

Deep Learning-Based Multi-Class Tomato Leaf Disease Classification for Precision Agriculture

Original Scientific Paper

R. Meenakshi

Chennai Institute of Technology
Department of Computer Science and Engineering
Chennai, Tamil Nadu, India
meenakshir@citchennai.net

Sneha Joshi

Malla Reddy College of Engineering
Department of Mathematics
Hyderabad, Telangana, India
drjoshisneha27@gmail.com

R. Shobiga

J.J. College of Engineering and Technology
Department of Electronics and Communication
Engineering
Tiruchirappalli, Tamil Nadu, India
shobigaraj123@gmail.com

E. Vijaya Babu

VNR Vignana Jyothi Institute of Engineering and
Technology
Department of Electronics and Communication
Engineering
Hyderabad, Telangana, India
vijayababu.e@gmail.com

M. Radhika

R.M.D. Engineering College
Department of Information Technology
Kavaraipettai, Chennai, Tamil Nadu, India
radikamoorthi@gmail.com

C. Srinivasan*

Saveetha School of Engineering, Saveetha Institute of
Medical and Technical Sciences,
Saveetha University, Department of Computer Science
and Engineering
Chennai, Tamil Nadu, India
srinivasanchelliah@gmail.com

*Corresponding author

Abstract – Diseases affecting tomato leaves substantially decrease crop output and quality, resulting in economic losses if not identified quickly. Manual inspection is labour-intensive and sometimes imprecise, making it inappropriate for extensive agricultural operations. This research presents a deep learning framework using DenseNet-121 for the automatic multi-class classification of tomato leaf diseases. The model was trained and validated using the PlantVillage dataset, which contains more than 54,000 labelled pictures of tomato leaves classified into Early Blight, Late Blight, Leaf Mold, and Healthy categories. Training utilized RMSprop optimization, categorical cross-entropy loss, and data augmentation methods such as rotation, flipping, and brightness modifications to enhance generalization. The proposed approach achieved an accuracy of 99.17%, precision of 98.34%, and F1-score of 98.33%, surpassing baseline models. In addition to numerical performance, the system facilitates practical applications: it may be included in mobile applications, IoT-based monitoring systems, or cloud platforms, enabling farmers to identify illnesses in real time, implement early treatments, and mitigate crop losses. This study illustrates the capabilities of AI-enhanced precision agriculture, providing a scalable and dependable approach for sustainable crop management.

Keywords: Tomato Leaf Disease, Deep Learning, DenseNet-121, Multi-Class Classification, Automated Disease Detection.

Received: August 27, 2025; Received in revised form: October 8, 2025; Accepted: October 9, 2025

1. INTRODUCTION

Plant diseases greatly impact agricultural output, resulting in considerable yield losses and economic challenges for farmers. Despite their widespread cultivation, tomatoes are susceptible to several foliar diseases, such as mosaic viruses. Identifying this disease quickly

and accurately is crucial for administering timely treatments and minimizing crop loss. The use of human examiners in traditional disease detection methods is labour-intensive, prone to error, and time-consuming. Due to automated, accurate, and economical classification models made possible by recent advances in artificial intelligence (AI), particularly DL, plant dis-

ease detection has been revolutionized. Due to their ability to extract and interpret complex patterns from leaf images, Convolutional Neural Networks (CNNs) have demonstrated remarkable effectiveness in disease detection using images. This research presents a DL-driven multi-class classification method for tomato leaf diseases in precision agriculture. Technology uses a CNN-based methodology to accurately classify various tomato leaf diseases, facilitating informed decision-making for farmers and agricultural specialists. Using an improved DL architecture for training, the proposed model employs a dataset of pre-processed images of tomato leaves to enhance feature extraction. The use of AI-driven technology in precision agriculture optimizes disease control tactics, reduces chemical pesticide usage, and promotes overall crop health and output. The paper presents key contributions as follows.

Standard CNNs, such as VGG, AlexNet, ResNet, and GoogleNet, with transfer learning, are widely used for various image classification tasks. However, these architectures are unable to fully leverage the feature reuse and connection patterns provided by the DenseNet architecture. Hence, the proposed work selects DenseNet-121 as the base model for classifying tomato leaf diseases, demonstrating a measurable improvement in accuracy compared to existing models, thereby advancing the field of automated plant disease detection in precision agriculture.

- Developed a framework using DenseNet-121 for the classification of many tomato leaf diseases.
- Utilized dense connections to improve feature reuse and reduce gradient vanishing in deep networks.
- Implemented data augmentation and optimized training procedures to enhance model generalization
- Provided the methodology for cross-crop adaptation, facilitating wider use in precision agriculture.

Cross-crop relevance enhances motivation by demonstrating deep learning models' capacity to generalise beyond tomato disease detection, hence evidencing their resilience, scalability, and adaptability across diverse agricultural settings, underlining wider practicality. Existing works may be classified as CNN-based, YOLO-based, and hybrid models. Convolutional Neural Networks face overfitting and limited feature extraction, YOLO might miss subtle patterns, and hybrid models enhance representation while augmenting complexity. DenseNet-121 addresses these limitations by efficient and robust classification. To detect and classify various tomato leaf diseases, an improved capsule neural network (CapsNet) is employed in this computer vision approach using standard dataset images [1]. This research proposes the use of a deep multi-scale CNN as a new method for automatically classifying tomato leaf diseases into multiple classes [2]. The YOLOv8 model was employed to extract leaf images from tomato plant images, encompassing seven disease clas-

sifications, and multiple DL models were trained to utilize images [3]. Using several YOLO models trained on the features of images of diseases affecting tomatoes, this work investigates these diseases in their natural habitats [4]. Classifying diseases affecting tomato plant leaves using CNN, a DL approach mostly used for image identification and processing data from pixels [5]. The substantial diversity and complexity of tomato leaf diseases are the primary obstacles to their detection, making precise diagnosis challenging. This problem can be addressed by implementing models based on CNN and U-Net architecture [6]. To classify tomato leaves, a CNN and BiLSTM model can work together. Results were improved when using the proposed hybrid learning disease-finding approach for a tomato leaf condition [7]. To detect pathogens in tomatoes, this research proposes an edge-based processing architecture that utilizes a CNN. This method conserves bandwidth and reduces latency by placing computing resources closer to the data source [8].

Examining the performance of InceptionV3, VGG-16, EfficientNetB7, ResNet-50, and VGG-19, among other fine-tuned pre-trained models, in classifying diseases in tomato leaves [9]. This research, utilizing CNN and, more specifically, ResNet-18, may transform the way tomato crop diseases are detected [10]. Using the VGG-16 CNN architecture, trained to detect diseases in tomato leaves [11], applies precision agriculture principles. In this case, it details an SVM-based model that can accurately identify tomato leaf diseases and classify them to a very high degree of reliability [12]. With minimal labelled data, this study introduces a novel method for classifying tomato leaf diseases using a YOLOv8-based iterative learning approach [13]. To classify tomato leaf diseases, this study employed an inexpensive, proprietary CNN model and two TL-based models, namely VGG 16 and VGG 19 [14]. To classify images of tomato leaves, a YOLO v5 DL model was trained [15]. This study aims to determine the optimal hyperparameters for leaf section classification and disease detection. The purpose of this empirical comparison was to classify diseases in tomato leaves using a machine learning (ML) approach [16].

To classify images depicting diseases that affect tomato leaves, the K-Nearest Neighbour (K-NN) method is proposed for use, along with feature extraction [17]. Based on the findings, the KNN algorithm is capable of accurately identifying and classifying diseases affecting tomato leaves. To classify diseases in tomato leaves, a new approach is devised that utilizes deep CNN methods [18]. One potential attempt is the classification of diseases affecting tomato leaves. To classify the data, the methods shown in [19] use SVM and KNN. Tomato leaf disease detection and classification using a hybrid DL approach. This hybrid system is created by combining SVM with a convolutional attention module (CBAM) [20]. The InceptionResNetV2 CNN model is trained using a transfer learning technique to identify diseases in images of rice leaves [21]. This work uses VGG16 CNN

for disease classification of corn leaves; to improve accuracy and decrease training time, it uses Bayesian optimisation and transfer learning [22]. DenseNet-121 addresses issues in previous CNN architectures, including vanishing gradients, limited feature representation, and overfitting. The extensive connections improve gradient flow and feature reuse, increase computing efficiency, and promote generalisation under diverse settings, resulting in better disease classification.

2. PROPOSED SYSTEM

To classify tomato leaf diseases, the proposed method uses DenseNet-121, a DL model, to improve precision agriculture. Fig. 1 depicts a sequential flow of tomato leaf images with preprocessing, feature extraction using DenseNet-121, classification into four classes, and assessment for real-time disease detection implementation.

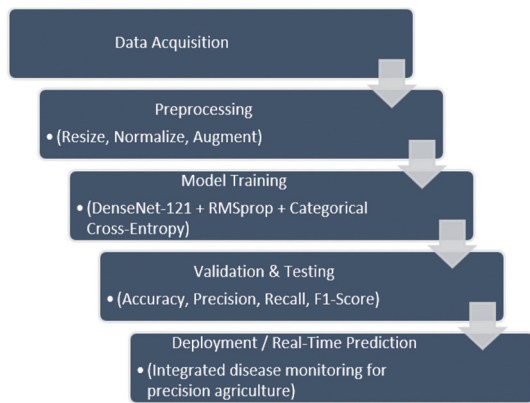


Fig. 1. Workflow of DenseNet-121-Based Multi-Class Disease Detection

The process begins with the input of leaf images, which undergo image preprocessing steps such as resizing to 224×224 pixels, normalization, and data augmentation to enhance model generalization. Data augmentation simulates real-world variations: rotation handles diverse leaf angles, flipping addresses mirrored patterns, and brightness modulation mimics changing lighting conditions. These methods improve model generalization, increase dataset variety, and maintain disease-specific features for resilient classification. These pre-processed images are then passed through the proposed DenseNet-121 model for robust feature extraction, leveraging its dense connectivity for efficient learning. The extracted features are fed into a classification layer, consisting of a fully connected layer followed by a softmax activation function, which outputs the probability distribution across various disease classes. The final output predictions classify the leaf as Healthy, Early Blight, Late Blight, or Leaf Mold, enabling effective plant disease diagnosis. This structured approach ensures the accurate and scalable identification of tomato leaf diseases. Each stage in the proposed DenseNet-121 model is described comprehensively in the following sub-sections.

2.1. DATA ACQUISITION AND PREPROCESSING

The proposed method uses the PlantVillage dataset, which contains annotated images of diseased tomato leaves. Before training, images are subjected to preprocessing to provide consistency. Each image is scaled to 224×224 pixels, the usual input dimension for DenseNet-121, then normalized using Min-Max Scaling to adjust as a result, improving training stability, with pixel values ranging from 0 to 1. The normalization procedure is defined as follows:

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (1)$$

where X is the original pixel intensity, X_{min} and X_{max} represent the minimum and maximum pixel values, and X' is the normalized output. When data is transformed using operations such as rotation (R), flipping (F), and brightness adjustment (B), it improves generalizability. This process is represented as:

$$X_{aug} = R(F(B(X))) \quad (2)$$

These techniques provide the model with exposed diverse visual representations, mitigating overfitting and enhancing robustness.

2.2. DENSENET-121 MODEL ARCHITECTURE

DenseNet can be used for classifying tomato leaf diseases due to its dense connection, which enhances feature reuse and facilitates gradient flow. It provides enhanced feature extraction with a reduced number of parameters and minimizes overfitting. The architecture collects complex and comprehensive features in tomato leaves, thus improving robustness and accuracy in diagnosing various disease types from leaf images. The base model for this system is DenseNet-121, a deep CNN known for its effective feature extraction and enhanced gradient flow. Compared to traditional CNNs, DenseNet-121 uses a dense connection, allowing each layer to receive inputs from all preceding levels. This configuration is technically denoted as:

$$X_l = H_l([X_0, X_1, \dots, X_{l-1}]) \quad (3)$$

where X_l is the feature map at layer l , H_l is the transformation function comprising batch normalization, Rectified Linear Unit (ReLU) activation, and convolution, and $[X_0, X_1, \dots, X_{l-1}]$ denotes the concatenation of outputs from preceding layers. The activation function used by the model is the PReLU, which is defined as:

$$ReLU(x) = \max(0, x) \quad (4)$$

Computing complexity is reduced by adding a Global Average Pooling (GAP) layer after the model's numerous dense blocks. For multi-class classification, the last completely linked layer employs the Softmax activation function.

$$P(y_i) = \frac{e^{z_i}}{\sum_{j=1}^N e^{z_j}} \quad (5)$$

where z_i represents the raw output score for class i , N is the total number of classes, and $P(y_i)$ is the predicted probability. Fig. 2 shows the DenseNet-121 architecture diagram. In Fig. 2, MPL represents the Max Pooling Layer, and APL represents the Average Pooling Layer. z_i represents the raw output score for class i , N is the total number of classes, and $P(y_i)$ is the predicted probability. Fig. 2 shows the DenseNet-121 architecture diagram. In Fig. 2, MPL represents the Max Pooling Layer, and APL represents the Average Pooling Layer.

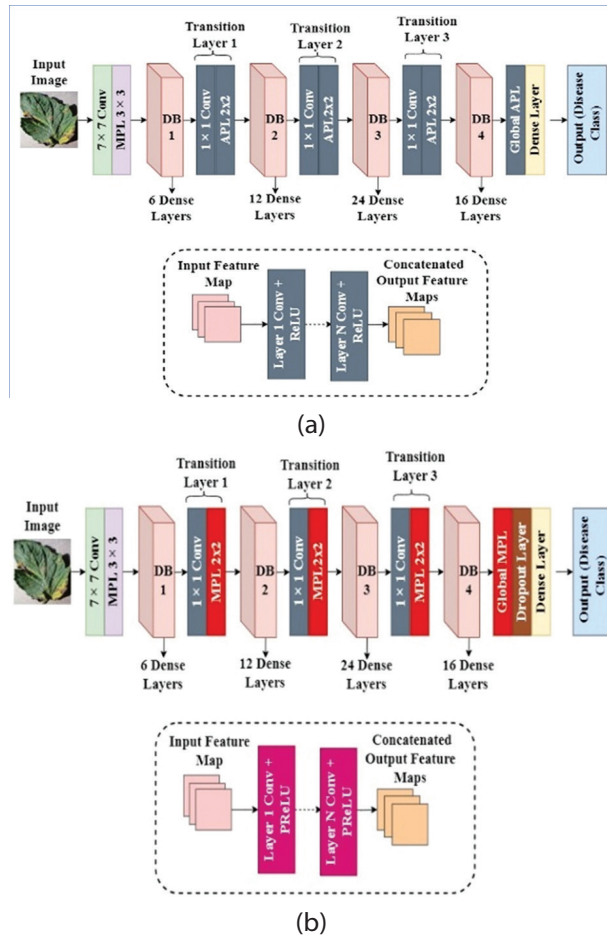


Fig. 2. DenseNet-121 architectures for tomato leaf disease classification. (a) Conventional DenseNet-121, (b) Modified DenseNet-121

The modified DenseNet-121 architecture includes several enhancements for improved performance. ReLU activations are replaced with Parametric ReLU (PReLU) to facilitate adaptive learning of activation parameters. A dropout layer is included to reduce overfitting by randomly deactivating neurons throughout the training process. Furthermore, all average pooling layers are replaced with max pooling layers to more effectively capture dominant features. As the ReLU is less vulnerable to the vanishing gradient problem, PReLU is employed, allowing the learning of negative slopes in the proposed DenseNet-121 model for tomato leaf disease classification. It is defined as

$$PReLU(x) = \begin{cases} \max(0, x) & x \geq 0 \\ ax & x < 0 \end{cases} \quad (6)$$

where a is the learnable parameter. The modified DenseNet-121 for tomato leaf disease classification uses PReLU to reduce the dying neuron issue, thereby improving gradient propagation and feature extraction. Studies indicate [23] that PReLU outperforms standard ReLU, enhancing convergence and resilience in complex classification problems.

DenseNet architecture enhances feature transmission and model efficiency via dense connectivity. The architecture is designed with many Dense Blocks, with each layer receiving input from all the levels below it and sending its feature maps to all the layers above it. Reduced parameter counts and improved gradient flow are the results of extensive feature reuse, which is advantageous for training deep networks. Every layer in a Dense Block comprises a 1×1 convolution (bottleneck layer) succeeded by a 3×3 convolution, both employing PReLU activation. The layers are concatenated instead of summed, thereby maintaining the learned features from the preceding layers. Transition Layers are employed between Dense Blocks to regulate the model's complexity. A Transition Layer comprises reducing the number of feature maps by a 1×1 convolution, followed by a 2×2 average pooling operation to decrease the spatial dimensions. This architecture offers improved generalization, accelerated convergence, and superior performance in complex image classification problems, such as detecting diseases in tomato leaves. It is both profound and concise, rendering it suitable for precision agriculture applications. Table 1 shows the output shape and number of parameters for each DenseNet-121 layer used for disease classification.

Table 1. Summary of the Proposed Densenet-121 Architecture

| Layer (Type) | Output Shape | Parameters |
|---------------------------|----------------------|------------|
| Input Layer | (None, 224, 224, 3) | 0 |
| Conv2D + BatchNorm + ReLU | (None, 112, 112, 64) | 9,408 |
| MPL | (None, 56, 56, 64) | 0 |
| Dense Block 1 (6 layers) | (None, 56, 56, 256) | ~27K |
| Transition Layer 1 | (None, 28, 28, 128) | ~33K |
| Dense Block 2 (12 layers) | (None, 28, 28, 512) | ~208K |
| Transition Layer 2 | (None, 14, 14, 256) | ~66K |
| Dense Block 3 (24 layers) | (None, 7, 7, 512) | ~857K |
| Transition Layer 3 | (None, 7, 7, 512) | ~132K |
| Dense Block 4 (16 layers) | (None, 7, 7, 1024) | ~590K |
| PReLU | (7, 7, 1024) | - |
| GlobalMPL | (None, 1024) | 0 |
| Fully Connected (Dense) | (None, 4) | ~4,100 |
| Softmax Activation | (None, 4) | 0 |

2.3. MODEL TRAINING

Splitting the dataset into training and validation sets helps to better understand the model and evaluate its performance.

The model uses the RMSprop optimizer, which adaptively modifies learning rates to ensure steady convergence. The formula for weight updates in RMSprop is:

$$v_t = \beta v_{t-1} + (1 - \beta) g_t^2 \quad (7)$$

$$w_t = w_{t-1} - \frac{\eta}{\sqrt{v_t + \epsilon}} g_t \quad (8)$$

where the following variables are v_t , which stands for the moving average of squared gradients, g_t for the gradient at time t , w_t for updated weights, η for the learning rate, and ϵ for the constant that guarantees numerical stability.

2.4. DISEASE PREDICTION

The trained model is included in a mobile application or cloud-based system for immediate disease diagnosis in real-time deployment. Agricultural experts can record leaf images using a smartphone or camera, which are then analyzed by the trained DenseNet-121 model for disease classification. The classification uses the Softmax function, which generates probability ratings for each disease group. The disease with the greatest potential is chosen as the definitive prognosis. Upon disease detection, the system provides proposed treatments and preventative methods, facilitating early intervention and minimizing crop loss. Table 2 outlines the key parameters of DenseNet-121 used in the classification system for tomato leaf diseases.

Table 2. DenseNet-121 Parameters

| Parameter | Value |
|---------------------------------|------------------------------|
| Input Size | 224 × 224 × 3 (PlantVillage) |
| Layers | 121 |
| Dense Blocks | 4 |
| Activation | PReLU |
| Optimizer | RMSprop |
| Loss Function | Categorical Cross-Entropy |
| Batch Size | 32 |
| Learning Rate | 0.001 |
| Epochs | 20 |
| Classes | 4 (3 diseases + healthy) |
| Dropout | 0.2 |
| Learning parameter (a) in PReLU | 0.01 |

The hyperparameter values shown in Table 2 are chosen to enhance the system accuracy and training stability based on the following criteria: A 224×224 input is compatible with the DenseNet-121 architecture. It uses RMSprop for adaptive learning rates, and categorical cross-entropy is optimal for multi-class problems. The hyperparameters learning rate, batch size, and dropout were selected experimentally with iterative evaluation and validation performance, rather than using formal optimization methods such as grid search or Bayesian optimization. An organized method for classifying tomato leaf diseases using DenseNet-121 is shown in Table 3.

Table 3. Workflow of DenseNet-121-Based Tomato Leaf Disease Classification

| Step | Description |
|----------------------------|---|
| 1. Data Acquisition | Collect tomato leaf images from the PlantVillage dataset. |
| 2. Data Preprocessing | Normalize the pixel values, resize the images to 224 × 224, and use data augmentation. |
| 3. Model Selection | Use DenseNet-121 for efficient feature extraction and classification. |
| 4. Feature Extraction | By utilizing convolutional layers, DenseNet-121 can extract hierarchical characteristics from input images. |
| 5. Classification | A fully connected Softmax layer classifies various tomato leaves. |
| 6. Training & Optimization | For the model's training, use the RMSprop optimizer and the categorical cross-entropy loss. |
| 7. Model Evaluation | Performs evaluations based on F1-score, recall, accuracy, and precision. |
| 8. Deployment | Utilize the trained model to detect diseases in tomato leaves in real time. |

3. RESULTS AND DISCUSSIONS

ML methods for plant disease identification can benefit from the extensive, freely available PlantVillage dataset [28]. Covering 38 classes and 14 different crop species, it contains almost 54,000 high-quality images of healthy and sick plant leaves, such as tomatoes, potatoes, apples, and grapes. Each image is annotated with the relevant plant species and disease conditions, establishing a robust foundation for training and assessing DL models in plant pathology. Tomato leaf subsets from the PlantVillage dataset were utilized in this research. This subset comprises 4,600 images divided into four classes: Early Blight (800 images), Late Blight (1,500 images), Leaf Mold (800 images), and Healthy Leaves (1,500 images). Each class was increased to 1,500 images using data augmentation techniques to address class imbalance. For the DenseNet-121 architecture to recognize images were all resized to 224×224 pixels. An effective DL model for accurate and reliable disease classification of tomato leaves was developed with the help of this balanced and pre-processed dataset.

The images in PlantVillage are acquired in a research setting with consistent backgrounds, which may introduce dataset bias and limit generalization to outdoor conditions. Incorporating field-collected images, advanced augmentation, or domain adaptation may enhance model robustness and practical application. The model was trained on an NVIDIA RTX 3080 GPU (10 GB VRAM) with TensorFlow 2.12 and Keras on Ubuntu 22.04. Training for 20 epochs with a batch size of 32 took around 2.5 hours, using CUDA 11.8, cuDNN 8.6, NumPy, Pandas, and scikit-learn.

The model performs well on the PlantVillage dataset but may be overfit to controlled environments, resulting in poor performance in real-world situations characterised by changing lighting and occlusions. Validating the model on real-world data is essential to ensure its performance in practical scenarios.

tion on varied field datasets and the use of methods such as data augmentation are crucial for enhancing generalisation and practical applicability. Sample images of different tomato leaf diseases from the PlantVillage dataset are shown in Fig. 3.

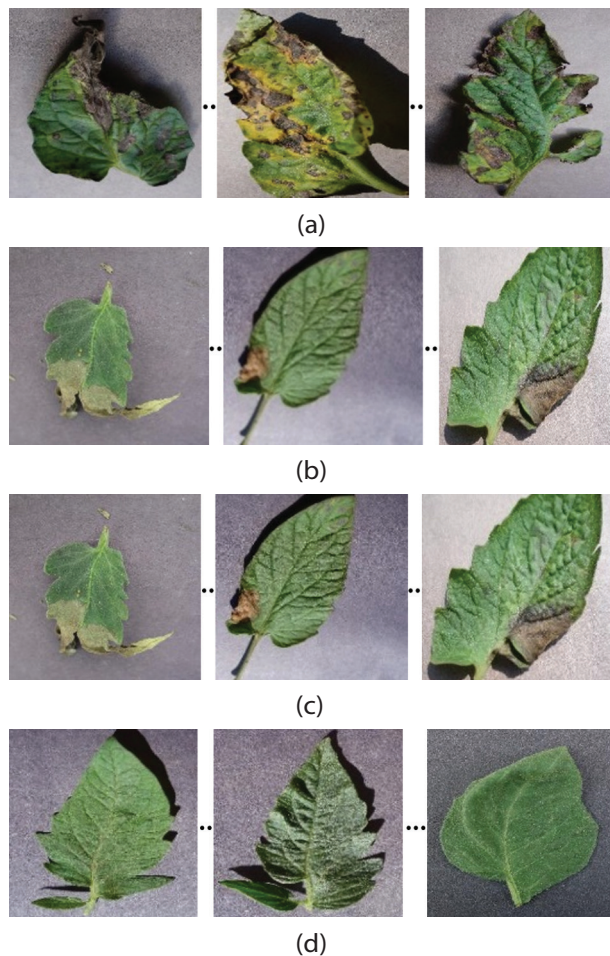


Fig. 3. (a) Early Blight Affected Tomato Leaf, (b) Early Blight Affected Tomato Leaf, (c) Leaf Mold Symptoms on Tomato Leaf, (d) Healthy Tomato Leaf

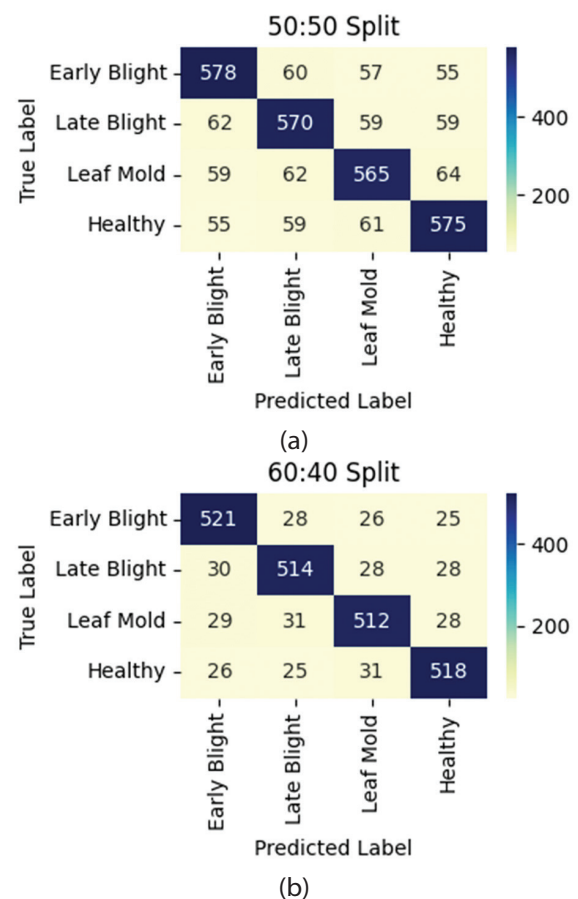
To address the class imbalance in the PlantVillage dataset, data augmentation techniques were applied to underrepresented classes, increasing each class to 1,500 images. This resulted in a balanced dataset that included equal representation of Early Blight, Late Blight, Leaf Mold, and Healthy Leaves. Balancing of the dataset increased the DenseNet-121 model's performance, enhancing its ability to generalize better and classify tomato leaf diseases. Data augmentation was employed to address class imbalance in the PlantVillage subset, specifically for the Early Blight and Leaf Mold classes, which initially consisted of 800 samples each. Original images per class were chosen, normalized by scaling pixel intensities to the [0, 1] range, and then augmented using a 90° rotation operation. The augmentation method was repeated iteratively to generate 700 additional samples per class, resulting in 1,500 images for each class. The augmentation preserved disease-specific characteristics while provid-

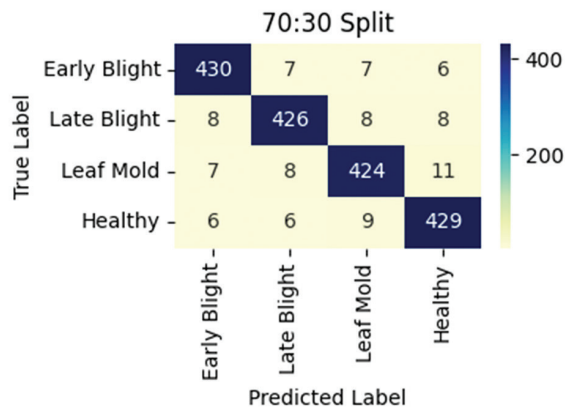
ing spatial variability, improving the resilience of the DenseNet-121 model, mitigating overfitting, and ensuring improved generalization across all tomato leaf disease classes. Fig. 4 shows the rotated images of tomato leaves used for performance evaluation.



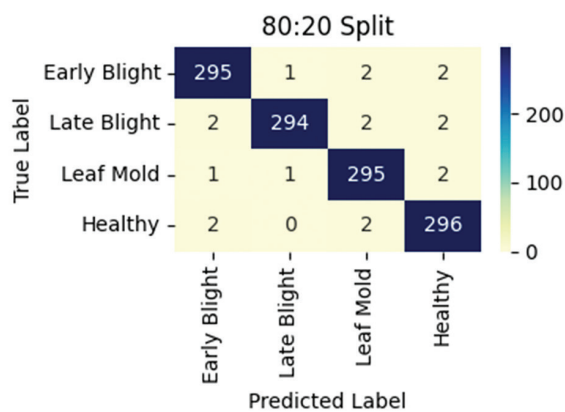
Fig. 4. Data Augmented Images of Tomato Leaves

A multi-class confusion matrix is essential to evaluate classification models with more than two classes. It provides comprehensive insights into the predictions for each class, detects misclassifications, and facilitates the computation of class-wise metrics such as precision and recall. This facilitates the identification of model deficiencies, enhances performance, and provides balanced evaluation. Fig. 5 shows DenseNet-121 with diverse training/testing splits, facilitating the evaluation of the efficiency of the model across varying data utilization, hence enhancing accuracy and generalization through the modification of training data ratios.





(c)



(d)

Fig. 5. (a) DenseNet-121 (50:50 Split), (b) DenseNet-121 (60:40 Split), (c) DenseNet-121 (70:30 Split), (d) DenseNet-121 (80:20 Split)

With an increase in training data quantity, the model demonstrates enhanced classification performance and a decrease in misclassifications. The 50:50 split (Fig. 5a) shows a considerably larger degree of class confusion, indicating less model confidence. The 70:30 and 80:20 splits (Fig. 5c and 5d) indicate robust diagonals with negligible off-diagonal values, indicating extremely precise forecasts for all classes. The 80:20 division archives near-perfect classification, with most misclassifications limited to only 1–2 instances per class. Overall, the confusion matrices show that model accuracy and robustness improve significantly with larger training datasets.

The performance of Equations 8 to 11 of the proposed DenseNet-121-based tomato leaf disease classification system was evaluated using traditional ML metrics to ensure the accuracy and reliability of the results.

$$Recall = \frac{TP}{TP + FN} \quad (9)$$

$$Precision = \frac{TP}{TP + FP} \quad (10)$$

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

$$F1\ Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (12)$$

The classification error is the ratio of incorrect predictions generated by a classification model. It can be derived from Eqn. 11 and is expressed as:

$$Classification\ Error = 1 - Accuracy \quad (13)$$

The performance metric assesses the performance of the model for classifying diseases in tomato leaves, shown in Tables 4 to 7 for different random splits.

Table 4. Class-wise evaluation metrics using 50:50 random split images

| Class | TP | FP | FN | TN | Accuracy (%) | Precision (%) | Recall (%) | F1-Score (%) |
|----------------|-----|-----|-----|------|--------------|---------------|------------|--------------|
| Early Blight | 578 | 176 | 172 | 2074 | 88.40 | 76.66 | 77.07 | 76.86 |
| Late Blight | 570 | 181 | 180 | 2069 | 87.97 | 75.90 | 76.00 | 75.94 |
| Leaf Mold | 565 | 177 | 185 | 2073 | 87.93 | 76.15 | 75.33 | 75.73 |
| Healthy Leaves | 575 | 178 | 175 | 2072 | 88.23 | 76.36 | 76.67 | 76.51 |
| Overall | | | | | 88.13 | 76.27 | 76.27 | 76.26 |

Table 5. Class-wise evaluation metrics using 60:40 random split images

| Class | TP | FP | FN | TN | Accuracy (%) | Precision (%) | Recall (%) | F1-Score (%) |
|----------------|-----|----|----|------|--------------|---------------|------------|--------------|
| Early Blight | 521 | 85 | 79 | 1715 | 93.17 | 85.97 | 86.83 | 86.40 |
| Late Blight | 514 | 84 | 86 | 1716 | 92.22 | 85.95 | 85.67 | 85.80 |
| Leaf Mold | 512 | 85 | 88 | 1715 | 92.79 | 85.76 | 85.33 | 85.54 |
| Healthy Leaves | 518 | 81 | 82 | 1719 | 93.21 | 86.48 | 86.33 | 86.40 |
| Overall | | | | | 93.02 | 86.04 | 86.04 | 86.04 |

Table 6. Class-wise evaluation metrics using 70:30 random split images

| Class | TP | FP | FN | TN | Accuracy (%) | Precision (%) | Recall (%) | F1-Score (%) |
|----------------|-----|----|----|------|--------------|---------------|------------|--------------|
| Early Blight | 430 | 21 | 20 | 1329 | 97.72 | 95.34 | 95.56 | 95.44 |
| Late Blight | 426 | 21 | 24 | 1329 | 97.50 | 95.30 | 94.67 | 94.98 |
| Leaf Mold | 424 | 24 | 26 | 1326 | 97.22 | 94.64 | 94.22 | 94.98 |
| Healthy Leaves | 429 | 25 | 21 | 1325 | 97.44 | 94.49 | 95.33 | 94.43 |
| Overall | | | | | 97.47 | 94.95 | 94.94 | 94.94 |

Table 7. Class-wise evaluation metrics using 80:20 random split images

| Class | TP | FP | FN | TN | Accuracy (%) | Precision (%) | Recall (%) | F1-Score (%) |
|----------------|-----|----|----|-----|--------------|---------------|------------|--------------|
| Early Blight | 295 | 5 | 5 | 895 | 99.17 | 98.33 | 98.33 | 98.33 |
| Late Blight | 294 | 3 | 6 | 897 | 99.25 | 98.99 | 98.00 | 98.49 |
| Leaf Mold | 295 | 6 | 5 | 894 | 99.08 | 98.01 | 98.33 | 98.16 |
| Healthy Leaves | 296 | 6 | 4 | 894 | 99.17 | 98.01 | 98.67 | 98.33 |
| Overall | | | | | 99.17 | 98.34 | 98.33 | 98.33 |

It can be seen from Table 4 to Table 7 that a distinct improvement is obtained with an increase in training data. The model with a 50:50 distribution achieves a moderate accuracy of 88.13% and balanced precision and recall (~76%). As the distribution changes to 60:40 and 70:30, both accuracy and F1-score significantly increase, reaching ~97% in the 70:30 configuration. At the 80:20 ratio, the model achieves optimal performance with nearly 99.17% accuracy and F1 scores exceeding 98% across all classes. The performance analysis indicates that enhanced training data improves model generalization, accuracy, and class-wise robustness in plant disease identification.

The performance of the DenseNet-121 model during training and validation across 20 epochs is shown in Fig. 6. The training accuracy rises consistently, reflecting enhanced model learning, whilst the validation accuracy exhibits minor fluctuations, suggesting the model's generalization capability. Provide performance indicators as mean \pm standard deviation across multiple runs to reflect variability and statistical significance. For instance, express accuracy as $99.17 \pm 0.12\%$. This methodology illustrates model stability and dependability, ensuring that enhancements above baseline models represent actual performance increases rather than chance variations.

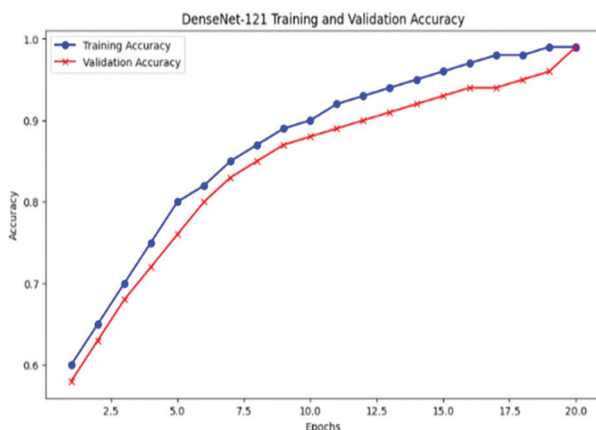


Fig. 6. DenseNet-121 Accuracy Progression

Fig. 7 depicts the decrease in training and validation loss across the epochs. The training loss decreases consistently, indicating successful learning, while the validation loss concurrently declines, albeit at a more gradual pace, implying potential overfitting or minor fluctuations in the validation data.

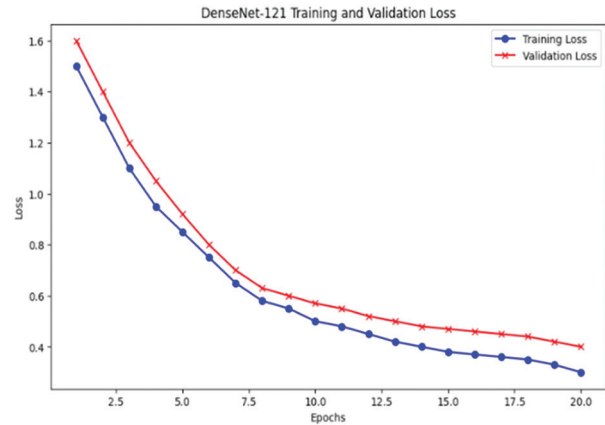


Fig. 7. DenseNet-121 Loss Curve

The DenseNet-121 model demonstrated a significant increase in training accuracy from 60% to 99% and validation accuracy from 58% to 98%. The training loss decreased from 1.5 to 0.3, while the validation loss reduced from 1.6 to 0.4, indicating successful optimization. Fig. 8 illustrates a comparative analysis of performance measures across DenseNet-121 versions, demonstrating incremental improvements in accuracy, precision, and F1-score from the baseline to the completely upgraded architecture.

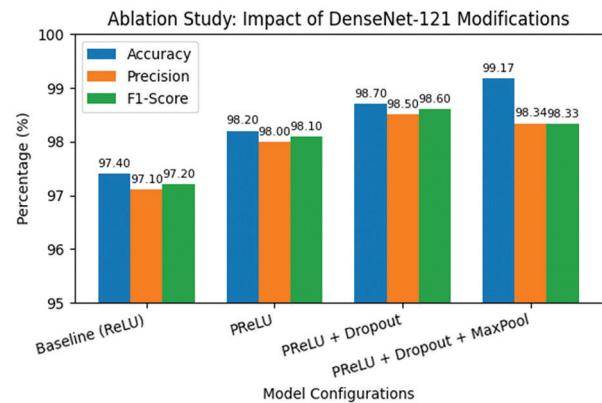


Fig. 8. Ablation Study of DenseNet-121 Modifications for Tomato Leaf Disease Classification

Fig. 9. shows the proposed DenseNet-121 improvement over other models, ViT, in terms of performance metrics to classify tomato leaf diseases into many classes.

Table 8 clearly shows that the proposed DenseNet-121-based system outperforms existing models, including traditional CNNs and sophisticated designs such as ViT and EfficientNetV2B7, across all critical performance criteria.

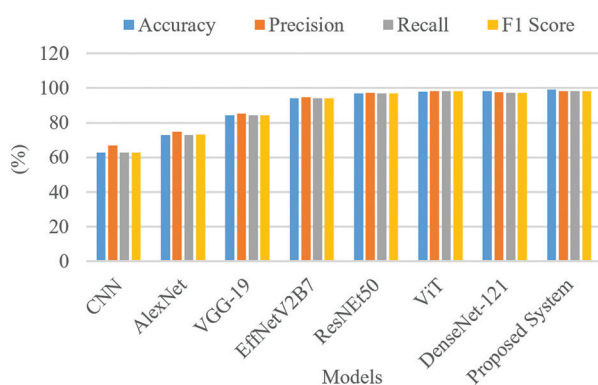


Fig. 9. Performance Comparison of Existing DL Models

Table 8. Comparison of Model Performance for Tomato Leaf Disease Classification

| Models | Accuracy | Precision | Recall | F1 Score |
|------------------------|----------|-----------|--------|----------|
| CNN [23] | 62.8 | 66.8 | 62.8 | 62.89 |
| AlexNet [23] | 72.86 | 74.89 | 72.86 | 73.33 |
| VGG-19 [23] | 84.29 | 85.28 | 84.29 | 84.34 |
| EffNetV2B7 [23] | 94.29 | 94.89 | 94.29 | 94.07 |
| ResNet50 [23] | 97.14 | 97.27 | 97.14 | 97.14 |
| ViT [24] | 98 | 98.3 | 98.33 | 98.2 |
| InceptionResNetV2 [21] | 95.67 | - | - | - |
| VGG16 [22] | 97 | - | - | - |
| DenseNet-121 | 98.4 | 97.7 | 97.3 | 97.4 |
| Proposed System | 99.17 | 98.34 | 98.33 | 98.33 |

The proposed DenseNet-121 improves other existing models in accuracy, achieving 99.17% and demonstrating the highest classification performance. While ResNet50 exhibits somewhat superior precision, DenseNet-121 achieves a more favourable equilibrium across all metrics. CNN and AlexNet demonstrate markedly inferior performance, rendering them inadequate for intricate multi-class classification in agriculture. DenseNet-121 offers enhanced feature reuse and gradient flow, improving its learning efficiency and robustness in the successful diagnosis of several tomato leaf diseases. The work indicates potential mobile or cloud implementation, although it requires empirical confirmation outside of controlled dataset tests. Future work should include prototype testing on mobile devices or cloud platforms to assess performance, latency, and usability in real-world agricultural settings.

4. CONCLUSIONS

The DenseNet-121 framework effectively identifies tomato leaf diseases by the extraction of deep hierarchical features, exhibiting excellent accuracy and robust performance. The research highlights practical applications for precision agriculture, such as incorporation into mobile apps, IoT monitoring systems, and cloud platforms for real-time disease diagnosis, early intervention, and minimized crop loss. Despite these advantages, limits are present. The PlantVillage data-

set is gathered in a laboratory setting, characterized by consistent backdrops and controlled lighting, perhaps restricting its applicability to field conditions that include noise, occlusions, and variable illumination. Training deep networks incurs significant computational expenses, which may limit accessibility for small-scale farmers or resource-constrained environments. Future studies should rectify these shortcomings by augmenting the dataset with field-acquired photos from other cultivars and environmental conditions, while also integrating multi-disease and co-infection situations. Methods, including domain adaptation, transfer learning, and lightweight designs, may enhance robustness and efficiency. The practical implementation and evaluation will further substantiate the approach, improving its scalability and relevance for sustainable, data-driven agriculture.

REFERENCES

- [1] L. M. Abouelmagd, M. Y. Shams, H. S. Marie, A. E. Hassanien, "An Optimized Capsule Neural Networks for Tomato Leaf Disease Classification", *EURASIP Journal on Image and Video Processing*, Vol. 2024, No. 1. 2024, p. 2.
- [2] E. Elfatimi, R. Eryigit, L. Elfatimi, "Deep Multi-Scale Convolutional Neural Networks for Automated Classification of Multi-Class Leaf Diseases in Tomatoes", *Neural Computing and Applications*, Vol. 36, No. 2, 2024, pp. 803-822.
- [3] N. B. Korade, M. B. Salunke, A. A. Bhosle, G. G. Asalkar, D. M. Joshi, A. S. Patil, S. M. Sangve, "Tomato Leaf Disease Detection with YOLOV8 Leaf Extraction, Resnet-50 Classification, and Gpt-3.5 for Treatment Recommendations", *International Research Journal of Multidisciplinary Scope*, Vol. 6, No. 1, 2025, pp. 879-891.
- [4] N. S. Kumar, J. Sony, A. Premkumar, J. J. Nair, "Transfer Learning-based Object Detection Models for Improved Diagnosis of Tomato Leaf Disease", *Procedia Computer Science*, Vol. 235, 2024, pp. 3025-3034.
- [5] S. Amutha, D. Nandhini, "Classification and Validation of Tomato Leaf Disease Using Deep Learning Techniques", *International Journal on Engineering Artificial Intelligence Management, Decision Support, and Policies*, Vol. 1, No. 1, 2024, pp. 41-60.
- [6] K. Lavanya, P. Maniar, T. Rokade, N. Jain, R. Lakshmanan, "Customized U-NET CNN Model for Tomato Leaf-Based Disease Classification", *Journal of the Balkan Tribological Association*, Vol. 30, No. 3, 2024, pp. 1-9.

- [7] V. Jayanthi, M. Kanchana, "Predictive Tomato Leaf Disease Detection and Classification: A Hybrid Deep Learning Framework", Proceedings of the International Conference on Communication and Computational Technologies, Jaipur, India, 7-8 January 2024, pp. 11-23.
- [8] J. Lenin, S. Muthumarakshmi, V. S. Prabhu, "PlantNet: A Deep Learning Model for Early Detection of Plant Diseases", International Journal of Advances in Signal and Image Sciences, Vol. 10, No. 2, 2024, pp. 37-47.
- [9] A. Debnath, J. M. Gil, Y. C. Byun, "Enhancing Tomato Leaf Disease Classification with Augmentation Techniques in Deep Learning", Proceedings of the International Conference on Frontier Computing, Singapore, 10-14 July 2024, pp. 45-50.
- [10] M. S. Asha, H. K. Yogish, "Deep Learning for Early Detection of Tomato Leaf Diseases: A ResNet-18 Approach for Sustainable Agriculture", International Journal of Advanced Computer Science & Applications, Vol. 15, No. 1, 2024, pp. 876-883.
- [11] A. Chakraborty, A. Chakraborty, A. Sobhan, A. Pathak, "Deep Learning for Precision Agriculture: Detecting Tomato Leaf Diseases with VGG-16 Model", International Journal of Computer Applications, Vol. 975, 2024, pp. 30-37.
- [12] M. N. Uddin et al. "A Novel Approach for Tomato Leaf Disease Detection and Classification Using Deep Learning", Proceedings of the Multimedia University Engineering Conference, Cyberjaya, Malaysia, 23-25 July 2024, pp. 1-6.
- [13] J. Lee, N. Kim, "Utilizing Minimal Label Data for Tomato Leaf Disease Classification: An Approach through Recursive Learning Based on YOLOv8", The Journal of Big Data, Vol. 9, No. 1, 2024, pp. 61-73.
- [14] S. Srinivasan, S. LK, T. Alavanthar, C. Srinivasan, S. Murugan, S. Sujatha, "IoT-Enabled Horticultural Lighting for Optimizing Plant Growth and Agriculture Operations", Proceedings of the 2nd International Conference on Networking and Communications, Chennai, India, 2-4 April 2024, pp. 1-7.
- [15] R. Rajamohan B. C. Latha, "An Optimized YOLO v5 Model for Tomato Leaf Disease Classification with Field Dataset", Engineering, Technology & Applied Science Research, Vol. 13, No. 6, 2023, pp. 12033-12038.
- [16] A. Sreedevi, C. Manike, "Development of Weighted Ensemble Transfer Learning for Tomato Leaf Disease Classification Solving Low-Resolution Problems", The Imaging Science Journal, Vol. 71, No. 2, 2023, pp. 161-187.
- [17] P. Madhiraju, R. Sampathrajan, M. Pattnaik, P. Rajalakshmi, K. Dhivya, S. Murugan, "IoT-Driven Defect Detection System for Plastic Recycling Plants Based on Convolutional Neural Networks", Proceedings of the 2nd International Conference on Self-Sustainable Artificial Intelligence Systems, Erode, India, 23-25 October 2024, pp. 1075-1080.
- [18] S. Sagar, J. Singh, "An Experimental Study of Tomato Viral Leaf Diseases Detection Using Machine Learning Classification Techniques", Bulletin of Electrical Engineering and Informatics, Vol. 12, No. 1, 2023, pp. 451-461.
- [19] P. F. I. R. Zer, F. N. Tambunan, R. Rosnelly, W. Wanayumini, "Comparison of Tomato Leaf Disease Classification Accuracy Using Support Vector Machine and K-Nearest Neighbor Methods", Synchronous: Journal and Research of Informatics Engineering, Vol. 7, No. 2, 2023, pp. 939-947.
- [20] M. Altalak, M. A. Uddin, A. Alajmi, A. Rizg, "A Hybrid Approach for the Detection and Classification of Tomato Leaf Diseases", Applied Sciences, Vol. 12, No. 16, 2022, p. 8182.
- [21] N. Krishnamoorthy, L. N. Prasad, C. P. Kumar, B. Subedi, H. B. Abraha, V. E. Sathishkumar, "Rice Leaf Diseases Prediction Using Deep Neural Networks with Transfer Learning", Environmental Research, Vol. 198, 2021, p. 111275.
- [22] M. Subramanian, L. V. N. Prasad, B. Janakiramaiah, A. M. Babu, V. E. Satishkumar, "Hyperparameter Optimization for Transfer Learning of VGG16 for Disease Identification in Corn Leaves Using Bayesian Optimization", Big Data, Vol. 10, No. 3, 2022, pp. 215-229.
- [23] S. R. Dubey, S. K. Singh, B. B. Chaudhuri, "Activation functions in deep learning: A comprehensive survey and benchmark", Neurocomputing, Vol. 503, 2022, pp. 92-108.
- [24] S. A. Alshammari, "TLDViT: A Vision Transformer Model for Tomato Leaf Disease Classification", International Journal of Advanced Computer Science & Applications, Vol. 15, No. 12, 2024, pp. 841-848.