INTERNATIONAL JOURNAL OF LATEST TECHNOLOGY IN ENGINEERING, MANAGEMENT & APPLIED SCIENCE (IJLTEMAS)

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XIII, Issue XII, December 2024

Smart Farming in Bangladesh: Mobile Application for Tomato Leaf Disease Detection Using a Hybrid VGG16-CNN Model

Mrittika Mahbub[1], Md. Habib Ehsanul Hoque[2], Mst. Rehena Khatun[3]

[1]Lecturer, Dept. of Computer Science and Engineering, Pundra University of Science & Technology, Bangladesh

[2,3]Assistant Professor, Dept. of Computer Science and Engineering, Pundra University of Science & Technology, Bangladesh

DOI : https://doi.org/10.51583/IJLTEMAS.2024.131220

Received: 30 December 2024; Accepted: 03 January 2025; Published: 11 January 2025

Abstract: Tomato cultivation (Solanum lycopersicum L.) is highly significant due to its considerable economic value, high consumer demand, and critical role in supporting the livelihoods of farmers in Bangladesh. However, the majority of Bangladeshi farmers rely on traditional, manual methods for detecting tomato leaf diseases, relying on visual inspection and personal experience. Limited resources and a lack of awareness about advanced technologies further hinder the adoption of efficient disease detection methods. Computer vision, a cutting-edge technology, enables the automated identification and classification of tomato leaf diseases, holding significant promise for improving agricultural productivity and farmers' livelihoods. This study focuses on developing a robust disease detection framework involving image acquisition, preprocessing, and feature extraction using a VGG16-CNN hybrid model, integrated with smartphone applications for real-time detection. To address the limitations faced by local farmers and plant enthusiasts unfamiliar with such technology, a diverse dataset of approximately 16,824 images was created, comprising field images and online sources. The proposed method leverages VGG16 for feature extraction, achieving enhanced performance through additional fine-tuned layers that form a hybrid model. This approach delivers an accuracy of 98%, with an F1 score of 98%. These findings highlight the potential of the proposed system to significantly mitigate the impacts of tomato leaf diseases, thereby improving tomato cultivation and production outcomes.

Keywords: Smart Farming, Tomato Leaf Diseases, Hybrid Model, Early Detection, Classification, Mobile Application.

I. Introduction

Tomatoes are a key vegetable in many countries and rank as the second most widely cultivated crop worldwide, following potatoes. They are used extensively to make sauces, ketchup, chutneys, juices, and pastes and can be eaten raw, ripe, cooked, or in salads. In addition to their colour, flavour, and appearance, tomatoes are prized for their vitamin C content. Tomato cultivation is a vital agricultural activityin Bangladesh, spanning 40,000 hectares and yielding about 255,000 metric tons annually, primarily during the Rabi season (November to February). [1] Key producing districts include Bagerhat, Satkhira, Khulna, Rajshahi, Natore, Chapainawabganj, and parts of Dhaka, Dinajpur, Pabna, Narsingdi, and Mymensingh. [2] As a cash crop, tomatoes provide significant income, support rural livelihoods, and cater to local markets and processing industries, with ongoing efforts to boost yield and extend the growing season. Bangladesh's agrarian economy relies heavily on agriculture for employ-ment and food security, supporting a significant portion of the population. Tomatoes, a staple vegetable, are regarded as "poor man's apple" due to their versatility, affordability, and nutritional value. Enhancing tomato production through improved practices and technological interventions is crucial for meeting growing demand. Over 50 tomato varieties, including hybrids created by BARI and BINA for off-season growing, are cultivated year-round in Bangladesh. In order to ensure consistent output and profitability, popular kinds are categorised as early, main season, late winter, and year-round types. [3] Both farmers and tree aficionados in our nation love tomatoes. Many women in rural areas use available spaces, such gardens and courtyards, to grow tomatoes around their homes. In addition to increasing their household's food supply, this approach promotes sustainability and pride. Similar to this, rooftop gardening has gained popularity in metropolitan places with limited space, and tomatoes are a popular choice because of their ease of growing and culinary flexibility. Tomato plants are highly susceptible to various diseases, particularly those affecting their leaves, which often hinder production and cause significant economic losses. In many cases, the lack of proper disease recognition leads to mismanagement, resulting in further damage to healthy plants. Since tomato plants are sensitive, incorrect diagnoses or the use of inappropriate pesticides can exacerbate the problem. Most farmers, rural women, and rooftop gardeners in our country lack awareness of modern techniques and tools to identify and manage these diseases effectively. Numerous leaf diseases that negatively affect crop health and productivity are serious obstacles to tomato farming in Bangladesh. The main illnesses are Mosaic Virus, Early Blight, Late Blight, Bacterial Wilt, Tomato White Mold, and Tomato Yellow Leaf Curl Virus. [4]Tomato leaf diseases are ideal for image-based analysis because they frequently exhibit obvious symptoms like discoloration and patches. With AI-driven image-based solutions [5] that offer accurate, economical, and environmentally friendly ways to diagnose diseases utilizing computer vision, machine learning, and deep learning techniques [6], digital technology advancements have completely changed agriculture. These contemporary

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XIII, Issue XII, December 2024

Fig. 1. An overview of the targeted procedure

approaches outperform conventional ones by providing excellent accuracy even under challenging circumstances. User friendly smartphones and online applications have made these tools more accessible, enabling farmers and plant enthusiasts to diagnose diseases simply by capturing images of affected leaves. Food security, sustainable farming, and crop protection are all greatly improved by this innovation. Figure 1 illustrates the proposed system for real-time tomato leaf disease detection and classification, optimized for mobile visualization. This paper introduces a mobile application for tomato leaf disease detection powered by a hybrid VGG16-CNN model. The system uses VGG16 [7] for feature extraction and custom CNN layers [8] for precise classification, ensuring high accuracy and efficiency. Users can capture leaf images via smartphones for real-time disease diagnosis, enabling prompt action. The model identifies healthy and diseased leaves, pinpoints specific diseases, and suggests treatments, providing practical insights for farmers. This approach is validated on a robust dataset and offers a scalable solution for identifying diverse plant diseases, advancing sustainable farming practices.

II. Literature Review

Tomato crops are highly susceptible to diseases that can significantly affect their growth and yield. Early detection of tomato leaf diseases is essential for timely intervention and preventing pathogen spread. Traditional manual inspection methods are labor-intensive and subjective, while recent advances in image processing, machine learning, and mobile technologies have enabled automated, efficient, and scalable disease detection systems. This review focuses on deep learning techniques, particularly Convolutional Neural Networks (CNNs) and VGG16, and their integration with mobile applications for real-time tomato leaf disease identification. Ferentinos et al. (2018) developed a CNN model for detecting diseases across multiple crops [6], including tomatoes, by analyzing a large dataset of labeled leaf images. The model demonstrated exceptional accuracy in disease classification by recognizing unique characteristics specific to each disease. Similarly, Moosavi et al. Similarly, S.AM et al.(2024) [9] present an adaptive ensemble model combining fine-tuned VGG-16 and NASNet architectures with exponential moving average fusion and enhanced gradient optimization, achieving 98.7% accuracy in tomato leaf disease classification. Although highly accurate, the model's complexity and dataset diversity may limit its scalability and real-world generalizability. Ramcharan et al. (2020) extended CNN's application to real-time disease detection by integrating the model into a smartphone app [10]. The app allowed farmers to capture and analyze leaf images, providing immediate diagnostic feedback. Similarly, Ranganajan et al. (2018) [11] achieved high precision in classifying tomato leaf diseases by utilizing optimized pre-trained AlexNet and VGG16 models on the PlantVillage dataset. Limitations include dataset constraints under controlled conditions, potential generalization issues in real-world scenarios, and high computational requirements for fine-tuning.

Transfer learning has further enhanced the performance of deep learning models in plant disease detection. Iffaty et al.(2023) [12] applied transfer learning using the VGG16 model for classifying tomato leaf diseases, leveraging its pre-trained architecture to achieve high accuracy on a limited dataset. The study demonstrated the effectiveness of transfer learning in reducing computational costs and training data requirements while maintaining robust disease classification performance.

Several mobile applications have successfully integrated deep learning models for practical use. For example, Plant Village, introduced by Hughes et al. (2015), employs CNN-based models for detecting plant diseases, including those affecting tomatoes [13]. The app provides a simple interface for farmers to upload leaf images and receive rapid disease predictions. In the same way, Debnath et al. (2023) [14] introduced a smartphone-based system for detecting tomato leaf diseases using the EfficientNetV2B2 model, which achieved high ac-curacy. The study emphasizes explainability in AI, ensuring farmers can interpret the model's predictions effectively. Mohanty et al. (2016) [5] also demonstrated the utility of CNNs for plant disease detection, highlighting the scalability of auto-mated diagnostic systems. However, their models were trained on a relatively small dataset, potentially limiting performance across diverse environments and plant species. In addition, high computational requirements presented challenges for the deployment of mobile devices. Recent research has focused on enhancing tomato leaf disease detection in Bangladesh, utilizing advanced deep-learning models. A. Sobur and colleagues recently (2024) [15] conducted a comparative analysis of advanced deep learning models for detecting tomato leaf diseases across various climatic conditions. The authors pro-pose a novel hybrid approach, combining multiple deep learning techniques, to improve detection accuracy across varied environments. The study highlights the effectiveness of these models in handling diverse challenges posed by the variation in climate, offering insights into the optimal application of AI for agricultural disease detection. Meanwhile, Chowdhury et al. (2021) [15] suggested that adapting the model with region-specific data could enhance its effectiveness in Bangladesh, tackling local agricultural challenges.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XIII, Issue XII, December 2024

III. Proposed Methodology

We proposed a hybrid VGG16-CNN model with a mobile app interaction for the detection of tomato leaf diseases in the context of smart farming in Bangladesh. The methodology is divided into several key steps to implement the proposed strategy, as shown in Fig. 2. The following subsections provide a detailed explanation of each step.

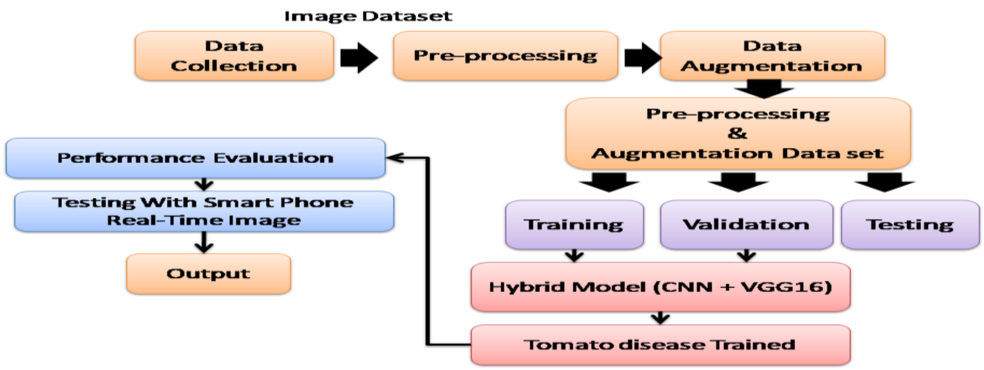


Fig. 2. Abstract Representation of the Proposed Methodology.

A. Data Collection

For this work, a dataset of 16,800 tomato leaf images, including healthy and diseased leaves, was compiled from publicly available sources and local farming communities in Bangladesh. Images were six categorized based on common tomato leaf diseases such as bacterial spot, early blight, late blight, yellow leaf curl virus, mosaic virus, white mold, and last of all healthy leaf. The images related to tomato diseases have been gathered, focusing on the most prevalent tomato diseases in Bangladesh. To ensure variety, images were taken from multiple angles at different times of day and across varied environments and climates. To provide a diverse dataset, several images of both healthy and sick tomato leaves are collected from internet resources like Kaggle.com [16], Mendeley data set [17] etc. Table 1 presents six diseases, along with images of healthy leaves, their contributing factors, symptoms, and the quantity collected. The table highlights that common diseases affecting tomato plants in Bangladesh include yellow leaf curl, bacterial spot, and blight, while white mold disease occurs less frequently.

B. Data Augmentation

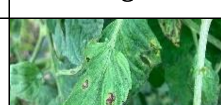
Transformations were applied to the original images in the dataset to generate a larger and more diverse collection. This approach helps prevent overfitting and improves the model's ability to generalize. Techniques such as rotation, flipping, scaling, and noise addition were used to expand the dataset, reducing the likelihood of the model becoming too specific to the training data and enhancing its robustness. The following augmentation parameters were utilized:

Rotation: Each image was randomly rotated within a range of $\pm 30^{\circ}$. Mathematically, rotation can be represented as:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

where $\theta \epsilon [-30^{\circ}, 30^{\circ}]$

Table 1: Details of Bangladeshi Tomato Leaf Diseases

Related Information of Tomato Leaf Disease				
Disease	**Contributing Factors**	**Images**	**Symptoms**	**Quantity**
Bacterial Spot	• High humidity • Poor airflow		• Leaf spots • Yellow halos	2721
Mosaic Virus	• Infection • Contamination		• Mosaic patterns • Leaf distortion	2405
Late Blight	• Moisture • Infected seeds		• Dark green lesions • Fungal growth	2319

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XIII, Issue XII, December 2024

Early Blight	• Microclimate • Inoculum		• Dark brown circular spots • Premature leaf drop	2474
Yellow Leaf Curl Virus	• Whitefly transmission • Pest management gap		• Upward curling and rolling • Yellowing in margin of young leaves.	3211
White mold	• Water-soaked lesions • Stem collapse		• White, Cottony Growth • Water-Soaked Lesions	1282
Additionally the healthy leaf images				2412
Total Images				**16824**

Width and Height Shift:
Images were shifted horizontally and vertically by up to 20% of their dimensions. The translation matrix is given by:

$$T= \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix}$$

where $\in$[−0.2 × width, 0.2 × width] and [−0.2 ×height, 0.2 × height], respectively.

Shear:
Images were sheared by up to 20%. The shear transformation is represented as:

$$S = \begin{bmatrix} 1 & \tan \emptyset & 0 \\ \tan \varphi & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

where $\emptyset, \varphi \epsilon [-20°, 20°.$

Zoom:
Images were scaled by a zoom factor within a range of 20%.

The zoom matrix is expressed as:

where. $z \, \epsilon [0.8,1.2]$

$$Z = \begin{bmatrix} z & 0 & 0 \\ 0 & z & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Horizontal Flip:

Images were flipped horizontally with a probability of 50%.

This operation mirrors the image along the vertical axis:

Horizontal Flip:

Fill Mode: $x' = -x, y' = y.$

Any empty pixels created during transformations were filled using nearest neighbor interpolation to ensure a seamless appearance. These augmentations enhanced the dataset by adding variability, helping the model improve its ability to generalize to new, unseen data.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XIII, Issue XII, December 2024

C. Normalization

Normalization is a crucial preprocessing step in deep learn-ing that rescales image pixel values to a standardized range, typically [0, 1]. This process ensures that the input values are within a smaller, consistent range, which can significantly improve the model's convergence rate during training and enhance numerical stability.

The normalization formula for a pixel value x is given by:

where:

$$x_{normalized} = \frac{x}{255}$$

- x is the original pixel intensity value (ranging from *0* to *255* for 8-bit images).
- x normalized is the resulting value after normalization, confined to the range [0, 1].

By normalizing the pixel values, the deep learning model can process the data more efficiently, reducing the risk of issues caused by varying scales in the input data.

D. VGG16 and CNN Architectural Concept

1) VGG16 Architecture: One well-known convolutional neural network (CNN) architecture that has had a big impact on deep learning, especially in image recognition applications, is **VGG16**. Three fully connected layers and thirteen convolutional layers make up its sixteen weight layers. Figure 3 illustrates the architectural overview of the 16-Layer VGG16 Model and its characteristics include:

• **Deep Architecture:** To learn more complex patterns in the data, VGG16 uses tiny receptive fields (3×3 fil-ters) [18] in each convolutional layer.

• **Layer Stacking:** VGG16 extracts increasingly higher-level features by stacking numerous convolutional layers, beginning with edges and textures and progressing to intricate patterns.

• **Pooling Layers:** By reducing spatial dimensions, max pooling layers maintain important features and guarantee computational efficiency.

• **Fully Connected Layers:** These layers combine the learnt characteristics for classification tasks at the con-clusion of the architecture.

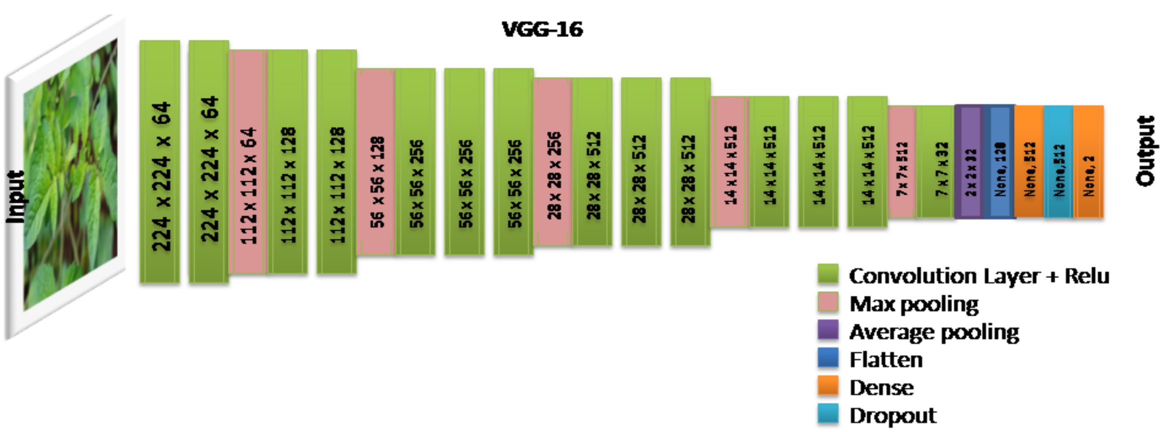


Fig. 3. Architectural Overview of the 16-Layer VGG16 Model

2) Convolutional Neural Network (CNN): CNN stands for **Convolutional Neural Network**. CNNs form the foundation of contemporary computer vision and image processing applications. They consist of layers [19] specifically designed to handle spatial data, such as images. Key components include:

• **Convolutional Layers:** These layers apply filters to input images to capture spatial hierarchies and feature patterns, such as edges, textures, and shapes.

• **Activation Functions:** Non-linear functions, such as ReLU (Rectified Linear Unit), introduce non-linearity, enabling the network to model complex relationships.

• **Pooling Layers:** Techniques like max pooling reduce the spatial dimensions of feature maps, enhancing computational efficiency and promoting generalization.

• **Fully Connected Layers:** These layers finalize the classification process by transforming high-dimensional feature maps into class probabilities.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XIII, Issue XII, December 2024

E. Tomato Leaf Disease Detection with VGG16-CNN Hybrid

The Hybrid VGG16-CNN model combines the pre-trained VGG16 network with custom layers [20], leveraging transfer learning to optimize the architecture for domain-specific features in the classification of seven tomato leaf disease categories.

1) Pre-trained VGG16 Backbone: The model incorporates the VGG16 architecture pre-trained on the ImageNet dataset, where the top classification layers are removed to use the network as a feature extractor. The output of the VGG16 convolutional base can be represented as:

$$F = f\,\mathrm{VGG16}(I)$$

where:

• I represents the input image.

• F is the feature map generated by the VGG16 convolutional layers.

• fVGG16 denotes the VGG16 feature extraction function.

2) Custom Layers: Custom layers are added to refine the extracted features and enhance the model's capability for domain-specific tasks:

• **Batch Normalization:** Normalizes the activations to improve training stability and convergence speed.

• **Global Average Pooling:** Reduces the spatial dimensions of the feature maps by computing the average of each feature map:

where:

$$g_i = \frac{1}{h \times w} \sum_{h=1}^{H} \sum_{w=1}^{W} F_{i,h,w}$$

– gi is the global average of the i-th feature map.

– H and W are the height and width of the feature map.

– Fi,h,w is the activation of the i-th feature map at position (h, w).

• **Dense Layers:** Fully connected layers with ReLU activation refine the extracted features:

$$z_j = \max\,(0, \sum_i w_{ij} g_i + b_j)$$

where:

– zj is the output of the j-th dense neuron.

– wij and bj are the weights and biases of the neuron.

• **Dropout:** Randomly drops a fraction of the neurons during training to prevent overfitting.

3) Output Layer: The final layer is a dense layer with a softmax activation function for multi-class classification. The probability distribution over the C classes is computed as:

$$p(c_k|x) = \frac{exp(z_k)}{\sum_{j=1}^{C} exp(z_j)}$$

• z_k is the logit for class k.

• $p(ck|x)$ is the predicted probability of class k given input x.

• $C = 7$, the total number of classes (tomato leaf disease categories).

F. Deployment as a Mobile Application

1) Model Export: The trained hybrid VGG16-CNN model was exported in a lightweight format (e.g., TensorFlow Lite) optimized for mobile devices. Techniques like quantization were applied to reduce model size while maintaining accuracy.

2) Mobile Application Development: A user-friendly mo-bile app was developed with features such as real-time image capture, disease detection, and actionable recommendations for disease management. The app supports offline functionality for remote use and provides results with confidence scores.

3) Localization*:* The app is tailored for Bangladeshi farmers with a Bengali interface and local farming terminologies. Region-specific recommendations are provided, considering local agricultural practices and resources.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XIII, Issue XII, December 2024

4) Benefits: This app empowers farmers to identify and manage tomato diseases quickly, enhancing yields and reducing losses. Future updates will include additional languages, predictive analytics, and cloud-based model improvements.

IV. Result and Discussion

We used a virtual computer with a P100 GPU on Kaggle to carry out our research. Another tool used to evaluate the system's performance has an Intel Core i5 9th Gen CPU and 8 GB of RAM. A 50MP camera with a live feed of 30 frames per second was used to capture images. In our experiments, we created datasets, tested the Hybrid model, evaluated its accuracy, and integrated it into mobile apps for real-time feedback.

A. Dataset Preparation

The dataset comprised a total of 16,824 images, with approximately 75% sourced online and the remaining 25% captured in the field. Of these, 13,757 images (80% of the dataset) were designated for training, encompassing seven classes of Bangladeshi tomato leaf diseases. Figure 4 illustrates examples of various types of diseases in the dataset. For testing, the remaining 3,067 images (20% of the dataset) were utilized. The final column of Table I provides the distribution of images across each class.

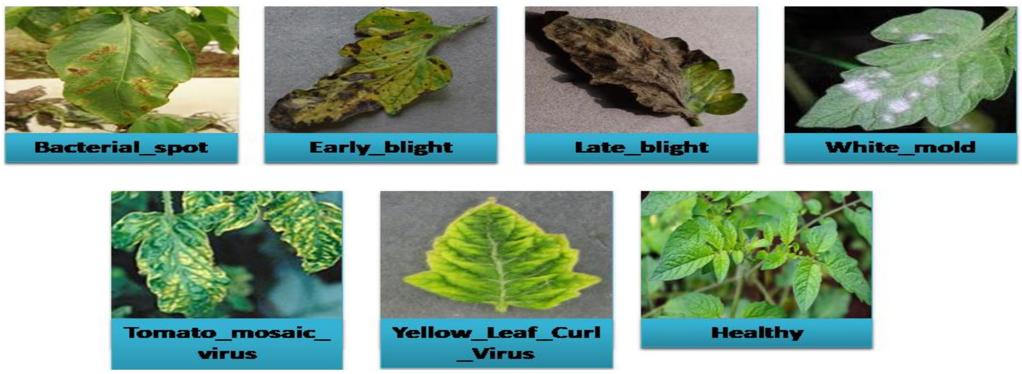


Fig. 4. A sample from our dataset showcasing Bangladeshi tomato leaf diseases

B. Model Training and Validation

1) Training: The hybrid model was trained using the Adam optimizer, which adapts the learning rate for each parameter dynamically. The learning rate for training was set to $\alpha = 0.0001$, and the categorical cross-entropy loss function was used to optimize the model. The loss function is defined as:

$$L = -\frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{C} y_{i,k}\log\left(p_{i,k}\right),$$

where:

- N is the number of samples in the training set.
- C is the total number of classes.
- $y_{i,k}$ is a binary indicator (1 if the true class is k, 0 otherwise) for the i-th sample.
- $p_{i,k}$ is the predicted probability for class k for the i-th sample, calculated using the softmax function:

$$p_{i,k} = \frac{exp\left(z_{i,k}\right)}{\sum_{j=1 exp}^{C}\left(z_{i,j}\right)}.$$

To prevent overfitting, early stopping was employed. This method monitors validation loss and halts training when per-formance stops improving. To obtain the best fit and avoid both overfitting and underfitting, the model was trained over 50 epochs.

2) Fine-Tuning: After the initial training, fine-tuning was performed by partially unfreezing the base VGG16 layers. Specifically:

- The first 15 layers of the VGG16 backbone were kept frozen to retain learned features from the ImageNet dataset.
- The remaining layers were unfrozen and trained with a reduced learning rate of $\alpha = 10^{-5}$ to refine domainspecific feature representations.

C. Model Accuracy Measurement

The model was evaluated using the following metrics:

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XIII, Issue XII, December 2024

Accuracy: Accuracy measures the proportion of correctly classified samples and is defined as:

$$Accuracy = \frac{Number\ of\ Correct\ Predictions}{Total\ Number\ of\ Predictions} = \frac{\sum_{i=1}^{N} \mathbb{1}\left(\hat{y}_i = y_i\right)}{N}$$

where:

- $\hat{y}_i$ is the predicted label for the *i*-th sample.
- *yi* is the true label for the *i*-th sample.
- $\mathbb{1}$ is the indicator function, which equals 1 if $\hat{y}_i = yi$, and 0 otherwise.

Precision, Recall, and F1-Score: These metrics are particularly useful for imbalanced datasets:

$$\text{Precision} = \frac{\text{True Positives(TP)}}{\text{True Positives (TP)} + \text{False Positives (FP)}},$$

$$\text{Recall} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Negatives (FN)}},$$

$$\text{F1} - \text{Score} = 2.\frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}.$$

Confusion Matrix: A confusion matrix was used to visualize classification performance across all classes. Each element *Mi,j* of the matrix represents the number of samples of class *i* predicted as class *j*. It provides a clear view of true positives, false positives, and false negatives.

$$M = \begin{bmatrix} M_{1,1} & M_{1,2} & \cdots & M_{1,C} \\ M_{2,1} & M_{2,2} & \cdots & M_{2,C} \\ \vdots & \vdots & \ddots & \vdots \\ M_{C,1} & M_{C,2} & \cdots & M_{C,C} \end{bmatrix}$$

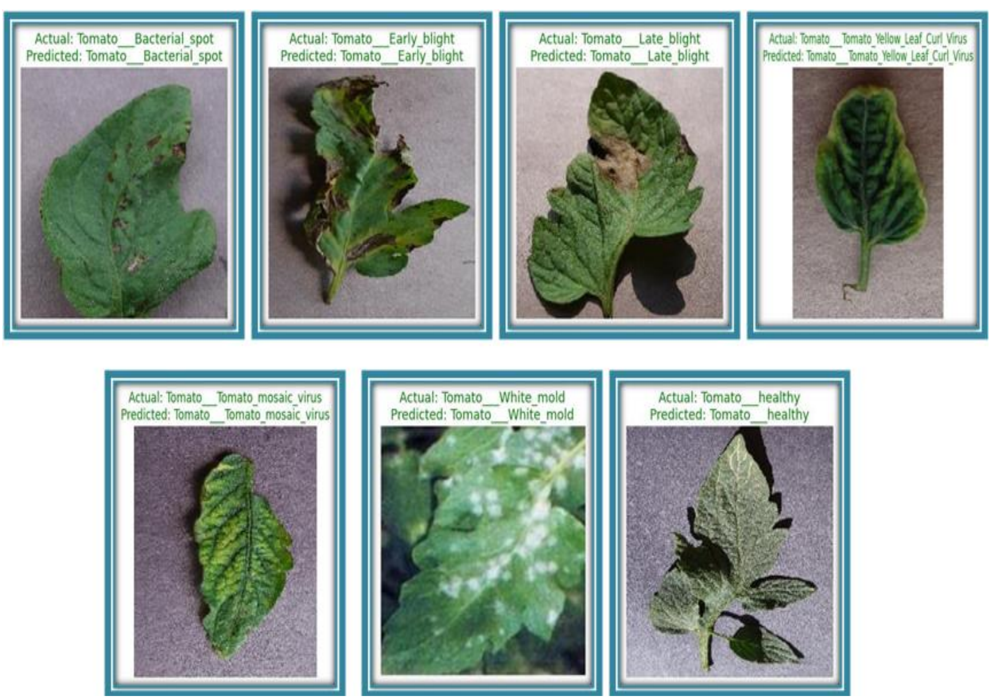


Fig. 5. Our test dataset's sample detection outcomes

INTERNATIONAL JOURNAL OF LATEST TECHNOLOGY IN ENGINEERING, MANAGEMENT & APPLIED SCIENCE (IJLTEMAS)

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XIII, Issue XII, December 2024

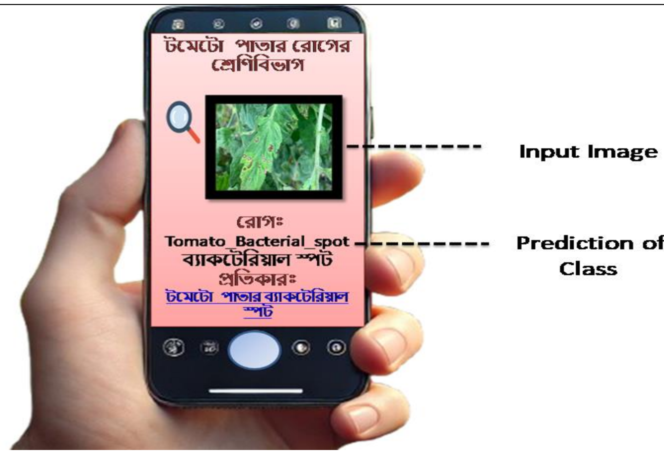


Fig. 6. Final Outcomes as Presented to the End-User.

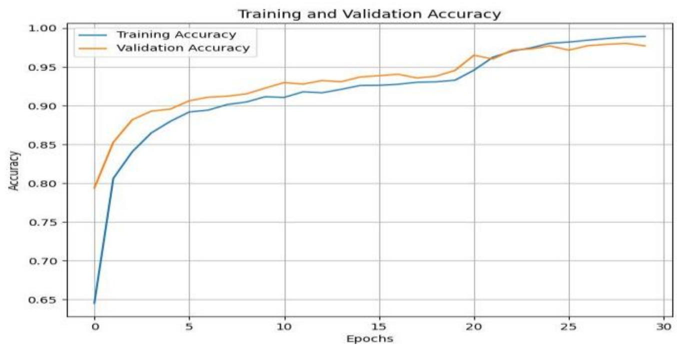


Fig. 7:Training and Validation Accuracy for Tomato Leaf Disease Detection

Fig. 8: Training and Validation Loss Analysis

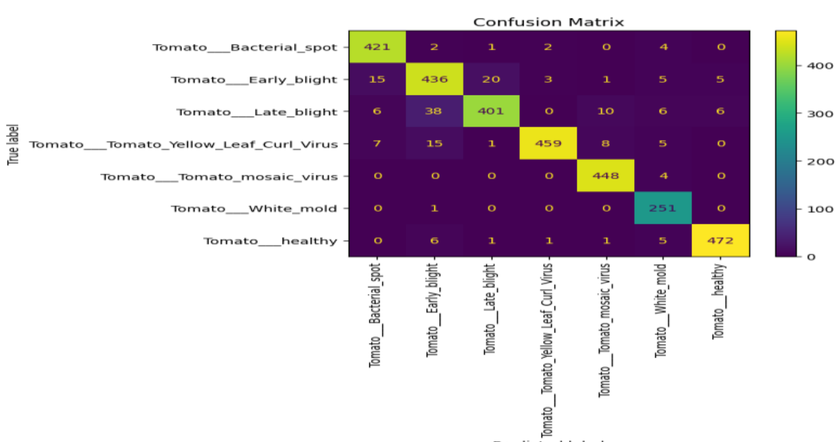


Fig. 9. Confusion Matrix for Tomato Leaf Disease Classification

INTERNATIONAL JOURNAL OF LATEST TECHNOLOGY IN ENGINEERING, MANAGEMENT & APPLIED SCIENCE (IJLTEMAS)

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XIII, Issue XII, December 2024

Table 2 : Classification report for tomato leaf disease detection.

Class	Precision	Recall	F1-Score	Support
Bacterial_spot	0.99	0.97	0.98	430
Early_blight	0.97	0.97	0.97	485
Late_blight	0.99	0.96	0.97	467
Yellow_Leaf_Curl	0.99	0.99	0.99	495
Virus Mosaic_	1.00	0.99	1.00	452
Virus	0.90	1.00	0.95	252
White_Mold	0.99	0.99	0.99	486
Healthy				
Accuracy	0.98	0.98	0.98	3067
Macro Average	0.98	0.98	0.98	3067
Weighted			0.98	3067
Average				

Explanation of Metrics

• **Precision**: Represents the proportion of true positive predictions among all positive predictions for each class. High precision indicates that the model is good at avoiding false positives.

• **Recall**: Measures the proportion of actual positive cases that were correctly identified by the model. High recall ensures that the model detects most of the positive cases.

• **F1-Score**: The harmonic mean of precision and recall, providing a balanced measure of the model's performance. A higher F1-score indicates a better balance between precision and recall.

• **Support**: The number of true instances for each class in the dataset. This helps in understanding how well the model performs on classes with different amounts of data.

Overall Performance

The overall accuracy of the model is 98%, with a balanced performance across all disease categories. The weighted and macro averages also show consistent model performance across the classes. Figures 5 and 6 illustrate the sample detection outcomes from our test dataset and the final outcomes as presented to the end-user, respectively. Figure 7 shows the training and validation accuracy for tomato leaf disease detection, Figure 8 highlights the analysis of training and validation loss, and Figure 9 presents the confusion matrix for the model's performance. The model demonstrates excellent performance, particularly for critical diseases such as Tomato Mosaic Virus and Tomato Yellow Leaf Curl Virus, achieving near-perfect F1-scores in these cases. The high recall values indicate that the model is very effective at detecting positive instances, while the precision values show it avoids false positives.

V. Conclusion

This research demonstrates the effectiveness of the proposed hybrid VGG16-CNN model for detecting and classifying tomato leaf diseases with high accuracy. The model achieved an overall accuracy of 98%, with precision, recall, and F1 scores consistently above 95% across most categories. This performance highlights the model's ability to accurately identify and differentiate complex disease patterns, such as Tomato Yellow Leaf Curl Virus and Tomato Mosaic Virus, which are often challenging due to their visual similarities. The primary goal is to develop a reliable framework to assist both farmers and plant enthusiasts in early disease diagnosis and yield optimization. Ongoing efforts include integrating the disease detection model with predictive analytics to forecast outbreaks based on environmental factors like humidity, temperature, and soil quality. Future plans also involve expanding the model to include leaf diseases of other crops, creating a generalized system for multi-crop disease detection. Additionally, the model will be integrated into mobile and IoT devices for real-time field detection, ensuring immediate access to results for farmers. Incorporating explainable AI techniques will provide insights into the model's decision-making process, helping users understand the reasoning behind predictions. By addressing these areas, the model can further enhance its practical applications, offering a comprehensive, scalable solution for precision agriculture and sustainable farming practices.

References

1. IndexBox, "Bangladesh's tomato market report 2024 - prices, size, fore-cast, and companies," https://www.indexbox.io/store/bangladesh-tomato-market-report-analysis-and-forecast-to-2020/, accessed: 2024-12-23.
2. Fresh Plaza, "Tomato farmers bangladesh see bumper production and fair prices," https://www.freshplaza.com/europe/article/9589231/tomato- farmers-bangladesh-see-bumper-production-and-fair-prices/, accessed: 2024-12-23.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XIII, Issue XII, December 2024

3. Bangladesh Agro-Meteorological Information Service (BAMIS),“Tomato,” https://www.bamis.gov.bd/en/ crops/ view/ 13/,accessed:2024-2-23.
4. L. Depenbusch, T. Sequeros, P. Schreinemachers, M. Sharif, K. Man-namparambath, N. Uddin, and P. Hanson, “Tomato pests and diseases in bangladesh and india: farmers’ management and potential economic gains from insect resistant varieties and integrated pest management,” International Journal of Pest Management, pp. 1–15, 2023.
5. S. P. Mohanty, D. P. Hughes, and M. Salath´e, “Using deep learning for mage-based plant disease detection,” Frontiers in plant science, vol. 7, p. 1419, 2016.
6. K. P. Ferentinos, “Deep learning models for plant disease detection and diagnosis,” Computers and electronics in agriculture, vol. 145, pp. 311–318, 2018.
7. A. Sofiane, B. Mostefa, and B. Soumia, “Deep learning model based on vgg16 for tomato leaf diseases detection and categorization,” in 2024 2nd International Conference on Electrical Engineering and Automatic Control(ICEEAC). IEEE, 2024, pp. 1–6.
8. M. Agarwal, A. Singh, S. Arjaria, A. Sinha, and S. Gupta, “Toled: Tomato leaf disease detection using convolution neural network,” Pro-cedia Computer Science, vol. 167, pp. 293–301, 2020.
9. S. AM, P. JOE IR, S. Venkatraman, P. Kumar S et al., “Improved tomato leaf disease classification through adaptive ensemble models with exponential moving average fusion and enhanced weighted gradient optimization,” Frontiers in Plant Science, vol. 15, p. 1382416, 2024.
10. T. Ramcharan, O. Nolan, C. Y. Lai, N. Prabhu, R. Krishnamurthy, A. G.Richter, D. Jyothish, H. K.Kanthimathinathan, S. B. Welch, S. Hackett et al., “Paediatric inflammatory multisystem syndrome: temporally associated with sars-cov-2 (pims-ts): cardiac features, management and short-term outcomes at a uk tertiary paediatric hospital,” Pediatric cardiology, vol. 41, pp. 1391–1401, 2020.
11. A. K. Rangarajan, R. Purushothaman, and A. Ramesh, “Tomato crop disease classification using pre-trained deep learning algorithm,” Proce-dia computer science, vol. 133, pp. 1040–1047, 2018.
12. E. D. N. Iffaty, P. Sukmasetya, M. R. A. Yudianto et al., “Transfer learning vgg16 for image classification of tomato leaf disease,” in Proceedings Universitas Muhammadiyah Yogyakarta Undergraduate Conference, vol. 3, no. 2, 2023, pp. 35–46.
13. D. Hughes, M. Salath´e et al., “An open access repository of images on plant health to enable the development of mobile disease diagnostics,”arXiv preprint arXiv:1511.08060, 2015.
14. A. Debnath, M. M. Hasan, M. Raihan, N. Samrat, M. M. Alsulami, M. Masud, and A. K. Bairagi, “A smartphone-based detection system for tomato leaf disease using efficientnetv2b2 and its explainability with artificial intelligence (ai),” Sensors, vol. 23, no. 21, p. 8685, 2023.
15. A. Sobur, M. H. Kabir, M. Z. Hossain, A. Hossain, and I. C. Rana, “En-hancing tomato leaf disease detection in varied climates: A comparative study of advanced deep learning models with a novel hybrid approach,” International Journal of Creative Research Thoughts— IJCRT, vol. 12, no. 2, 2024.
16. K. Bhalerao, “Tomato leaf dataset,” 2021, ac-cessed: December 27, 2024. [Online]. Available: https://www.kaggle.com/datasets/kaustubhb999/tomatoleaf
17. M. Saravanan and K. S, “Tomato leaf dataset for plant disease detection,” 2020, accessed: December 27, 2024. [Online]. Available: https://data.mendeley.com/datasets/zfv4jj7855/1
18. Geeks for Geeks, “Vgg-16 cnn model,” 2024, accessed: 2024-12-28.[Online]. Available: https://www.geeksforgeeks.org/vgg-16-cnn-model/
19. Data Camp, “Introduction to convolutional neural networks (cnns),” 2024, accessed: 2024-12-28. [Online]. Avail-able: https://www.datacamp.com/tutorial/introduction-to-convolutional-neural-networks-cnns
20. K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” arXiv preprint arXiv: 1409.1556, 2014.[Online]. Available: https://arxiv.org/abs/1409.1556