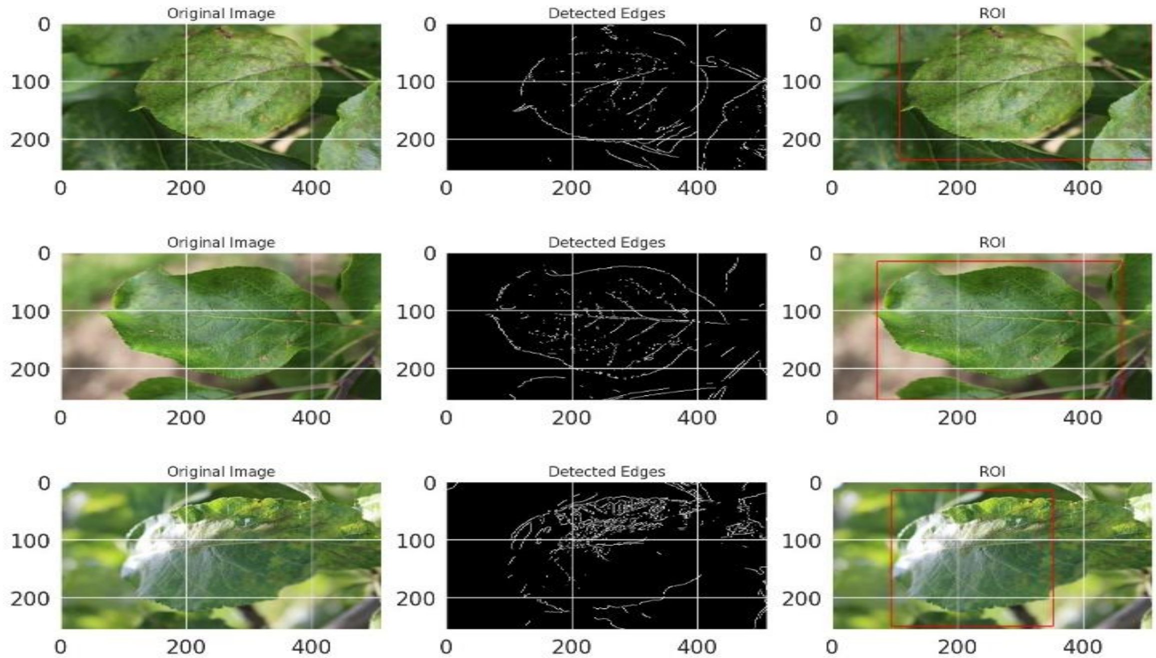


Figure 5. Canny edge detection on the apple leaf images.

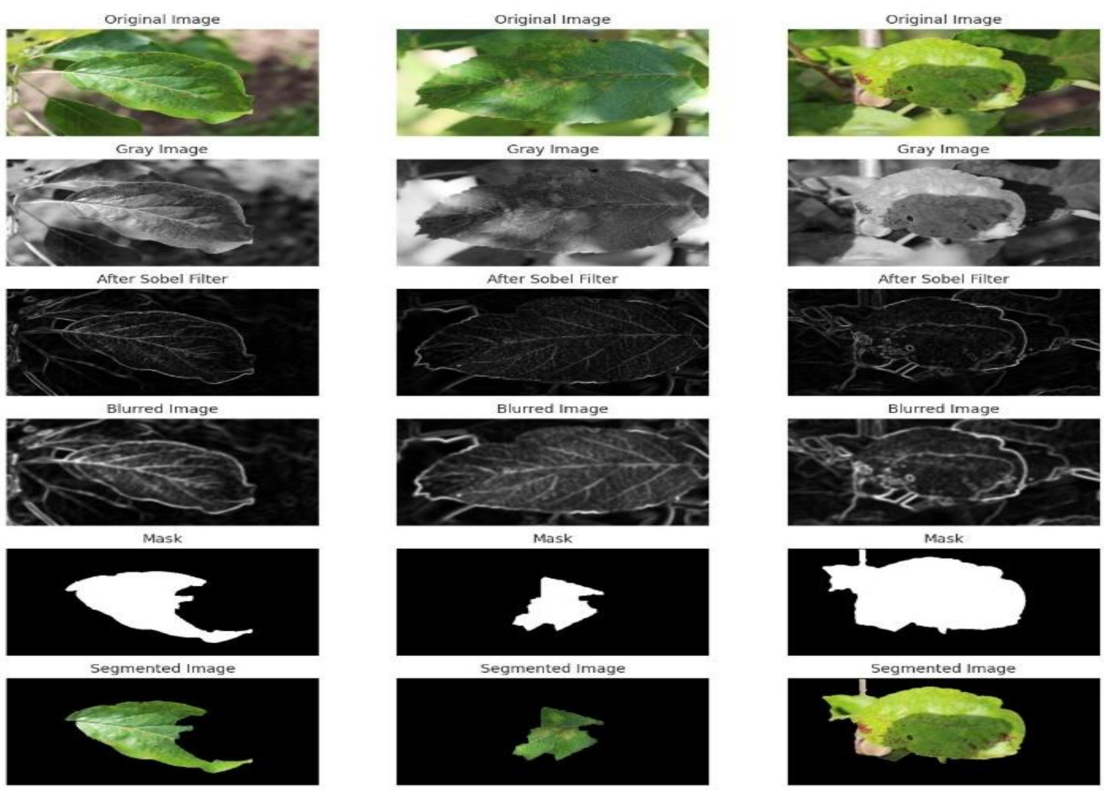


Figure 6. Watershed Transformation operation on the edge emphasized images.

The main principle of watershed transformation is to view an image as a topographical map, with lowlands represented by a low intensity and the highlands by a high level of intensity. The segmentation starts by adding various colored water to every isolated valley (the local minimum). According to the neighbouring elevations, the water begins to blend as it rises, coming from many valleys, each with its distinct color. Blockades control wherever the water mixes. Until all the heights submerge, it keeps filling the water and constructing blockades. These barriers determine the segmentation. Canny edge detection indicates object borders, assisting in accurate feature extraction for classification applications. Watershed transformation narrows these borders, allowing for more precise object segmentation. It improves the model's capability to discriminate between classes by generating a more detailed and meaningful image representation. Finally, it improves the model's accuracy by minimizing noise and emphasizing the essential features for image classification.

3.3 Formulation of Classification Problem

The plant disease detection strategy in this study follows a supervised learning approach for identifying diseases in apple leaves. Assuming that there are T_s training instances in apple leaf dataset D= {P, Q}. Here, P represents the all the apple leaf images in the dataset and Q denotes the labels according to their actual classes. P is denoted as, $P = \{p_1 + p_2 + p_3 + \cdots p_{Ts}\}$ and Q is denoted as $Q = \{q_1 + q_2 + q_3 + \cdots q_{Ts}\}$. The $Q_s \in [1,2,3,4]$, where '1' represents common rust, '2' represents scab, '3' represents multiple diseases, and '4' represents healthy class. The output of the classifier is expressed as a function, $f^{(x)} : P \rightarrow R$. In this, (x) denotes the parameters. Depending on these parameters, the predicted label may differ from the actual label. This difference is the prediction error which has to be minimized by tuning the parameters repeatedly to enhance the classification accuracy of the model.

3.4 Inception-v3 Model

The Inception-v3 architecture is chosen as the primary model for apple leaf disease detection due to its proven effectiveness in image classification tasks. With its roots as a Google Net module, Inception-v3 is a CNN model useful in object recognition and image processing (Szegedy et al., 2015). Inception v3 is preferred over other pre-trained models for image classification due to its balance of precision and processing capability. Its inception modules make it easier to extract detailed characteristics across various levels, which improves performance. Also, the broad use and availability of previously trained weights make integration and deployment easier in many different kinds of applications. Inception-v3 architecture (Szegedy et al., 2016) for a multiclass classification problem considered in this study (**Figure 7**) allows greater network depth while restricting the generation of enormous parameters. It is more efficient and has a deeper structure than the previous versions without compromising computational speed. It leverages auxiliary classifiers to regularize and involves less computational cost.

The foundation of Inception-v3 is implementing multiple inception modules that are made up of many convolutions with variable-size kernels for obtaining features across different levels (**Figure 8** to **Figure 10**). Reduction modules (Figure 11, Figure 12) scale down spatial dimensions and lower the feature map's resolutions with max pooling and 1×1 convolutions. Major optimizations in the Inception-v3 model are the replacement of 5×5 conv2D layers with two 3×3 conv2D layers to decrease the computing costs and offer a 28% proportional gain. For more factorization, a 1×3 conv2D layer and a 3×1 conv2D layer replace 3×3 conv2D layers and a 1×7 conv2D layer and 7×1 conv2D layer replaces 7×7 conv2D layers. As a result, a Bi-layered network with an identical response field as in 3×3 conv2D and 7×7 conv2D emerges. If both input and output filter counts are the same, the bi-layered approach is 33% less expensive. The auxiliary classifiers support the network to learn discriminatory features at intermediary levels. It regularises the network by preventing it from depending too much on only one path of data flow.

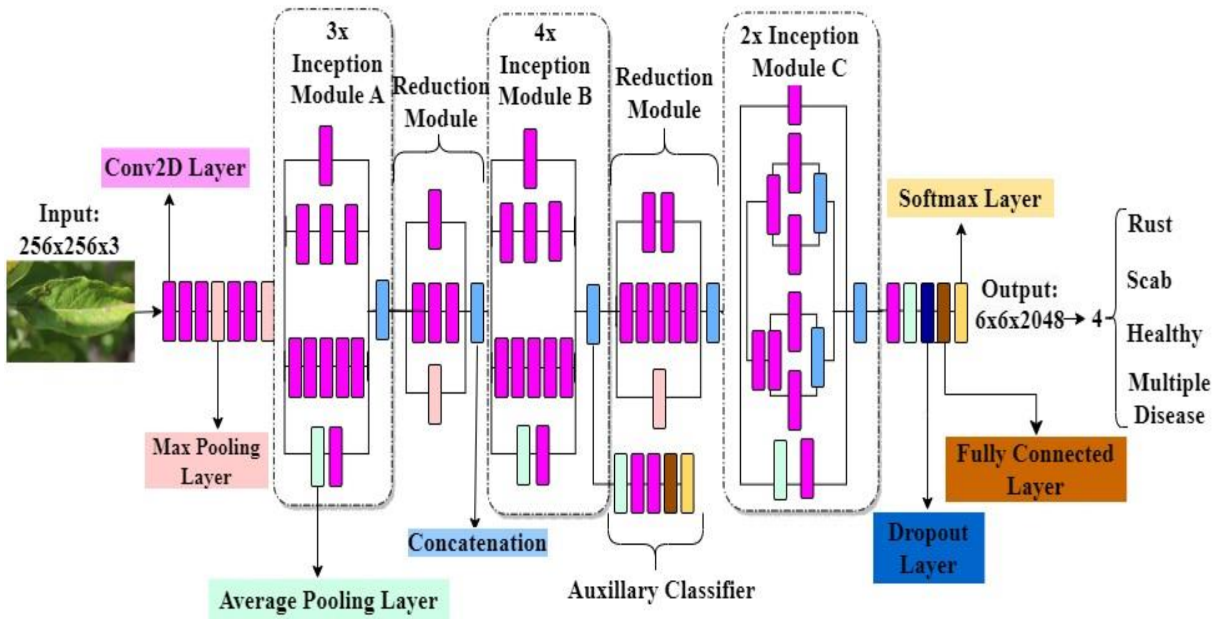


Figure 7. Inception-v3 architecture (Szegedy et al., 2016).

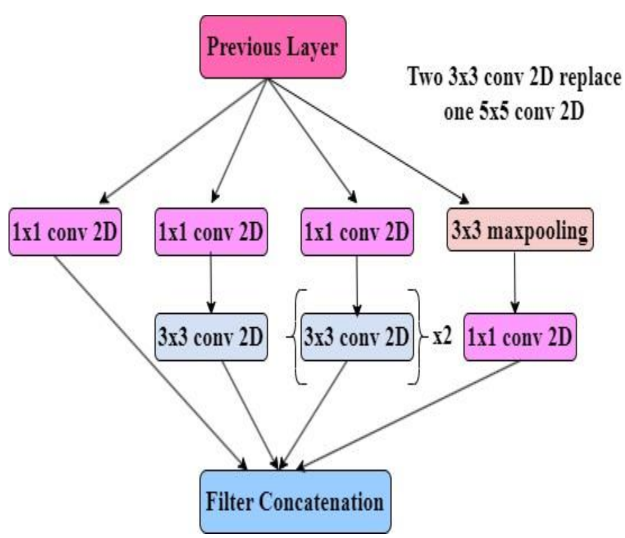


Figure 8. Inception module A (Szegedy et al., 2016).

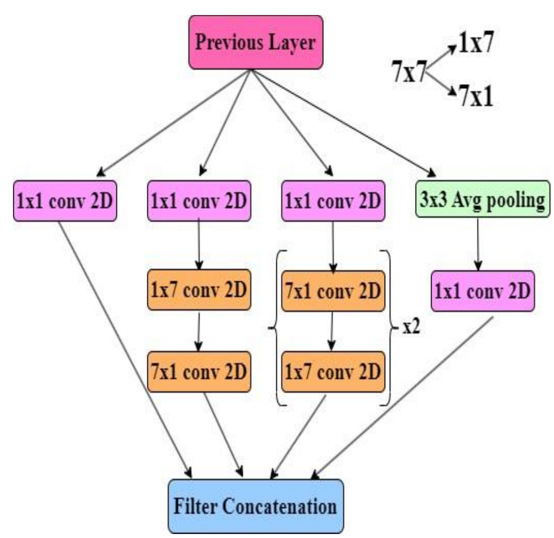


Figure 9. Inception module B (Szegedy et al., 2016).

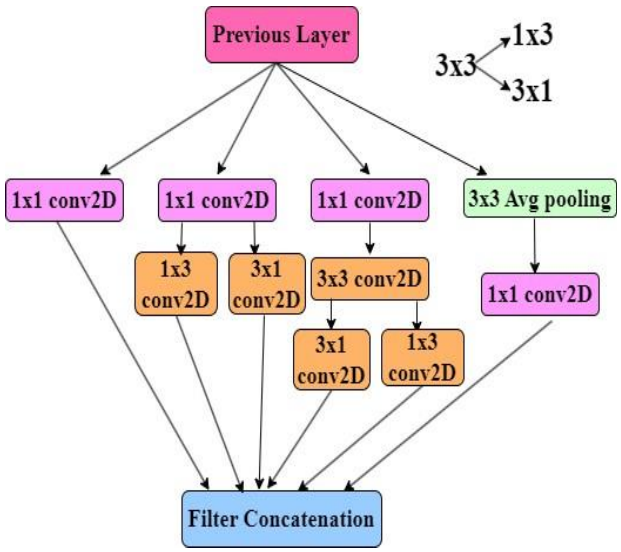


Figure 10. Inception module C (Szegedy et al., 2016).

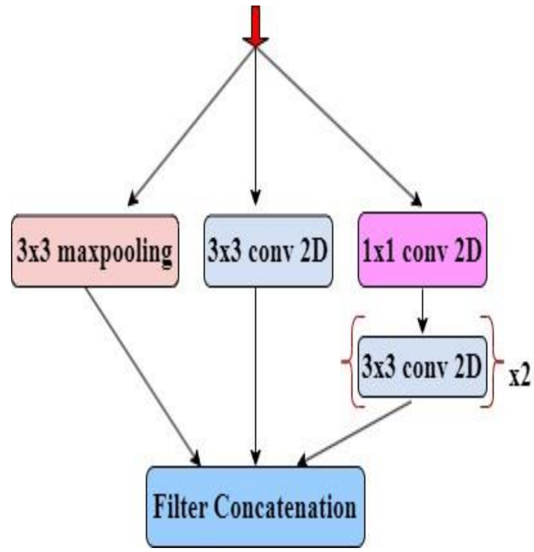


Figure 11. Reduction module A (Szegedy et al., 2016).

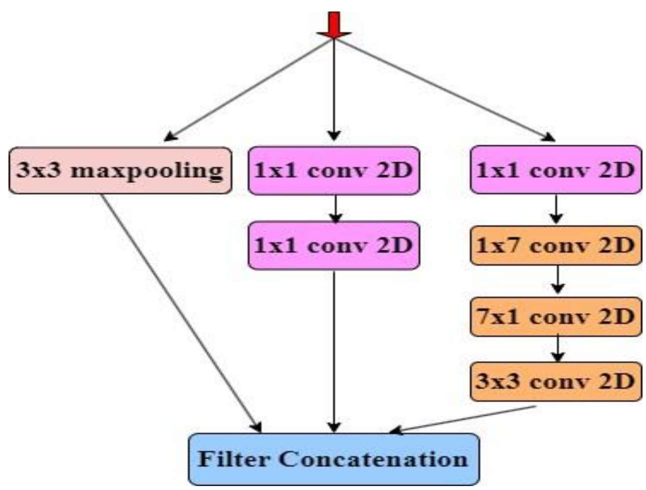


Figure 12. Reduction module B (Szegedy et al., 2016).

The Inception-v3 model is pre-trained on a large dataset, such as ImageNet, and serves as a powerful feature extractor.

3.5 Methodology

This section explains the proposed plant disease detection methodology. The authors initialize the Inception-v3 model with pre-trained weights and perform fine-tuning on pre-processed and segmented images. During fine-tuning, we freeze the initial layers to retain the learned features from ImageNet and only allow the last few layers to adapt to our specific disease detection task. The original Inception-v3 architecture comprises 107 layers, including conv2D, pooling, and fully connected layers. The convolution layers in Inception-v3 consist of variable size filters such as 1×1 filter, 3×3 filter, and 5×5 filter, facilitating multi-scale feature extraction. During fine-tuning for plant disease detection, filter sizes are adjusted and customized in specific layers to capture disease-specific patterns at different scales. The model is fine-tuned by retraining specific layers while freezing most of the pre-trained network. During this fine-tuning procedure, a batch size of 64 is employed, and 15 extra layers are modified to customize the Inception-v3 model to detect apple plant diseases. To achieve the best-performing arrangement for the current classification task, experiments are run with different batch sizes and hyper settings. These fine-tuned layers contain adjustments to filter sizes and learning rates, which improve the model's ability to detect disease patterns in plant images. A batch size of 64 works adequately for the model and dataset through experimentation. For training the model, an Adam optimizer and a cyclic learning rate ranging from 0.0001 to 0.001 are used. It allows for dynamic adjustment of the learning rate in the training process. The loss function used is categorical cross-entropy. Training spans 20 epochs as the loss was stabilized and did not improve further. The experimental steps for plant disease detection are presented in Algorithm 1. The hyperparameters and their values used in model training are presented in (**Table 1**). **Figure 13** illustrates the framework for the recommended plant disease detection approach.

Algorithm 1 Experimental steps for apple leaf disease detection

Input: Apple Leaf images
Output: common rust, scab, multiple disease, and heathy leaf classification
begin
    $D_{train} \leftarrow$ Training Leaf Images
    $R_{train} \leftarrow$ Resize (D_{train})
    $C_{train} \leftarrow$ Canny_Edge_Detect (R_{train})
    $S_{train} \leftarrow$ Watershed_Transform (C_{train})
    Apply stratified five-fold cross-validation
    Initialize k=1
    while(k<=5)
    begin
        $T_{train}, T_{valid} \leftarrow$ Split S_{train} into Train and Validation sets
        $AUG_{train} \leftarrow$ Perform data augmentation on T_{train}
        $CNN_{finetuned} \leftarrow$ Load Inception-v3 model
        $Disease\_Detect\_Model \leftarrow$ Train ($CNN_{finetuned}, AUG_{train}, T_{valid}$)
        $R_{Precision, Recall, F1\ score, Accuracy} \leftarrow$ Predict ($Disease\_Detect\_Model, T_{valid}$)
        Print $R_{Precision, Recall, F1\ Score, Accuracy}$
    end
end
    Computer average of all folds
    Print result

Table 1. Hyper parameters used in model training.

Hyperparameters	**Assigned Values**
Optimizer	Adam
Loss Function	Categorical Cross-Entropy
Epochs	20
Batch-Size	64
Learning Rate	Cyclic Learning Rate (0.0001 to 0.001)

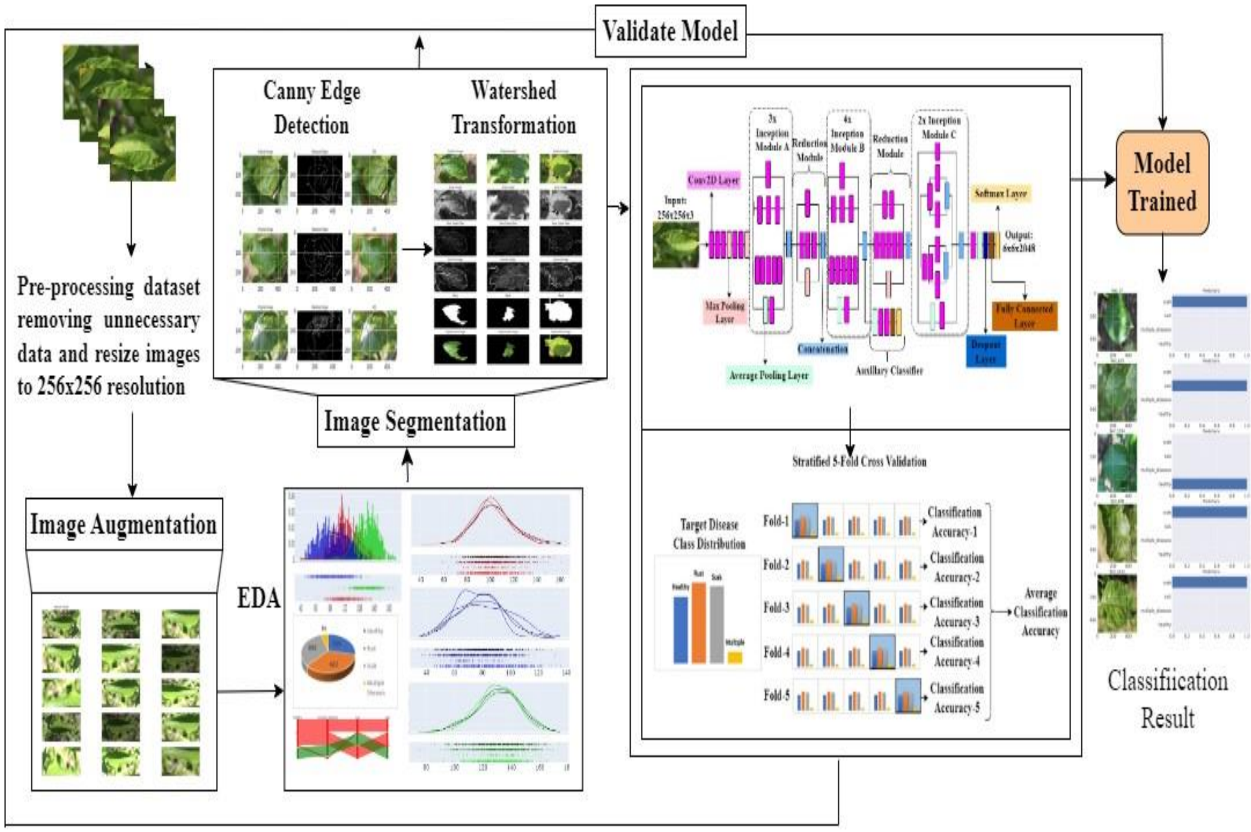


Figure 13. Proposed plant disease detection framework.

The first layer is the fine-tuned conv2D layer is adjusted to a filter size of 3×3 to capture intermediate-level features specific to plant disease. This layer is followed by a second fine-tuned conv2D layer with a filter size of 3×3 to capture complementary features in various disease patterns. A max-pooling layer was introduced after the fine-tuned convolutional layers to down-sample the learned features without changing filter sizes. After the pooling operation, a conv2D layer utilizes a 5×5 filter size, emphasizing the extraction of more complex spatial patterns related to certain plant diseases. Another fine-tuned conv2D layer follows with a 3×3 filter size, further enhancing the capture of diverse disease-related features. Then a pooling layer down samples without altering the filter sizes of the learned features. After the pooling layer, there are two Conv2D layers with 3×3 and 5×5 filter sizes respectively. These conv2D layers are followed by a max pool layer, a global averaging, a dense, and then a dropout layer to prevent overfitting without impacting filter sizes. It includes another dense layer followed by a final SoftMax layer for four-class classification. The activation layer finalizes the model's predictions and maintains filter sizes from the preceding layers. The SoftMax activation function provides probability scores for each class.

4. Performance Evaluation

This section covers the evaluation metrics that are used to evaluate the model's predictive performance, implementation details and the result of the experiments.

4.1 Evaluation Metrics

Precision: It represents the ratio of correctly identified positive cases (TP) to the total number of cases identified as positive, including both true and false positives (FP).

$$Precision = \frac{TP}{TP+FP} \tag{1}$$

Recall: It measures the proportion of real cases that were fittingly identified by a model, emphasizing the model's ability to capture all positive instances.

$$Recall = \frac{TP}{TP+False\ Negatives(FN)} \tag{2}$$

F1 score: The F1 score strikes an equilibrium between precision and recall providing one value that represents the harmonic mean of these two values and an overall evaluation of a model's performance.

$$F1\ score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{3}$$

Accuracy: It is a commonly used evaluation metric for a multi class classification. It measures the proportion of correctly categorised cases from the total examples in the data.

$$Accuracy = \frac{TP+True\ Negatives(TN)}{TP+TN+FN+FP} \tag{4}$$

4.2 Implementation Details

The model is trained using Google Colab with a Tesla K80 GPU and 12 GB RAM. The Keras package is used for implementing the model in Python. The model is trained using the cyclic learning rate and an Adam optimizer with a categorical cross-entropy loss function. Cyclic learning rate speeds up training by enabling the learning rate to oscillate between the lowest and the highest values. Relative to the fixed learning rate, this method assists in the model's rapid convergence. It drives the model to analyze various areas of the loss terrain, improving generalization ability. The cyclic learning rate is immune to hyper parameter selection. Finding an ideal learning rate becomes simple as it self-adapts the learning rate through the training progression. It has a regularisation impact that prevents overfitting and results in higher-quality models. Following the experiment (**Figure 14**), it is apparent from evaluations of the slopes of the loss function at various training stages that the learning rate was too low and that the loss function failed to improve between 10^{-5} and 10^{-4}. So, a learning rate range between 10^{-4} and 10^{-3} is chosen for model training. There is a significant loss reduction. Then the learning rate exceeds the optimum range and starts to diverge.

Adam optimizer is a prevalent optimization algorithm that integrates the advantages of AdaGrad and RMSProp optimization algorithms. Adam keeps a running average of both the gradients and the subsequent moments of the gradients' magnitude, adapting the learning rate for each parameter individually (Reyad et al., 2023). This enables the optimizer to perform effectively on a diverse set of optimization tasks while handling sparse gradients effectively. The optimizer work as follows:

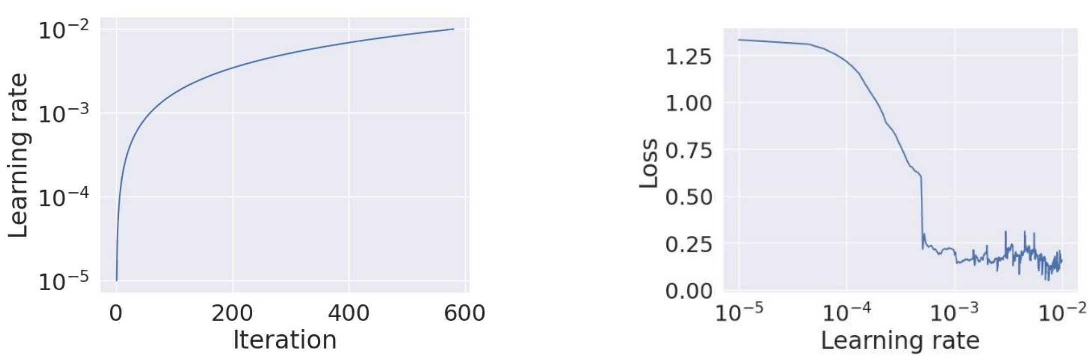


Figure 14. Cyclic learning rate and loss.

Initialize parameters: θ (model parameters)

- Initialize first moment variable: $m = 0$
- Initialize second moment variable: $v = 0$
- Initialize time step: $t = 0$
- Compute gradients.
- Update First Moment (Mean).
- Update Second Moment (Uncentered Variance) Estimate.
- Perform bias Correction.
- Update parameters.

The Adam optimizer uses the following equations for parameter updates:

$$g = \nabla\theta J(\theta) \tag{5}$$

$$m = \beta_1 * m + (1 - \beta_1) * g \tag{6}$$

$$v = \beta_2 * v + (1 - \beta_2) * g^2 \tag{7}$$

$$\widehat{m} = \frac{m}{(1 - {\beta_1}^t)} \tag{8}$$

$$\hat{v} = \frac{m}{(1 - {\beta_2}^t)} \tag{9}$$

$$\theta = \theta - \alpha * \frac{\widehat{m}}{(\sqrt{\hat{v}} + \varepsilon)} \tag{10}$$

In the above equations, the parameters m and v are running averages of the first and subsequent moment, respectively. $J(\theta)$ is the loss function, $\beta 1$, β_2 are exponential rates in decay for first and subsequent moment r. $\widehat{m}$ and $\hat{v}$ are bias-correct first and second moment estimates. These moment approximations are partial towards zero in the early training steps, specifically when β_1 and β_2 are close to 1. α is the learning rate, and ε is a small constant to prevent divide by zero condition. The superscript $"t"$ represents the current time step, and $\nabla\theta\, J(\theta)$ denotes the gradient of loss with respect to θ.

Categorical Cross-Entropy (CCE) loss function used in this study to compute the difference between the expected and true probability distributions of classes.

$$CCE = -\sum_{n=1}^{\infty}\left(y_{true} * log(y_{pred})\right) \tag{11}$$

Here y_{true} is the categorical ground truth label vector and y_{pred} is the predicted probability distribution over classes.

The plant pathology dataset considered in the study is unbalanced. Only 5% of the total leaf images belong to the multiple-disease category. Therefore, the authors applied stratified K-Fold cross-validation to address the problem with the classification of unbalanced class distribution in the dataset. In this study, stratified 5-fold cross-validation (**Figure 15**) divides the dataset into five equally sized random groups, ensuring that the proportion of data of each class in every group is nearly proportional in the overall dataset. The technique is repeated five times, employing four groups for training and one for testing and changing the test set around so that every group acts as the test set just once. After each fold, the model's precision, recall, F1 score, and accuracy are calculated. After performing 5-folds the average values of precision, recall, F1 score, and accuracy are generated that give validation of the model performance.

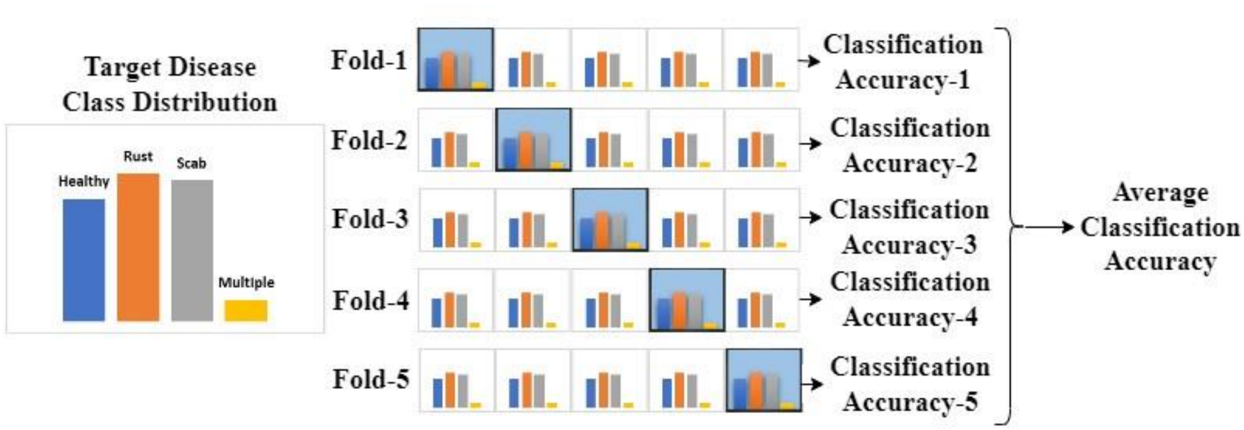


Figure 15. Stratified 5-fold cross-validation.

5. Results

Here, we provide the results of the experiments for image classification. After cross-validation, the model was trained for 15 steps with 20 epochs, to assess its performance. The training process resulted in the convergence of the Inception-v3 model after 20 epochs. A stable decrease in the training loss indicates that the model was effectively learning during training. The accuracy of the training set reached 100% at the end of training. (**Figure 16**) shows training Loss and training Accuracy. After applying the cross-validation, the best values for precision, F1 score, and accuracy are achieved in fold-4 as 94%, 89%, and 96.4% respectively. The best recall value of 92% is achieved in fold-1. The minimum precision value of 73%, minimum F1 score value of 81%, and minimum accuracy of 92.4% are reported in fold-1. The lowest value of 83% for recall is received in fold-4. The model achieved the overall cross-validated average 84.6% precision, 87.4% recall, 85% F1-score, and 94.76% accuracy. **Table 2** presents the image classification results based on precision, recall, F1 score, and accuracy for each fold during the 5-fold stratified cross-validation process. These results demonstrate the consistency of the model's performance across different folds and indicate the model's ability to generalize well to different subsets of the dataset. **Figure 17** and **Figure 18** presents average cross-validation accuracy and loss. **Figure 19** visually shows the classification performance of the model on five leaf images randomly selected from the test data.

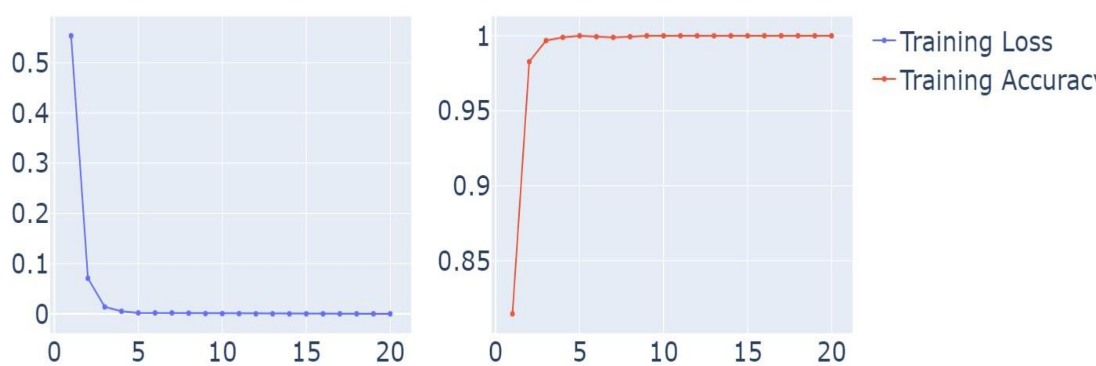


Figure 16. Training loss and accuracy.

Table 2. Results obtained by the proposed model for plant disease detection on validation set. Accuracy represents the overall accuracy of all healthy, rust, scab, and multiple disease classes.

S. No.	Fold Number	Precision	Recall	F1-Score	Accuracy
1.	Fold-1	73%	92%	81%	92.4%
2.	Fold-2	79%	90%	83%	93.5%
3.	Fold-3	88%	87%	87%	96.2%
4.	Fold-4	94%	83%	89%	96.4%
5.	Fold-5	89%	85%	85%	95.3%
6.	Average	84.6%	87.4%	85%	94.7%

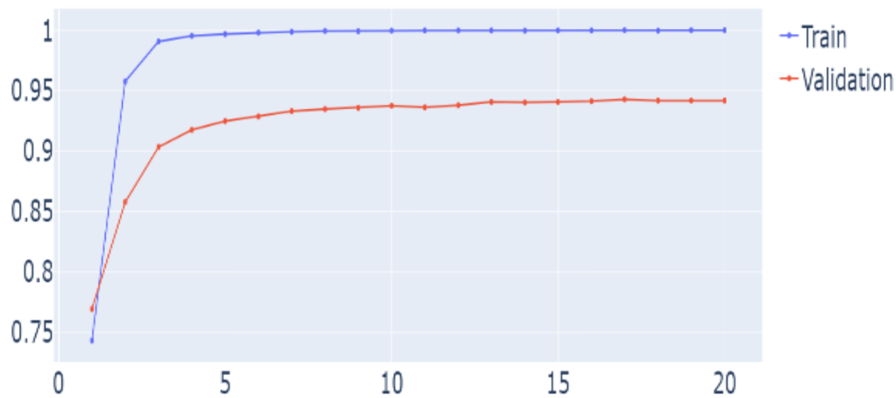


Figure 17. Average cross-validation accuracy.

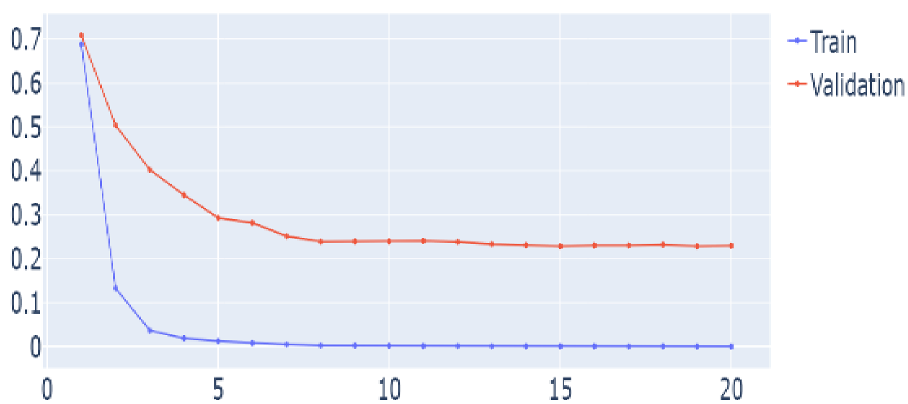


Figure 18. Average cross-validation loss.

6. Discussion

In this discussion, we delve into the implications of our findings in the proposed approach. The exploratory data analysis and channel distributions provided valuable insights into the dataset's characteristics. Understanding the color distributions of healthy and diseased leaves informed our pre-processing strategy and model design, contributing to the model's successful performance. The stratified 5-fold cross-validation yielded consistent and reliable performance evaluation of the proposed model. It ensures the appropriate distribution of the classes at every fold, decreasing biases in model evaluation for datasets with imbalances. It provides a reliable measure of model performance by averaging findings across numerous folds, which improves generalizability. Furthermore, it optimizes data to use during the training and validation process, resulting in a more reliable assessment of model performance. To compare the proposed work with previous works on plant disease detection some papers are identified in the literature.

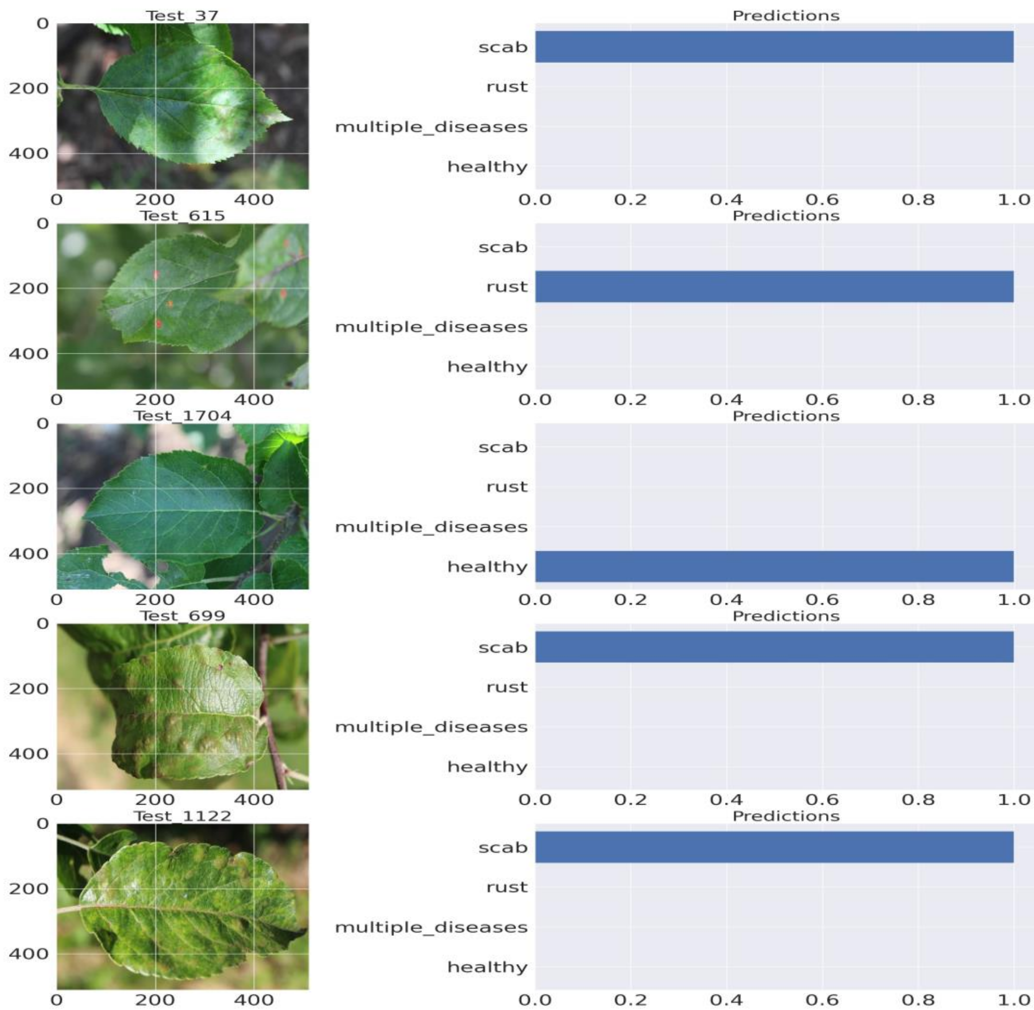


Figure 19. Model classification performance on test data.

Jin et al. (2018) proposed a 2DCNN with GRU and achieved an F1 score of 75% and an overall accuracy of 74.30%. Sudhesh et al. (2023) proposed an XceptionNet-based leaf disease classification model and obtained 81.40% accuracy. A faster RCNN model was proposed by Khan et al. (2022) and used a self-collected strawberry dataset. The model achieved a precision of 42% and an accuracy of 88%. Another research study (Hu et al., 2019; Sibiya & Sumbwanyambe, 2021) used VGG16 to classify plant leaf images on the tea dataset and maize dataset obtained from the plant village dataset. They obtained 90% and 89% accuracy respectively. Chen et al. (2020) proposed a VGG and inception-based model and achieved 60.76% recall and 80.38% accuracy. Abbas et al. (2021) utilized an efficientNet-B3 model to classify self-collected strawberry leaves images and reported an 81% F1 score, 80% recall, 83% precision, and 86% accuracy. Another recent study, Hassan & Maji (2022) introduced an inception-based CNN model with residual learning on the cassava dataset and achieved a 66.91% F1 score, 72.63% recall, 62.03% precision, and 76.59% accuracy. Lv & Su (2024) proposed a CNN model using YOLO-v5 with an attention mechanism and transformer-based encoder network. The model showed improvement in performance over the original YOLO-v5. However, the model achieved a recall of 69.5%, precision of 70.9%, and accuracy of 92.4% on the plant pathology dataset. To the best of our knowledge, no research study in the existing literature on plant disease detection models integrated canny edge detection, watershed transformation, and inception v3 for the classification of plant leaves. The proposed inception v3 based model with canny edge detection and watershed transformation outperformed these recent plant disease detection models and achieved 84.6% precision, 87.4% recall, 85% F1-score, and 94.76% accuracy. In the proposed methodology for apple leaf disease detection, applying canny edge detection on the input images extracts the salient edges in the image and provides valuable structural information for disease identification. Based on the edge gradients, watershed transformations isolate the individual region of interest inside the image providing the most relevant and quality features to the inception v3 model for improved classification performance. The proposed model achieved better precision, accuracy, F1-score, and recall values than those presented by the previous studies. The results are more generalized as they are cross-validated and the average of precision, accuracy, F1-score, and recall values achieved in five folds are presented as the final resultant values. The high accuracy achieved in detecting various apple leaf diseases validates the model's robustness and generalization capability. It confirmed the model's ability to correctly identify diseased leaves. Focusing the study on detecting a small number of the most common and impactful apple diseases can lead to a more immediately useful tool for farmers, compared to a very broad classifier. This research study represents a significant advancement in the field of plant disease detection, surpassing several recent previous works (**Table 3**). To practically use the proposed model, it needs a user-friendly mobile application for quick uploading of plant leaf images and integration with present farming equipment and facilities to be easily adopted by the farmers. Regular testing and model refinement, considering adaptability and affordability, is critical for broad adoption and sustained viability in farming practices.

Table 3. Evaluation of the suggested model's performance in relation to previous studies based on F1-score, recall, precision and accuracy.

S. No.	Article	Model	Dataset	F1-score	Recall	Precision	Accuracy
1.	Jin et al. (2018)	2DCNN-GRU	Own dataset	75.00%	-	-	74.30%
2.	Hu et al. (2019)	VGG16	Own dataset (Tea leaves)	--	--	-	90.00%
3.	Chen et al. (2020)	VGG + Inception	Plant Village dataset	-	60.76%	-	80.38%
4.	Sibiya & Sumbwanyambe (2021)	VGG16	Plant Village dataset (Maize)	-	-	-	89.00%
5.	Abbas et al. (2021)	EfficientNet-B3	Own dataset (Strawberry)	81.00%	81.00%	83.00%	86.00%
6.	Khan et al. (2022)	Faster RCNN	Own dataset (Apple)	-	-	42.00%	88.00%
7.	Hassan & Maji (2022)	Inception based CNN with residual connection	Cassava Dataset	66.91%	72.63%	62.03%	76.59%
8.	Sudhesh et al. (2023)	XceptionNet	Rice dataset	-	-	-	81.40%
9.	Lv & Su (2024)	YOLO-v5 with attention and transformer block	Plant Pathology dataset	-	69.50%	70.90%	92.40%
10.	**Proposed Model**	**Inception-v3**	**Plant Pathology Dataset**	**85.00%**	**87.40%**	**84.60%**	**94.76%**

Although the proposed model obtained good results there are some areas in which the model may be improved. Most of the images are optimally segmented, with a small number of images having an extra region marked as a leaf. Leaves with multiple diseases show the segmentation result is a little bit poor. However, the experimental results demonstrate that slightly inadequate segmentation ability has no significant impact on disease categorization. Following are a few limitations of the proposed approach.

- Although exploratory data analysis and visualizing channel distribution in images provide insight into dataset properties, sometimes datasets may have heterogeneity, causing biases in training and testing the model.
- It is observed in the experiments that the segmentation approach works better when the images are of excellent resolution with the desired leaf in the image's central part. There are some instances where it does not produce the expected results.
- The performance analysis that uses stratified 5-fold cross-validation showed acceptable results but the model evaluation is limited to only one dataset. The model's effectiveness has to be validated if is used for plant image datasets with diversified features like different apple varieties, severity of diseases, or specific environmental conditions.

7. Conclusion

This paper proposes an innovative strategy for detecting apple disease employing the Inception-v3 design, which relies on canny edge detection and watershed transformation. The results demonstrate the efficacy and stability of the proposed strategy through experimentation and evaluation with stratified 5-fold cross-validation. The model exhibited great accuracy in recognizing many apple leaf diseases, outperforming existing methodologies and showcasing its utility for agricultural uses. Also, the exploratory data analysis helped identify aspects of the data. Channel distributions provided insights into the color distribution of healthy and diseased leaves. Finally, we illustrate an efficient method for detecting apple leaf disease using the inception-v3 design and image segmentation methods. The apple dataset used in this research study consists of four classes of leaves. Focusing on fewer classes can result in a more relevant, practical, and in-depth analysis of these specific classes, including identifying their features, causes, and appropriate mitigation measures. In some farming situations, focusing on the detection of a few common diseases may be more applicable and instantly useful to farmers than a broader classification strategy that covers many different kinds of diseases. The suggested approach offers various opportunities for subsequent studies, from model refinements to practical applications, thus leading to better and more profitable agricultural operations.

The following aspects may expand this investigation in the future.

- Field verification and testing in actual apple farms are required to evaluate the model efficacy in realistic conditions. Factors like computer resources and processing speed are crucial when deploying the model for farmers.
- Since the current investigation examines apple leaves only, expanding it to other crops might extend its effect on agriculture.
- The validation of the proposed approach on multiple datasets from different geographic locations and environments may help in evaluating its reliability and generalizability across a variety of conditions.
- Attention mechanisms can be applied to increase disease detection ability and emphasize relevant regions for improved classification precision.

Conflict of Interest

The authors guarantee that they have no conflicts of interest to disclose for this work.

Acknowledgments
No funding from public, private, or nonprofit organisations was given to this research. The editors and anonymous reviewers are appreciated for their insightful feedback that helped make this work better. The authors thank them for their contributions.

References

Abbas, I., Liu, J., Amin, M., Tariq, A., & Tunio, M.H. (2021). Strawberry fungal leaf scorch disease identification in real-time strawberry field using deep learning architectures. *Plants*, *10*(12), 2643. https://doi.org/10.3390/plants10122643.

Atole, R.R., & Park, D. (2018). A multiclass deep convolutional neural network classifier for detection of common rice plant anomalies. *International Journal of Advanced Computer Science and Applications*, *9*(1), 67-70.

Barbedo, J.G.A. (2019). Plant disease identification from individual lesions and spots using deep learning. *Biosystems Engineering*, *180*, 96-107. https://doi.org/10.1016/j.biosystemseng.2019.02.002.

Chen, J., Chen, J., Zhang, D., Sun, Y., & Nanehkaran, 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.

Chlap, P., Min, H., Vandenberg, N., Dowling, J., Holloway, L., & Haworth, A. (2021). A review of medical image data augmentation techniques for deep learning applications. *Journal of Medical Imaging and Radiation Oncology*, *65*(5), 545-563. https://doi.org/10.1111/1754-9485.13261.

Dinh, H.X., Singh, D., Periyannan, S., Park, R.F., & Pourkheirandish, M. (2020). Molecular genetics of leaf rust resistance in wheat and barley. *Theoretical and Applied Genetics*, *133*, 2035-2050. https://doi.org/10.1007/s00122-020-03570-8.

Ferentinos, K.P. (2018). Deep learning models for plant disease detection and diagnosis. *Computers and Electronics in Agriculture*, *145*, 311-318. https://doi.org/10.1016/j.compag.2018.01.009.

Hari, P., & Singh, M.P. (2023). A lightweight convolutional neural network for disease detection of fruit leaves. *Neural Computing and Applications*, *35*(20), 14855-14866. https://doi.org/10.1007/s00521-023-08496-y.

Hassan, S.M., & Maji, A.K. (2022). Deep feature-based plant disease identification using machine learning classifier. *Innovations in Systems and Software Engineering*, 1-11. https://doi.org/10.1007/s11334-022-00513-y.

Hu, G., Wu, H., Zhang, Y., & Wan, M. (2019). A low shot learning method for tea leaf's disease identification. *Computers and Electronics in Agriculture*, *163*, 104852. https://doi.org/10.1016/j.compag.2019.104852.

Jakjoud, F., Hatim, A., & Bouaddi, A. (2019). Detection of diseases on tomato leaves based on Sub–Classifiers Fuzzy Combination. *International Journal of Innovative Technology and Exploring Engineering, 8*(5), 735-739.

Jin, X., Jie, L., Wang, S., Qi, H.J., & Li, S.W. (2018). Classifying wheat hyperspectral pixels of healthy heads and Fusarium head blight disease using a deep neural network in the wild field. *Remote Sensing*, *10*(3), 395. https://doi.org/10.3390/rs10030395.

Khan, A.I., Quadri, S.M.K., Banday, S., & Shah, J.L. (2022). Deep diagnosis: A real-time apple leaf disease detection system based on deep learning. *Computers and Electronics in Agriculture*, *198*, 107093. http://dx.doi.org/10.2139/ssrn.4019467.

Lv, M., & Su, W.H. (2024). YOLOV5-CBAM-C3TR: An optimized model based on transformer module and attention mechanism for apple leaf disease detection. *Frontiers in Plant Science*, *14*, 1323301. https://doi.org/10.3389/fpls.2023.1323301.

Math, R.M., & Dharwadkar, N.V. (2022). Early detection and identification of grape diseases using convolutional neural networks. *Journal of Plant Diseases and Protection*, *129*(3), 521-532. https://doi.org/10.1007/s41348-022-00589-5.

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.

Rahman, S.U., Alam, F., Ahmad, N., & Arshad, S. (2023). Image processing-based system for the detection, identification and treatment of tomato leaf diseases. *Multimedia Tools and Applications*, *82*(6), 9431-9445. https://doi.org/10.1007/s11042-022-13715-0.

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*, 1-27. https://doi.org/10.1007/s11042-023-14969-y.

Reyad, M., Sarhan, A.M., & Arafa, M. (2023). A modified Adam algorithm for deep neural network optimization. *Neural Computing and Applications*, 1-18. https://doi.org/10.1007/s00521-023-08568-z.

Sankaran, S., Mishra, A., Ehsani, R., & Davis, C. (2010). A review of advanced techniques for detecting plant diseases. *Computers and Electronics in Agriculture*, *72*(1), 1-13. https://doi.org/10.1016/j.compag.2010.02.007.

Sibiya, M., & Sumbwanyambe, M. (2021). Automatic fuzzy logic-based maize common rust disease severity predictions with thresholding and deep learning. *Pathogens*, *10*(2), 131. https://doi.org/10.3390/pathogens10020131.

Sudhesh, K.M., Sowmya, V., Kurian, S., & Sikha, O.K. (2023). AI based rice leaf disease identification enhanced by Dynamic Mode Decomposition. *Engineering Applications of Artificial Intelligence*, *120*, 105836. https://doi.org/10.1016/j.engappai.2023.105836.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., & Rabinovich, A. (2015). Going deeper with convolutions. *In proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (pp.1-9). https://doi.org/10.1109/CVPR.2015.7298594.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the inception architecture for computer vision. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 2818-2826). https://ieeexplore.ieee.org/document/7780677.

Thakur, P.S., Sheorey, T., & Ojha, A. (2023). VGG-ICNN: A Lightweight CNN model for crop disease identification. *Multimedia Tools and Applications*, *82*(1), 497-520. https://doi.org/10.1007/s11042-022-13144-z.

Thapa, R., Snavely, N., Belongie, S., & Khan, A. (2020). The plant pathology 2020 challenge dataset to classify foliar disease of apples. *ArXiv Preprint ArXiv:2004.11958*. https://doi.org/10.48550/arXiv.2004.11958.

Too, E.C., Yujian, L., Njuki, S., & Yingchun, L. (2019). A comparative study of fine-tuning deep learning models for plant disease identification. *Computers and Electronics in Agriculture*, *161*, 272-279. https://doi.org/10.1016/j.compag.2018.03.032.

Zeng, W., & 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.

Zhang, X., Qiao, Y., Meng, F., Fan, C., & Zhang, M. (2018). Identification of maize leaf diseases using improved deep convolutional neural networks. *IEEE Access*, *6*, 30370-30377. https://doi.org/10.1109/ACCESS.2018.2844405.

Publisher's Note- Ram Arti Publishers remains neutral regarding jurisdictional claims in published maps and institutional affiliations.