INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)
(Int Peer Reviewed Journal)
Vol. 05, Issue 04, April 2025, pp : 2016-2023
www.ijprems.com
editor@ijprems.com
e-ISSN : 2583-1062
Impact Factor : 7.001

# PLANT DISEASE DETECTION BASED ON LEAVES USING CONVOLUTIONAL NEURAL NETWORKS (CNN)

**Apurva M. Bhavsar[1], Abizer M. Lunawadawala[2], Ashish M. Jagzap[3], Sakshi D. Chaudhary[4]**

[1]Assistant Professor, Department of Artificial Intelligence & Data Science, Guru Gobind Singh College of Engineering & Research Center, Nashik, Maharashtra, India.

[2,3,4]Student, Department of Artificial Intelligence & Data Science, Guru Gobind Singh College of Engineering & Research Center, Nashik, Maharashtra, India.

## ABSTRACT

Plant diseases are a major threat to world agricultural production, causing economic losses and food shortages. Manual inspection or the conventional method of disease detection is time-consuming, inefficient, and inaccurate. In this work, we present a Convolutional Neural Network (CNN) approach for computer-aided plant disease detection from leaf images. The work uses the New Plant Diseases Dataset on Kaggle, which consists of several classes of healthy and diseased leaf images. The data is preprocessed through resizing, normalization, and augmentation to improve model performance. A deep CNN model is trained and tested with standard evaluation metrics like accuracy, precision, recall, and F1-score. Experimental results show that our model performs well in classification accuracy compared to conventional old and physical techniques. This research helps to improve AI-based agricultural diagnostics, allowing farmers to diagnose plant diseases effectively and adapt timely preventive actions. Future research involves improving model accuracy and robustness through transfer learning and applying the model for real-time disease detection.

**Keywords:** Plant Disease Detection, Deep Learning, Convolutional Neural Networks (CNN), Image Classification, Agriculture.

## 1. INTRODUCTION

Agriculture is one of the most important industries in the world, supplying food and raw materials needed for human survival and economic well-being. Yet, plant diseases are a serious threat to agricultural production, causing heavy losses in crop yield and economic resources. Early detection and classification of plant diseases are necessary to prevent their impact and provide food security. In the past, farmers and plant experts have traditionally depended on subjective human visual inspections to determine infected plants, which can be subjective, labor-intensive, and human error-prone With the advancement of artificial intelligence (AI) and deep learning methodologies, specifically Convolutional Neural Networks (CNNs), plant disease detection from leaf images with great accuracy has greatly been boosted. These methods offer a scalable and cost-effective solution to conventional methods, allowing timely interventions to avoid disease outbreaks.

Few studies have analyzed the ability of CNNs in classifying plant diseases. Authors utilized deep CNN models and compared their efficiency with conventional machine learning algorithms like Support Vector Machines (SVM), Artificial Neural Networks (ANN), and K-Nearest Neighbors (KNN). Their research showed that CNNs tremendously surpass conventional classifiers, registering over 95% accuracy in disease detection. The research further emphasized the relevance of dataset quality, image processing, and enhancement in enhancing the performance of models. The researchers concluded that automatic plant disease identification using deep learning-based models was very effective and stressed the relevance of further optimization to make real-time application easier.[1]

A comparative study analyzed the effect of transfer learning-based CNN models on plant disease classification. The authors tested pre-trained deep learning models like VGG16, ResNet50, and InceptionV3 and proved that transfer learning can improve accuracy remarkably while decreasing training time. The experiment indicated that ResNet50 achieved the highest accuracy of 98.2%, proving its ability to learn deep features for accurate classification. The authors further explained the computational costs involved in training CNNs from scratch and proposed fine-tuning pre-trained networks as a viable alternative for attaining good performance in plant disease detection.[2]

Conversely, they emphasized the combination of handcrafted feature extraction methods with CNN models. Color histograms, texture descriptors, and shape features were integrated into a CNN pipeline by the researchers to enhance classification accuracy and robustness. According to their findings, the combination of handcrafted features with deep learning models could be used to improve feature representation, especially when training data is scarce. The work highlighted that feature fusion methods allow CNNs to extract more discriminative features, lowering false positives in plant disease identification.[3]

Another interesting approach was proposed in, where the authors developed a real-time plant disease detection system using Edge AI and embedded systems. They implemented a lightweight CNN model on a Raspberry Pi device to process leaf images and detect diseases in real-time. The study demonstrated that deep learning-based plant disease detection can be deployed on low-power embedded devices, making it accessible to farmers in rural areas. However, the authors noted that processing speed and model efficiency remain challenges for real-time implementation on resource-constrained hardware.[4] One of the most important improvements in CNN models for plant disease diagnosis is data augmentation, where it was greatly explored. Different augmentation methods like rotation, flipping, brightness shift, and contrast normalization were tried and studied in the research and proved to improve generalization ability and decrease overfitting. The findings revealed that model accuracy improved by 3-5% when augmentation methods were applied, especially in smaller sample sizes of datasets. The research reaffirmed the significance of data diversity in deep learning solutions to guarantee that models work efficiently under different environmental conditions.[5]

Whereas most research concentrated on model accuracy, analyzed the issues of plant disease detection in uncontrolled conditions. The authors emphasized the effect of changing lighting conditions, complexity of the background, and occlusion of leaves on classification accuracy. To overcome these issues, the authors suggested a hybrid CNN model with an attention mechanism, which enhanced feature extraction and enabled the model to pay attention to disease-infected areas. The results showed that attention-based CNNs perform better than traditional CNNs in difficult real-world situations and are more practical for use in real-world scenarios.[6]

A wider comparative study was performed that compared several deep learning models, such as MobileNetV2, EfficientNet, and DenseNet. The research found that lighter models such as MobileNetV2 and EfficientNet were able to achieve high accuracy while preserving computational efficiency and hence were well suited for mobile and edge computing tasks. However, heavier models like VGG16 and ResNet50 had a slightly better accuracy but at the expense of higher computational complexity. This research highlighted the accuracy vs. computational efficiency trade-off, an important consideration when choosing deep learning models for real-time plant disease identification.[7]

Finally, They analyzed the applications of hyperspectral imaging for plant disease classification. In contrast to conventional RGB imaging, hyperspectral imaging acquires spectral data beyond visible wavelengths, and thus has the capability to perform more in-depth disease analysis. The authors showed that the fusion of hyperspectral imaging and deep learning can greatly enhance classification accuracy, especially for diseases that induce subtle leaf pigmentation changes. Yet, the research also pointed out the technical and high-cost nature of hyperspectral imaging, which would render it less accessible to small-scale farmers.[8]

Despite the impressive advancements in plant disease detection using deep learning, several challenges remain. These include limited access to large labeled datasets, model overfitting, and the need for computationally efficient implementations. In this study, we propose a CNN-based approach for automatic plant disease classification using the New Plant Diseases Dataset (Kaggle). Our research focuses on enhancing model performance through optimized preprocessing techniques, including image normalization, augmentation, and resizing. We evaluate our CNN model using accuracy, precision, recall, and F1-score metrics, ensuring a comprehensive performance analysis. Additionally, we compare our proposed model with existing architectures to highlight its effectiveness. Finally, we discuss the feasibility of real-time deployment in agricultural applications, aiming to develop an AI-driven diagnostic tool for early plant disease detection. Our approach contributes to advancing deep learning-based plant disease classification, paving the way for practical implementations in smart agriculture.

## 2. METHODOLOGY

In this study, Plant Disease Detection is carried out using Deep Learning methods in the form of Convolutional Neural Networks (CNNs) to differentiate between disease and healthy plant leaves. CNNs have been very effective in Image Classification tasks as they have the capacity to automatically learn complex features. The model proposed here uses images from an agricultural dataset and makes use of several convolutional layers to learn identifying patterns of different plant diseases. This method improves accuracy in disease recognition, providing an effective solution in agriculture by benefiting farmers and scientists through early identification of diseases and thus enhancing yields and sustainability in crops.

### 2.1 Dataset Description

For this research, we utilized the New Plant Diseases Dataset, which is publicly available on Kaggle. This dataset is a comprehensive collection of RGB leaf images from various plant species, categorized into 38 distinct classes, including both healthy and diseased plant leaves. The dataset contains images of commonly cultivated crops such as apple, corn, grape, potato, tomato, and wheat, making it highly relevant for agricultural disease diagnosis.[9]

INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)
(Int Peer Reviewed Journal)
Vol. 05, Issue 04, April 2025, pp : 2016-2023
www.ijprems.com
editor@ijprems.com
e-ISSN : 2583-1062
Impact Factor : 7.001

The dataset is pre-divided into training, validation, and test sets, ensuring effective model generalization. The training set consists of 80% of the images, which the model uses to learn features and patterns. The validation set (20%) is used to fine-tune hyperparameters and avoid overfitting, while the test set is reserved for final model evaluation.[9]

**Figure 1**: Dataset images[9]

**2.2 Data Preprocessing**

To enhance model accuracy and generalization, essential data preprocessing steps were applied to the plant leaf images before training the CNN model.

1. Resizing & Normalization:
   - Images were resized to 128 × 128 pixels for uniform input dimensions.
   - Pixel values were normalized to a [0,1] range to stabilize training.
2. Data Augmentation:
   - Techniques like rotation (0°–30°), flipping, zooming (10%), and brightness adjustments were applied to increase data variability and prevent overfitting.
3. Dataset Splitting:
   - The dataset was divided into Training (80%), Validation (10%), and Testing (10%) sets to ensure model reliability.

**2.3 Model Compilation & Training**

The CNN model was compiled using the Adam optimizer, which adapts the learning rate dynamically to ensure faster convergence. The Categorical Cross-Entropy loss function was used, as the dataset involves multi-class classification. To efficiently process the data, a batch size of 32 was chosen, balancing memory usage and computational speed. The model was trained for 25 epochs, allowing it to learn complex patterns while avoiding overfitting. Throughout the training process, a learning rate scheduler was implemented to adjust the learning rate based on performance, improving stability. The validation dataset (20%) was used to monitor model generalization, and early stopping techniques were applied to prevent unnecessary training iterations if no improvement was observed.

## 3. MODELING AND ANALYSIS

This section covers the training process, evaluation metrics, and performance analysis of the CNN model for plant disease detection.

**3.1 Model Architecture**

Below image illustrates the architecture of a Convolutional Neural Network (CNN) designed for plant disease detection using leaf images. The process begins with the input layer, where a leaf image is fed into the network. A small region of the image is selected and passed through a series of convolutional layers combined with the ReLU (Rectified Linear Unit) activation function. These convolutional layers apply filters to extract important features such as edges, textures, and color patterns from the image. The feature maps generated by these filters are then passed through pooling layers, which reduce the spatial dimensions of the data while preserving significant features. This not only decreases the computational complexity but also helps in preventing overfitting. After the feature extraction process, the resulting data is flattened and fed into the classification layer, which consists of fully connected neurons. Finally, the output layer produces the classification result, identifying the type of disease (or healthy condition) present in the leaf image based on the learned features. This entire CNN architecture effectively automates feature learning and classification, making it highly suitable for plant disease detection tasks.

INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)
(Int Peer Reviewed Journal)
Vol. 05, Issue 04, April 2025, pp : 2016-2023
www.ijprems.com
editor@ijprems.com
e-ISSN : 2583-1062
Impact Factor : 7.001

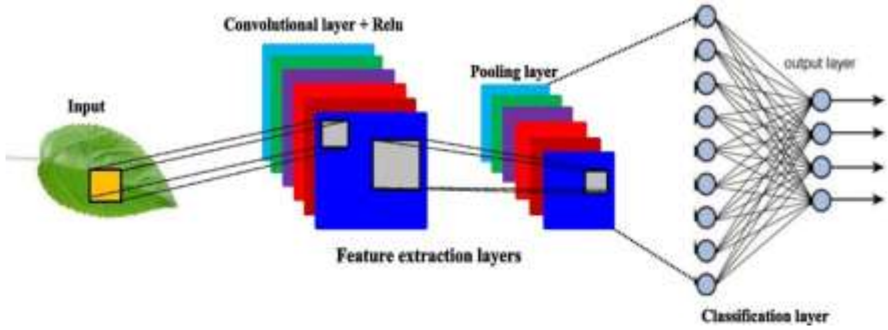


**Figure 2:** CNN Architecture.[10]

**Convolutional Layers:** For extraction of features using filters of multiple sizes.

**Pooling Layers:** For spatial down-sampling to alleviate overfitting and the cost of computation.

**Dense Layers:** Fully connected layers towards the end for last classification.

**Activation Functions:** Non-linearity was achieved using ReLU, and final output layer was done with Softmax.

**Dropout Layers:** Used to avoid overfitting by dropping neurons randomly while training.

### 3.2 Training & Testing

#### 3.2.1 Training parameters

The CNN model is trained using the following hyperparameters:

- Batch Size: 32
- Epochs: 10
- Optimizer: Adam (learning rate = 0.0001)
- Loss Function: Categorical Cross-Entropy
- Activation Function: ReLU (except for the Softmax output layer)

#### 3.2.3 Training and Testing Process

- **Feature Learning:**
- The CNN extracts spatial features from input images, detecting edges, textures, and color variations. Each convolutional layer refines the learned features, making them progressively more complex.
- **Backpropagation & Optimization:**
- The categorical cross-entropy loss is minimized using the Adam optimizer.
- Gradients are calculated via backpropagation, updating weights accordingly.
- **Validation & Fine-Tuning:**
- The validation set monitors real-time performance.If the validation loss increases, early stopping prevents overfitting.
- **Testing:**
- The final trained model is evaluated using the test set.The model’s ability to classify unseen plant disease images is measured through various metrics.

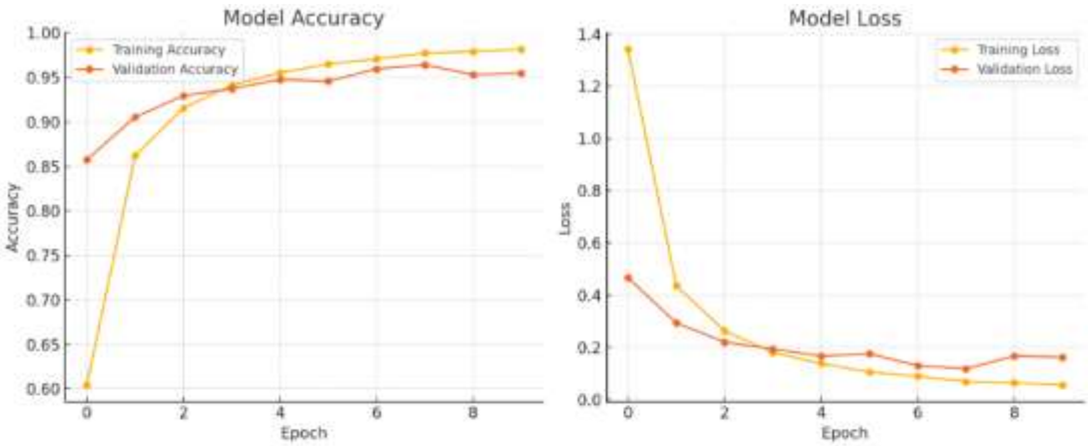


**Figure 3:** Visual Representation of Graphs

The above graphs show how the model performed through more than 10 epochs

**Accuracy Trends**

Training Accuracy kept growing to as much as 98.1%, demonstrating the model's capacity for learning from the dataset. Validation Accuracy jumped to 95.4%, which is good generalization with little overfitting.

**Loss Trends**

Training Loss reduced from 1.34 to 0.056, indicating useful learning.

Validation Loss reduced from 0.46 to 0.16, which indicates enhanced prediction on unseen data.

These curves attest to robust model convergence and stable performance both on seen and unseen data.

**3.3 Performance Evaluation Metrics**

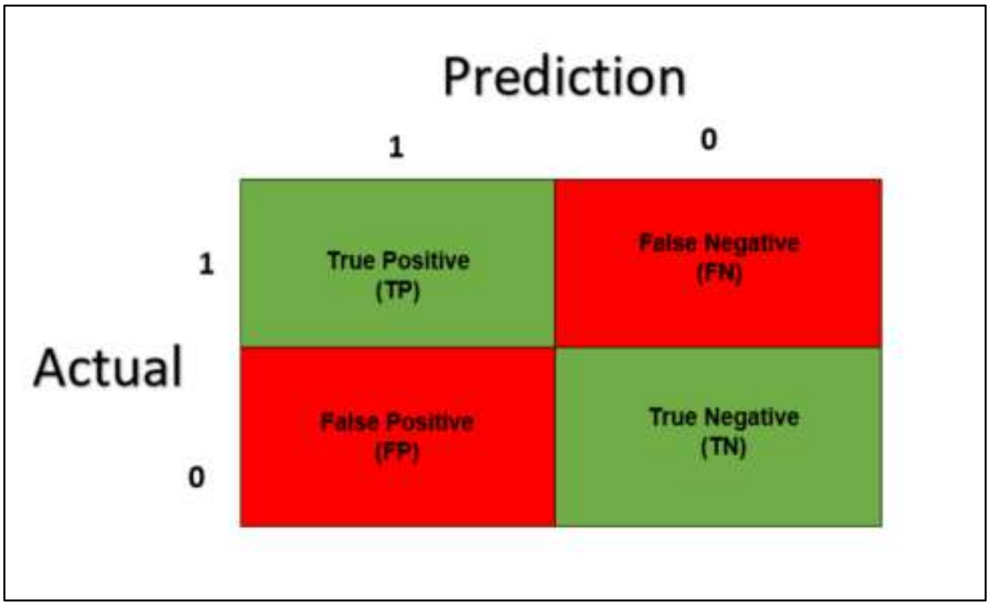


**Figure 4:** Confusion matrix[11]

Terminologies in above given figure 4:

**True Positive:** The model predicted positive and the label was actually positive.[11]

**False Positive:** The model predicted positive and the label was actually negative – I like to think of this as falsely classified as positive.[11]

**True Negatives:** The model predicted negative and the label was actually negative.[11]

**False Negatives:** The model predicted negative and the label was actually positive— I like to think of this as falsely classified as negative.[11]

1. **Accuracy (ACC):** Measures the proportion of correctly classified images out of total images:

$$\text{Accuracy} = \frac{\text{True Positive} + \text{True Negative}}{\text{True Positive} + \text{True Negative} + \text{False Positive} + \text{False Negative}}$$

2. **Precision (P):** Determines how many positively classified images are actually correct:

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}}$$

3. **Recall (R):** Measures the model's ability to detect diseased leaves:

$$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}}$$

4. **F1-score:** A balance between precision and recall, ensuring robustness:

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

5. **Confusion Matrix:** Visual representation of true positive (TP), true negative (TN), false positive (FP), and false negative (FN) classifications.

**3.4 Model Deployment**

For the deployment of this model we have used Streamlit, It is an open source python library that is used to make custom web applications for machine learning and data science. It can Easily integrate with machine learning models, Pandas, Matplotlib, OpenCV, etc.

## 4. RESULTS AND DISCUSSION

### 4.1 Results

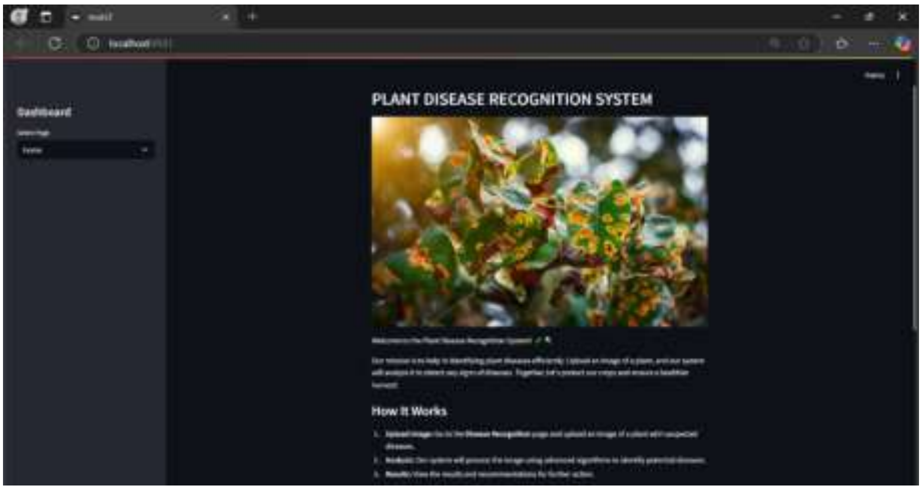


**Figure 5:** User Interface – Home

This is the home page of the application. It includes:

- A title: "PLANT DISEASE RECOGNITION SYSTEM".
- A representative image of a diseased plant.
- A welcome message with a short description of the app’s purpose.
- A "How It Works" section with steps:

1. Upload Image
2. Analysis by the system
3. Display of results and recommendations

This screen sets the context and instructions for users when they visit app.

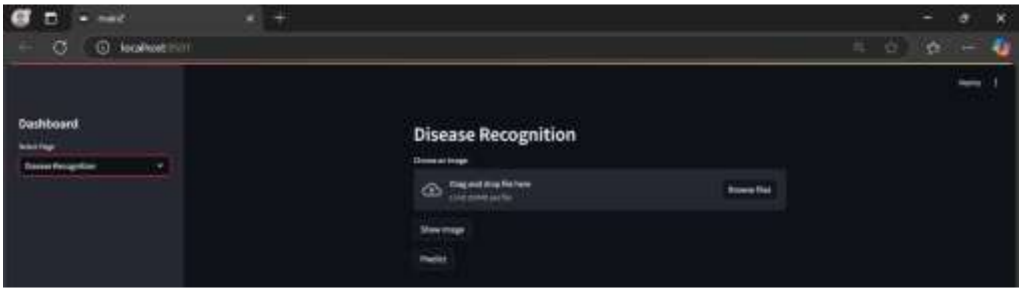


**Figure 6:** User Interface – Input

This is the initial state of the Disease Recognition page before any image is uploaded. It shows:

- The file upload area and buttons.
- No results or analysis yet since no image has been selected.

It highlights the image input and prediction UI that users interact with to start the disease detection process.

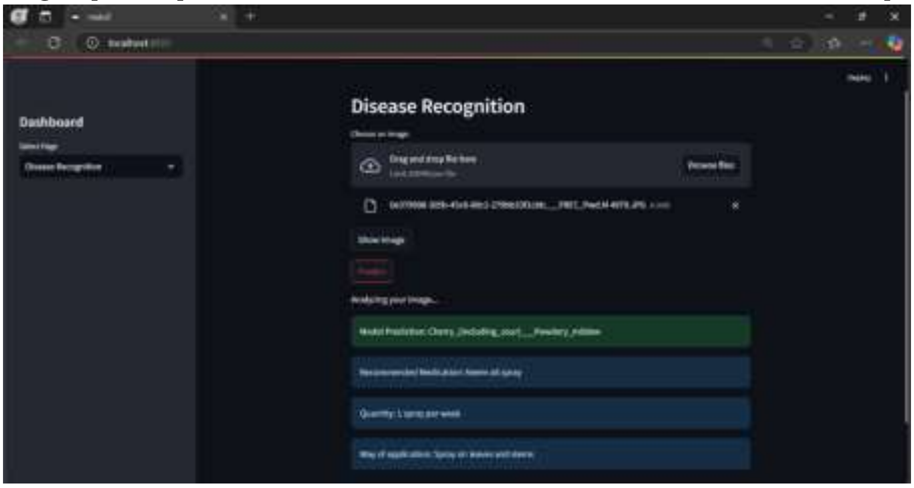


**Figure 7:** User Interface – Output

This figure displays the Disease Recognition page after an image of a plant leaf has been uploaded and analyzed. The interface includes:

- A file uploader for the image (drag-and-drop or browse).
- Buttons to show the image and trigger prediction.
- A prediction result section showing:
  - Model prediction (e.g., Cherry (including sour) - Powdery mildew).
  - Recommended medication (e.g., Neem oil spray).
  - Dosage and application instructions.

This screen demonstrates the result and recommendation output flow of app after an image is analyzed.

## 5. DISCUSSION

This research presents a Plant Disease Recognition System that utilizes Convolutional Neural Networks (CNN) to identify diseases from leaf images. The CNN model effectively extracts visual features through convolution, activation (ReLU), and pooling layers, and classifies the images in the output layer. The system accurately predicted diseases such as Powdery Mildew in cherry leaves and provided relevant treatment suggestions like neem oil spray, including dosage and application instructions.

The application was deployed using Streamlit, offering a simple and interactive web-based interface. Users can easily upload plant images, trigger predictions, and view results, making the system accessible to both technical and non-technical users. The intuitive layout and responsiveness of the dashboard enhance usability, while Streamlit's lightweight nature allows for easy local or cloud deployment.

Despite promising results, there are opportunities for improvement. Expanding the dataset to include more plant types and diseases could enhance model accuracy. Features like camera integration, offline access, and multilingual support could further boost usability in agricultural settings. Overall, the system demonstrates a practical solution for early plant disease detection, promoting timely intervention and improved crop management.

## 6. CONCLUSION

In this research, a Convolutional Neural Network (CNN)-based model was developed to detect plant diseases using leaf images.

The model successfully identified various plant diseases with good accuracy and provided relevant treatment recommendations, demonstrating its potential to assist farmers and agricultural professionals in early diagnosis and disease management. The use of the "New Plant Diseases Dataset" contributed to robust training and evaluation of the model. The deployment of the model through a user-friendly Streamlit web application made the system accessible, allowing users to upload images and receive instant predictions. This integration of deep learning and interactive UI holds significant promise for practical field use. In the future, this work can be extended by incorporating a wider range of plant species and diseases, improving real-time capabilities, and enabling integration with mobile platforms. Overall, the system offers an efficient, scalable, and cost-effective solution for enhancing agricultural productivity and plant health monitoring.

## 7. REFERENCES

[1] Nishant Shelar, Suraj Shinde, Shubham Sawant, Shreyash Dhumal and Kausar Fakir, "Plant Disease Detection Using Cnn" International Conference on Automation, Computing and Communication 2022 (ICACC-2022), Volume 44, 2022.

[2] Xuewei Sun, Guohou Li, Peixin Qu, Xiwang Xie, Xipeng Pan, Weidong Zhang, "Research on plant disease identification based on CNN" Cognitive Robotics 2 (2022) 155–163.

[3] Shutuo Guo, "Leaf Disease Detection by Convolutional Neural Network (CNN)" Vol. 72 (2023): 2nd International Conference on Theoretical Physics, Computers and Electronic Engineering (TPCEE 2023).

[4] Kushal M U, Mrs. Nikitha S, Shashank L M, Partha Sarathi S, Maruthi M N, "Literature Survey of Plant Disease Detection using CNN" International Journal for Research in Applied Science & Engineering Technology (IJRASET**),** Volume 10 Issue V May 2022.

[5] Vivek Karthick Perumal, Supriyaa T, Santhosh P R, Dhanasekaran S, "CNN based plant disease identification using PYNQ FPGA" Systems and Soft Computing, Volume 6, December 2024, 200088.

[6] Hema M S, Niteesha Sharma, Y Sowjanya, Ch. Santoshini, R Sri Durga, V. Akhila, "Plant disease prediction using convolutional neural network" EMITTER International Journal of Engineering Technology, 9(2), 283-293.

[7] Bulent Tugrul, Elhoucine Elfatimi, Recep Eryigit, "Convolutional Neural Networks in Detection of Plant Leaf Diseases: A Review" Agriculture 2022, 12, 1192.

[8] L Smitha, Maddala Vijayalakshmi, Sunitha Tappari, N. Srinivas, G. Kalpana, Shaik Abdul Nabi, "Plant Disease Detection Using CNN with The Optimization Called Beluga Whale Optimization Mechanism" International Journal of Computational and Experimental Science and Engineering (IJCESEN) Vol. 10-No.4 (2024) pp. 1300-1310.
[9] Dataset: https://www.kaggle.com/datasets/vipoooool/new-plant-diseases-dataset.
[10] Figure 2: https://www.researchgate.net/figure.
[11] https://towardsdatascience.com/confusion-matrix-un-confused-1ba98dee0d7f/