iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 Issue: IX Month of publication: September 2025

DOI: https://doi.org/10.22214/ijraset.2025.74149

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue IX Sep 2025- Available at www.ijraset.com

Enhanced Plant Leaf Disease Detection Using CLAHE-Processed Images and Custom CNN Deep Learning Architecture

Neelam Sulaiya[1], Shyamol Banerjee[2]
[1]Research Scholar, [2]Asst. Prof, Dept. of Computer Science and Engg., Shriram College of Engineering & Management, Gwalior

Abstract: This study aims to develop an efficient and accurate deep learning-based model for the classification of plant leaf diseases using Convolutional Neural Networks (CNN). The objective is to automate disease detection in agricultural crops to assist farmers and agricultural experts in early and reliable diagnosis. The model is trained on the publicly available "Plant Village CLAHE Processed Data" dataset, which includes high-resolution RGB images of healthy and diseased plant leaves. Images are preprocessed through resizing (128×128), normalized, and split into training, validation, and test sets. Data augmentation techniques such as flipping, zooming, and rotation are used to improve generalization. A custom CNN architecture comprising convolutional, pooling, dense, and dropout layers is employed and trained using the Adam optimizer. Exploratory Data Analysis (EDA) ensures data quality and balance. The model achieves impressive results, with 93% test accuracy, 91% precision, 93% recall, and an F1-score of 92%, indicating robust performance in identifying diverse plant diseases. Training accuracy reached 94.64% with a validation accuracy of 92.95%, confirming minimal overfitting. These results validate the model's reliability for practical use in smart farming solutions, especially in mobile or IoT-based applications for real-time disease monitoring and precision agriculture.
Keywords: Plant Disease Classification**, **Convolutional Neural Network (CNN)**, **Deep Learning in Agriculture**, **Image Preprocessing (CLAHE)** and **Smart Farming Applications.

I. INTRODUCTION

Particularly in nations where a sizable fraction of the population depends on farming for livelihood, agriculture is the pillar of world food security and economic growth. Nonetheless, one of the ongoing difficulties in agriculture is the predominance of plant diseases, which can cause major crop yield losses, worse product quality, and higher production costs[1]–[4]. The Food and Agriculture Organisation (FAO) estimates that annually between 20–40% of crop output losses are caused by plant diseases. Thus, early identification of plant illnesses and precise diagnosis of them are absolutely vital to reduce damage, guarantee good crop development, and preserve agricultural output. Historically, crop disease detection has depended on manual techniques such farmers' eye inspections or agricultural specialists' consultations[5], [6]. While effective in some cases, these methods are time-consuming, labor-intensive, and highly subjective, depending on the experience and expertise of the observer. Moreover, in remote or under-resourced farming communities, access to expert advice is often limited. With deep learning rising as a top answer, these constraints have driven researchers and engineers to investigate automated, efficient, scalable methods of disease diagnosis. A subset of machine learning, deep learning has transformed computer interpretation of data—especially images. Because they can automatically learn spatial hierarchies of features from raw pixel data, Convolutional Neural Networks (CNNs) have shown to be rather effective among the several deep learning architectures for image-based tasks. Inspired by the human visual system, CNNs can remarkably quickly and precisely examine visual images. By analysing digital photographs of afflicted leaves, CNNs have shown amazing ability in the context of agriculture in identifying and classifying plant leaf illnesses. Especially in the early phases of illness development, this invention helps robots to detect delicate visual signals such discolouration, texture alterations, and shape defects that could be challenging for the human eye[7]–[9].

Usually, CNNs identify plant leaf diseases by training a model on a vast collection of annotated leaf photos where every image is labelled based on the type of disease it reflects or marks as healthy. By means of layers of convolution, pooling, and activation functions, the CNN automatically extracts and learns pertinent characteristics during training. Once taught, the model can very accurately classify fresh, unseen leaf images[10], [11].

Plant disease classification has seen the application of several CNN architectures including LeNet, AlexNet, VGGNet, GoogLeNet, ResNet, and EfficientNet each with varying trade-offs between model complexity, computational efficiency, and prediction performance. Among other crops, tomatoes, potatoes, maize, grapes, apples, and rice have all had illnesses found using these models effectively identified in them.

Research showing classification accuracy higher than 90% shows CNN-based methods' feasibility in practical agricultural environments. Integration of CNN-based plant disease diagnosis into useful farming applications has transforming power. Farmers can photograph suspected leaves and get immediate diagnosis feedback by including trained models into mobile apps, handheld devices, or unmanned aerial vehicles (drones). Such instruments democratise knowledge about plant diseases, therefore empowering even smallholder farmers in remote areas to make wise choices on disease control. Greenhouses and big farms can also use automated detection systems for constant surveillance, therefore lowering the requirement for human monitoring and allowing quick interventions. Together with Internet of Things (IoT) devices and cloud-based platforms, these systems can also enable pattern analysis, extensive illness surveillance, and the creation of predictive models for next pandemics.

CNN-based illness detection systems present various difficulties in their use notwithstanding the encouraging developments. First of all, CNN performance is highly influenced by the quality and volume of training data. Although publicly available datasets like PlantVillage have proved very helpful for model building, they may feature photographs taken under controlled conditions that might not generalise well to field settings with different illumination, background clutter, and occlusion. Extensive data augmentation, domain adaption, and maybe multimodal data (e.g., environmental parameters or spectral imaging) are needed to guarantee model robustness in many real-world scenarios. Second, interpretability of deep learning models still causes issues especially for end users like farmers who would need justification for the model's choices. By stressing image areas that affect model predictions, recent initiatives in explainable artificial intelligence (XAI) seek to solve this problem and thereby improve user confidence and system transparency[12]–[15].

Furthermore taken into account in putting these technologies broadly into use are data protection, digital literacy, and infrastructure constraints in rural locations. Successful adoption and sustained usage of CNN-based plant disease detection systems depend on cooperation amongst artificial intelligence researchers, agricultural scientists, government agencies, and local communities. Furthermore accelerating the shift to intelligent farming methods are regulations supporting open access to agricultural data, subsidised technology deployment, and farmer training programs. An important advance in the application of artificial intelligence in agriculture is the detection of plant leaf diseases using convolutional neural networks. CNNs provide a strong weapon to increase crop health, lower production losses, and assist the lives of farmers all around by providing accurate, rapid, scalable disease identification. Deep learning methods hold the potential to revolutionise conventional farming into a smarter, more resilient, and data-driven company as they develop and interact with other developing technologies such IoT, cloud computing, and precision agriculture. Harnessing the full possibilities of these developments for world agricultural sustainability will depend on closing the gap between research and practical application.

II. LITERATURE REVIEW

Joshi 2025 et al. This study applies a YOLOv8n model to the PlantDoc dataset for detecting seven tomato diseases. A hybrid data augmentation strategy increased dataset size, achieving 96.5% mAP and 95% F1-score. The model demonstrates high accuracy and adaptability, supporting precision agriculture through scalable and efficient disease detection systems[16].

Pradeep 2025 et al. This work outperformed conventional classifiers by diagnosis of pepper leaf illnesses using CNN architecture with 99.65% accuracy. Model performance enhanced via segmentation and preprocessing. The results highlight CNN's practical possibilities in automated, high-precision plant disease diagnostics for contemporary agricultural techniques as well as its ability to capture intricate patterns[17].

Habaragamuwa 2024 et al. Plant leaf pictures were classified using a variational autoencoder (VAE) model to improve interpretability. Applied to the PlantVillage collection, the model balanced explainability with accuracy. This approach supports the creation of explainable artificial intelligence tools for agriculture by being relevant to many crops and image classification problems[18].

Kaur 2023 et al. Integrating Ant Colony Optimisation with CNN for plant disease classification, the ACO-CNN model It improves accuracy by removing noise and useless traits before classification. Analysed under several performance criteria, this technology shows better diagnostic capacity than conventional techniques and presents a fresh approach for the identification of plant diseases[6].

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue IX Sep 2025- Available at www.ijraset.com

Alshammari 2023 et al. This work presents HL-FO, a hybrid Lion–Firefly optimisation technique coupled with CNN for olive leaf disease classification. Under ROC, F1-score, and accuracy the model stresses feature selection to prevent overfitting. In agricultural diagnostics, the method promotes sophisticated decision-making and improves classification dependability[19]

.

Table 1 Literature Summary

Author / Year	Methodology	Results	Research Gap	Limitations
Bhatti et al., 2024[20]	Introduced ETFG model combining 3D-CNN, fuzzy C-means clustering, PCA, and GAT for hyperspectral image classification in agriculture.	Achieved optimal spatial-spectral classification with improved accuracy and dimensionality reduction.	Lack of benchmarking against real-field agricultural image datasets.	High model complexity; performance on noisy field data not evaluated.
Li et al., 2024[21]	Proposed PL-DINO object detection using CBAM with ResNet50 and Equalization Loss for class imbalance.	Achieved 70.3% mAP on PlantDoc; outperformed YOLOv7 and Faster R-CNN in real-world leaf detection.	Limited evaluation across multiple plant species and varied environmental conditions.	Performance may decline with highly imbalanced or occluded leaf datasets.
Sharma et al., 2024[22]	Employed image processing for rose plant disease detection using segmentation, feature extraction, and classification stages.	Enabled early identification of rose diseases, improving disease management.	Focused only on rose plants; lacks generalization to other crops or disease types.	Manual segmentation and preprocessing stages may limit scalability.
Mane et al., 2024[23]	Developed a hybrid CNN+SVM/KNN model for basil leaf disease detection using a custom dataset and balanced sampling.	Achieved 95.02% accuracy across five basil leaf classes.	No standard basil leaf dataset; model not compared with advanced deep learning models.	Dataset is limited in size and diversity; custom data may bias generalization.
Ouamane et al., 2024[24]	Proposed HOWSVD-MDA tensor subspace learning model for tomato disease classification using PlantVillage and Taiwan datasets.	Achieved 98.36% and 89.39% accuracy on PlantVillage and Taiwan datasets respectively.	Needs validation across additional datasets and more plant species.	Model performance may vary significantly depending on dataset quality and dimensional consistency.

.

III. METHODOLOGY

This work uses the Plant Village CLAHE-processed dataset to classify plant leaf diseases using a custom CNN. Images are loaded and scaled to 128x128 pixels then split into training, validation, and test groups. Using batching and prefetching guarantees effective data handling; augmenting techniques include rescaling, flipping, rotation, and zooming improve model generalisation. Class balance and data quality are evaluated in exploratory data analysis (EDA). Multiple convolutional and pooling layers abound in the CNN design, then dense and dropout layers extract strong features. The model is trained with the Adam optimiser then tested for accuracy, precision, recall, and F1-score.

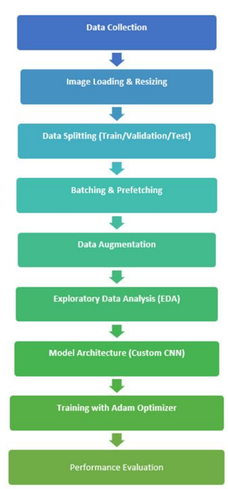

Fig. 1 Proposed Flowchart

A. Data Collection

The dataset used in this study is the publicly available “Plant Village CLAHE Processed Data” hosted on Kaggle (https://www.kaggle.com/datasets/rahimanshu/plant-village-clahe-processed-data). Along with healthy samples, it has RGB photos of plant leaves afflicted by several illnesses. Using CLAHE (Contrast Limited Adaptive Histogram Equalisation), a method that increases local contrast and sharpens the visibility of fine details in the photos, each has been preprocessed. By means of class-wise folders, the dataset is arranged such that each image can be automatically labelled depending on its folder name. This ordered approach supports models of supervised deep learning rather successfully. The dataset fits for building strong classification models thanks to the large spectrum of classes and balanced distribution. The dataset lessens the effort of hand cleaning or labelling since it includes preprocessed, high-quality, cleaned images. Furthermore, the availability of such consistent data guarantees that the model can be faithfully compared with other studies employing the same dataset and increases reproducibility. The dataset offers a complete visual depiction of plant disease categories, which helps to build a scalable, generalisable model for agricultural uses.

B. Data Preprocessing

1) Image Loading and Resizing: Using TensorFlow's image_dataset_from_directory() method—which reads files from the directory structure and assigns labels depending on folder names—images are loaded. To keep consistency, all photos are downsized to 128×128 pixels at this stage. CNNs depend on fixed-size input to preserve consistent kernel application and tensor forms over the model, so this resizing is absolutely essential. While keeping enough resolution for good feature extraction, standardising input dimensions also aids in lowering the computational complexity, memory use, and training time. Different image sizes without scaling could introduce mistakes during model construction and compromise learning performance.
2) Data Splitting (Train, Validation, Test): Using the validation_split argument, the dataset is split 80% for training and 20% for first validation following loading. Using.take() and.skip(), 10% of the training subset is further split to function as a manual validation or testing set. Three separate, non-overlapping datasets—training, validation, as well as test—come of this. Maintaining these partitions guarantees objective performance measurement and eliminates data leaks. It also lets hyperparameter tuning free from polluting the test set, set aside for last model evaluation.
3) Batching and Prefetching: To enable good model training, the datasets—training, validation, and test—are batch-sized at 32 in order. Batching enables the model concurrently process smaller sets of data, hence reducing memory load and increasing training speed. Every dataset also makes use of TensorFlow's prefetch (buffer_size=AUTOTUNE). Running concurrently with model training, prefetching loads and preprocesses data, therefore reducing input bottlenecks and optimising GPU consumption. Especially in cases of large image datasets, these techniques ensure that the data pipeline remains optimal, hence accelerating the training process without compromising stability or model performance.
4) Data Augmentation: Using tf.keras. Sequential, a data augmentation process is created to artificially extend the dataset and enhance generalisation. To normalise pixel values to the [0, 1] range, rescaling (1./255) is included; random flip ("horizontal") is used to replicate mirror views; random rotation and random zoom help to account for actual variance. Although augmentation is applied at runtime during training rather than directly during loading. This method guarantees the model sees different inputs while maintaining validation/test consistency, hence lowering overfit and enhancing real-world performance.

C. Exploratory Data Analysis (EDA)

Before model development, exploratory data analysis (EDA) is used to study the properties of the dataset. Top concerns are determining class distribution, visualising representative images, and identifying likely abnormalities. Plotting the photo count for each class helps one to verify class balance and avoid model bias towards over-represented categories. Sample photos from every class are on display for a visual review of quality, clarity, and originality. This stage additionally ensures that CLAHE preprocessing has increased the contrast satisfactorially for the aim of obtaining discriminative features. EDA also encompasses search for abnormalities, missing or corrupted files, or repetitive images that could distort the learning process. EDA insights direct decisions about preprocessing, augmentation, and modelling approaches. Should class imbalance be observed, for example, additional augmentation could be done for under-represented groups. EDA serves as a quality check and gives hope that the dataset is fit for a CNN model of high performance. All things considered, it provides a vital link between raw data and model development that guarantees data readiness and best performance downstream.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue IX Sep 2025- Available at www.ijraset.com

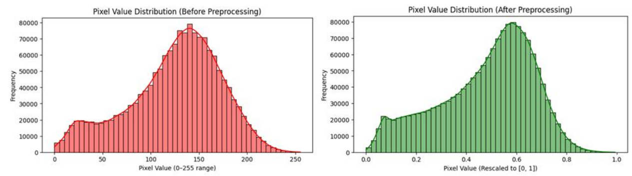

Fig. 2 Histograms of Pixel

The left histogram, colored red, shows the distribution of pixel values before preprocessing, ranging from 0 to 255. The right histogram, colored green, displays the distribution of pixel values after preprocessing, with the values rescaled to a range of 0 to 1. Both histograms illustrate the frequency of pixel values within their respective ranges.

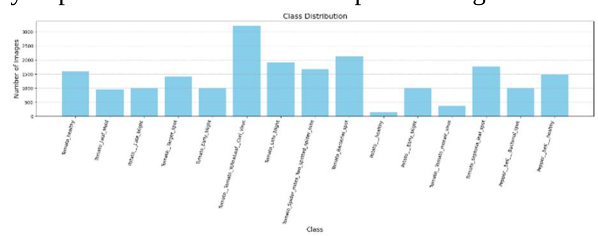

Fig. 3 Class Distribution of each Classes

This figure shows the number of images available per class, confirming balanced distribution across all disease and healthy categories.

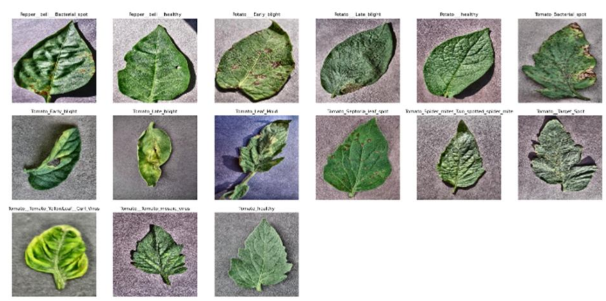

Fig. 4 Sample Images of Each classes

Representative images from each class demonstrate visual differences in disease patterns, aiding in model learning and human interpretation.

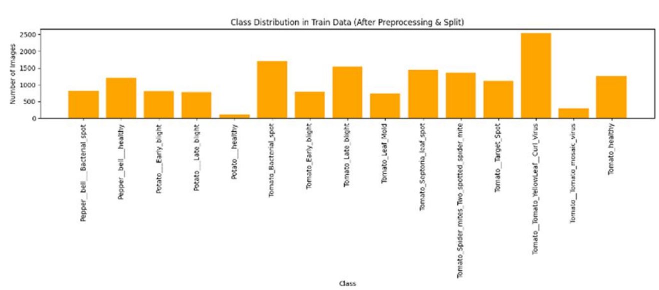

Fig. 5 Class Distribution in Train Data

Displays how training data is distributed among classes after splitting, ensuring the model trains on a balanced and fair dataset.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue IX Sep 2025- Available at www.ijraset.com

Fig. 6 Sample Augmented images (After Preprocessing)

Illustrates the augmented images using flipping, rotation, and zooming, which enhance dataset diversity and improve model generalization.

D. Model Architecture

To increase generalisation, the proposed custom CNN model employs rescaling, flipping, rotation, and zoom and processes RGB images resized for homogeneity. It consists of several convolutional layers that gradually extract intricate information, each followed by pooling layers to lower spatial dimensionality. Flattened and processed through a fully connected layer with activation and dropout for regularisation, the obtained features are Class probability for multi-class classification are produced by the last softmax layer. The model is trained to maximise accuracy by use of an adaptive optimiser and suitable loss function.

1) Input Layer: The model starts with an explicit input layer accepting images of shape (128, 128, 3), thereby representing 128x128 RGB images. This guarantees, for convolutional neural networks, a constant shape and format for all images supplied into the model. Explicit definition of the input layer improves model readability and tool compatibility as well. summary() and transfer learning extensions.

2) Data Augmentation Layer: The model uses a data augmentation layer comprising multiple transformations immediately following the input: pixel value rescaling (1./255), random horizontal flipping, random rotation, and random zoom. By making the model less sensitive to the precise location or orientation of features in the training images, these augmentations replicate real-world variability and help minimise overfitting. Crucially, this increase is limited to training, so evaluation data stays unmodified for proper validation.

3) Convolutional and Pooling Layers: The model derives spatial characteristics from the input photos by use of three blocks of convolutional and max pooling layers. After max pooling to lower dimensionality, the first Conv2D layer employs 32 filters using a 3x3 kernel size. Following max pooling layers, the second and third convolutional layers progressively raise the number of filters to 64 and 128 correspondingly. Beginning from edges and textures in the early layers, these layers climb hierarchically to identify ever more abstract visual patterns, then more intricate forms and structures in the deeper layers.

4) Flatten and Dense Feature Extraction: The Flatten layer flattens the feature mappings into a 1D vector following the convolutionary layers. This lets the data flow into completely connected, Dense layers. ReLU activation in the first dense layer generates non-linearity and aids in learning intricate patterns by means of 128 units. It is marked "features," suggesting, should necessary visualisation or feature extraction employ it.

5) Dropout Regularization: A Dropout layer featuring a dropout rate of 0.5 is included to fight overfitting. 50% of the neurones in the dense layer are thus randomly silenced at every phase during training. Dropout serves as a regularising tool, pushing the model towards learning more generalisable patterns rather than memorising the training set.

Table 2. Hyper parameter Details

Hyperparameter	Value	Description
Batch Size	32	Number of samples processed before updating the model weights.
Epochs	100	Number of complete passes through the entire training dataset.
Image Size	128×128	Dimension to which all input images are resized.
Optimizer	Adam	Optimization algorithm used to update model weights.
Loss Function	Sparse Categorical Crossentropy	Measures the difference between predicted and true labels for multi-class classification.
Dropout Rate	0.5	Fraction of neurons randomly dropped during training to reduce overfitting.

IV. RESULTS AND DISCUSSIONS

Standard performance measures—accuracy, precision, recall, and F1-score—evaluated the suggested Plant Disease-CNN model. These measures give a whole picture of the model's capacity to appropriately classify several disease categories from leaf photos. The findings show that the model performs well in all evaluation stages and is hence dependable as well.

A. Performance Metrics Explained

1) Accuracy: The fraction of all predictions that are accurate overall.
2) Precision: High accuracy means less false positives; the ratio of accurately predicted positive observations to the total expected positives determines this.
3) Recall: Correctly projected positive observations to all actual positives; strong recall reduces false negatives.

B. Training and Validation Performance

TABLE 3. PERFORMANCE EVALUATION OF TEST RESULTS

Model	Accuracy	Precision	Recall	F1-Score
Plant disease-CNN Model	93	91	93	92

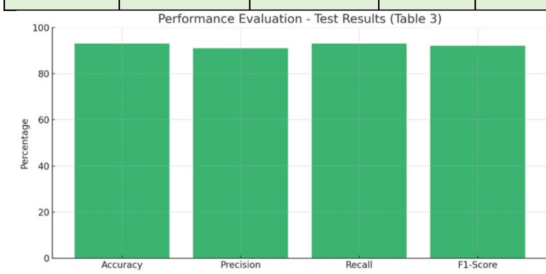

Fig. 7 . Performance Evaluation of Test Results

Using important classification measures—accuracy, precision, recall, and F1-score—table shows the performance evaluation of the proposed Plant Disease-CNN model on the test set. With an accuracy of 93%, the model effectively categorised 93 of every 100 test photos. This great accuracy captures the general success of the program in recognising plant diseases from leaf photos. With a precision of 91%, the model was accurate 91% of the time when it projected a certain condition, so implying a low false positive rate and a great degree of prediction confidence. Strong sensitivity and low false negatives are shown by the model's successful detection of 93% of all genuine illness cases in the dataset. Even if class distributions vary, the F1-score of 92% shows that the model works consistently across several plant disease classes by balancing precision and recall. These measures taken together verify that the model not only generates correct forecasts but also generallyizes effectively to unprocessed data. Such consistent and excellent performance across all criteria suggests that the CNN model is rather dependable for practical agricultural uses, including mobile and IoT-based disease monitoring systems in farms and rural places.

TABLE 4. PERFORMANCE EVALUATION OF TRAINING RESULTS

Model	Accuracy	Loss	Validation Accuracy	Validation Loss
Plant disease-CNN Model	94.64	0.1793	92.95	0.2758

Fig. 8 Performance Evaluation of Training Results

Table 4 offers a complete picture of Plant Disease-CNN model performance in training and validation stages. With a training accuracy of 94.64%, the model learnt patterns from the training data rather successfully. This great accuracy shows that most of the photos the model trained on could be properly categorised by it. The training loss of 0.1793 underlines even more how well the model reduced the error between its forecasts and actual labels during development. More crucially, the excellent generalising capacity of the model to unseen data shown by the validation accuracy of 92.95%, which is near to the training accuracy. This little variation in training and validation accuracy implies that the model did not overfit the training data, which is a crucial need for strong performance in practical settings. Furthermore indicating steady learning over the epochs, the validation loss of 0.2758 is rather modest and well-aligned with the training loss. These findings show generally that the model kept consistency between the training and validation phases, learning discriminative features successfully while avoiding memorising of the training set. This harmony between loss and precision helps the model to be suitable for use in systems of real-time plant disease monitoring and agricultural diagnostics.

C. Visualization and Insights

The confusion matrix and performance graphs (e.g., accuracy and loss curves) provide further insight. The confusion matrix reveals which classes are most and least accurately predicted, helping identify opportunities for improvement. The accuracy and loss curves show a smooth convergence, further validating that the training process was stable.

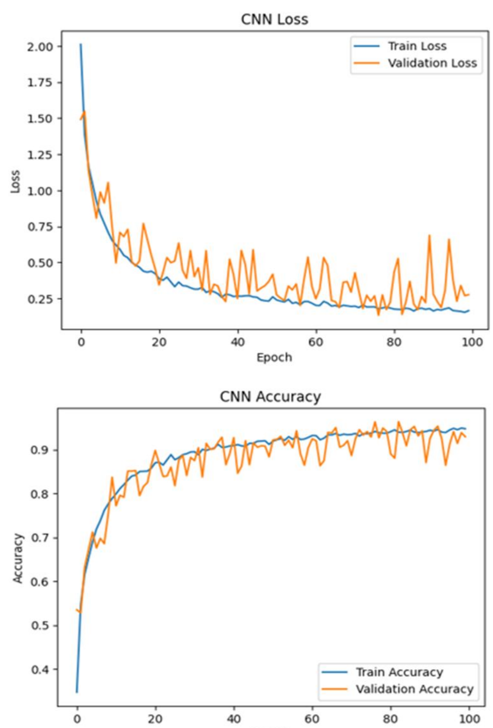

Fig. 9 Model Performance graphs

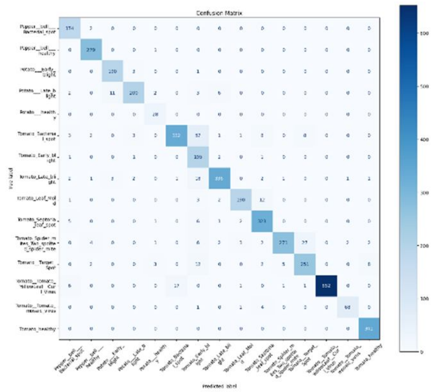

Fig. 10 Confusion matrix of Proposed Model

These findings demonstrate the model's robustness, suggesting it is suitable for real-world deployment in precision agriculture. However, further improvements can be achieved by increasing data diversity, especially with field images under natural conditions.

Table 5 Comparative Analysis Between Existing Models And Proposed

Ref.	Dataset Used	Model Type	Accuracy (%)	Validation Accuracy (%)	Loss
[25]	PlantVillage	Basic CNN	86.00	84.20	0.41
[26]	PlantVillage	LeNet-5	88.72	87.10	0.38
Proposed work	PlantVillage (CLAHE Preprocessed)	Custom CNN	94.64	92.95	0.1793

Table 5 compares the proposed custom CNN model with two currently in use CNN-based plant disease detection methods. Using the PlantVillage dataset, the current research obtained rather lower training and validation accuracies—86.00% and 84.20%, respectively. Their models depended on simple designs with no preprocessing and no complex augmentation or contrast enhancement methods. Their weak generalising capacity caused greater loss values (0.41 and 0.38), thereby indicating less effective learning. By using CLAHE (Contrast Limited Adaptive Histogram Equalisation), the Proposed Model presents several innovations: it increases picture contrast, hence boosting feature visibility and supporting improved learning by means of enhanced image contrast. Designed for efficiency, the proprietary CNN architecture combines augmentation methods including flipping, zooming, and rotation to increase generalisation and resilience. Furthermore used to reduce overfitting is dropout regularisation. The suggested model thus achieves a much smaller loss of 0.1793, 92.95% validation accuracy, and 94.64% training accuracy. These findings unequivocally show the originality and superiority of the suggested model in terms of accuracy, training stability, and real-world application, hence better fit for deployment in systems of precision agriculture.

V. CONCLUSION

This work uses a proprietary convolutional neural network (CNN) to show an efficient and accurate deep learning method for the classification of plant leaf diseases. Using the PlantVillage dataset processed by CLAHE, the model solves typical constraints in agricultural disease diagnosis like low picture contrast and model overfitting. To improve generalisation, the approach consists in important processes including image scaling, normalisation, and data augmentation methods including flipping, zooming, and rotation. Trained using the Adam optimiser, a well crafted CNN architecture including convolutional, pooling, dense, and dropout layers guarantees effective feature extraction and model stability. With precision of 91%, recall of 93%, and an F1-score of 92%, the model attained a test accuracy of 93% suggesting strong and balanced performance. With low loss values, training and validation accuracies of 94.64% and 92.95% respectively demonstrate minimum overfitting and dependable learning.

Comparatively, the suggested model shows better loss values and much outperforms other current CNN-based methods, which usually had lower accuracies between 86% and 88%. This model is unique, lightweight, and fit for implementation in smart agriculture applications including mobile and IoT-based real-time plant disease monitoring systems by means of CLAHE preprocessing, extensive augmentation, and a bespoke CNN.

REFERENCES

[1] J. Liu and X. Wang, "Plant diseases and pests detection based on deep learning : a review," Plant Methods, pp. 1–18, 2021, doi: 10.1186/s13007-021-00722-9.
[2] Y. Guo et al., "Plant Disease Identification Based on Deep Learning Algorithm in Smart Farming," Discret. Dyn. Nat. Soc., vol. 2020, 2020, doi: 10.1155/2020/2479172.
[3] K. Golhani, S. K. Balasundram, G. Vadamalai, and B. Pradhan, "A review of neural networks in plant disease detection using hyperspectral data," Inf. Process. Agric., vol. 5, no. 3, pp. 354–371, 2018, doi: 10.1016/j.inpa.2018.05.002.
[4] K. Zhang, Q. Wu, A. Liu, and X. Meng, "Can deep learning identify tomato leaf disease?," Adv. Multimed., vol. 2018, 2018, doi: 10.1155/2018/6710865.
[5] S. Prajapati, S. Qureshi, Y. Rao, S. Nadkarni, M. Retharekar, and A. Avhad, "Plant Disease Identification Using Deep Learning," 2023 4th Int. Conf. Emerg. Technol. INCET 2023, vol. 90, no. February, pp. 249–257, 2023, doi: 10.1109/INCET57972.2023.10170463.
[6] Y. M. Abd Algani, O. J. Marquez Caro, L. M. Robladillo Bravo, C. Kaur, M. S. Al Ansari, and B. Kiran Bala, "Leaf disease identification and classification using optimized deep learning," Meas. Sensors, vol. 25, no. September 2022, p. 100643, 2023, doi: 10.1016/j.measen.2022.100643.
[7] S. S. Bhoomika and K. M. Poornima, "Plant Leaf Disease Detection and Classification Using Deep Learning Technique," Lect. Notes Networks Syst., vol. 494, no. 3, pp. 73–83, 2023, doi: 10.1007/978-981-19-4863-3_7.
[8] M. Bouni, B. Hssina, K. Douzi, and S. Douzi, "Impact of Pretrained Deep Neural Networks for Tomato Leaf Disease Prediction," J. Electr. Comput. Eng., vol. 2023, 2023, doi: 10.1155/2023/5051005.
[9] B. N. Kahsay and O. D. Makinde, "Ecoepidemiological Model and Optimal Control Analysis of Tomato Yellow Leaf Curl Virus Disease in Tomato Plant," J. Appl. Math., vol. 2023, 2023, doi: 10.1155/2023/4066236.
[10] K. Academy, E. Balakannan, K. Academy, and E. Maragatharajan, "A Novel Method for Predicting Plant Leaf Disease Based on Machine Learning and Deep Learning Techniques," no. Ml, pp. 1–24, 2023.
[11] V. Balaji et al., "Deep Transfer Learning Technique for Multimodal Disease Classification in Plant Images," Contrast Media Mol. Imaging, vol. 2023, p. 5644727, 2023, doi: 10.1155/2023/5644727.
[12] B. Panigrahi, K. C. R. Kathala, and M. Sujatha, "A Machine Learning-Based Comparative Approach to Predict the Crop Yield Using Supervised Learning With Regression Models," Procedia Comput. Sci., vol. 218, no. 2022, pp. 2684–2693, 2023, doi: 10.1016/j.procs.2023.01.241.
[13] D. Paudel et al., "Machine learning for regional crop yield forecasting in Europe," F. Crop. Res., vol. 276, no. December 2021, p. 108377, 2022, doi: 10.1016/j.fcr.2021.108377.
[14] A. Tripathi, R. K. Tiwari, and S. P. Tiwari, "A deep learning multi-layer perceptron and remote sensing approach for soil health based crop yield estimation," Int. J. Appl. Earth Obs. Geoinf., vol. 113, no. April, p. 102959, 2022, doi: 10.1016/j.jag.2022.102959.
[15] K. Jhajharia, P. Mathur, S. Jain, and S. Nijhawan, "Crop Yield Prediction using Machine Learning and Deep Learning Techniques," Procedia Comput. Sci., vol. 218, pp. 406–417, 2022, doi: 10.1016/j.procs.2023.01.023.
[16] K. Joshi et al., "Precision diagnosis of tomato diseases for sustainable agriculture through deep learning approach with hybrid data augmentation," Curr. Plant Biol., vol. 41, no. August 2024, p. 100437, 2025, doi: 10.1016/j.cpb.2025.100437.
[17] S. Pradeep and J. Joseph, "Enhancing pepper growth and yield through disease identification in plants using leaf-based deep learning techniques," pp. 54–59, 2025, doi: 10.1201/9781003559115-10.
[18] H. Habaragamuwa, Y. Oishi, and K. Tanaka, "Achieving explainability for plant disease classification with disentangled variational autoencoders," Eng. Appl. Artif. Intell., vol. 133, no. PA, p. 107982, 2024, doi: 10.1016/j.engappai.2024.107982.
[19] H. H. Alshammari and A. Alzahrani, "Employing a hybrid lion-firefly algorithm for recognition and classification of olive leaf disease in Saudi Arabia," Alexandria Eng. J., vol. 84, no. October, pp. 215–226, 2023, doi: 10.1016/j.aej.2023.10.057.
[20] M. A. Bhatti et al., "Advanced Plant Disease Segmentation in Precision Agriculture using Optimal Dimensionality Reduction with Fuzzy C-Means Clustering and Deep Learning," IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens., vol. 17, pp. 18264–18277, 2024, doi: 10.1109/JSTARS.2024.3437469.
[21] W. Li, L. Zhu, and J. Liu, "PL-DINO: An Improved Transformer-Based Method for Plant Leaf Disease Detection," Agric., vol. 14, no. 5, pp. 1–14, 2024, doi: 10.3390/agriculture14050691.
[22] A. Sharma et al., "Rose Plant Disease Detection Using Image Processing and Machine Learning," Commun. Comput. Inf. Sci., vol. 2050 CCIS, pp. 69–85, 2024, doi: 10.1007/978-3-031-58953-9_6.
[23] D. Mane, M. Deore, R. Ashtagi, S. Shinde, and Y. Gurav, "Basil plant leaf disease detection using amalgam based deep learning models," J. Auton. Intell., vol. 7, no. 1, pp. 1–13, 2024, doi: 10.32629/jai.v7i1.1002.
[24] A. Ouamane et al., "Enhancing Plant Disease Detection: A Novel CNN-Based Approach with Tensor Subspace Learning and HOWSVD-MD," pp. 1–17, 2024, [Online]. Available: http://arxiv.org/abs/2405.20058
[25] A. Gautam, K. Kinjalk, A. Kumar, and V. Priye, "FBG-Based Respiration Rate Sensing with Arduino Interface," IEEE Sens. J., vol. 20, no. 16, pp. 9209–9217, Aug. 2020, doi: 10.1109/JSEN.2020.2989004.
[26] "(PDF) Plant species recognition using modified LeNet-5 CNN architecture." https://www.researchgate.net/publication/343471599_Plant_species_recognition_using_modified_LeNet-5_CNN_architecture (accessed Jun. 14, 2025).