International Journal of Applied Mathematics
Volume 38 No. 12s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

MATHEMATICAL MODEL OF HYBRID DEEP LEARNING FOR MULTI-CLASSIFICATION OF PLANT DISEASES

Prabhat Kumar Srivastava[1], Dipak Kumar Ghosh[1], Amit Arora[1]

[1]Department of Electrical, Electronics and Communication Engineering, Galgotias University, Greater Noida, India.

Email: srivastavapk1@gmail.com, dipakkumar05.ghosh@gmail.com, amit06arora@gmail.com

Corresponding Author: Amit Arora, Email:amit06arora@gmail.com

Abstract

Plant diseases significantly impact on global agricultural productivity and crop yield which is a threat to food security. It is imperative to have effective disease management through timely and accurate identification of these diseases. It reduces crop losses and enabling farmers to take appropriate action. Existing AI models struggle with challenges such as poor generalization to complex leaf patterns, difficulty in distinguishing between visually similar diseases, and limited scalability across multiple crop species. These issues have been addressed and presented in this work where a hybrid model that integrates Mobile Network Version-2 (MNV2) and machine learning such as Support Vector Machine (SVM) learning . The MNV-2 is known for its ability to understand complicated patterns and identify images reliably. The hybrid model leverages the spatial feature extraction capabilities of MNV2 with SVM allowing for more effective learning of both low- and high-level discriminative features. The proposed approach has been evaluated on a multi-class dataset comprising 38 categories, which includes 30 diseased and 8 healthy class of various plant disease from the Plant Village dataset. Simulation results have shown that the hybrid model achieved an accuracy (Ac) of 98.95%, a Precision (Pr) of 99.0%, Recall (Re) of 98.9% and a F-1 Score of 99% representing improvement over standard transfer learning techniques. These results accentuate the model's possible for real-world solicitation, including amalgamation into mobile-based platforms to support agriculturalists with on-site analysis of the infection and strategies for timely involvement and accomplishment.

Keywords: Hybrid CNN Model; Support Vector Machine Learning, Depthwise Seperable Convolution, Bottelneck Design of MNV2 Architecture, Machine Learning

1. INTRODUCTION

Agriculture has long been a key support of the Indian budget, having a huge portion of its populace reliant on it for earning their livelihood. Farmers benefit from many options in choosing the suitable crops and determining the suitable pesticides for their plants. Disease in the plant has been a very crucial contributor to the loss of agricultural yield, significantly affecting both the quality and quantity of the production of crops [1]. These losses can bring

Received: August 08, 2025

International Journal of Applied Mathematics
Volume 38 No. 12s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

about food shortages, economic instability, and increased unemployment in agrarian economies, especially in developing countries like India, where agriculture is the primary source of livelihood. Farmers are often constrained by a lack of knowledge and access to disease detection facilities, leading to delayed results and/or incorrect interpretation [2].

Crops are generally susceptible to a wide variety of diseases that can be caused by viruses, fungi, bacteria, molds, and many other adverse weather conditions involving excessive humidity, rainfall, and fluctuating temperatures that might favor the spread of the pathogen. All these factors threaten not only agricultural productivity but also food security at the national level and the economic welfare of farming communities.

Conventional disease detection methods required a tremendous amount of work and also required excessive processing time. It usually relies on expert visual inspection, which, being labour-intensive and error-prone, is not feasible in most remote or resource-limited regions. Image processing techniques have produced promising advancements in automatic disease detection; however, they are highly vulnerable to changes in image quality due to variations in leaf shape, color, texture, and background noise. Even though these techniques offer some better utility but still they lack in scalability and robustness especially in dynamically changing agricultural environments. Above limitations are addressed through artificial intelligence (AI), in particular the deep learning, which has now evolved as a promising solution for accurate and efficient disease diagnosis using plant leaf images.

The Machine Learning Algorithms (traditional) such as Artificial Neural Networks (ANN), Extreme Machine Learning, Random Forest (RF), Support Vector Machines (SVM) and k-Nearest Neighbors (k-NN), have ber5en widely applied in earlier works. SVM has been used to identify diseases in few crops such as Potatoes, Grapes and Palm leaves [3] [4] [5]. These methods rely heavily on handcrafted features, which has a limitation in complex and diverse datasets [6]. It has been observed that ANN and SVM were combined to extract color and texture features which has yielded results with 92.17% accuracy [7]. Alternatively, the k-means clustering method applied in the Lab color space attained 94% accuracy in classifying rice blast [8]. Furthermore, on applying the vision-based approach for detecting chlorosis in black gram leaves using SVM, it has yielded an accuracy of 94.69% [9],[10],[11].

The Deep learning approach have further shown superior performance because of its ability to learn hierarchical features directly from data without manual intervention. A CNN-based model for classifying images of Vigna mungo leaf (black gram leaves) into healthy, mild, and severe categories which has yielded 97.403% accuracy [12]. EfficientNet, embedded with transfer learning, was also explored for disease classification using laboratory-acquired images [13]. Other techniques, such as hyperspectral imaging, have been utilized to identify plant species in complex field conditions [14], while a deep learning approach for bacteriosis detection in peach crops attained an accuracy of 97.75% [15]. The PD2 SE- Net, leveraging ResNet-50, was used

International Journal of Applied Mathematics
Volume 38 No. 12s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

for multi-crop disease severity estimation [16], [17]. A transfer learning method for cassava leaf disease detection attained an accuracy of 93% on unseen data [18] [19], [20], [21]. Even though these models have achieved significantly better accuracy across multiple crop species and varying environmental conditions, still it is a challenge for future.

In order to address above limitations, the this study introduces a hybrid deep model that combine as MNV2 CNN Model and Machine learning (ML) architecture. The local spatial feature extraction strengths and the deep residual learning capabilities of MNV2 have been leveraged, enabling it to effectively learn both low-level and high-level discriminative features.

Leaf images from eighteen different crops across 38 categories, including both healthy and diseased classes have been selected and a most diverse and curated dataset has been developed for training. The model's generalization ability is enhanced through the images collected from laboratory setups and real-field conditions which exhibit significant inter-class and intra-class variations. The dataset was categorized into two classes – training and testing and these data base randomly divided into 80% and 20% respectively.

The proposed hybrid MNV2 with SVM model has achieved an accuracy (Ac) of 98.95%, a precision (Pr) of 99.0%, recall (Re) of 98.9% and a F1 Score of 99% representing improvement over standard transfer learning techniques. The results emphasize how well the model performs in correctly identifying diseases in plants under realistic imaging conditions. Further, this framework can be incorporated into mobile apps or embedded vision systems to enable real-time monitoring of plant health and provide scalable support for farmers and agricultural experts in disease management.

The rest of the paper has been organized and the structure is outlined. Section 2 elaborates on data collection and the architecture of the deep neural network. The outcome of experimental results has been discussed in Section 3, and finally, the proposed work has been concluded in Section 4.

2. Materials and Methods

To develop a proper framework for automatic plant disease detection and classification, several key steps were undertaken, from data collection to model training and multi-class classification of images of plant leaves. In the following study, leaf images from 38 different classes across various plant species were taken +from online standard datasets. These were taken under varying resolutions and conditions and then standardized to match the dimension of input required by the deep learning model. This finalized dataset was split into training, validation, and test sets to enable robust training and testing. A deep convolutional neural network was then trained over multiple epochs to learn the disease-specific features present in the images of leaves. A random (80-20) % validation was done to ensure thorough evaluation, and in addition to this, testing for generalization was performed on unseen data from all 38 classes.

Received: August 08, 2025

International Journal of Applied Mathematics
Volume 38 No. 12s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

2.1 Materials:

A complete dataset of leaf images was used for training and testing the disease classification model. This dataset consists of healthy and diseased conditions of several crops in 38 distinct classes. These were obtained through a rich variety of image types and environmental conditions from several publicly available datasets and field-collected samples for wide coverage. These images cater to a wide range of type of diseases as well as plant species that has been captured in diverse backgrounds including lab settings and natural field conditions. All images were resized and normalized to meet the input size and normalisation requirements of the deep learning model, resulting in uniform training and testing over the extended dataset.

This work is based on a rich dataset containing 38 classes of plant diseases and healthy conditions. These classes span various crops, with each category containing labeled images such as [0] Apple_ Apple Scab (200 out of 630); [1] Apple_ Black Rot (200 out of 621); [2] Apple_ Cedar Apple Rust (200 out of 275); [3] Apple_ Healthy (200 out of 1645); [4] Blueberry_Healthy (200 out of 1502); [5] Cherry_ Powdery Mildew (200 out of 1052); [6] Cherry_Healthy (200 out of 1773); [7] Corn_Cercospora Leaf Spot (Gray Leaf Spot) (200 out of 513); [8] Corn_Common Rust (200 out of 1192); [9] Corn_Northern Leaf Blight (200 out of 985); [10] Corn_Healthy (200 out of 1162); [11] Grape_ Black Rot (200 out of 1180); [12] Grape_ Esca (Black Measles) (200 out of 1383); [13] Grape_ Leaf Blight (200 out of 1076); [14] Grape_ Healthy (200 out of 1646); [15] Peach_ Bacterial Spot (200 out of 2296); [16] Peach_ Healthy (200 out of 2229); [17] Pepper Bell_Bacterial Spot (200 out of 1000); [18]Pepper Bell_Healthy (200 out of 1000); [19] Potato_Early Blight (200 out of 1000); [20]Potato_Late Blight (200 out of 1000); [21] Potato_Healthy (200 out of 1000); [22] Raspberry_Healthy (200 out of 371); [23] Soybean_ Healthy (200 out of 5090); [24] Squash_ owdery Mildew (200 out of 1835); [25] Strawberry_ Leaf Scorch (200 out of 1109); [26] Strawberry_ Healthy (200 out of 1500); [27] Tomato_Bacterial Spot (200 out of 2127); [28] Tomato_Early Blight (200 out of 1000); [29] Tomato_Late Blight (200 out of 1909); [30] Tomato_Leaf Mold (200 out of 952); [31] Tomato_Septoria Leaf Spot (200 out of 1771); [32] Tomato_pider Mites (Two-Spotted) (200 out of 1676); [34] Tomato_Target Spot (200 out of 1404); [35] Tomato_Tomato Mosaic Virus (200 out of 373); [36] Tomato_Tomato Yellow Leaf Curl Virus (200 out of 5357); [37] Tomato_Healthy (200 out of 1591);

2.2 Methods

The proposed model follows a structured approach in detecting disease cells in plants, as illustrated in Figure 1. Input: The model takes various types of images as input for analysis. Preprocessing: The input images undergo preprocessing to enhance quality and ensure the data is suitable for further processing. Feature Extraction: Features are extracted from the preprocessed images using the MNv2 CNN model, which captures important plant diseases characteristics. Machine Learning Classification: The extracted features are then fed into various machine learning models, including Extreme Learning Machine (ELM), SVM, K-

Received: August 08, 2025

International Journal of Applied Mathematics
Volume 38 No. 12s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

NN, and RF 25], [26], [27], [28]. These models are used to classify the plant disease based on the extracted features.

2.2.1 MobileNetV2 Architecture

MobileNet-v2 primarily consists of convolution, depthwise separable convolutions, inverted residuals, bottleneck design, and further linear bottlenecks associated with squeeze and excitation. This helps lower parameters and lessens the total computations while trying to retain most of its capabilities of complex feature capturing. The structure of MNv2 architecture is as shown in figure 2 and follow as

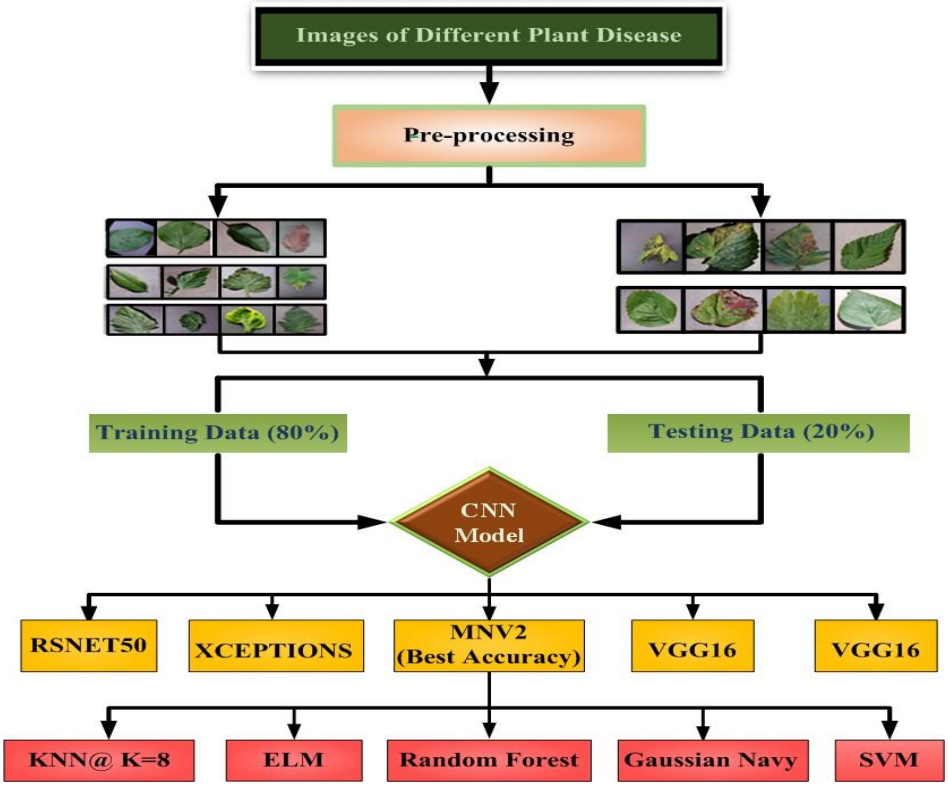

Fig.1 Frame work of proposed research work

Depthwise Separable Convolution

Depthwise separable convolution (DSC), employed in MobileNetV2, is a factorization method intended to lower the computational cost of average convolutions. It breaks down a traditional convolution into two step such as depthwise convolution and pointwise convolution, thereby reducing the overall number of computations and cultivating model efficiency.

Inverted Residuals

The most important parts of Mobileetv2 architecture are the Inverted residuals, which helped to improve the accuracy of the model [12]. They proposed a bottleneck construction

International Journal of Applied Mathematics
Volume 38 No. 12s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

that first enlarges the channel dimensions previously applying depthwise separable convolutions, permitting the model to capture richer features and advance its representation capability.

Bottleneck Design

It additional minimizes computational cost by put on 1×1 convolutions to decrease the channel sizes before depthwise separable convolutions. The bottleneck strategy in MobileNetV2 confirms an effective trade-off between classical dimensions and accuracy.

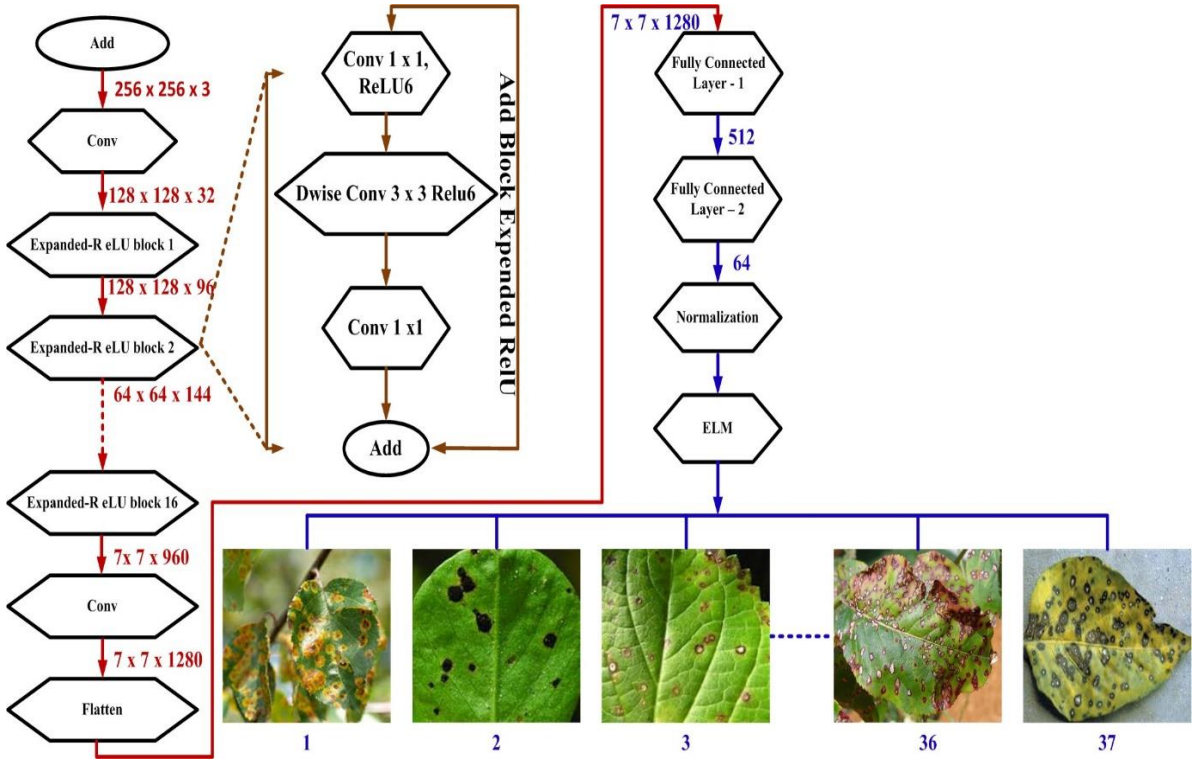

Figure 2: Structure and features distribution in MNv2 CNN Transfer Learning Model

Linear Bottlenecks

To alleviate the loss of information in the process of bottlenecks, MobileNet-v2 introduced linear bottlenecks. The use of linear instead of nonlinear activations results in much more information being preserved and the ability to capture more fine-grained details by the model.

Squeeze-and-Excitation (SE) Blocks

Added Squeeze-and-Excitation blocks in the network structure have further improved the feature representation capability of MobileNet-v2. This block performs adaptive recalibration of channel-wise feature responses; hence, allowing the model to focus on more informative features by suppressing less relevant ones.

Data Preparation

Received: August 08, 2025

International Journal of Applied Mathematics
Volume 38 No. 12s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

Before training MNV2, data preprocessing is vital. This includes making the images, dividing the dataset into training and testing sets, and applying data increase to advance the model's generalization ability.

Transfer Learning

A very common approach while dealing with MNV2 is transfer learning, where the concept itself allows utilizing pre-trained models on very large-scale datasets. The training can be considerably sped up by initializing the model with pre-trained weights, while the model can gain from knowledge learned from the source dataset.

Fine-tuning

Fine-tuning of MobileNetV2 is the process where the model gets trained on a target dataset, but for some layers, the pre-trained weights are kept fixed. In that way, it learns dataset-specific characteristics yet remembers the knowledge it had acquired from the source dataset.

Hyperparameter Tuning

Hyperparameter tuning becomes important in bringing out the optimal performance of MobileNetV2. This study has identified some of the important parameters, namely learning rate, batch size, and regularization techniques, for tuning to obtain optimal results. Apply methods such as grid or arbitrary search to select the best blend of hyperparameters.

2.2.2 Extreme Learning Machine

The ELM is a feedforward neural network with a single hidden layer, it has been introduced to address certain limitations of conventional neural network training methods. The major drawbacks are related to their convergence being too slow, mainly because of overfitting. In ELM, first, the input and hidden layer weights are arbitrarily allocated and fixed, while between hidden layer and output the weights are analytically determined. This may result in much faster training with better generalization in many applications.

According to [18], [29], [30] basic of ELM can be divided into 3 phases, for the training data attributes (X) and class output (T) sets represent as $\{X_K, T_K\}_{K=1,2\ldots,M}$ and $X_K = \{x_{K1}, x_{K2}, \ldots\ldots, x_{Kn}\}^t \in R^n$, Here M, K and t indicates number of data samples, quantity of input attributes to the n input data layer and transpose of the input features matrix. Denotation $T_K \in \{-1,1\}$ denotes output consistent to the sample X_K for binary classification. ELM consists of m There are a quantity of hidden layer nodes with activation function G (.) between the input and output layers. The structure of ELM is shown in **Figure 3**

The following steps are involved in the learning of ELM

1. First take the arbitrary value of input weights $(W) = \{W_{m1}, W_{m2}, \ldots\ldots, W_{mn}\}^t$ and biases $B = \{B_1, B_2, \ldots\ldots B_m\}$ of hidden nodes (N). This value remains constant during the training and validation of ELM.

Received: August 08, 2025

International Journal of Applied Mathematics
Volume 38 No. 12s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

2. Calculate the hidden layer (N) Output by utilizing the activation function $G(.)$ as $G(W\ X\ +B)$.

$$N = \begin{bmatrix} \emptyset(W_1X_1 + B_1,) & \cdots & \emptyset(W_mX_1 + B_m) \\ \vdots & \ldots & \vdots \\ \emptyset(W_1X_n + B_1,) & \cdots & \emptyset(W_mX_n + B_m) \end{bmatrix}_{n\times m} \quad (1)$$

3. Calculate the output weight between hidden layer and output $\beta = N^{\dagger}T$, where $N^{\dagger}$ denotes Moore-Penrose generalized inverse of N and $\beta = \{\beta_1, \beta_2, \ldots\ldots\ldots\beta_m\}^t$weight .

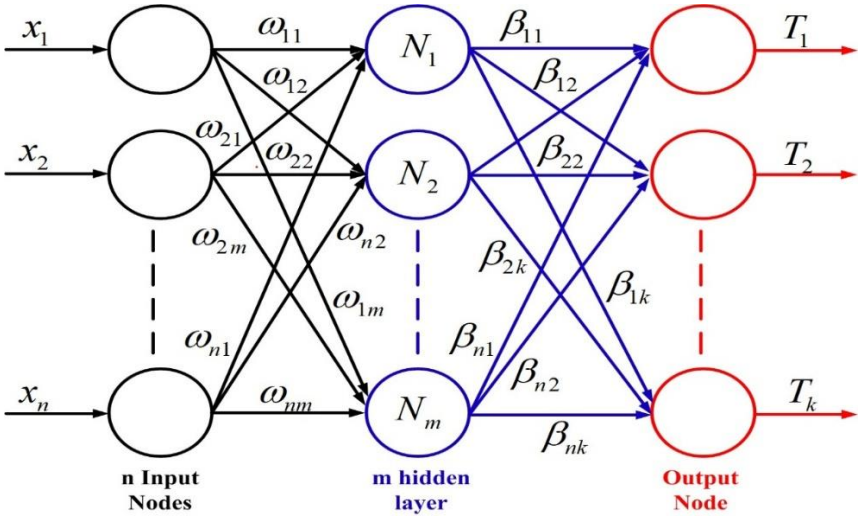

Figure 3. Extreme Learning Machine Construction for multi-classification.

2.2.3 K-NEAREST NEIGHBORS (KNN) ALGORITHM

K-Nearest Neighbors is, however, a very simple, non-parametric, supervised learning technique that can be used for either classification or regression. It predicts the output variable by looking for the most similarities to the k-nearest neighbors of a data point within the feature space.

Does not make any assumptions about the underlying data distribution. Learns directly from the training data without building an explicit model. Stores the entire training dataset and makes predictions at runtime. Decision-making depends on the proximity of data points to their neighbors. Classifies or predicts using a similarity metric (e.g., Euclidean distance). k is the number of nearest data points for making the prediction. Calculate the distance between the two point and all points in the training and testing dataset using a metric such as: Euclidean distance: $d(p,q) = \sqrt{\sum_{i=1}^{n}(p_i - q_i)^2}$ and Manhattan distance: $\sum_{i=1}^{n}|p_i - q_i|$, where p and q represents two points [31].

2.2.4 SUPPORT VECTOR MACHINE (SVM)

The SVM is one of the most extensively used supervised learning method. It is primarily working for classification in ML [32]. In classification, SVM search for to construct the best

Received: August 08, 2025

International Journal of Applied Mathematics
Volume 38 No. 12s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

decision boundary, or hyperplane, in an n-dimensional space to accurately divides classes and classify new data points.

In this way, a plane known as a best decision boundary is drawn through the SVM, selecting extreme points/vector that helps it construct the hyperplane. Those cases are named support vectors hence the algorithm name being a SVM.

A rational choice for the best hyperplane in SVM is the one that maximizes the dividation margin between the two classes. This supreme-margin hyperplane is defined such that the aloofness between the hyperplane and the closest data points from each class is maximized [33].

Consider the following diagram, where there are two different classes of categorization using a decision boundary or hyperplane. Classification by linear SVM can be performed by drawing a straight line between the two classes as shown Figure 4.

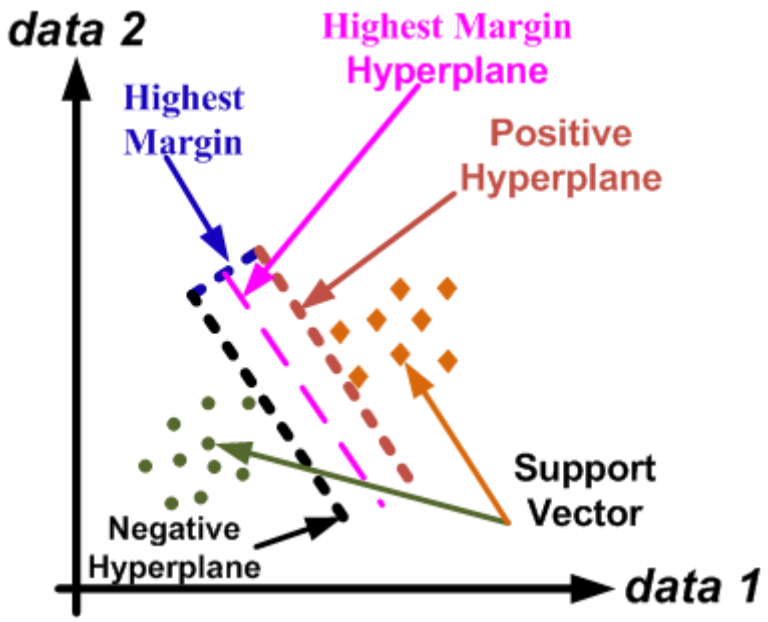

Figure. 4 Structure of SVM

The definitions of terms used in SVM are described as follows:

Hyperplane: In a feature space, the hyperplane is the divide the boundary that splits all data values of different classes. If the classification is linear, then this is a linear equation and can be written as wx+b=0. The data points nearby to the hyperplane, which straight influence its position, are known as Support Vectors [34].

Margin: In SVM, the margin denotes to the gap between the hyperplane and its nearby support vectors. The main objective of the SVM algorithm is always to make the most of the margin, because a larger width often means good classification.

Kernel: In SVM, a kernel is a mathematical function that converts the input data into a

International Journal of Applied Mathematics
Volume 38 No. 12s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

higher-dimensional feature space, sanctioning the model to discrete data points that are not linearly separate in the original domain. Commonly used kernels contain linear, polynomial, radial basis function (RBF), and sigmoid.

Hard Margin (HM): In SVM, the hard margin denotes to the extreme-margin hyperplane that has entirely separated the points of different classes labels without misclassifications.

Soft Margin: Whenever the data includes outliers or there is incomplete separability of the data, the soft margin concept in SVM is implemented. In the method, slack variable for every data point that undergoes certain misclassifications against a margin maximization process has to be considered balancing minimization violations [35].

In n-dimensional space, several lines or decision boundaries can segregate the classes, but we need to find the best decision boundary, known as a hyperplane, which will help in classifying data points. The aim of it is to construct a hyperplane that make the most of the margin, i.e., the space between the nearby data points of different classes.

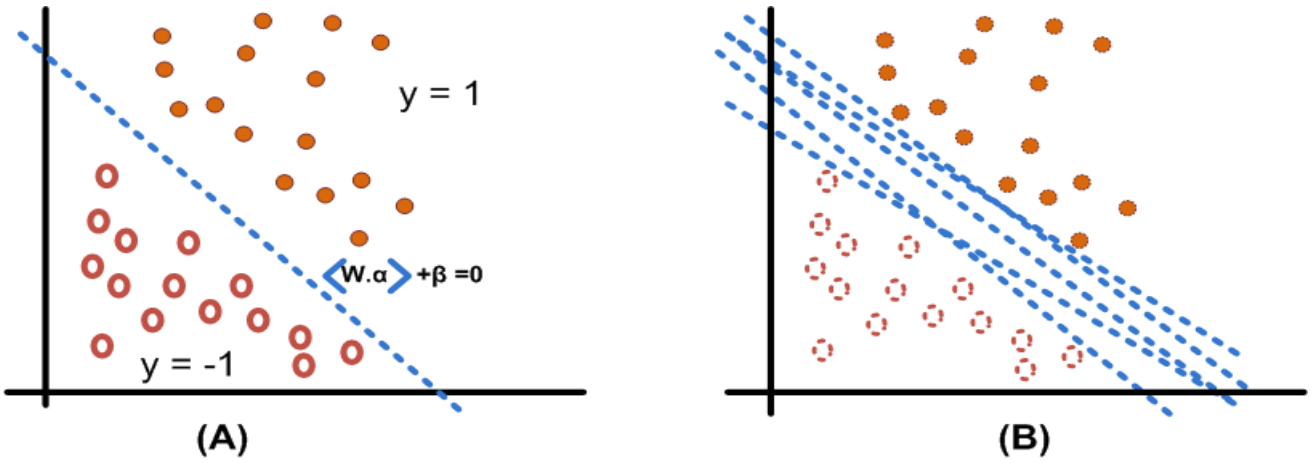

Fig. 5. (A) A linearly distinguishable data sets points and (B) possible decision separation of data points boundaries

As shown in Fig. 5(B), many lines can divide the positive and negative data sets points. The test is deciding which one to choice. SVM resolve this by selecting the hyperplane that maximizes the margin the parting between the two classes labels which will be properly defined later.

As shown in Fig. 5(A) and 5 (B), a hyperplane classifier applies only in cases when the positive and negative data are linearly separated. Nonlinearly separable data set, or those requiring nonlinear decision boundaries are dealt with kernel functions by SVM. If data are not linearly separable, then a new variable is created by SVM using a kernel.

Let us take a binary classification task where the two classes labels are denoted by +1 and -1. For a training dataset with feature vectors X and their associated class labels Y, the straight line hyperplane can then be expressed as:

$$w^T x + b = 0$$

Received: August 08, 2025

International Journal of Applied Mathematics
Volume 38 No. 12s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

The vector w serves as the normal to the hyperplane. The equation includes the dot product, while the parameter b represents the bias term, representing the shift of the hyperplane from the origin in the direction of W.

It can easily calculate the distance of a data point x_i from the decision boundary as:

$$d_i = \frac{w^T x_i + b}{\|\omega\|}$$

where ||w|| is the Euclidean norm (distance) of the weight vector w.

SVM as linear classifier:

$$\hat{y} = \begin{cases} one, \ w^T x + b \geq 0 \\ zero \ , \ w^T x + b < 0 \end{cases} \quad (2)$$

Optimization: For Hard margin (HM):

$$\underset{w,b}{minimize} \frac{1}{2} w^T w = \underset{w,b}{minimize} \frac{1}{2} \|w\|$$

subject to $y_i(w^T x_i + b) \geq 1 \ for \ i = 1, 2, 3, \ . \ . \ . \ , m$ (3)

Here, the symbol t_i signifies the target (output) variable or label of the i^{th} training instance [36]. And t_i=-1 for negative incidences (when y_i= ZERO) and t_i=1 positive examples (when y_i = ONE), respectively. Since we want the decision boundary satisfying constraint: $t_i(w^T x_i + b) \geq 1$

For Soft margin (SM) :

$$\underset{w,b}{minimize} \frac{1}{2} w^T w + C \sum_{i=1}^{m} \xi_i \quad (4)$$

subject to $y_i(w^T x_i + b) \geq 1 - \ \xi_i \ and \ \xi_i \ \geq \ 0 \ \ for \ i = 1, 2, 3, \ . \ . \ . \ , m$

where C - tradeoff parameter (Slack penalty)- chosen by cross-validation, ξ_i - "slack variables" (Allow "error" in classification)

Dual Problem: The solution to SVM can be determined when the dual problem of the optimization problem involved in finding the Lagrange multipliers corresponding to the support vectors is considered [37]. To achieve this, the following is the dual objective function maximized at the optimal Lagrange multipliers α(i).

$$\underset{\alpha}{maximize}: \frac{1}{2} \sum_{i \to m} \sum_{j \to m} \alpha_i \alpha_j t_i t_j \, K(x_i, x_j) - \sum_{i \to m} \alpha_i \quad (5)$$

$$\underset{\alpha}{maximize}: \frac{1}{2} \sum_{i \to m} . \sum_{j \to m} \alpha_i \alpha_j t_i t_j \, K(x_i, x_j) - \sum_{i \to m} \alpha_i \quad (6)$$

where,

α_i is the Lagrange multiplier associated with the i^{th} training sample;

Received: August 08, 2025

International Journal of Applied Mathematics
Volume 38 No. 12s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

$K(x_i, x_j)$ is the kernel function that calculates the similarity between two samples x_i and x_j. It enables SVM to handle nonlinear classification problems by implicitly mapping the samples into a higher-dimensional feature space;

$\sum\alpha_i$ = the summation of all the Lagrange multipliers.

After solving the dual design and obtaining the optimal Lagrange multipliers, the SVM decision edge can be uttered using these multipliers beside with the support vectors. The support vectors correspond to the training samples where αi>0, and the decision boundary is defined as

$$w = \sum_{i \to m} \alpha_i t_i K(x_i, x) + b \quad (7)$$

$$t_i(w^T x_i - b) = 1 \Leftrightarrow b = w^T x_i - t_i \quad (8)$$

2.2.5 RANDOM FOREST (RF)

Some of the most famous supervised machine learning algorithms include random forests, capable of solving regression and classification problems. The method depend on on ensemble learning, where numerous classifiers are combined to challenge complex problems and improve model performance. In RF classification, multiple decision trees are built using arbitrarily selected subdivisions of data and features. Each tree acts as a self-determining expert, donating its classification decision. The final estimate is determined by majority voting across all trees. Growing the number of trees generally advances accuracy while also tumbling the risk of overfitting.

Flow diagram explanation the employed of RF algorithm is shown in Figure 6.

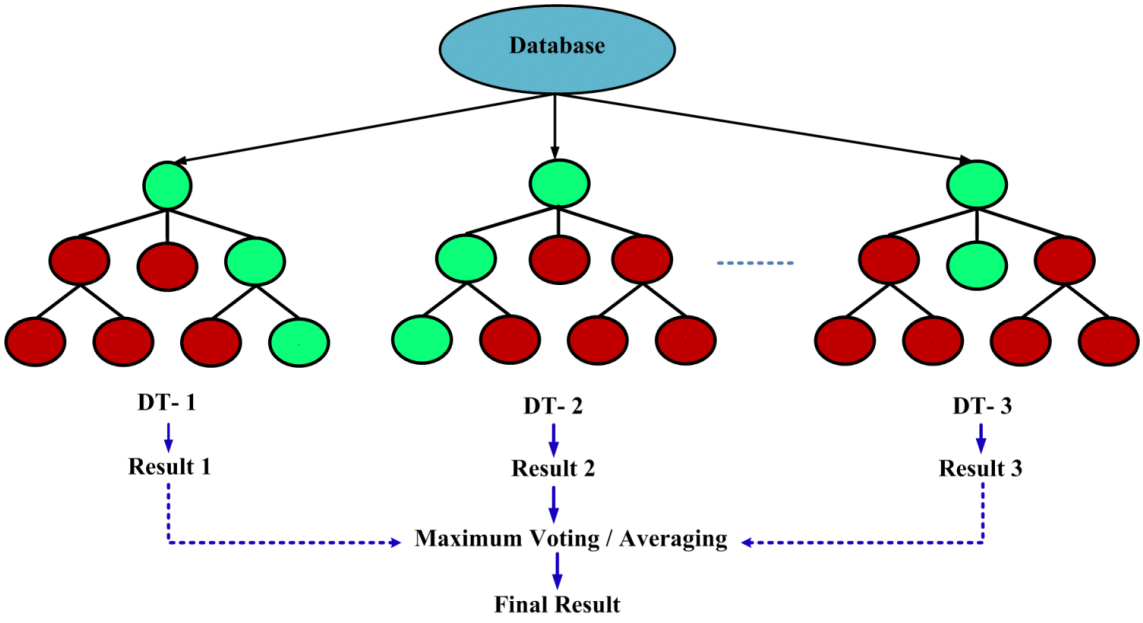

Fig. 6 Random Forest algorithm

Received: August 08, 2025

International Journal of Applied Mathematics
Volume 38 No. 12s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

In a Random Forest (RF), the general prediction is attained by combining the outputs of multiple decision trees some may classify properly while others may not, but together they yield a reliable result. For the classifier to perform efficiently, the dataset should cover expressive feature values that allow precise predictions rather than assumptions, and the distinct trees' estimates should preserve slight correlation with each other.

The RF algorithm operates in two key stages: first, constructing the forest by producing N decision trees, and second, producing guesses using the trees created in the initial stage. The process of the algorithm can be outlined as follows.

Multiple bootstrap datasets are produced from the original training data sets. A bootstrap model is formed through arbitrary sampling with spare, meaning some data sets points may appear numerous times while others may be left out. Each verdict tree in the RF is then trained on a separate bootstrap sample.

Unlike traditional decision trees that estimate all features, the RF algorithm arbitrarily selects a subset of features at each node. This extra layer of arbitrariness helps decrease correlation amongst trees and improves the simplification ability of the final model.

Such random feature selection is one of the major differences between random forests and bagging with decision trees. Each tree in the forest is grown to maximum without pruning, so as to enable each tree to overfit on its respective bootstrap sample. For classification problems, the outputs from all trees are combined by taking a majority vote. Whichever class gets the most votes becomes the final prediction.

A Random Forest is a classifier formed by an ensemble of tree-based models { h(x, Θk), k = 1,...} where the {Θk} represents an independent and identically distributed (iid) random vector, Each tree contributes a single vote for the most likely class of the input x, as illustrated in Figure 7.

Every tree is trained on a bootstrap sample along with a random variable Θ_k; since these random variables are iid, each produces a classifier h(x, Θ_k), where x is the input vector. After repeating this process kkk times, we obtain a collection of classifiers {h_1 (x), h_2(x), . . ., h_k(x)}, Together, they form an ensemble model where the final prediction is determined by majority voting, expressed through the decision function as:

$$\arg \max_{y} \sum_{i=1}^{k} I(h_i(x) = y) \quad (9)$$

Here, H(x) denotes the joint classification model, hi signifies a distinct decision tree, Y is the output variable, and I(·) denotes to the indicator function. Each tree donates a vote for a given input, and the final classification is resolute by selecting the majority result, as illustrated in Figure 7.

Received: August 08, 2025

International Journal of Applied Mathematics
Volume 38 No. 12s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

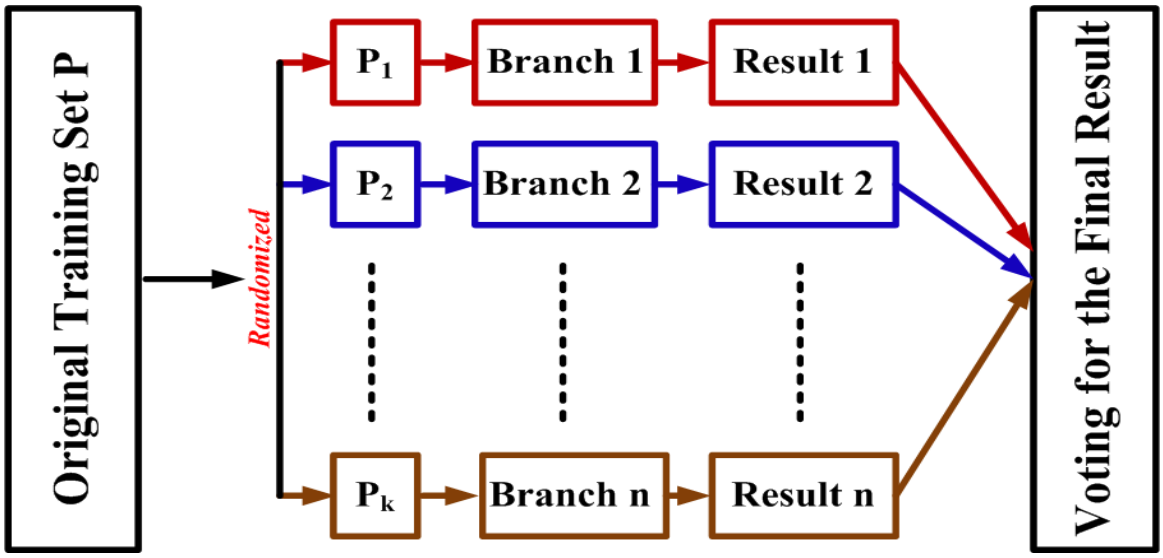

Fig. 7 RF Schematic with mathematical description

In Random Forests, the margin function evaluates how much the average number of votes for the correct class (X, Y) surpasses the votes for any incorrect class. The margin function can be defined as:

$$mg(X,Y) = av_k I(h_i(X) = Y) - max_{j \neq Y}\, av_k I(h_k(X) = j) \quad (10)$$

A larger margin value designates greater classification accurateness and higher confidence in the prediction. The simplification error of the classifier is defined as:

$$PE^* = P_{X,Y}(mg(X,Y) < 0) \quad (11)$$

when the number of decision tree is big enough, $h_k(x) = h(x, \Theta k)$ follows the Strong Law of Large Numbers. Leo Breiman demonstrated two key results: first, R F do not suffer from overfitting; and second, they converge to a limiting value of the generalization error. As the number of trees increases, for almost every sequence Θ_1, the value of PE approaches this limit.

$$P_{X,Y}(P_\theta(h_k(X,\theta) = Y) - max_{j \neq Y}\, P_\theta(h(X,\theta) = j) < 0 \quad (12)$$

Another is the upper bound of the generalization error is exist, and

$$PE^* \leq \bar{\rho}(1 - s^2)/s^2 \quad (13)$$

s is the strength of the set of classifiers $\{h_k(X,\theta)\}, \bar{\rho}$ is the mean value of the correlation. This equation reveals that the generalization error of RF depends on two aspects: one is the strength of every tree in the forest; the other is the dependence between trees. Obviously, the smaller this value is, the better the results of random forest.

2.2.6 PERFORMANCE METRICS

The pseudo-inverse method ensures minimal overfitting, leading to robust generalization.

Received: August 08, 2025

International Journal of Applied Mathematics
Volume 38 No. 12s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

Performance metrics include: Mean Squared Error (MSE) for ELM as

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \quad (14)$$

Accuracy for classification:

$$Accuracy\ (Ac) = \frac{TP+TN}{TP+TN+FP+FN} \quad (15)$$

$$\text{Precision (Pr)} = \text{TP} / (\text{TP} + \text{FP}) \quad (16)$$

$$\text{Recall (Re)} = \text{TP} / (\text{TP} + \text{FN})$$

$$\text{F1-score} = 2 \times (\text{Pr} \times \text{Re}) / (\text{Pr} + \text{Re}) \quad (17)$$

where, TP, TN, FP, FN are true positives, true negatives, false positives, and false negatives, respectively.

3 Result and Discussion

The Fig. 8 depicts the train accuracy observed by applying four different Convolutional Neural Network (CNN) models such as MobileNetV2, VGG16, DenseNet 201 and Xception for classifying different plant disease data set. Each bar represents average test accuracy, with exact percents labeled in the Fig. 9.

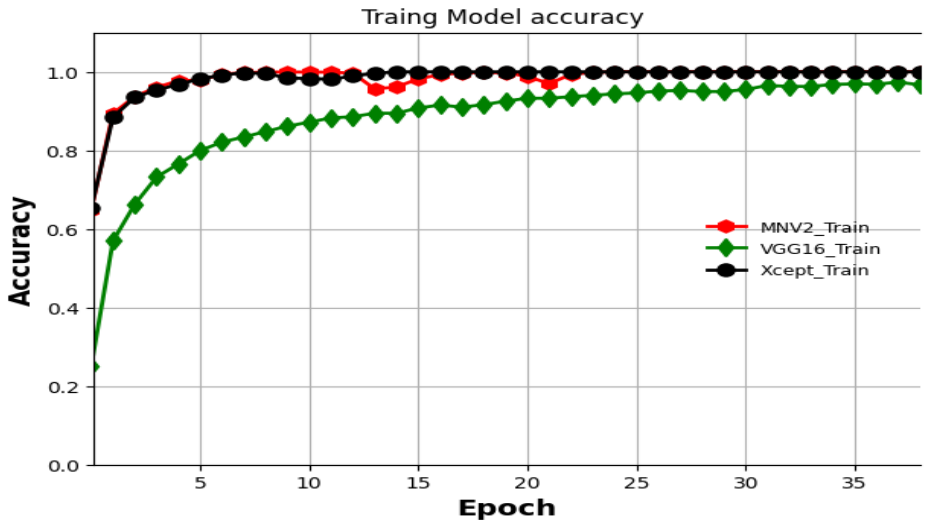

Fig. 8 CNN Models Training Accuracy

This is further elaborated by the training accuracy shown in Fig. 8, which describes the performance dynamics of the four models during the training phase. MobileNetV2, represented by the red color, maintains consistent superiority in rapid convergence, achieving near-perfect training as epochs progress, and aligns well with its high test accuracy of

International Journal of Applied Mathematics
Volume 38 No. 12s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

93.684%. The black curve represents Xception, and it steadily increases training accuracy. VGG16 is represented by a green curve and shows stable but lower training accuracy compared to MobileNetV2 and Xception. This behavior points out its struggles in adapting to specialized natures of various plant disease classification, and this reflects in its low-test accuracy of 89.144%. In general, the graph reinforces the earlier discussion of the clear dominance of MobileNetV2 and Xception both in training and test phases while comparing the limitations of VGG16 and DenseNet 201 in handling this classification task. Later in the next section, we computed the training loss of CNN models.

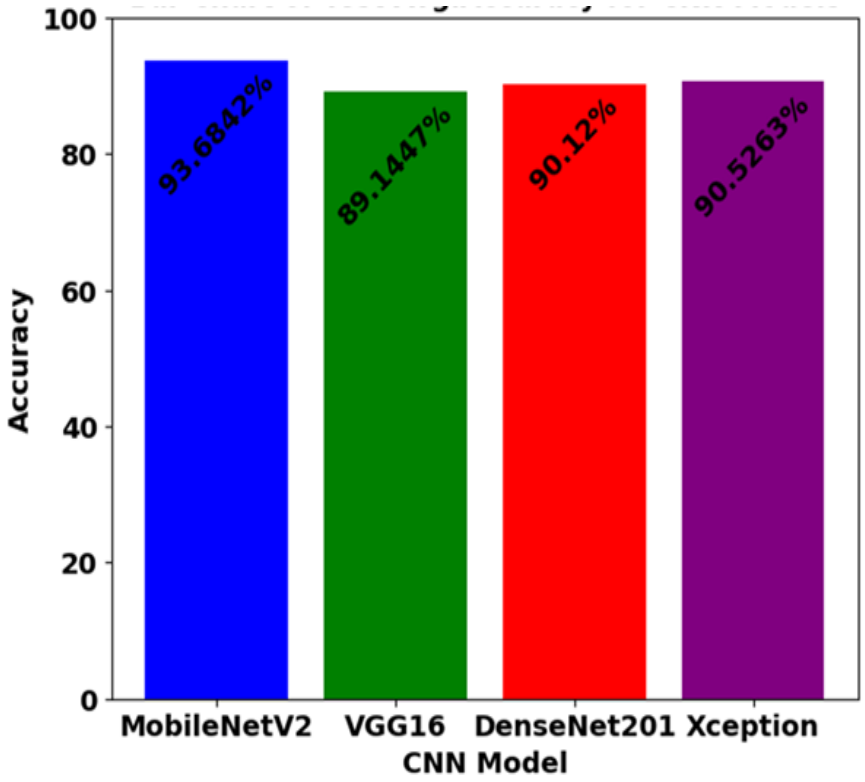

Fig. 9 Bar chart of CNN models for test accuracy

In the classification of various plant disease, MobileNetV2 emerged as the most effective model, achieving the highest average test accuracy of 93.684%. Xception followed closely with an accuracy of 90.526%, which effectively captures intricate patterns in plant disease images. VGG16, with a moderate accuracy of 89.144%, performed reasonably. Conversely, DenseNet50, despite its depth and residual learning capabilities, underperformed significantly with an accuracy of 90.12%, likely due to overfitting and poor generalization to the dataset. Overall, lightweight models like MobileNetV2 and Xception displayed superior performance. This analysis highlights the superiority of MobileNetV2 for the classification of various plant disease images, achieving the highest test accuracy. The training loss of each CNN model was observed.

Fig.10 depicts the training loss curves for three different CNN models: MobileNetV2, VGG16, and Xception.

The x-axis represents the number of epochs, and the y-axis represents the loss value.

Received: August 08, 2025

International Journal of Applied Mathematics
Volume 38 No. 12s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

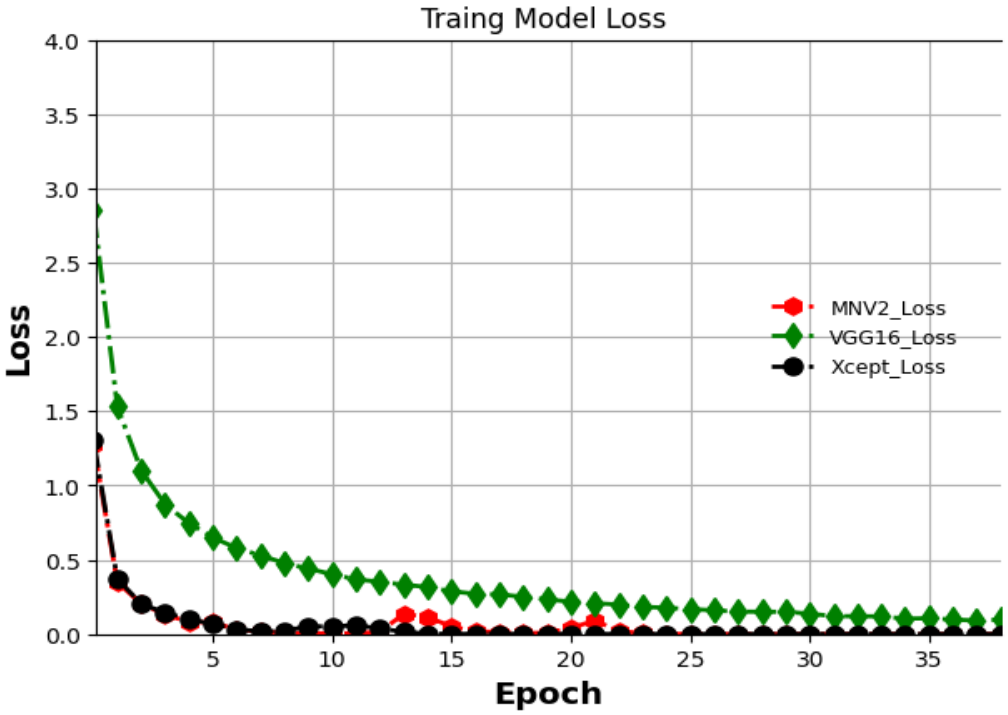

Fig.10 Training Loss Comparison for CNN Models

We observed the **MobileNetV2** model exhibits the highest initial loss among the four models. It shows a rapid decrease in loss during the initial epochs, indicating fast convergence. It can be seen that the loss curve converges and stabilizes around the 15th epoch, which shows that the model reached a minimum. The VGG16 model was then observed, which had a moderate initial loss to start. Also, its loss curve has a gradual decline throughout the training process. The curve converges at approximately the 25th epoch, showing that the model has reached stability. Then, the DenseNet201 was observed, starting with the least initial loss amongst the four models. The initial stages of training saw a rapid drop in loss for it. Its loss curve leveled out similarly to MobileNetV2 around the 15th epoch. Finally, the Xception model started off with a reasonable initial loss. Its loss curve also showed a gradual decline in the range throughout the training process, similar to VGG16. This curve converges at approximately the 25th epoch, showing convergence.

As a result, this study identified the convergence of MobileNetV2 and DenseNet201, which seemed to converge faster than VGG16 and Xception. Then, model complexity was observed as the models might influence their training dynamics. VGG16 and Xception, being deeper and more complex models, might require more epochs to converge.

The Fig. 11 confusion matrix indicates that the MNV2 model exhibits strong overall performance in classifying white blood cells. Most predictions are correct, as seen by the high values along the diagonal of the matrix.

Received: August 08, 2025

International Journal of Applied Mathematics
Volume 38 No. 12s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

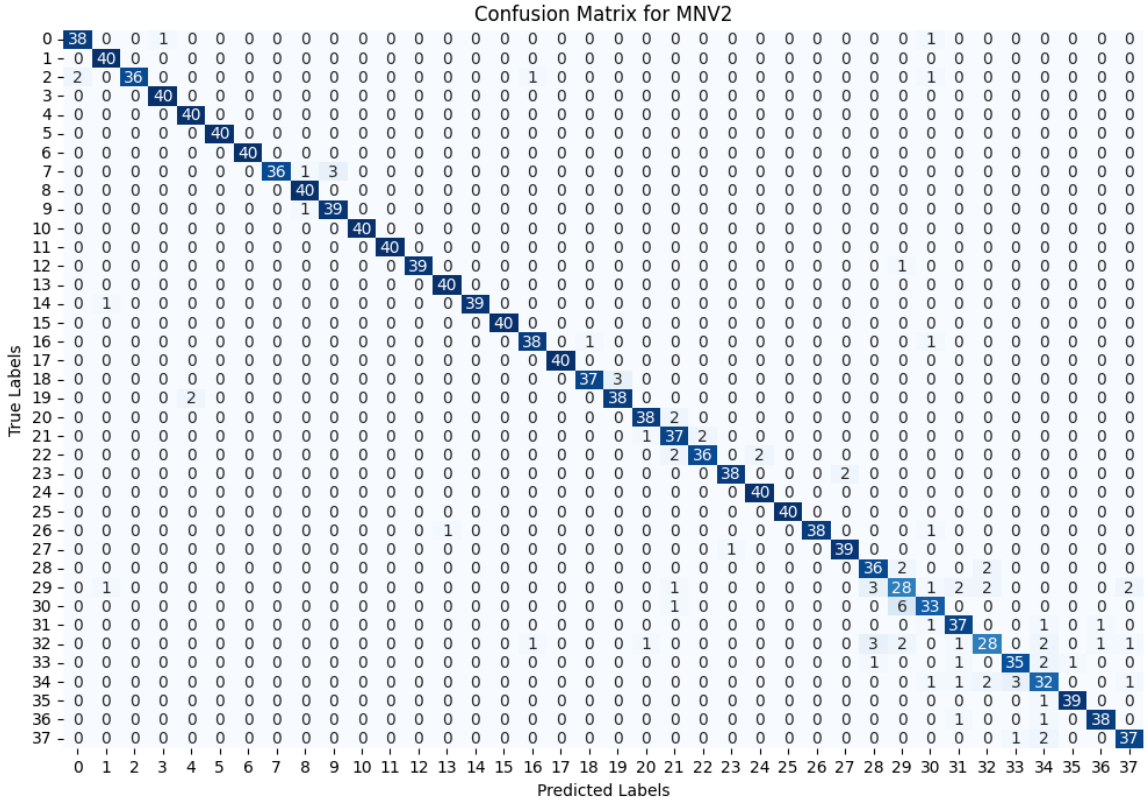

Fig.11 Confusion matrix of MNV2 Model for plant diseases Classification

The model achieves an overall accuracy of 93.684% on the test set. The model performs well on the diagonal value of the matrix is 40 with perfect precision and recall. The model struggles with the 36 and 38 diagonal value with lower precision and recall. The diagonal value 40 shows the highest precision and F1 score. The confusion matrix reveals that the model tends to misclassify if the diagonal value is less than 38 and vice versa. Both MNV2 and Xception exhibit similar strong performance for plant disease data base. However, MNV2 shows slightly better performance in classifying the plant disease data base.

The confusion matrix for MNV2 + KNN (k=8) is illustrated in Fig.12. It indicates that the proposed model performs exceptionally well in classifying 37 plant disease categories. Almost all values are concentrated along the diagonal, indicating strong agreement between true and predicted labels. Indeed, many classes like 2,4,5,6,8,10-15,19, and 24-27 achieved perfect accuracy (40/40 correct predictions), which in turn reflects the ability of the model to distinguish between visually quite different diseases. However, a few classes suffered from limited and imbalanced training data; minor misclassifications can be seen for classes including 7, 22, 28, 32, 33, and 34, which amounts to 1-3 samples per class. Importantly, since errors are scattered rather than systematic, no major cross-class confusion is observed – which confirms that MobileNetV2 features provide strong separability across most categories. Overall, the confusion matrix aligns with the high average F1-score (≈0.986) reported earlier, highlighting that MNV2 + KNN is highly effective for plant disease detection. A few challenging classes requires further refinement through feature enhancement or data augmentation.

Received: August 08, 2025

International Journal of Applied Mathematics
Volume 38 No. 12s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

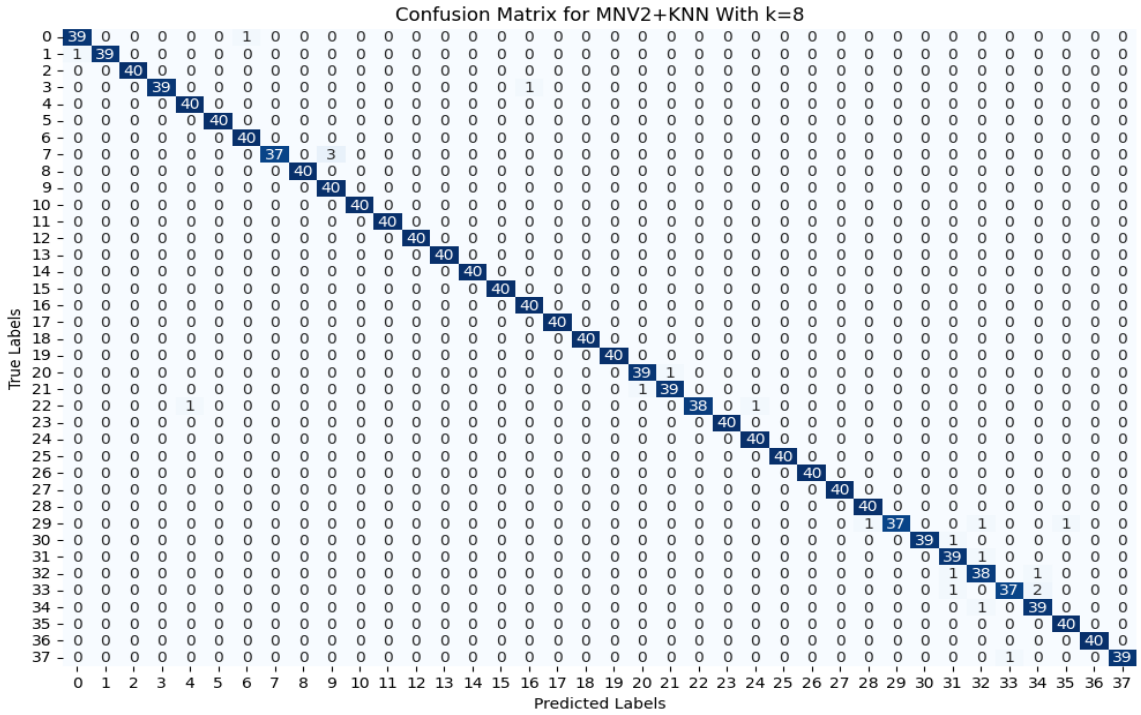

Fig.12 Confusion matrix of MNV2+ KNN Model for plant diseases Classification

The bar chart as shown in Fig. 13 describes the average test accuracy achieved by different combinations of MobileNetV2 (MNV2) deep feature extraction with machine learning classifiers for plant disease detection. Four models have been compared: MNV2 + ELM, MNV2 + KNN (k=8), MNV2 + SVM, and MNV2 + Random Forest**.** Out of all the four models**,** MNV2 + SVM achieves the highest accuracy at 98.95%**,** which highlights the effectiveness of Support Vector Machines in handling high-dimensional feature spaces generated by deep CNNs and in creating optimal decision boundaries between visually similar disease classes. Another model MNV2 + KNN has also performed very well with accuracy of 98.55%, benefiting from the strong discriminative power of MNV2 embeddings that make instance-based classification highly effective. The next model MNV2 + Random Forest achieved 98.03% accuracy, which shows its strength as a stable ensemble method but it performs slightly lower than KNN and SVM due to its limitations in capturing subtle inter-class variations. Finally, the lowest accuracy of 97.35% is recorded by MNV2 + ELM, though still strong, which might have resulted due to ELM's reliance on random weight initialization, sometimes leading to less consistent generalization. Overall, it is evident from the bar chart that MNV2 + SVM stands out as the most reliable and effective approach for plant disease classification in comparison to various other combination models which yield high accuracy above 97%. This model is **clo**sely followed by KNN, confirming the advantage of pairing deep CNN feature extraction with robust machine learning classifiers.

Received: August 08, 2025

International Journal of Applied Mathematics
Volume 38 No. 12s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

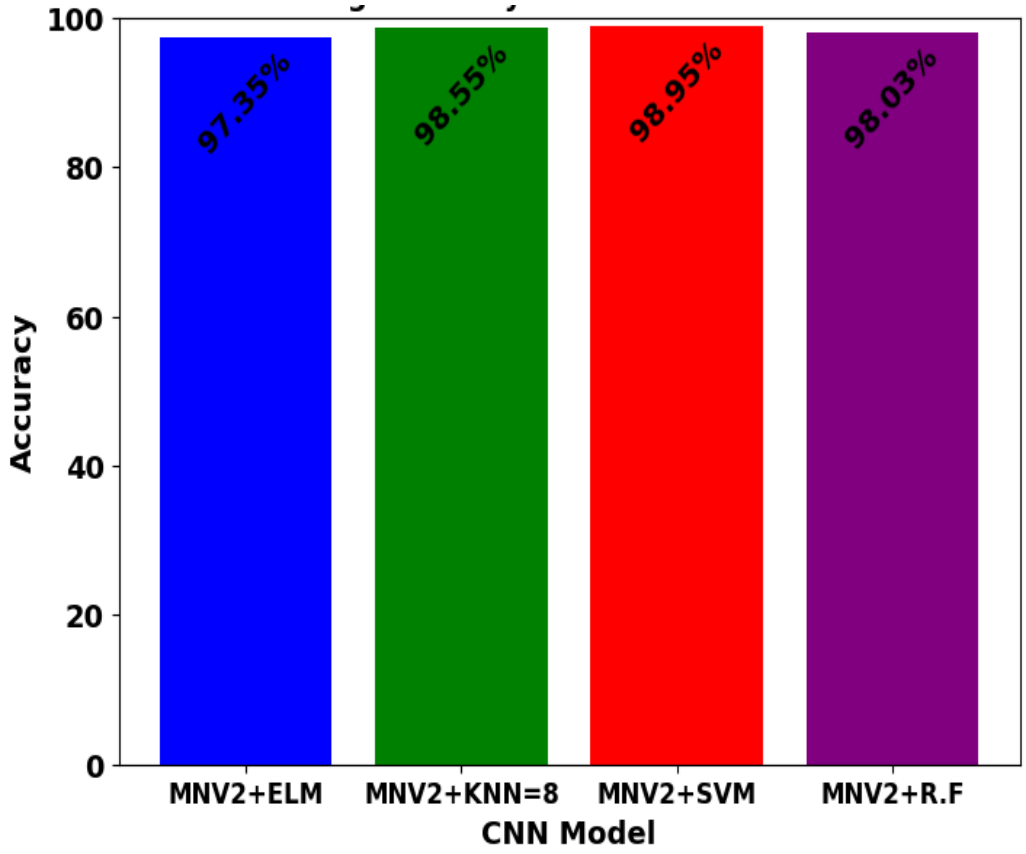

Fig.13 Test Accuracy Comparison of CNN+ML Models

Figure – 14 demonstrates the confusion matrix for MNV2 + SVM with a linear kernel. This model achieves extremely high accuracy in plant disease classification across the 37 categories. Almost all predictions lie along the diagonal, which clearly depicts that the true and predicted labels are in strong agreement. Many classes, such as 0, 1, 2, 4, 5, 6, 8, 10–15, 17, and 23–27, are classified perfectly with 100% correct predictions 40/40, which shows that the linear SVM in combination with MobileNetV2 deep features, is able to construct highly discriminative decision boundaries. In few classes, Minor misclassifications do appear, for example, class 3 (39/40), class 7 (39/40), class 19 (38/40), class 21 (39/40), class 28 (38/40), and classes 30–34 where one to three samples are misclassified. The scattered errors indicate that some plant diseases share similar visual symptoms. This makes them harder to separate even with deep features, or that there may be slight class imbalance in the dataset. Importantly, there is no evidence of systematic cross-class confusion, which confirms that the linear kernel SVM is effective at separating most plant disease categories. Overall, this confusion matrix supports the previously reported high average precision, recall, and F1-score (≈0.99), highlighting that MNV2 + linear SVM is one of the most reliable combinations for plant disease detection. It only have a very few challenging classes which requires further refinement.

Received: August 08, 2025

International Journal of Applied Mathematics
Volume 38 No. 12s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

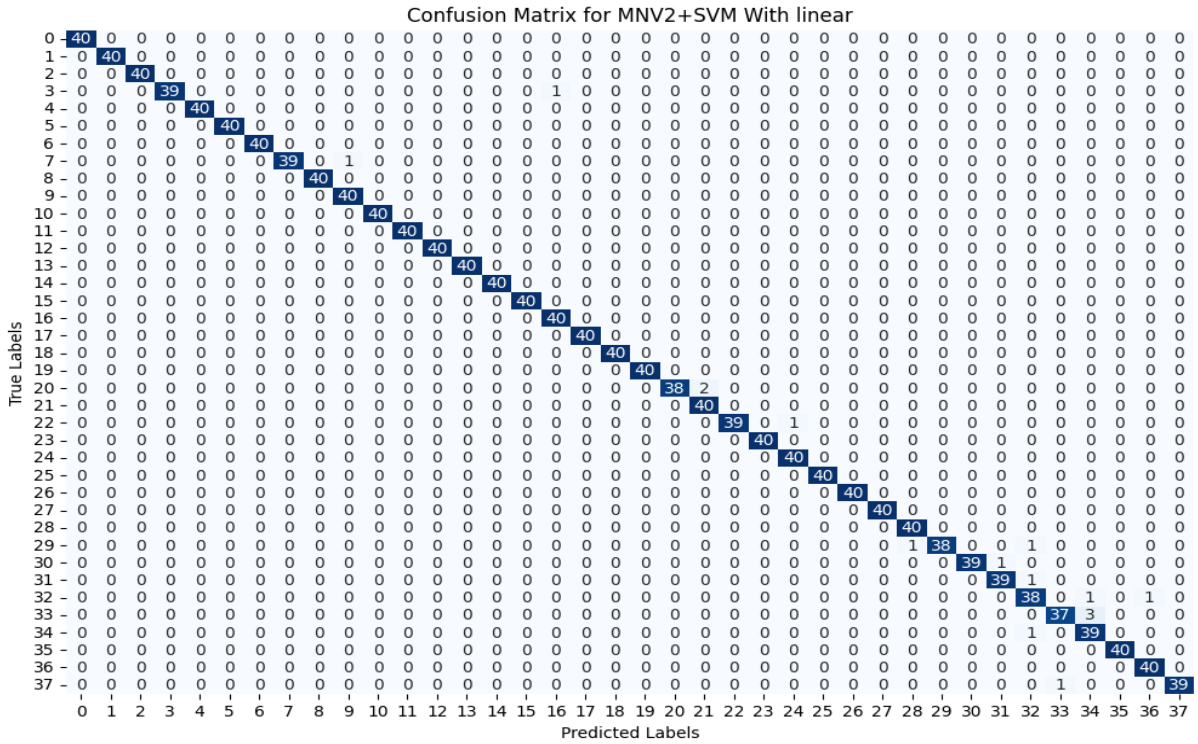

Fig.14 Confusion matrix of MNV2+ KNN Model for plant diseases Classification

The simulation results of plant disease classification across 37 categories (labelled 0–37) using MobileNetV2 (MNV2) as the feature extractor and various machine learning algorithms as classifiers, are summarized in Table-1. The selected classifiers include Extreme Learning Machine (ELM), K-Nearest Neighbours (KNN), Support Vector Classifier (SVC), Random Forest, and Gaussian Naïve Bayes. The performance has been evaluated in terms of precision, recall, and F1-score. Overall, the results demonstrate that the choice of classifier plays a significant role in determining the accuracy of disease recognition, even though all methods use the same deep feature backbone (MNV2). Among the models, SVC delivered the best performance with an average precision, recall, and F1-score of around 0.99, followed closely by KNN with 0.986, Random Forest with 0.98, while ELM and Gaussian Naïve Bayes achieved relatively lower scores of about 0.966–0.968. These variations are indicative of the strengths and limitations of each classifier when applied to complex, multi-class plant disease datasets.

The superior performance of MNV2+SVC can be explained by the properties inherent in the Support Vector Classifier. SVC is particularly effective in high-dimensional feature spaces, which becomes crucial when dealing with deep learning features extracted by MobileNetV2. It constructs the optimal decision boundary to maximize the margin between classes, hence being inherently robust against class overlaps and noisy samples. This explains why SVC achieved almost perfect precision and recall in many categories, especially for diseases with highly discriminative patterns. KNN also competed well because the classification of KNN is distance-based; when MobileNetV2 produces compact and meaningful embeddings, KNN can classify diseases effectively based on the comparison of feature similarity. Random Forest, while stable and interpretable, is based on ensembles of decision trees that

Received: August 08, 2025

International Journal of Applied Mathematics
Volume 38 No. 12s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

occasionally may fail to capture the subtle differences between classes which look similar in appearance.

In contrast, MNV2 + ELM and MNV2 + Gaussian Naïve Bayes had some drawbacks. Although ELM has been efficient in computation, it could suffer from random weight initialization in the hidden layers, which may lead to unstable results in highly complex datasets. Gaussian Naïve Bayes, by its design, assumes feature independence and relies on probabilistic decisions. However, deep features extracted by MobileNetV2 are often correlated and non-linear, making this assumption quite unrealistic. Due to this, both ELM and Naïve Bayes obtained lower performances, especially for some of the hard classes such as labels 28, 29, 31, 32, and 34, where the F1-score fell down to approximately 0.88-0.94. Hence, it may be concluded that while features from deep learning are powerful, simpler classifiers bereft of feature dependency handling capabilities may not perform under multi-class plant disease datasets.

Finally, per-class result analysis discloses that not all classes of plant diseases are equally easy to classify. Several classes like 10-13, 23, 25, and 26 were classified perfectly by all classifiers, which supported the fact that those diseases have typical visual symptoms, easily modeled by MNV2 features. On the other hand, diseases from classes 28, 29, 31, 32, and 34 were difficult to classify, possibly because of high intra-class variation, inter-class similarity, or imbalanced data distribution. These cases emphasized the need for robust classifiers and sufficient training samples. To conclude, from Table 1, it can be evinced that leveraging MobileNetV2 with more advanced classifiers, especially SVC, leads to almost flawless plant disease detection, while simpler models like Naïve Bayes and ELM struggle to grasp substantial patterns in disease representation. Such results highlight the importance of strong feature extraction and the careful selection of a classifier in order to achieve a reliable and scalable agricultural disease monitoring system.

Table 1. Indicates the simulation result of plant disease classification on 37 categories (labelled 0-37 are shown in section 2.1)

MNV2 + ELM				**MNV2 + KNN**			**MNV2 + SVC**			**MNV2 + Random Forest**			**Gaussian Navy**		
Plant Disease	**Precision**	**Recall**	**F1-Score**	**Precision**	**Recall**	**F1-Score**	**Precision**	**Recall**	**F1-Score**	**Precision**	**Recall**	**F1-Score**	**Precision**	**Recall**	**F1-Score**
0	0.93	0.95	0.94	0.97	0.97	0.97	1	1	1	1	0.95	0.97	0.91	0.97	0.94
1	0.93	0.97	0.95	1	0.97	0.99	1	1	1	0.98	1	0.99	0.95	1	0.98
2	0.	0.	0.	1	1	1	1	1	1	0.9	1	0.	0.9	0.	0.

International Journal of Applied Mathematics
Volume 38 No. 12s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

	97	97	97							8		99	7	95	96
3	0.97	0.95	0.96	1	0.97	0.99	1	0.97	0.99	1	0.97	0.99	1	0.97	0.99
4	1	0.95	0.97	0.98	1	0.99	1	1	1	1	1	1	1	1	1
5	0.97	0.97	0.97	1	1	1	1	1	1	0.98	1	0.99	0.98	1	0.99
6	1	1	1	0.98	1	0.99	1	1	1	1	1	1	0.98	1	0.99
7	1	0.93	0.96	1	0.93	0.96	1	0.97	0.99	0.97	0.9	0.94	0.97	0.95	0.96
8	1	1	1	1	1	1	1	1	1	0.98	1	0.99	1	0.97	0.99
9	0.93	1	0.96	0.93	1	0.96	0.98	1	0.99	0.93	0.97	0.95	0.93	0.93	0.93
10	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
11	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
12	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
13	1	1	1	1	1	1	1	1	1	1	1	1	1	0.95	0.97
14	1	0.97	0.99	1	1	1	1	1	1	1	1	1	1	1	1
15	0.93	1	0.96	1	1	1	1	1	1	0.98	1	0.99	0.98	1	0.99
16	0.95	0.97	0.96	0.98	1	0.99	0.98	1	0.99	0.95	0.95	0.95	0.95	0.93	0.94
17	0.95	0.97	0.96	1	1	1	1	1	1	1	1	1	1	1	1
18	0.98	1	0.99	1	1	1	1	1	1	1	1	1	0.93	1	0.96
19	0.97	0.97	0.97	1	1	1	1	1	1	1	1	1	1	1	1
20	0.97	0.95	0.96	0.97	0.97	0.97	1	0.95	0.97	0.97	0.95	0.96	0.95	0.97	0.96

Received: August 08, 2025

International Journal of Applied Mathematics
Volume 38 No. 12s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

21	0.95	0.95	0.95	0.97	0.97	0.97	0.95	1	0.98	0.97	0.97	0.97	0.97	0.95	0.96
22	1	0.97	0.99	1	0.95	0.97	1	0.97	0.99	1	0.97	0.99	0.97	0.97	0.97
23	0.98	1	0.99	1	1	1	1	1	1	1	1	1	1	1	1
24	0.97	0.95	0.96	0.98	1	0.99	0.98	1	0.99	0.97	0.97	0.97	0.97	0.97	0.97
25	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
26	1	1	1	1	1	1	1	1	1	1	0.97	0.99	1	0.95	0.97
27	1	0.97	0.99	1	1	1	1	1	1	1	1	1	1	1	1
28	0.89	1	0.94	0.98	1	0.99	0.98	1	0.99	0.95	0.97	0.96	0.95	0.93	0.94
29	0.95	0.88	0.91	1	0.93	0.96	1	0.95	0.97	0.95	0.93	0.94	0.94	0.85	0.89
30	0.97	0.9	0.94	1	0.97	0.99	1	0.97	0.99	0.95	0.95	0.95	0.89	1	0.94
31	0.89	0.97	0.93	0.93	0.97	0.95	0.97	0.97	0.97	0.97	0.95	0.96	1	0.95	0.97
32	0.97	0.8	0.88	0.93	0.95	0.94	0.93	0.95	0.94	0.93	0.95	0.94	0.92	0.88	0.9
33	0.93	0.97	0.95	0.97	0.93	0.95	0.97	0.93	0.95	0.97	0.93	0.95	0.95	0.9	0.92
34	0.97	0.88	0.92	0.93	0.97	0.95	0.91	0.97	0.94	0.93	1	0.96	0.85	1	0.92
35	0.93	1	0.96	0.98	1	0.99	1	1	1	0.98	1	0.99	0.98	1	0.99
36	0.95	0.97	0.96	1	1	1	0.98	1	0.99	0.98	1	0.99	0.97	0.97	0.97
37	0.97	0.97	0.97	1	0.97	0.99	1	0.97	0.99	1	0.97	0.99	1	0.9	0.95
Avera	0.96	0.96	0.96	0.986	0.984	0.986	0.99	0.989	0.99	0.98	0.979	0.98	0.97	0.96	0.96

Received: August 08, 2025

International Journal of Applied Mathematics
Volume 38 No. 12s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

ge	8	6	6											8	8

3 Conclusion and Future Works

These results clearly indicate that the integration of deep feature extraction by MobileNetV2 with machine learning classifiers provides very accurate plant disease detection. The best performance was realized by SVM at 98.95%, closely followed by KNN and Random Forest, while ELM performed comparatively lower but above 97%. Similarly, the near-perfect classification shown by the confusion matrices and accuracy comparisons underlines the robustness of deep CNN features in capturing disease-specific patterns and the selection of appropriate classifiers to attain such optimal results. However, the visually similar disease classes caused some misclassifications, showing that further refinement is needed. In addition, the approach can be extended as future scope by incorporating larger and more diverse datasets, fine-grained classification methods, and advanced techniques such as ensemble deep learning, attention mechanisms, or transformer-based architectures to further improve accuracy and generalization. Additionally, the integration of this system into real-time mobile or IoT-based applications could provide farmers with practical, on-field disease diagnosis tools that would make the solution more scalable and impactful for precision agriculture.

Statement:

Conflicts of interest: All authors declare that they have no conflicts of interest.

References:

1. Yang, X., & Guo, W. (2017). Automatic image-based plant disease severity estimation using deep learning. Computational Intelligence and Neuroscience, 2017, 2917536. https://doi.org/10.1155/2017/2917536.
2. Padol, P., & Yadav, A. (2016). SVM classifier based grape leaf disease detection. 2016 Conference on Advances in Signal Processing (CASP), Pune, India, 175–179. https://doi.org/10.1109/CASP.2016.7746160.
3. Patil, J. K., & Kumar, R. (2017). An advanced technique for leaf disease detection using k-means clustering. International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, 6(1), 138–142.
4. Masazhar, M. H., & Kamal, M. M. (2017). Palm leaf disease detection using support vector machine. Indonesian Journal of Electrical Engineering and Computer Science, 5(3), 566–572.
5. Demilie, W.B. Plant disease detection and classification techniques: a comparative study of the performances. *J Big Data* **11**, 5 (2024).
6. Sujatha, R., et al. "Advancing plant leaf disease detection integrating machine learning and deep learning." *Scientific Reports* 15.1 (2025): 11552.

Received: August 08, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

7. Sarkar, Chittabarni, et al. "Leaf disease detection using machine learning and deep learning: Review and challenges." *Applied Soft Computing* 145 (2023): 110534.
8. Sujatha, Radhakrishnan, et al. "Performance of deep learning vs machine learning in plant leaf disease detection." *Microprocessors and Microsystems* 80 (2021): 103615.
9. Singh, V., & Misra, A. K. (2017). Detection of plant leaf diseases using image segmentation and soft computing techniques. Information Processing in Agriculture, 4(1), 41–49.
10. Shoaib, Muhammad, et al. "An advanced deep learning models-based plant disease detection: A review of recent research." *Frontiers in Plant Science* 14 (2023): 1158933.
11. Haridasan, Amritha, Jeena Thomas, and Ebin Deni Raj. "Deep learning system for paddy plant disease detection and classification." *Environmental monitoring and assessment* 195.1 (2023): 120.
12. Li, Lili, Shujuan Zhang, and Bin Wang. "Plant disease detection and classification by deep learning—a review." *IEEE access* 9 (2021): 56683-56698.
13. Moupojou, Emmanuel, et al. "FieldPlant: A dataset of field plant images for plant disease detection and classification with deep learning." *IEEE Access* 11 (2023): 35398-35410.
14. E. Moupojou *et al*., "FieldPlant: A Dataset of Field Plant Images for Plant Disease Detection and Classification With Deep Learning," in *IEEE Access*, vol. 11, pp. 35398-35410, 2023, doi: 10.1109/ACCESS.2023.3263042.
15. Pujari, J. D., Yakkundimath, R., & Byadgi, A. S. (2016). Image processing based detection of fungal diseases in plants. Procedia Computer Science, 46, 1802–1808.
16. Larijani, H. T., et al. (2019). Classification of rice blast disease using K-means clustering algorithm in Lab color space. International Journal of Scientific & Technology Research, 8(9), 302–307.
17. Pandey, S., et al. (2021). Automated detection of chlorosis in black gram leaves using SVM. Ecological Informatics, 64, 101351.
18. Joshi, D., et al. (2021). CNN-based plant leaf disease classification for Vigna mungo. Materials Today: Proceedings, 47, 427–433.
19. Atila, U., et al. (2021). Plant leaf disease classification using EfficientNet and transfer learning. Computers and Electronics in Agriculture, 190, 106527.
20. Liu, Y., et al. (2021). Hyperspectral imaging for field-based plant species classification using deep learning. Biosystems Engineering, 207, 50–60.
21. Yadav, S. P., et al. (2021). Deep learning approach for peach crop bacteriosis detection. Ecological Informatics, 62, 101285.
22. Liang, W., et al. (2019). PD2SE-Net: A novel hybrid CNN for plant disease severity estimation using ResNet-50. Computers and Electronics in Agriculture, 160, 12–20.
23. Ramcharan, A., et al. (2017). Transfer learning for cassava disease detection on mobile devices. Computers and Electronics in Agriculture, 136, 15–22.

Received: August 08, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

24. L. Li, S. Zhang and B. Wang, "Plant Disease Detection and Classification by Deep Learning—A Review," in *IEEE Access*, vol. 9, pp. 56683-56698, 2021, doi: 10.1109/ACCESS.2021.3069646.
25. Ali Hussein Ali, Ayman Youssef, Mahmoud Abdelal, Muhammad Adil Raja, An ensemble of deep learning architectures for accurate plant disease classification, Ecological Informatics, Volume 81, 2024, 102618, ISSN 1574-9541, https://doi.org/10.1016/j.ecoinf.2024.102618.
26. Jung, M., Song, J.S., Shin, AY. *et al.* Construction of deep learning-based disease detection model in plants. *Sci Rep* **13**, 7331 (2023). https://doi.org/10.1038/s41598-023-34549-2
27. A. Bhargava, A. Shukla, O. P. Goswami, M. H. Alsharif, P. Uthansakul and M. Uthansakul, "Plant Leaf Disease Detection, Classification, and Diagnosis Using Computer Vision and Artificial Intelligence: A Review," in *IEEE Access*, vol. 12, pp. 37443-37469, 2024, doi: 10.1109/ACCESS.2024.3373001
28. Ramesh, Shima, and Ramachandra Hebbar. "Plant disease detection using machine learning." *2018 International conference on design innovations for 3Cs compute communicate control (ICDI3C)*. IEEE, 2018.
29. Jackulin, C., and S. J. M. S. Murugavalli. "A comprehensive review on detection of plant disease using machine learning and deep learning approaches." *Measurement: Sensors* 24 (2022): 100441.
30. Ahmed, Imtiaz, and Pramod Kumar Yadav. "Plant disease detection using machine learning approaches." *Expert Systems* 40.5 (2023): e13136.
31. Vasavi, Pallepati, Arumugam Punitha, and T. Venkat Narayana Rao. "Crop leaf disease detection and classification using machine learning and deep learning algorithms by visual symptoms: A review." *International Journal of Electrical and Computer Engineering* 12.2 (2022): 2079.
32. Balafas, Vasileios, et al. "Machine learning and deep learning for plant disease classification and detection." *IEEE Access* 11 (2023): 114352-114377.
33. Panchal, Adesh V., et al. "Image-based plant diseases detection using deep learning." *Materials Today: Proceedings* 80 (2023): 3500-3506.
34. Chug, Anuradha, et al. "A novel framework for image-based plant disease detection using hybrid deep learning approach." *Soft Computing* 27.18 (2023): 13613-13638.
35. Khan, Rehan Ullah, et al. "Image-based detection of plant diseases: from classical machine learning to deep learning journey." *Wireless Communications and Mobile Computing* 2021.1 (2021): 5541859.
36. Wani, Javaid Ahmad, et al. "Machine learning and deep learning based computational techniques in automatic agricultural diseases detection: Methodologies, applications, and challenges." *Archives of Computational methods in Engineering* 29.1 (2022): 641-677.

Received: August 08, 2025