Variable convolution kernel with feature fusion and transfer learning for leaf disease classification

Swapna Jamal*, J. E Judith

(Computer Application, Noorul Islam Centre for Higher Education, Thackalai, Thuckalay, Kumaracoil, Tamilnadu 629180, India)

Abstract: The advancement of automated frameworks for detecting and classifying leaf diseases is extensively explored in contemporary agricultural practices. The effectiveness of a classifier relies on the feature extraction process. A novel Variable Convolution Kernel (VCK) feature extraction algorithm with Feature Fusion (FF) and Transfer Learning (TL) based disease classification is proposed. Sparse representation is obtained in the training stage by fusing features obtained through different filters. TL offers the benefit of leveraging pre-trained models on large datasets, saving significant time and computational resources when building and training new models for specific tasks. Mobilenet_v2 pretrained using ImageNet dataset can improve model performance, especially when dealing with limited training data, by transferring weights and features. A novel framework has been developed by incorporating CNN, TL, FF and tuning the hyperparameters. The underlying algorithm is known as FF-TL-CNN algorithm. The empirical investigation utilized the Plant Village dataset. The leaf disease categories examined in this study encompass early blight, black rot, bacterial spot, apple scab, cercospora leaf spot, and the category of healthy leaves. FF-TL-CNN outperformed other classifiers by attaining 98.85% accuracy, 98.63% precision, 98.41% recall and 99.32% F1-score with Plant Village dataset. The research findings demonstrate that the suggested deep learning model and algorithm have practical applications in real-world computer vision contexts, particularly in the field of agriculture.

Keywords: Variable Convolution Kernel, Feature Fusion, Transfer Learning, Leaf Disease Classification, Deep Learning, Hyperparameter Tuning.

Citation: Jamal, S., and J. E Judith. 2025. Variable convolution kernel with feature fusion and transfer learning for leaf disease classification. Agricultural Engineering International: CIGR Journal, 27(3):178-194.

1 Introduction

The economy of a country heavily reliant on agriculture is significantly influenced by farming, as approximately 70% of the population depends on it for their livelihood. India, for instance, boasts a vast diversity of plant cultivation. However, plant diseases caused by fungi, bacteria, and viruses pose a major threat to agricultural productivity. These diseases drastically reduce crop yields, especially when detected late, leading to plant death and economic losses. For example, little leaf disease in pine trees is a severe condition that stunts growth and results in the death of the tree within six years. Early detection and intervention are critical to minimizing agricultural losses and ensuring satisfactory crop production. Traditional methods of disease detection, which rely on manual inspection by experts, are labor-intensive, costly, and often inaccurate, particularly when dealing with large agricultural fields (Hassanien et al., 2017).

Automated systems for leaf disease detection offer a more efficient and accurate alternative, even in the early stages of disease development (Barbedo, 2018). Plant diseases typically manifest symptoms in

Received date: 2024-09-12 **Accepted date:** 2025-04-08

***Corresponding author: Swapna Jamal.** Computer Science-DIP, Noorul Islam Centre for Higher Education, Thackalai, Thuckalay, Kumaracoil, Tamilnadu, 629180, India. Email: swapnashan@gmail.com.

leaves, stems, flowers, and fruits, with leaves being the most accessible and early indicators of disease. Consequently, researchers have focused on developing methods for leaf disease classification using advanced algorithms, including supervised and unsupervised classifiers (Barbedo et al., 2016; Sharif et al., 2018). Supervised classifiers, in particular, rely on training with known data to classify unknown samples, making them highly effective for disease detection (Yang et al., 2015).

The integration of technological innovations in agriculture, such as precision farming, has the potential to reduce costs and increase productivity (Phadikar et al., 2012). Automated monitoring systems powered by artificial intelligence (AI) and deep learning (DL) are revolutionizing disease identification and enabling timely interventions (Al Bashish et al., 2011). DL techniques, particularly convolutional neural networks (CNNs), have shown remarkable promise in automating plant disease classification, leveraging advancements in computing power and machine learning algorithms (Gavhale and Gawande, 2014).

Traditional methods of disease detection, such as visual inspection, are prone to delays in identifying diseases, leading to significant crop damage (Zhang et al., 2019). Recent research has emphasized the use of technology-driven approaches to overcome these challenges. However, implementing machine learning (ML) and DL techniques for leaf disease detection presents several hurdles. These include the need for high-quality leaf images from remote locations and the challenge of ensuring image quality for further processing (Majumdar et al., 2014; Li et al., 2018). Additionally, the scarcity of large-scale training data for diverse crops limits the effectiveness of DL algorithms, underscoring the need for comprehensive datasets to improve detection accuracy (Singh et al., 2019).

Accurate identification of the region of interest in leaf images and the availability of robust training data is critical for effective disease detection. Digital image processing introduces complexities related to segmentation and processing accuracy, necessitating adaptive algorithms for machine interpretation of images (Dhingra et al., 2019). Despite these challenges, DL offers significant advantages in automating disease detection, with its growing prominence in AI-driven computer vision applications (Aparajita et al., 2017). The adoption of AI-driven solutions for real-time monitoring and disease detection holds immense potential for enhancing agricultural productivity and ecosystem health (Patil and Kumar, 2011).

Recent advancements in DL have further expanded its applications in agriculture. For instance, studies have demonstrated the effectiveness of CNNs and transfer learning (TL) in identifying plant diseases with high accuracy (Hassan et al., 2021; Majji and Kumaravelan, 2021). Additionally, the integration of multi-feature fusion techniques and hybrid models, such as CNN-LSTM networks, has improved the classification of complex agricultural datasets (Ding et al., 2022; Li et al., 2022). These innovations highlight the transformative potential of AI in addressing global agricultural challenges.

The global adoption of automated disease detection systems through technological innovation can significantly impact agricultural practices, enabling precision farming and reducing crop losses through timely interventions (Sullca et al., 2019). This research is motivated by the potential of AI-driven solutions to revolutionize agriculture and promote sustainable farming practices. The following sections are organized as follows: Section 2 provides an overview of related works on leaf disease classification. Section 3 introduces the proposed leaf detection and classification algorithm. Section 4 presents the experimental results and analysis of the proposed work. Finally, Section 5 concludes the study.

2 Related works

This section discusses several related works on leaf disease classification algorithms. Dhingra et al. (2019) introduced a method combining K-means segmentation, gray-level co-occurrence matrix, and support vector machine (SVM) classifiers for disease detection. Aparajita et al. (2017) developed an automated system for detecting late blight disease in potato leaves using adaptive thresholding to segment disease-affected regions. However, their work was limited to late blight disease in potatoes and did not address other diseases. Li et al. (2017) proposed an algorithm for detecting downy mildew disease in cucumber plants, while Kalaivani and Shantharajah (2017) presented a method for identifying diseases in brinjal leaves using a clustering algorithm and artificial neural networks (ANN). The segmentation was performed using the K-means clustering algorithm, and classification was achieved through a neural network (Patil and Kumar, 2011; Patil and Bodhe, 2011).

Paul and Munkvold (2005) utilized recursion and artificial neural networks to detect gray leaf spot in maize. Their approach involved correlation and recursion analysis to identify effective predictor variables, with 60% of the leaf images used for training and the remaining for testing. Chtioui et al. (1999) employed a generalized regression neural network (GRNN) to predict disease based on leaf wetness, as different diseases exhibit varying degrees of wetness. Pydipati et al. (2006) used discriminant analysis to detect citrus diseases, leveraging statistical classification algorithms with color and texture features.

Voulodimos et al. (2018) conducted a comprehensive study on DL detection models, including CNNs and recurrent neural networks (RNN), for agricultural applications. Their research highlighted the effectiveness of DL models, particularly CNNs, in detecting leaf diseases across various crops. They emphasized the importance of extensive training data and explored configurations such as dropouts to enhance model performance. Tiwari et al. (2020) investigated pre-trained DL models like VGG16 and emphasized the potential of CNN-based models for improving detection accuracy. Gandhi et al. (2018) integrated CNNs with Generative Adversarial Networks (GANs) to address data insufficiency through data augmentation, creating new training samples to improve model performance.

Karthik et al. (2020) incorporated a residual approach within a CNN model to detect tomato diseases, achieving high accuracy through optimized layer configurations and dropouts. Zhong and Zhao (2020) explored the use of DenseNet-121 for multi-label classification of apple diseases, accommodating multiple class labels to reflect the complexity of crop diseases. Sullca et al. (2019) developed a modified LeNet-CNN baseline for disease detection in maize, featuring multiple layers to handle input data effectively.

Recent advancements in DL have further improved leaf disease detection. For instance, Hassan et al. (2021) utilized CNNs and TL to achieve high accuracy in identifying plant diseases. Majji and Kumaravelan (2021) demonstrated the effectiveness of CNNs on the PlantVillage dataset for disease classification. Sethy et al. (2020) employed deep feature-based SVM for rice leaf disease identification, achieving robust performance. Liu and Wang (2020) proposed a MobileNetv2-YOLOv3 model for early detection of tomato gray leaf spot, highlighting the potential of lightweight models for real-time applications. Ding et al. (2022) introduced a multi-feature fusion approach combining graph neural networks (GNN) and CNNs for hyperspectral image classification, demonstrating improved accuracy in complex datasets.

These studies collectively underscore the importance of advancing CNN-based models and leveraging techniques such as TL, data augmentation, and multi-feature fusion to enhance leaf disease detection. However, existing approaches often face limitations such as suboptimal model configurations, underutilization of pre-trained models, and accuracy issues. This study aims to address these shortcomings by developing a comprehensive model that incorporates advanced CNN architectures, feature fusion, hyperparameter tuning, and TL g techniques to improve the accuracy and effectiveness of leaf disease detection.

3 Methodology

This section delivers an in-depth look into the specific procedures and practices utilized in each component of the proposed classification model. The initial phase, image acquisition, involves gathering data and is pivotal in shaping the dataset's quality and diversity. Following this, preprocessing steps, including augmentation and resizing, are applied to ensure that the dataset is well-prepared and optimized for CNN-based analysis. Dataset division is a crucial step, allowing data to be categorized into subsets for training, validation, and testing, facilitating robust model evaluation and generalization. The core of the methodology lies in the combination of CNN and TL and feature fusion (FF), responsible for extracting intricate features and patterns from the preprocessed data (Hassan et al., 2021). Finally, performance evaluation is essential for assessing the model's accuracy and effectiveness, offering insights into its suitability for real-world applications. The seamless integration of these sections within the block diagram ensures a systematic and comprehensive approach to image analysis, classification, and evaluation, paving the way for the subsequent detailed exploration of each component within this methodology. The proposed model's block diagram is depicted in Figure 1.

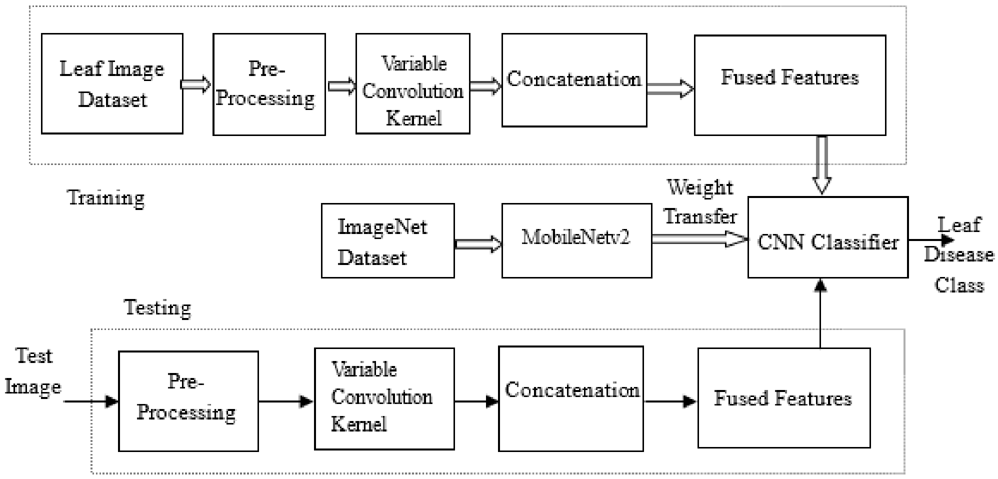
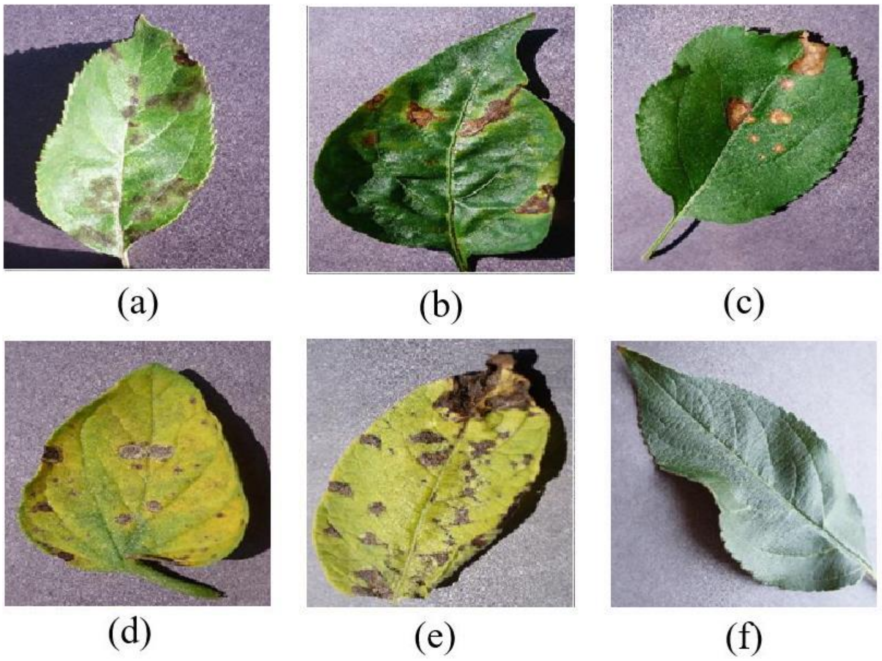


Figure 1 Block diagram of proposed FF-TL-CNN model for leaf disease classification



(a) apple scab (b) bacterial spot (c) black rot (d) cercospora leaf spot (e) early blight (f) healthy

Figure 2 Sample images

The proposed system encompasses the application of two distinct preprocessing techniques to the images sourced from the Plant Village dataset (Majji and Kumaravelan, 2021). Image augmentation is employed to expand the dataset by creating additional images. In the context of Deep Neural Networks, having a larger quantity of images is crucial for achieving optimal training and enhancing validation accuracy. The augmentation process encompasses various image processing techniques, such as flipping, rotation, and cropping, which contribute to diversifying the dataset and improving the model's capability to recognize different variations of the same object (Coulibaly et al., 2019).

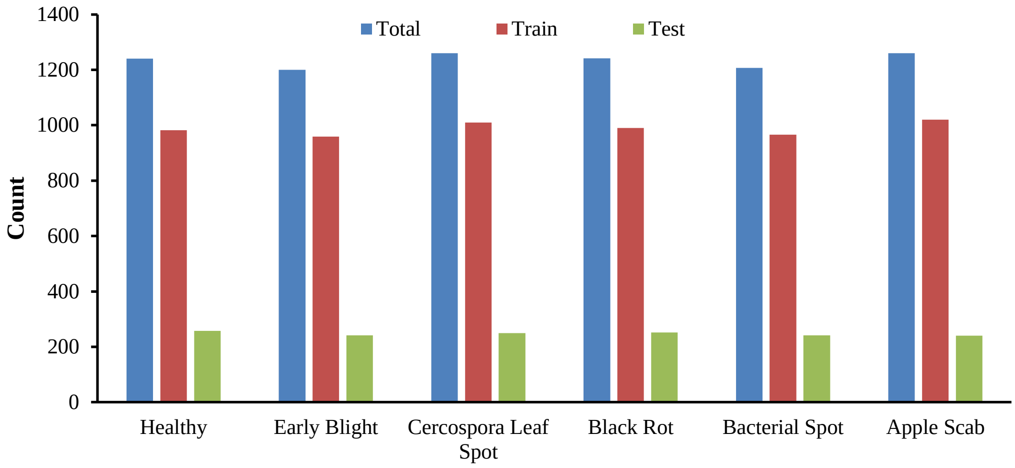


Figure 3 Leaf image class

Image resizing serves the purpose of reducing the computational complexity associated with the DL process. When dealing with large-sized images, processing a higher number of pixels simultaneously leads to increased computational time and complexity. To address this, the size of the leaf images is standardized to 256×256, which optimizes computational efficiency while retaining the essential information within the images. This resizing step ensures that the DL model can handle the data more efficiently without compromising its ability to extract meaningful features. Sample images from the Plant Village dataset are displayed in Figure 2.

In the development and evaluation of DL models, the process of dataset division plays a pivotal role. It involves a methodical partitioning of the dataset into discrete subsets, typically consisting of a training set and a testing set. More advanced techniques may also include a validation set in this partitioning. The training set serves as the cornerstone for instructing the model, enabling it to learn and recognize patterns, features, and relationships present in the images. Conversely, the testing set functions as an impartial benchmark for assessing the model's performance. This process ensures that the model doesn't merely memorize the images in the training set but can accurately make predictions in real-world applications. Effective dataset division is imperative for constructing robust, dependable, and efficient machine learning models, as it helps safeguard against issues like overfitting and facilitates the rigorous evaluation of a model's predictive capacity (Sethy et al., 2020). In the context of the proposed FF-TL-CNN model, 80% is allocated for training, while 20% is used for testing. Detailed information about the distribution of various image classes within the dataset can be found in Figure 3.

3.1 Convolutional neural network

The CNNs stands as an advanced DL framework capable of autonomously extracting meaningful insights from data, all without the need for human intervention. CNNs shine particularly in pattern recognition tasks, finding their strength in image analysis for the identification of objects, individuals, and scenes. These networks encompass a multitude of layers, often numbering in the tens or even hundreds, with each layer performing a distinct function in discerning different aspects of an image. In the training process, every image, regardless of its original resolution, undergoes a sequence of filtering operations. The output from each transformed image serves as the input for the subsequent layer. To capture intricate object characteristics, these filters typically commence by detecting fundamental attributes such as contours and edges. The fundamental structure of the CNN utilized in the leaf image classification application can be visualized in Figure 4.

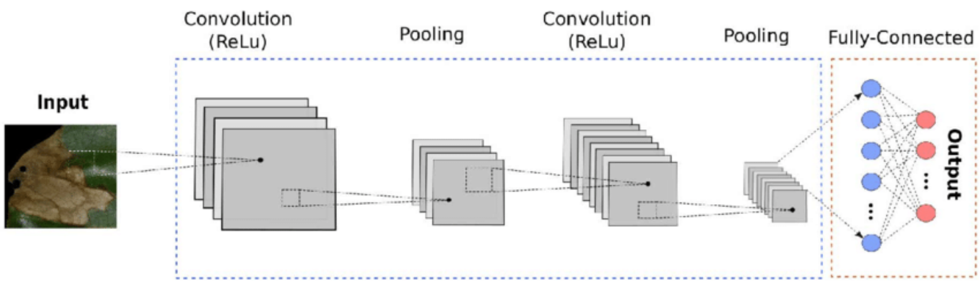


Figure 4 Basic structure of CNN for Leaf image classification

Traditional neural networks rely on matrix multiplications as their fundamental operation, CNNs introduce a distinctive technique known as convolution. In mathematical terms, convolution is an operation that combines two functions to create a third function, illustrating how the shape or characteristics of one function are influenced by the other. In the context of CNNs, this convolution operation plays a pivotal role in processing and extracting features from visual data, allowing the network to effectively analyze and recognize patterns and structures within images. In CNNs, the fundamental operation involves taking a filter or kernel, typically represented as 1x1, 3×3 or 5×5 matrices, and applying it to the input image. This process results in the generation of a convolved feature, which reflects the localized patterns and features in the input data that are captured by the filter. The convolved feature is then passed on to the next layer in the network. This progression through multiple layers, each performing their own convolutions and transformations, allows the CNN to extract hierarchical features and representations from the input image, enabling it to recognize complex patterns and objects within the image data. The convolution operation using a 3×3 kernel is depicted in Figure 5.

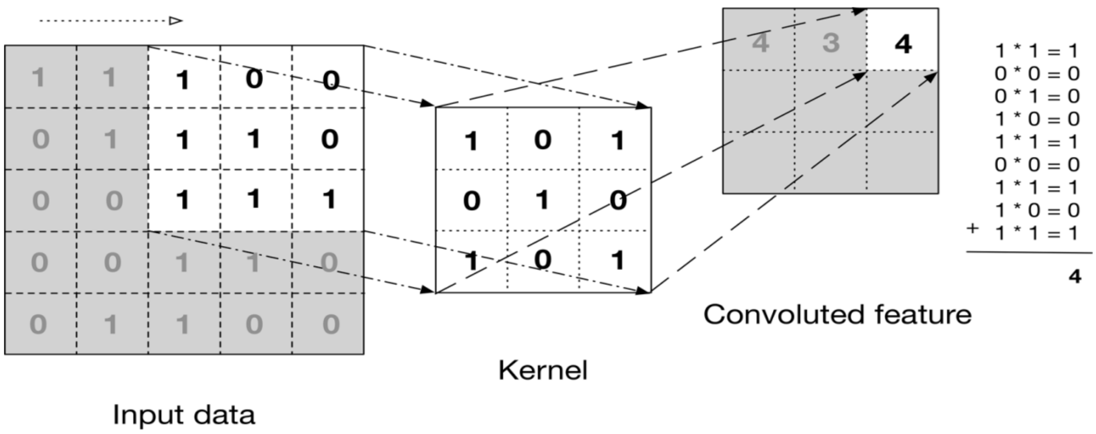


Figure 5 Convolution operation using 3×3 kernel

Max pooling entails selecting the maximum element from a designated region of the feature map covered by a filter, as illustrated in Figure 6. This operation plays a vital role in downsizing the spatial dimensions of the feature map, effectively reducing its size. In more detail, when a max-pooling layer is applied, it results in an output feature map that highlights the most prominent and significant features from the preceding feature map. By selecting the maximum value within each region, the max-pooling operation retains key information while reducing computational complexity. This allows the CNN to focus on essential features and patterns, leading to more efficient and effective feature extraction.

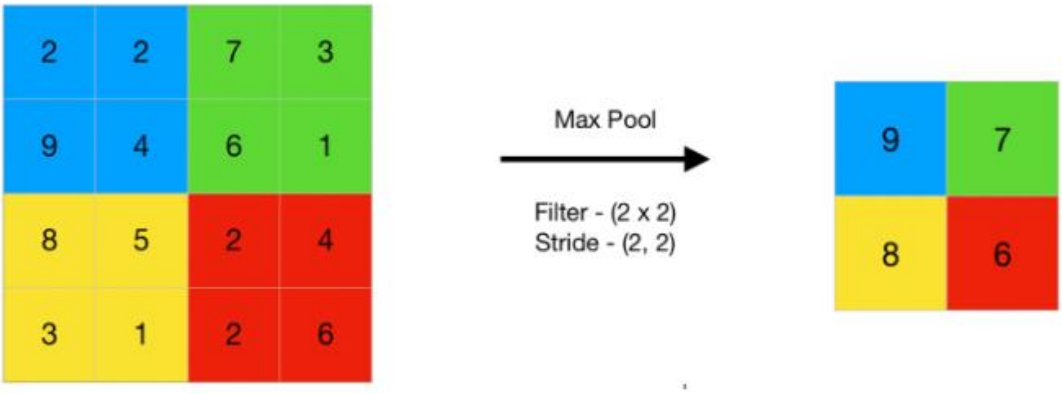


Figure 6 Max pooling operation

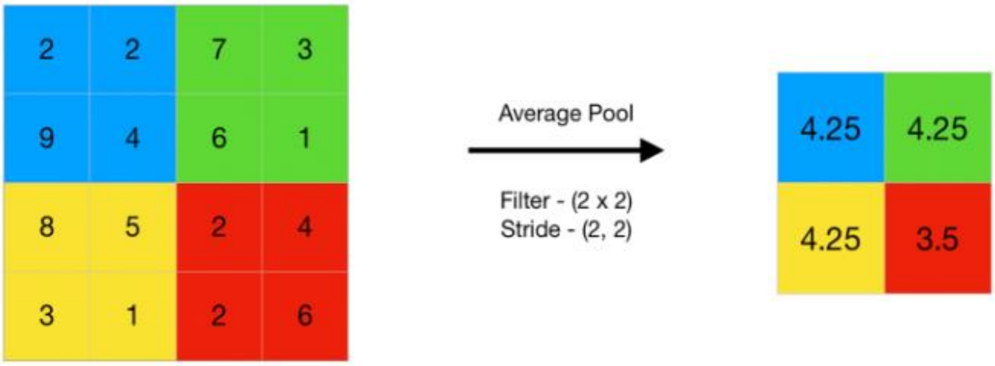


Figure 7 Average pooling operation

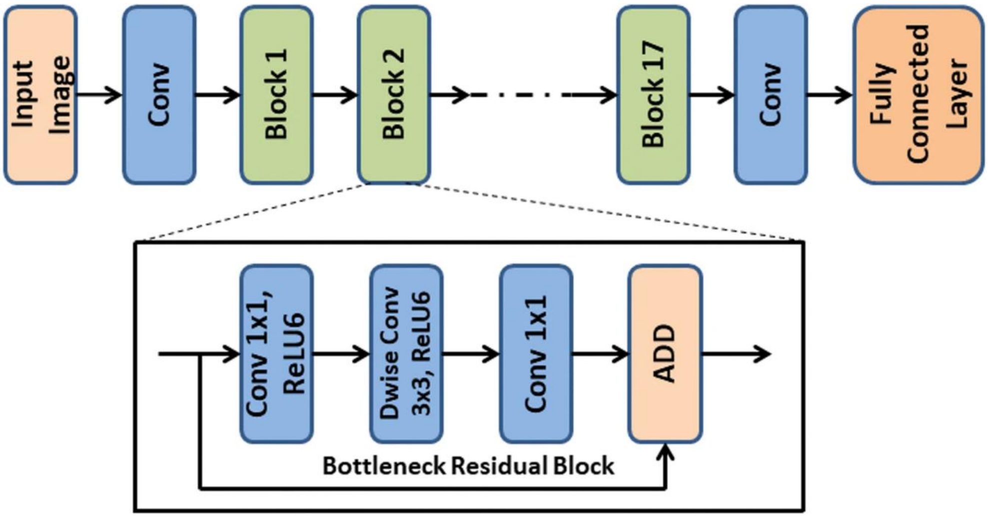


Figure 8 Architecture of MobileNetV2

Average pooling is a fundamental operation commonly used within CNNs. It entails calculating the average of the elements within a specific region of a feature map, as determined by the coverage of the filter. In more detail, when average pooling is applied, it calculates the mean value of the features within a patch or region of the feature map. This process effectively smoothens and reduces the spatial dimensions of the feature map, providing an aggregated representation of the underlying information. Average pooling is valuable for preserving essential information while decreasing computational complexity, making it a useful tool in the feature extraction process of CNNs. A sample average pooling operation is illustrated in Figure 7.

3.2 Transfer learning using MobileNetV2

TL allows to influence pre-trained models on huge datasets to solve new, related tasks more efficiently. MobileNetV2, as mentioned earlier, is an efficient CNN architecture that can be effectively used for TL. TL takes a pre-trained model, fine-tuning it on a new, smaller dataset, and adapting it to perform a specific task (Liu and Wang, 2020). Valuable features from a large dataset are copied and the model is built on this knowledge to improve performance on your task. The architecture of MobileNetv2 is illustrated in Figure 8.

MobileNetV2 is part of the MobileNet family of architectures developed by Google. Its primary focus is on efficient feature extraction while minimizing computational and memory resources. The fundamental concept behind MobileNetV2 is the application of depth-wise separable convolutions, which divides conventional convolutions into two distinct operations: depth-wise convolution and pointwise convolution. In the depth-wise convolution operation, a distinct convolutional filter is applied to each input channel. If the input has C channels and the filter size is $K \times K$, it results in C separate convolutions, one for each channel. For a depth-wise convolution operation, the output feature map Y at location (x, y) can be defined using Equation 1.

$$Y_{i,j,k}=\sum_{l=1}^{C} X_{x+i,y+j,l} \times K_{i,j,l,k} \tag{1}$$

Where, $Y_{i,j,k}$ represents output feature at (i, j) in channel k. $X_{x+i,y+j,l}$ represents input feature at $(x+i, y+j)$. $K_{i,j,l,k}$ defines depth-wise convolutional kernel for channel l to channel k. Pointwise convolution is essentially a 1×1 convolution applied to $Y_{i,j,k}$. It helps to mix the features obtained from depth-wise convolutions, increasing the network's representational power (Mahesh et al., 2023). For a point-wise convolution operation, the output feature map Y can be defined using Equation 2.

$$Y_{i,j,k}=\sum_{l=1}^{C} X_{i,j,l} \times K_{l,k} \tag{2}$$

The output of the pointwise convolution is then passed through batch normalization and ReLU activation, creating a new block of features. These blocks are stacked to form the MobileNetV2 architecture, and the network is characterized by its width multiplication factor (α) and resolution hyperparameter (ρ), which control channel count and resolution, respectively. The initial step involves utilizing a pre-trained MobileNetV2 model, which has undergone training on an extensive dataset for tasks such as image classification. We denote this pre-trained model as M_{pre}. To adapt M_{pre} for a specific task, it is necessary to fine-tune it on a new dataset related to the task. Fine-tuning refers to the process of adjusting the model's parameters based on the characteristics of the new dataset. The general idea is to retain the lower layers (early layers) of the MobileNetV2 because these layers capture more generic features, like edges and textures. It is possible to replace or fine-tune the upper layers to specialize in your task. The output of MobileNetV2, Y_{pre} before fine-tuning can be expressed using Equation 3.

$$Y_{pre} = M_{pre}(X) \tag{3}$$

Where, Y_{pre} is the output feature map before fine-tuning. M_{pre} represents the pre-trained MobileNetV2 model. X is the input image. After fine-tuning, a new output, Y_{fine} is generated, which is specialized for the given task as provided in Equation 4.

$$Y_{fine} = M_{fine}(X) \tag{4}$$

Where, Y_{fine} is the output feature map after fine-tuning. M_{fine} represents the fine-tuned MobileNetV2 model. During fine-tuning, a loss is evaluated in terms of cross-entropy for classification tasks or mean squared error for regression. Loss (L) computes the difference between ground truth and predicted labels as explained in Equation 5.

$$L = Loss\left(Y_{fine}, Y_{gt}\right) \tag{5}$$

MobileNetV2 achieves competitive accuracy while using significantly fewer parameters and computational resources compared to larger models, making it well-suited for resource-constrained environments. It enables real-time image classification on mobile devices, making it a crucial component for applications like leaf disease classification. MobileNetV2 is utilized as a feature extractor for TL tasks, and the proposed model can be fine-tuned for specific tasks (Mahesh et al., 2023).

3.3 Variable convolution kernel based feature fusion

To address potential issue of overfitting in the proposed architecture, the sizes of convolution kernels have been selected as 1×1, 3×3, and 5×5. This architectural design depicted in Figure 9 amalgamates all these layers, with their output filter banks fused through Depth Concatenation, resulting in a unified output vector, which can be provided as the input for the subsequent stage (Ding et al., 2022).

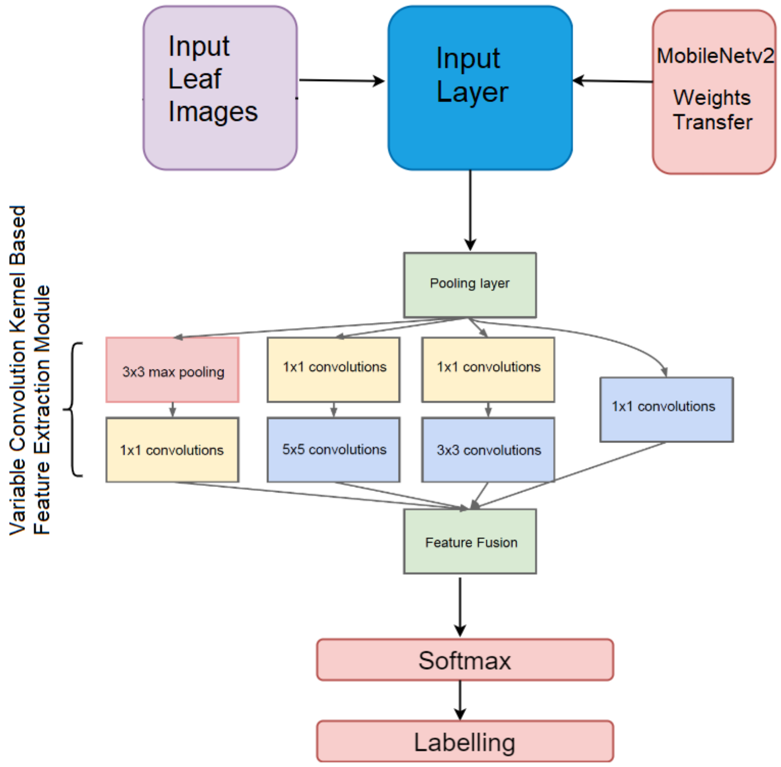


Figure 9 Proposed feature fusion architecture

Feature fusion in CNN is a technique used to combine information from different layers or pathways within the network. The goal is to enhance the network's ability to capture and integrate features at various levels of abstraction, leading to improved performance. In the proposed architecture, feature maps from different kernels (filters) are concatenated along the channel dimension (Li et al., 2022). This creates a feature map with a higher channel dimension, incorporating information from multiple sources. Feature concatenation is used in the design of modules, where parallel convolutional operations of different kernel sizes are performed, and their results are concatenated to capture features at multiple scales. Feature fusion with variable kernel can occur at different stages of a CNN as depicted in Figure 10.

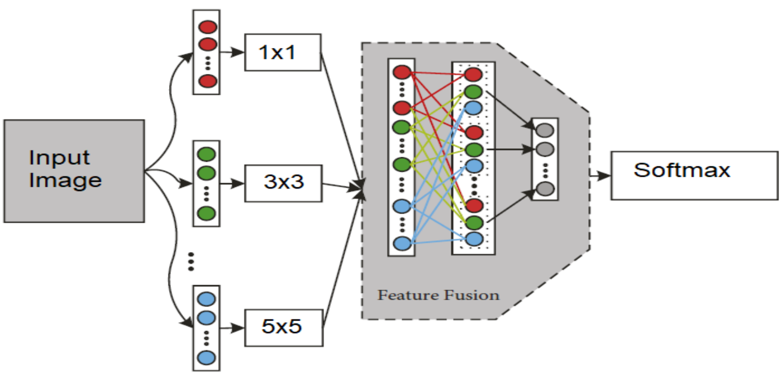


Figure 10 Proposed feature fusion process

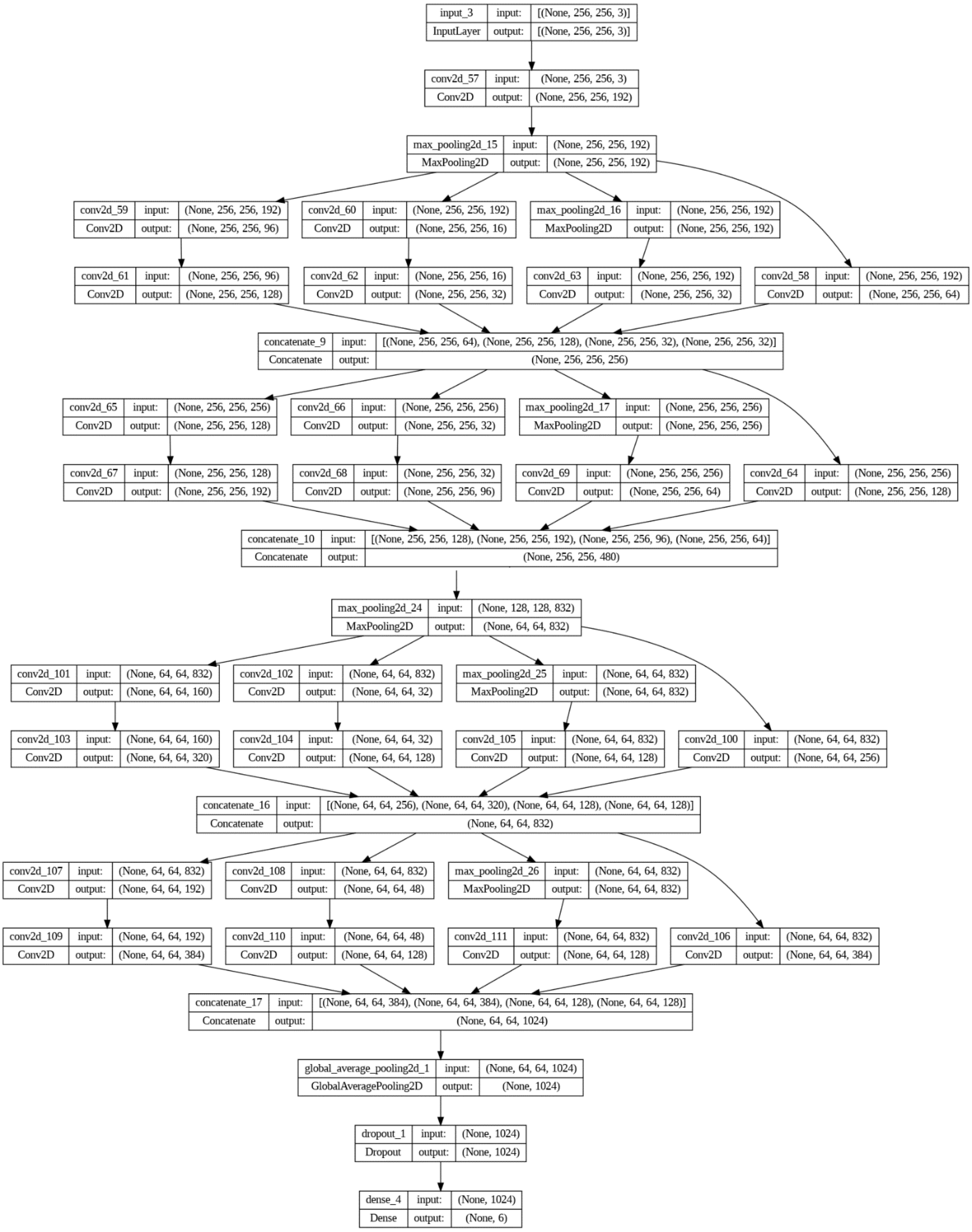


Figure 11 Proposed FF-TL-CNN model

The depth concatenation layer plays a crucial role in this process, as it takes inputs sharing the same height and width and concatenates them along the channel dimension. This approach, where computation modules are stacked upon each other, naturally leads to variations in output correlation statistics. Additionally, as we progress to higher layers, we anticipate a reduction in spatial concentration, suggesting an increasing ratio of 3×3 and 5×5 convolutions. It's noteworthy that the number of outputs grows from one stage to the next, contributing to the creation of an optimal sparse structure within the architecture (Ding et al., 2022). The proposed architecture introduces several distinctive features. These include an average pooling layer with a 5×5 filter size and a stride of 3, a 1×1 convolution that incorporates 128 filters for dimension reduction with rectified linear activation, a fully connected layer comprising 1024 units utilizing rectified linear activation, a dropout layer implementing a 70% ratio of dropped outputs, and a linear layer employing SoftMax loss for classification purposes. The proposed FF-TL-CNN classification algorithm is given below.

Algorithm 1: Proposed FF-TL-CNN Classifier

Input: X=Test Leaf Image.

Output: Disease Class (Y)

Step 1: Initialize the input train images (D).

Step 2: Generate ImageNet input weights from MobileNetV2.

Step 3: Extract features using variable kernel sizes (1x1, 3x3, 5,5).

Step 4: Concatenate the output of these filters using concatenation.

Step 5: Generate the sparse feature vector.

Step 6: Train the proposed model using these features.

Step 7: Adjust and tune the hyperparameters.

Step 8: Apply softmax function to the feature vector.

Step 9: Test image (X) is given as input and obtain disease class (Y) label as the output.

The FF-TL-CNN model proposed in this study employs a layered architecture, commencing with the initial layer serving as the input. Subsequent layers consist of a sequence of convolutional, max-pooling, dense, concatenation, average pooling, flatten, and fully connected layers, stacked in a cascading fashion. To enhance classification performance, the model adopts the Adam optimizer, well-suited for handling extensive datasets and parameter sets. The hyperparameters in this model are intuitively interpretable, simplifying the fine-tuning process. For the computation of loss during both training and validation, binary cross entropy (BCE) is utilized. BCE assesses the disparities between each estimated probability and the actual class output, considering the presence of six distinct classes. It assigns scores based on the extent of deviation from the predicted values. In Figure 5, we illustrate the FF-TL-CNN model tailored for the Plant Village dataset, which initializes with six distinct classes. The model undergoes a series of processing stages, encompassing convolutional layers and feature fusion, ultimately culminating in a final output with six classes and predictions for each class. Throughout the training and validation phases, the model operates with a batch size of 512 and a learning rate of 0.01, spanning a total of 30 epochs. The choice of 30 epochs is deliberate, representing the point at which the FF-TL-CNN model converges, stabilizing the loss and accuracy metrics to yield optimal results.

4 Results and discussion

In this research, the FF-TL-CNN approach is exclusively centered on leveraging multiple kernel-based feature fusion, TL and deeper layers to train the classifier using output features. While this strategy involves the training of sparse features, it still entails a considerable number of features. For instance, the feature set's dimension resulting from the concatenation layer comprises 28,874 features. To establish this training approach, we maintained an 80:20 split between the training and testing data. In addition, we ensured consistency in our parameter settings when comparing our results with prior research, employing both cross-validation and fixed partitioning methodologies. The implementation of the proposed FF-TL-CNN model was carried out in Python and evaluated using the Google Colab platform. It's essential to note that a lower learning rate contributes to the efficiency of FF-TL-CNN training, as an excessively high learning rate can lead to training stagnation with unsatisfactory results. Figure 12 depicts the graphical representation of the training and validation accuracy for the proposed FF-TL-CNN classifier. Figure 13 showcases the graphical representation of the training and validation loss for the proposed classifier.

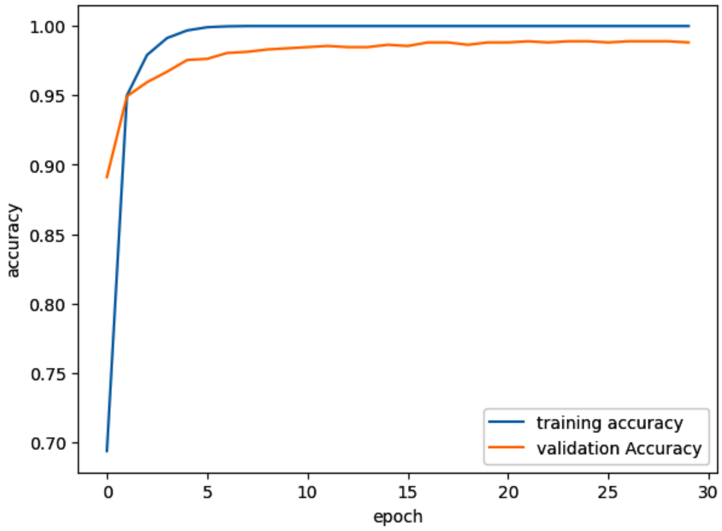


Figure 12 Train and validation accuracy of proposed FF-TL-CNN classifier

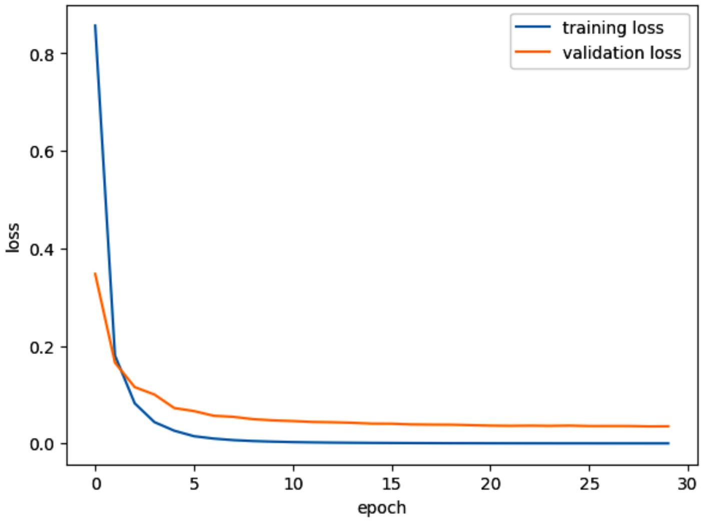


Figure 13 Train and validation loss of proposed FF-TL-CNN classifier

To assess the effectiveness of FF-TL-CNN model, we have calculated four crucial metrics: accuracy, recall, precision, and F1-score. These metrics are derived from the concepts of False Positive (P_f), False Negative (N_f), True Negative (N_t), and True Positive (P_t). The mathematical formulations for these performance measures are as follows:

$$Precision = \frac{P_t}{P_f + P_t} \tag{6}$$

$$Recall = \frac{P_t}{N_f + P_t} \tag{7}$$

$$Accuracy = \frac{N_t + P_t}{N_f + N_t + P_f + P_t} \tag{8}$$

$$F1-Score = \frac{2 \times P_t}{N_f + P_f + 2P_t} \tag{9}$$

Starting from the 5th epoch and beyond, the performance metrics consistently maintain high values, demonstrating the effectiveness of the TL and FF techniques in addressing the leaf disease categorization challenge. Particularly noteworthy is the remarkably low loss achieved by FF-TL-CNN model, used for classification, which stands at 0.13. Furthermore, the mean accuracy of this model reaches an impressive 98.85%. Additionally, the mean values for precision, recall, and F1-score are equally promising, measuring at 98.63%, 98.41%, and 99.32%, respectively. For a graphic depiction of the multiclass classification model's performance based on the proposed FF-TL-CNN approach, kindly refer to Figure 14. This stability in performance underscores the robustness of the model in addressing the leaf disease categorization task.

The efficiency of the proposed FF-TL-CNN model is evident in its accurate identification of different leaf disease classes. The classification report highlights the model's strong capability in distinguishing between these specific leaf disease classes. Moreover, when applied to the Plant Village dataset, the FF-TL-CNN model consistently demonstrates its effectiveness in classifying disease classes. To gain insight into the performance of individual classes, please refer to Figure 15.

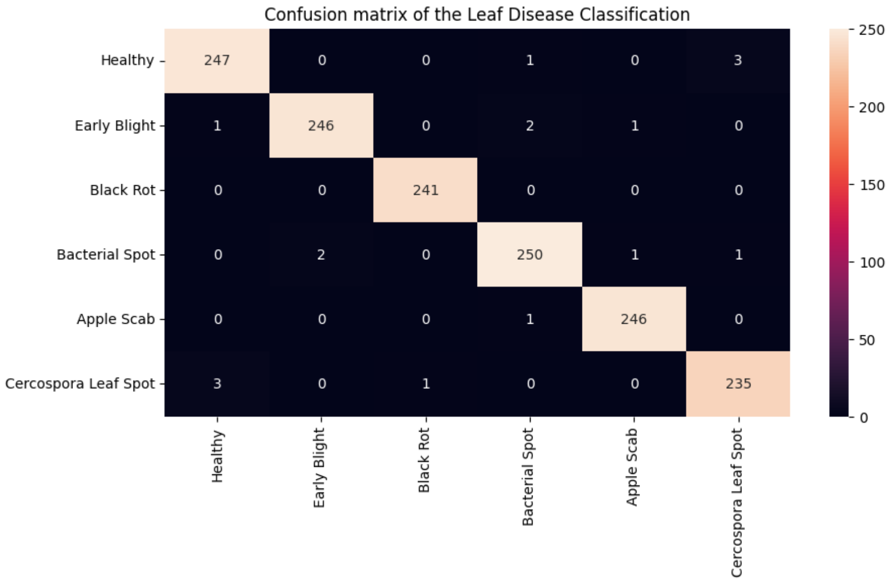


Figure 14 Confusion matrix of proposed FF-TL-CNN classifier

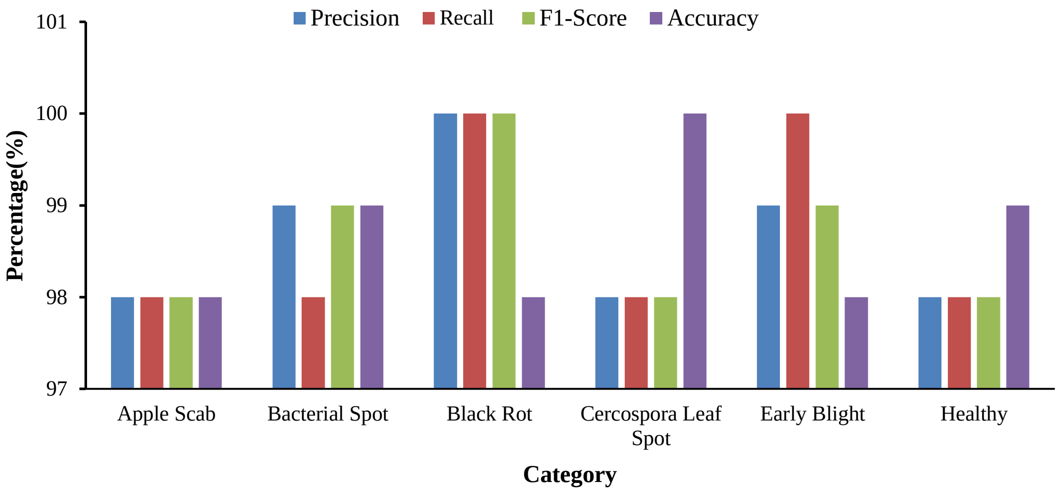


Figure 15 Category wise classification performance

To evaluate the effectiveness of the developed FF-TL-CNN approach, it is imperative to conduct a thorough assessment of its classification performance. In this regard, we carried out a comprehensive evaluation of the classification performance across various models using the Plant Village datasets. The findings of this evaluation are summarized in Table 1, which provides a comparative analysis of the efficiency of existing models based on the selected performance metrics.

Table 1 Comparison of leaf disease classification models

Model	Precision (%)	Recall (%)	F1-score (%)	Accuracy (%)
AlexNet	92.92	92.21	92.78	93.03
GoogleNet	90.73	91.09	89.56	90.31
ResNet 50	91.08	90.06	91.84	91.72
VGG16	91.27	91.42	90.37	91.33
Inception v3	90.92	89.06	89.74	90.81
CNN	91.95	90.40	91.47	90.94
FF-TL-CNN (Proposed)	98.63	98.41	99.32	98.85

When assessing classification accuracy, the FF-TL-CNN model stands out with an impressive score of 98.85%. Among the pre-trained models, we observe commendable accuracy rates with VGG16 (91.33%), ResNet50 (91.72%), and AlexNet (93.03%). It's noteworthy that the accuracy of the proposed FF-TL-CNN model surpasses that of AlexNet by a substantial margin of 5.25%. While shifting our focus to precision, the FF-TL-CNN model outperforms all other classifiers by offering a remarkable precision rate of 98.63%. In comparison, AlexNet achieves a precision of 92.92%, CNN attains 91.95%, and VGG16 scores 91.27%. The precision of FF-TL-CNN exceeds that of AlexNet by a significant 7.71%. These findings underscore the exceptional performance of the FF-TL-CNN model in achieving high accuracy and precision, outperforming its counterparts.

The recall value for the FF-TL-CNN model is particularly noteworthy, standing at an impressive 98.41%, marking it as the highest among all the models under consideration. Specifically, VGG16 exhibits a recall rate of 91.42%, and ResNet50 also records 91.42%, while AlexNet achieves a recall of 92.78%. In comparison, the FF-TL-CNN model's recall rate significantly outperforms that of AlexNet by a notable 6.2%. When we turn our attention to the F1-score, the FF-TL-CNN model emerges as the leader in this regard. While AlexNet demonstrates a respectable F1-score of 93.03%, the FF-TL-CNN model impressively attains an F1-score of 99.32%, representing a substantial 6.29% difference between these two approaches. This underscores the significance of certain key parameters and underscores the role of TL in mitigating overfitting and enhancing classification accuracy. It's essential to highlight that both the proposed model and AlexNet exhibit competence in detecting samples across substantial datasets. For a comprehensive comparative view of FF-TL-CNN's performance against state-of-the-art leaf disease classifiers, please refer to Figure 16. These findings highlight the exceptional recall and F1-score achieved by the FF-TL-CNN model, emphasizing the positive impact of its parameters and the contribution of EL in achieving superior performance.

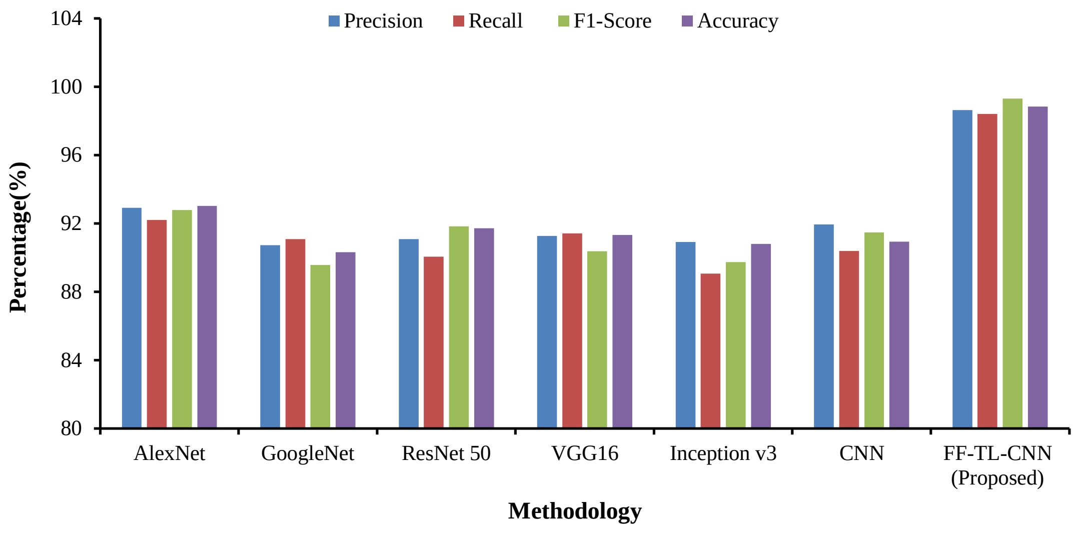


Figure 16 Comparison of classification performance

The effectiveness of FF-TL-CNN-based models lies in their utilization of extensive labeled datasets, allowing them to address highly complex problems with a high degree of automation. When FF-TL-CNN systems are combined with large datasets for classification, the entire classification process becomes automated. This eliminates the need for labor-intensive tasks like feature extraction, noise filtering, region of interest (ROI) delineation, or feature selection. As a result, predictions generated by FF-TL-CNN models are highly reproducible and free from bias, marking a significant advancement from earlier CNN approaches. Moreover, these models consistently achieve a remarkable level of accuracy, distinguishing them from previous CNN methodologies. The utilization of GPU resources within the Google Colab framework substantially reduces computation time. For instance, training the FF-TL-CNN on the Plant Village dataset required only 3 minutes and 54 seconds. Importantly, the performance metrics of the proposed multiclass classifier surpass those of existing models. For a visual representation of sample predictions alongside the corresponding ground truth, please refer to Figure 17. This emphasizes the efficiency and accuracy achieved by FF-TL-CNN-based models, paving the way for advancements in automated classification tasks.

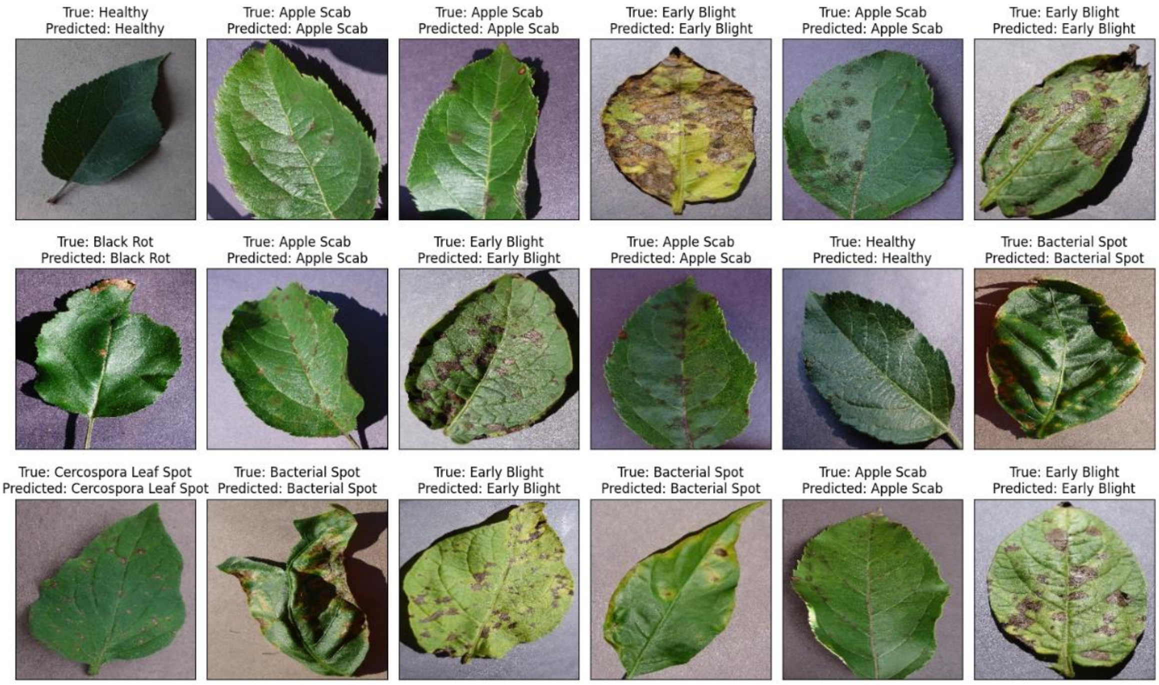


Figure 17 Ground truth and prediction results

5 Conclusion

This study delved into the exploration of several pre-trained CNN methods in conjunction with FF-TL-CNN for the purpose of classifying leaf diseases based on image data. The approach of combining CNN structures with multiple kernels, fusion and TL demonstrates remarkable efficiency in achieving peak classification performance. TL using ImageNet data and MobileNetV2 improved the learning capability of the proposed model. FF-TL-CNN, in particular, stands out by surpassing other classifiers with its outstanding accuracy of 98.85%, precision of 98.63%, recall of 98.41%, and an impressive F1-score of 99.32% when applied to the Plant Village dataset. The exceptional prediction accuracy of FF-TL-CNN is evidence to the effectiveness of optimization techniques employed in this work. One noteworthy aspect of FF-TL-CNN is its capacity to reduce the necessity for pre-processing stages, outperforming established techniques in this regard. Additionally, when compared to the pre-trained AlexNet classifier, FF-TL-CNN clearly demonstrates superior performance metrics. Looking ahead, future research endeavors will focus on the integration of these models into mobile platforms to enhance accessibility, the reduction of computing complexity, and further exploration of advanced methods for fine-tuning the models. This work not only highlights the achievements of FF-TL-CNN but also paves the way for continued progressions in image-based leaf disease classification.

References

Al Bashish, D., M. Braik, and S. Bani-Ahmad. 2011. Detection and classification of leaf diseases using K-means-based segmentation and neural networks-based classification. *Information Technology Journal*, 10(2): 267-275.

Aparajita, R. Sharma, A. Singh, M. K. Dutta, K. Ríha, and P. Kriz. 2017. Image processing based automated identification of late blight disease from leaf images of potato crops. In *2017 40th International Conference on Telecommunications and Signal Processing (TSP)*, 758-762. Barcelona, Spain, 5-7 July.

Barbedo, J. G. A. 2018. Factors influencing the use of deep learning for plant disease recognition. *Biosystems Engineering*, 172: 84-91.

Barbedo, J. G. A., L. V. Koenigkan, and T. T. Santos. 2016. Identifying multiple plant diseases using digital image processing. *Biosystems Engineering*, 147: 104-116.

Chtioui, Y., S. Panigrahi, and L. Francl. 1999. A generalized regression neural network and its application for leaf wetness prediction to forecast plant disease. *Chemometrics and Intelligent Laboratory Systems*, 48(1): 47-58.

Coulibaly, S., B. Kamsu-Foguem, D. Kamissoko, and D. Traore. 2019. Deep neural networks with transfer learning in millet crop images. *Computers in Industry*, 108: 115-120.

Dhingra, G., V. Kumar, and H. D. Joshi. 2019. A novel computer vision based neutrosophic approach for leaf disease identification and classification. *Measurement*, 135: 782-794.

Ding, Y., Z. Zhang, X. Zhao, D. Hong, W. Cai, C. Yu, N. Yang, and W. Cai. 2022. Multi-feature fusion: Graph neural network and CNN combining for hyperspectral image classification. *Neurocomputing*, 501: 246-257.

Gandhi, R., S. Nimbalkar, N. Yelamanchili, and S. Ponkshe. 2018. Plant disease detection using CNNs and GANs as an augmentative approach. In *2018 IEEE International Conference on Innovative Research and Development (ICIRD)*, 1-5. Bangkok, Thailand, 11-12 May.

Gavhale, K. R., and U. Gawande. 2014. An overview of the research on plant leaves disease detection using image processing techniques. *IOSR Journal of Computer Engineering*, 16(1): 10-16.

Hassan, S. M., A. K. Maji, M. Jasiński, Z. Leonowicz, and E. Jasińska. 2021. Identification of plant-leaf diseases using CNN and transfer-learning approach. *Electronics*, 10(12): 1388.

Hassanien, A. E., T. Gaber, U. Mokhtar, and H. Hefny. 2017. An improved moth flame optimization algorithm based on rough sets for tomato diseases detection. *Computers and Electronics in Agriculture*, 136: 86-96.

Kalaivani, S., and S. P. Shantharajah. 2017. Survey about segmentation feature extraction and classification of disease affected leaf using digital image. *Journal of Advanced Research in Dynamic Control System*, 14: 1694-1701.

Karthik, R., M. Hariharan, S. Anand, P. Mathikshara, A. Johnson, and R. Menaka. 2020. Attention embedded residual CNN for disease detection in tomato leaves. *Applied Soft Computing*, 86: 105933.

Li, J., P. Wang, and C. Geng. 2017. The disease assessment of cucumber downy mildew Based on image processing. In *2017 International Conference on Computer Network, Electronic and Automation (ICCNEA)*, 480-485. Xi'an, China, 23-25 September.

Li, H., M. Ding, R. Zhang, and C. Xiu. 2022. Motor imagery EEG classification algorithm based on CNN-LSTM feature fusion network. *Biomedical Signal Processing and Control*, 72(Part A): 103342.

Li, Z., B. Niu, F. Peng, G. Li, Z. Yang, and J. Wu. 2018. Classification of peanut images based on multi-features and SVM. *IFAC-PapersOnLine*, 51(17): 726-731.

Liu, J., and X. Wang. 2020. Early recognition of tomato gray leaf spot disease based on MobileNetv2-YOLOv3 model. *Plant Methods*, 16: 83.

Mahesh, T. R., R. Sivakami, I. Manimozhi, N. Krishnamoorthy, and B. Swapna. 2023. Early predictive model for detection of plant leaf diseases using MobileNetV2 architecture. *International Journal of Intelligent Systems and Applications in Engineering*, 11(2): 46-54.

Majji, V. A., and G. Kumaravelan. 2021. Detection and classification of plant leaf disease using convolutional neural network on plant village dataset. *International Journal of Innovative Research in Applied Sciences and Engineering (IJIRASE)*, 4(11): 931-935.

Majumdar, D., A. Ghosh, D. K. Kole, A. Chakraborty, and D. D. Majumder. 2014. Application of fuzzy C-Means clustering method to classify wheat leaf images based on the presence of rust disease. In *Proceedings of the 3rd International Conference on Frontiers of Intelligent Computing: Theory and Applications (FICTA) 2014*, 277-284. Bhubaneswar, Odisha, India, 14-15 November.

Patil, J. K., and R. Kumar. 2011. Advances in image processing for detection of plant diseases. *Journal of Advanced Bioinformatics Applications and Research*, 2(2): 135-141.

Patil, S. B., and S. K. Bodhe. 2011. Leaf disease severity measurement using image processing. *International Journal of Engineering and Technology*, 3(5): 297-301.

Paul, P. A., and G. P. Munkvold. 2005. Regression and artificial neural network modeling for the prediction of gray leaf spot of maize. *Phytopathology*, 95(4): 388-396.

Phadikar, S., J. Sil, and A. K. Das. 2012. Classification of rice leaf diseases based on morphological changes. *International Journal of Information and Electronics Engineering*, 2(3): 460-463.

Pydipati, R., T. F. Burks, and W. S. Lee. 2006. Identification of citrus disease using color texture features and discriminant analysis. *Computers and Electronics in Agriculture*, 52(1-2): 49-59.

Sethy, P. K., N. K. Barpanda, A. K. Rath, and S. K. Behera. 2020. Deep feature based rice leaf disease identification using support vector machine. *Computers and Electronics in Agriculture*, 175: 105527.

Sharif, M., M. A. Khan, Z. Iqbal, M. F. Azam, M. I. U. Lali, and M. Y. Javed. 2018. Detection and classification of citrus diseases in agriculture based on optimized weighted segmentation and feature selection. *Computers and Electronics in Agriculture*, 150: 220-234.

Singh, K., S. Kumar, and P. Kaur. 2019. Support vector machine classifier based detection of fungal rust disease in Pea Plant (*Pisam sativam*). *International Journal of Information Technology*, 11(3): 485-492.

Sullca, C., C. Molina, C. Rodriguez, and T. Fernandez. 2019. Diseases detection in blueberry leaves using computer vision and machine learning techniques. *International Journal of Machine Learning and Computing*, 9(5): 656-661.

Tiwari, D., M. Ashish, N. Gangwar, A. Sharma, S. Patel, and S. Bhardwaj. 2020. Potato leaf diseases detection using deep learning. In *2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS)*, 461-466. Madurai, India, 13-15 May.

Voulodimos, A., N. Doulamis, G. Bebis, and T. Stathaki. 2018. Recent developments in deep learning for engineering applications. *Computational Intelligence and Neuroscience*, 2018: 8141259.

Yang, C., G. N. Odvody, C. J. Fernandez, J. A. Landivar, R. R. Minzenmayer, and R. L. Nichols. 2015. Evaluating unsupervised and supervised image classification methods for mapping cotton root rot. *Precision Agriculture*, 16(2): 201-215.

Zhang, S., Z. You, and X. Wu. 2019. Plant disease leaf image segmentation based on superpixel clustering and EM algorithm. *Neural Computing and Applications*, 31(Suppl 2): 1225-1232.

Zhong, Y., and M. Zhao. 2020. Research on deep learning in apple leaf disease recognition. *Computers and Electronics in Agriculture*, 168: 105146.