Crop Disease Prediction System

Manoj Nath Goswami[1], Pooja Joshi[2], Rishabh Singhal[3]

Under the guidance of **Rahul Singh[4]**

Department of Computer Science Engineering, Graphic Era Hill University
Graphic Era University

ABSTRACT

Agriculture remains a foundational pillar of human society, providing essential food resources and playing a vital role in global economic development. Despite its importance, crops and foliage are frequently vulnerable to a range of diseases during their growth cycle, which can negatively impact both development and final yield. Timely and precise detection of such diseases is critical in minimizing damage and reducing agricultural losses. Historically, farmers have depended on manual inspection to detect and classify plant diseases. However, these traditional methods often suffer from inefficiencies, variability, and the potential for human error. Inaccurate or delayed diagnosis may result in substantial crop degradation and reduced output. The adoption of modern image processing methods in agriculture offers a promising path to improve the accuracy of disease detection, mitigate financial losses, and enhance crop production.

To address this challenge, a Crop Disease Prediction System has been developed, utilizing advanced deep learning technologies for plant disease identification. This system employs Convolutional Neural Networks (CNNs) within the TensorFlow framework to analyse plant images and classify potential diseases in real-time. It offers an effective tool for farmers, agronomists, and researchers to evaluate plant health swiftly and reliably. By submitting images of diseased crops, users receive predictions along with suggestions for preventive actions. CNNs, a popular choice in image recognition and classification tasks, are particularly effective when supported by ample training data and computational capacity. While these models yield high accuracy on specific datasets, their ability to generalize across diverse datasets can vary.

This research paper intends to serve as a reference for future work by presenting detailed insights into the accuracy, performance indicators, and outcomes of existing methods used for crop and leaf disease detection. It systematically reviews a variety of image-processing strategies powered by artificial intelligence (AI), analysing their advantages and constraints. The paper also delves into progress made in the fields of deep learning, computer vision, and machine learning, all of which contribute to improved precision and efficiency in plant disease diagnosis. The overarching objective is to aid ongoing research, promote the development of adaptable and efficient models, and ultimately boost agricultural productivity through accurate and timely disease recognition.

Keywords: Plant Disease, CNN, TensorFlow, Image Recognition

I. INTRODUCTION

In today's world, where agriculture is crucial to maintaining global food supplies, the creation and application of advanced technologies to protect crops have become increasingly important. Agricultural biodiversity is essential for delivering both food and raw resources, establishing its core role in the development of human civilization. Plant diseases can develop due to multiple causes, such as infections from fungi, bacteria, and nematodes, or due to unfavourable environmental conditions like imbalanced soil pH, extreme temperatures, and inconsistent moisture or humidity levels [1]. These diseases disrupt plant development and physiology, directly impacting the lives of people who depend on farming.

Despite progress in technology, many farmers still use traditional, manual techniques to detect and recognize plant diseases, especially in their initial phases. These methods can be unreliable and often contribute to a decrease in agricultural output. Given that agricultural production directly affects economic stability, it is vital to have reliable systems for early disease detection and classification to ensure sustainable growth in the farming sector [2]. Plant diseases continue to be a major challenge in modern agriculture, causing considerable crop damage and economic loss.

Evaluating how severely a plant is affected by disease is key to understanding the spread of infection, forecasting potential yield losses, and prescribing effective treatments. Quick and accurate assessment of disease severity can play a major role in reducing losses and improving productivity [3]. Traditionally, disease severity is determined through expert visual inspections, a method that is both time-consuming and expensive, often limiting its practicality in fast-paced agricultural settings [4]. With advancements in digital imaging and computer vision technologies, the development of automated systems for plant disease detection has become a necessity for precision farming, high-throughput phenotyping, and smart farming environments.

Typically, leaf tissues are the first to show signs of infection before the disease spreads, which results in deterioration of both crop quality and yield [5]. Innovations in deep learning have led to new approaches for detecting and classifying plant diseases using images of infected areas. Early detection plays a crucial role in maximizing productivity and ensuring healthy crop development [6]. However, variability in datasets presents a challenge in choosing the best image processing technique. The performance of a detection model relies heavily on factors such as dataset volume, resolution, and diversity. To achieve reliable results, deep learning models like Convolutional Neural Networks (CNNs) are often employed, as they are capable of learning detailed and complex features. Access to large, well-structured image datasets significantly boosts the precision and effectiveness of these detection systems, making them well-suited for use in agriculture [7].

II. METHODOLOGY

Various techniques can be employed for detecting plant diseases, but image classification using machine learning remains one of the most popular and effective methods. Below is a basic outline for developing a machine learning model tailored to identifying diseases in plants or crops:

1. **Data Collection and Preparation**: The initial phase involves assembling a well-rounded dataset that includes a wide range of plant images. A commonly used resource for this purpose is the New Plant Disease dataset, which comprises images of both healthy and infected plant specimens. These are categorized into 38 different classes, representing various plant species and diseases. After gathering the data, it's essential to clean and prepare it. This typically includes standardizing the image dimensions and applying techniques like rotation, flipping, or other augmentations to expand the dataset and improve model generalization.

2. **Data Preprocessing**: Once the images are collected, they must be processed to make them suitable for model training. This involves eliminating irrelevant content or visual noise from the images and refining the data to enhance clarity and quality. Steps such as normalization, contrast adjustments, and removing unwanted artifacts help ensure that the dataset is consistent and ready for use in the learning process.
3. To ensure consistent model training, pixel values in the images are often normalized, typically scaled between 0 and 1. This step helps accelerate convergence and avoids discrepancies caused by differing value ranges. If the dataset includes any textual metadata, text normalization techniques are applied to maintain uniformity. Additionally, feature extraction methods are used to identify and retain the most informative visual characteristics that assist the model in distinguishing between healthy and diseased plants.
4. Split the dataset into test dataset and training dataset respectively: After preprocessing, the dataset is split into two main subsets: training and testing datasets. The training dataset is used to train the model, allowing it to learn the patterns and features that differentiate healthy plants from diseased ones. A typical split ratio is 80% for training and 20% for testing, but this can vary depending on the total size of the dataset. The test dataset, which is kept separate from the training set, is used exclusively for evaluating the model's performance after it has been trained.

5. Transfer Learning: Transfer learning enables the reuse of an existing pre-trained model to solve a new but related problem. For plant disease classification, a Convolutional Neural Network (CNN) pre-trained on a large dataset such as ImageNet is often used. These models already understand general visual features like textures, shapes, and edges. By fine-tuning the top layers of the network, it can be adapted for the specific task of identifying plant diseases from leaf images.

6. Model Architecture:
   The architecture of the model is designed to efficiently process and classify the plant disease images. It typically consists of several convolutional layers with ReLU (Rectified Linear Unit) activation functions to extract relevant features from the images. These features are progressively abstracted through the network, allowing the model to learn increasingly complex patterns. To reduce the dimensionality of the feature maps and make the model computationally efficient, max-pooling layers are inserted after the convolutional layers. Dropout layers are also used to prevent overfitting, ensuring that the model doesn't memorize the training data but rather generalizes well to new, unseen images. Finally, the model ends with fully connected dense layers, which output the classification result for the plant's disease or health status, using a softmax activation function to predict the probability of each possible disease class.

7. Model Training : Training the model involves feeding the prepared training data into the network and adjusting the model's weight based on the difference between the predicted outputs and the actual labels. The goal is to minimize the loss function, which measures this error, and optimize the model's performance. During training, various techniques such as batch normalization and data augmentation may be applied to improve model stability and performance.

8. Model Evaluation: After the model is trained, its performance must be evaluated using the test dataset. The evaluation phase provides insights into how well the model generalizes to new, unseen data. Key evaluation metrics include accuracy, which indicates the percentage of correct predictions, and loss, which quantifies how far off the predictions are from the actual values. Additionally, a confusion matrix is generated to show how the model performs across different disease classes, highlighting which diseases are misclassified more frequently.

9. Deployment: Once the model has been trained and evaluated, it can be deployed in real-world applications. Deployment involves integrating the trained model into a frontend application, where

users can upload images of plants for disease detection. This application could be a mobile app or a desktop tool used by farmers to diagnose plant diseases in the field. The model can be deployed on a cloud platform or locally, depending on the application's requirements.

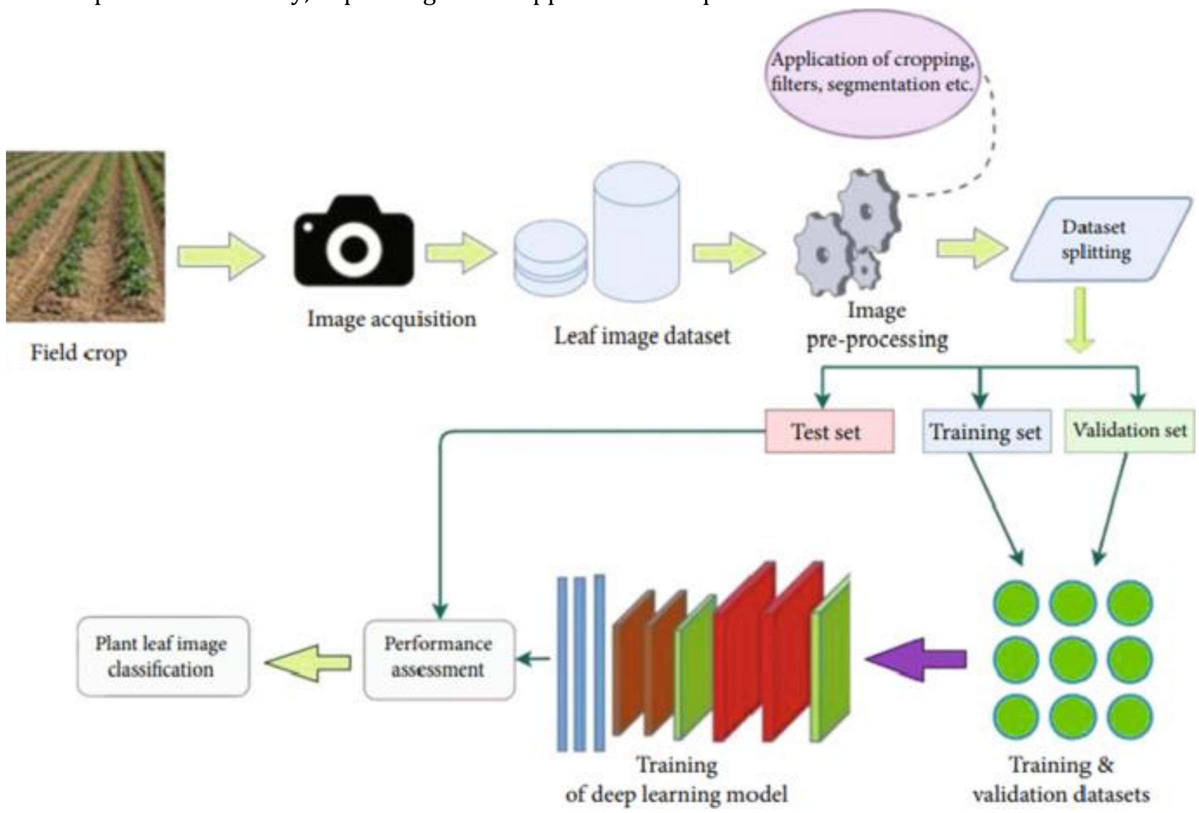


FIG.1

It's crucial to remember that the technique is only a general outline, and the specifics may change based on the demands of the unique application.

III. RELATED WORK

Early identification of plant diseases is essential for effective prevention and control, enabling farmers to implement timely interventions that minimize crop loss. This section highlights recent advancements in plant disease detection technologies. For instance, one study [8] applied a Support Vector Machine (SVM) model using hyperspectral imagery to detect diseases in sugar crops, resulting in a classification accuracy of 78% on the testing dataset. Another investigation [9] developed two diagnostic models for detecting huanglongbing disease in citrus plants—one using SVM and the other based on Artificial Neural Networks (ANN). These models achieved classification accuracies of 92.8% and 92.2%, respectively.

Further, a separate study [10] focused on the detection of tomato yellow leaf curl disease using an SVM with a quadratic kernel, yielding a classification accuracy of 92%. In [11], researchers examined and evaluated various image processing techniques and feature extraction methods for identifying plant diseases through leaf imagery. Another contribution [12] presented a Deep Convolutional Neural Network (DCNN) that identifies legume plant species by analysing vein morphology patterns. In [13], a detection system combining a region-based CNN with a single-shot multi-box detector was introduced for identifying plant diseases and pests. Moreover, Table 1 summarizes and contrasts the performance of several state-of-the-art DCNN models employed in different research efforts for plant disease classification.

Table 1. Comparison of different DCNN architecture.

Article	Year	Specie	Number of Classes	Number of Images	Architecture	Accuracy (%)
[22]	2017	Maize	2	1796	Custom	96.7
[23]	2017	Wheat	3	3500	Custom	81.04
[24]	2015	Cucumber	3	800	Custom	94.9
[25]	2016	Apple	5	1450	AlexNet	97.3
[26]	2018	Tomato	7	13,262	VGG16Net	97.29
[27]	2018	Maize	9	3060	GoogLeNet	98.9
[28]	2017	Tomato	9	14,828	GoogLeNet	99.18
[29]	2017	Rice	10	500	AlexNet	95.48
[2]	2016	Multiple	15	4483	CaffeNet	96.3
[30]	2016	Multiple	38	54,306	GoogLeNet	99.35
[7]	2018	Multiple	38	54,323	InceptionV3Net	99.76
[3]	2019	Multiple	39	61,486	Custom	96.46
[8]	2018	Multiple	58	87,848	VGG16Net	99.53
[31]	2019	Multiple	79	46,409	GoogLeNet	86.5
[15]	2020	Tomato	10	18,160	Custom	98.7
[14]	2021	Tomato	10	3000	Custom	98.49
[5]	2021	Tomato	10	18,345	AlexNet	98.0
[1]	2022	Multiple	38	240,000	Custom	98.41

Data augmentation plays a key role in increasing the diversity of training datasets without requiring new data acquisition. In [14], researchers investigated various augmentation methods used to train Deep Convolutional Neural Networks (DCNNs). Techniques including Generative Adversarial Networks (GANs), image flipping, cropping, shifting, Principal Component Analysis (PCA), colour adjustment, noise injection, and rotation were assessed. The study concluded that cropping, flipping, GANs, and rotation consistently led to better training outcomes. It was also found that using a combination of augmentation techniques yielded more robust results compared to applying them in isolation.

In [15], augmentation methods such as BIM, GANs, and Neural Style Transfer (NST) were applied to plant leaf disease classification. The findings demonstrated that integrated augmentation strategies significantly enhanced classification performance over individual methods. Similarly, the work in [16] emphasized the advantages of applying established augmentation techniques in deep learning workflows. In [17], a DCNN-based pest detection framework was introduced, utilizing datasets enhanced through GAN-based augmentation, which showed improved classification accuracy over non-augmented datasets. Further, [18] highlighted the positive impact of GAN-generated data in enhancing DCNN training, while also stressing the critical role of tuning hyperparameters to optimize model accuracy.

Collectively, these studies reinforce the importance of large and diverse datasets, well-designed augmentation strategies, and careful hyperparameter tuning in building effective plant disease detection systems. The following section discusses the proposed dataset and the DCNN model architecture developed for identifying plant diseases from leaf images.

IV. DISEASE SEVERITY ESTIMATION

Although significant progress has been made in disease identification, the estimation of disease severity has not received equivalent attention. Conventionally, severity assessment relies on visual rating scales and standardized area diagrams, which often suffer from subjectivity and inconsistencies due to human interpretation (Bock et al., 2010).

Assessing disease severity is vital, as it plays a key role in determining the timing, extent, and allocation of treatment resources. The standard metrics used for evaluating plant diseases are disease incidence and disease severity. Disease incidence is the percentage of infected plants among the total number surveyed, whereas disease severity indicates the percentage of plant tissue affected by the disease relative to the total tissue area (Bock et al., 2010).

Recent developments have aimed to automate severity estimation through image-based analysis. For example, Wang et al. (2017) utilized image segmentation to estimate lesion coverage in wheat, achieving a strong correlation coefficient of 0.88 with expert evaluations. In another study, Picon et al. (2019) employed deep learning with pixel-level segmentation for wheat disease assessment, reporting a mean absolute error of 8.7% compared to human ratings.

Mahlein (2016) explored the use of spectral vegetation indices to evaluate crop diseases across different growth stages using hyperspectral imaging. Liang et al. (2019) introduced a multi-task learning model that conducted both disease classification and severity estimation, reaching an accuracy of 87.9% in evaluating rice disease severity.

Nonetheless, many current solutions treat severity assessment as a supplementary task instead of integrating it as a core objective. Additionally, numerous methods are tailored to specific crop-disease pairs, limiting their adaptability to broader agricultural scenarios.

Severity Assessment Using Classification and Quantitative Metrics
Classification-Approach:

A dedicated branch of the convolutional neural network (CNN) was trained to categorize disease severity into three distinct classes, determined by expert-defined criteria:

- **Low Severity**: Less than 10% of plant area affected, with sparse and isolated lesion spots.
- **Moderate Severity**: Between 10% and 30% tissue damage, typically with more widespread and connected lesions.
- **High Severity**: More than 30% of tissue showing signs of damage, with significant lesion coverage and tissue destruction.

This branch utilizes the same foundational CNN architecture as the primary disease detection model but features an altered output layer with three neurons, each representing one severity category. To enhance efficiency, this severity estimation module shares the initial convolutional layers with the disease detection model, enabling a multi-task learning setup that leverages common feature representations.

$f(x)=max(0,x)$

ReLU introduces non-linearity into the model while mitigating issues related to vanishing gradients, making it particularly effective in deep networks.

Quantitative Assessment: Disease Severity Index (DSI)

To complement the classification-based method with a more precise evaluation, a **Disease Severity Index (DSI)** was implemented. This metric calculates the percentage of diseased tissue within an image through the following computational process:

1. **Segmentation** of plant regions to exclude background noise.
2. **Extraction of features** such as colour and texture to detect affected areas.
3. **Pixel-wise analysis** to determine the ratio of diseased to healthy plant tissue.
4. **Scaling** this ratio to a standardized percentage range (0%–100%).

The DSI is expressed mathematically as:

DSI = (Damaged Plant Area / Total Plant Area) × 100%

This formula provides an objective measure of how much of the plant is affected by disease, offering an alternative to the categorical classification system.

Disease Response Categories Based on Incidence Rates

Alongside severity metrics, plant disease responses were also categorized based on incidence levels to further support agronomic decision-making. The classification is as follows:

- **Free (F)**: No incidence observed
- **Resistant (R)**: 0.0% – 5.0% incidence
- **Moderately Resistant (MR)**: 5.1% – 15.0%
- **Moderately Susceptible (MS)**: 15.1% – 30.0%
- **Susceptible (S)**: 30.1% – 50.0%
- **Highly Susceptible (HS)**: Greater than 50.0%

IV. MATERIALS AND METHODS

This section presents the design and training methodology of the proposed Deep Convolutional Neural Network (DCNN) model, along with a description of the experimental setup and data preparation process. The plant leaf disease identification framework is structured into several key stages, beginning with dataset acquisition and ending with the final prediction phase. The model development was implemented using Python 3.7, with libraries such as TensorFlow 2.9.1, NumPy 1.19.2, Matplotlib 3.5.2, and OpenCV 4.5.5 for handling data processing and building the neural network.

All activities involving dataset handling, preprocessing, architecture design, and model inference were executed on an HP Z240 workstation featuring an Intel Core i7 CPU and 16GB RAM. For high-performance model training and comparative analysis with other state-of-the-art approaches, an NVIDIA DGX-1 deep learning system was utilized. This server hosts dual Intel Xeon E5-2694 v4 CPUs and eight NVIDIA Tesla P100 GPUs, providing substantial computational power for training deep networks efficiently.

The following subsections elaborate on each phase involved in the plant disease classification pipeline. The upcoming section focuses specifically on dataset preparation and preprocessing techniques.

Dataset Preparation and Preprocessing:

Leaf images representing both healthy and diseased conditions were collected from various publicly accessible sources [19]. The dataset spans two plant species, with each containing categories for both healthy foliage and frequently occurring diseases. Altogether, the dataset includes 38 unique leaf classes along with one extra category for non-leaf visuals. Initially, the collection featured around 87,000 images, encompassing both leaf and non-leaf types.

To achieve a balanced distribution across all classes and enhance model generalization, several data augmentation strategies were applied. These augmentation techniques helped increase the dataset's size and reduce the risk of overfitting by introducing diverse transformations in the training images. Augmented data was generated using methods like BIM (Basic Image Manipulation), DCGAN (Deep Convolutional Generative Adversarial Networks), and NST (Neural Style Transfer). The BIM method employed a variety of image transformations such as flipping, cropping, rotating, scaling, and Principal Component Analysis (PCA)-based color modification. PCA-based color augmentation adjusted pixel color intensities by leveraging the principal components of image color data. Other transformations introduced both spatial and color variations. Using BIM alone, the dataset was enriched with an additional 36,541 augmented images.

Model Design:

The proposed 14-DCNN model consists of five convolutional layers and five max-pooling layers. The input images are fed into the first two-dimensional Conv layer, and the output dimensions are determined using Equation (1):

$$Dimension\,(\mathrm{Conv}\,(n,k)) = \left(\left[\frac{n_w - f_w}{s} + 1\right], \left[\frac{n_h - f_h}{s} + 1\right], f_C\right) \tag{1}$$

Where nw and nh represent the width and height of the input image (128 × 128). Additionally, fw,fh and fc and denote the width, height, and channels of the convolutional kernel. The stride value () for this Conv layer is set to 1.

Convolutional and Pooling Layers

1. **First Conv and Pooling Layer:**
    - Input size: (128, 128, 4)
    - Kernel size: (3, 3, 4)
    - Output size: (126, 126, 4)
    - Max-pooling reduces it to: (63, 63, 4)
2. **Second Conv and Pooling Layer:**
    - Kernel size: (3, 3, 16)
    - Output size: (61, 61, 16)
    - Max-pooling reduces it to: (30, 30, 16)
3. **Third Conv and Pooling Layer:**
    - Kernel size: (3, 3, 32)
    - Output size: (28, 28, 32)
    - Max-pooling reduces it to: (14, 14, 32)
4. **Fourth Conv and Pooling Layer:**
    - Kernel size: (3, 3, 64)
    - Output size: (12, 12, 64)
    - Max-pooling reduces it to: (6, 6, 64)
5. **Fifth Conv and Pooling Layer:**
    - Kernel size: (3, 3, 128)
    - Output size: (4, 4, 128)
    - Max-pooling reduces it to: (2, 2, 128)

Activation Function The ReLU activation function is applied to all convolutional layers using Equation (3):

$$ReLu\,(x) = Max\,(0,\ x)$$

Fully Connected Layers After the final Conv and pooling layers, a flatten layer converts the three-dimensional data (2, 2, 128) into one-dimensional data (512). This is followed by a dense layer:

- **First Dense Layer:**
    - Input: 512
    - Output: 2048
    - Activation: ReLU
    - Equation (4):

$$z_j = ReLu\left(0,\ \sum_{i}^{512} b_j + x_i w_i\right)$$

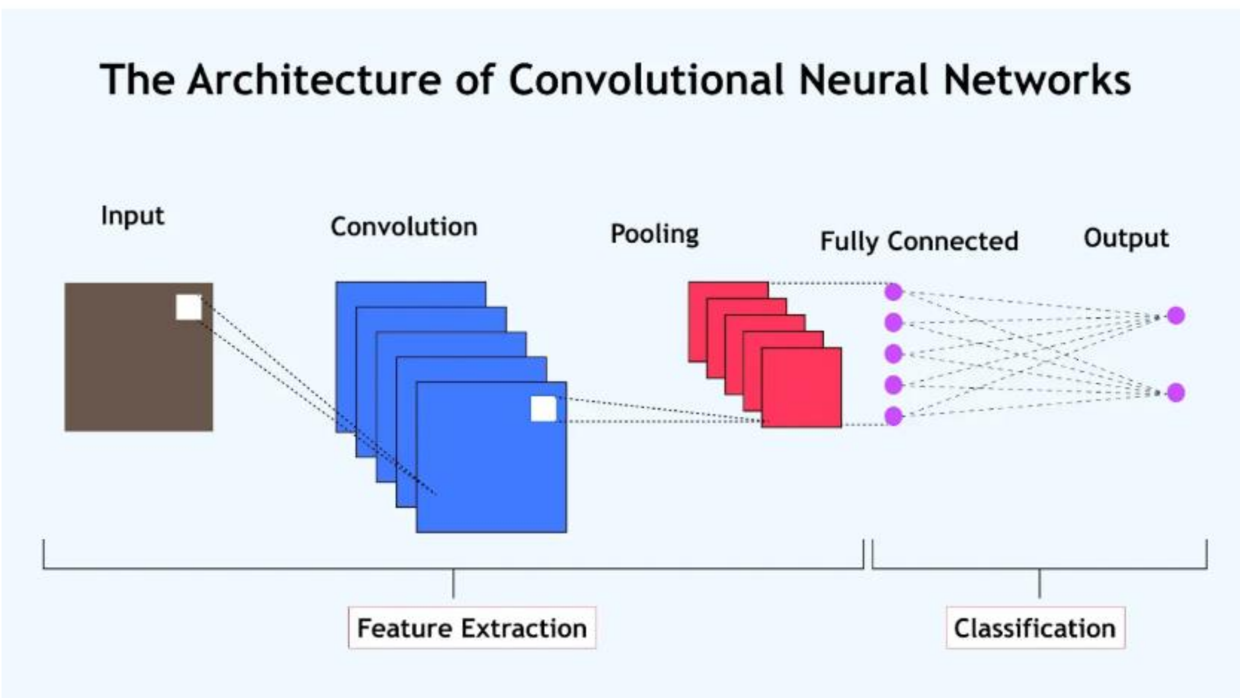
- **Dropout Layer:**
    - Used to prevent overfitting.
- **Second Dense Layer:**
    - Input: 2048
    - Output: 59 (corresponding to the 59 classes in the PlantDisease59 dataset)
    - Activation: Softmax
    - Equation (5):

$$softmax\left(\sigma\left(z_i\right)\right) = \frac{e^{z_i}}{\sum_{j=1}^{59} e^{z_j}}$$

- **Output Class:**
    - Determined using Equation (6):

$$Output\ Class\ \left(z_{out}\right) = max\left(z_1,\ z_2, \dots\ z_{59}\right)$$

Training Parameters The total number of training parameters in the 14-DCNN model is **5,424,583**.

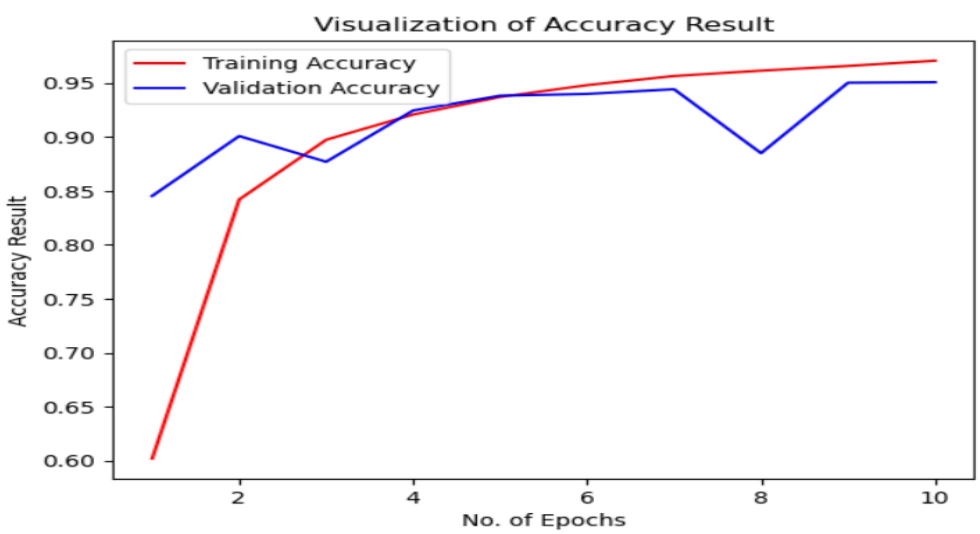


VI. MODEL TRAINING

The proposed 14-DCNN model was trained using optimized hyperparameters and an augmented dataset within a deep learning server environment. The training was conducted with various epoch values ranging from 1 to 10, with the highest validation accuracy and the lowest loss achieved at 10 epochs. The model took 7452 seconds to train with 10 epochs on the Nvidia DGX-1 deep learning server. Figure 3 illustrates the training and validation performance of the proposed 14-DCNN model for plant leaf disease identification.

The 14-DCNN model achieved an impressive training accuracy of 99.993% and a validation accuracy of 99.985%. These accuracy results surpass those of other DCNN models. The total training time of 7452 seconds for the 14-DCNN was shorter than that of transfer learning techniques, as the number of convolutional and pooling operations in the 14-DCNN is smaller compared to transfer learning models. Additionally, the architecture and trained weights of the 14-DCNN model were saved as a hierarchical data (H5) file, enabling future prediction tasks.

The model's performance was evaluated using both the training and validation datasets, yielding a training accuracy of 99.993% and a validation accuracy of 99.985%. To further assess classification performance, a confusion matrix and classification report were generated and visualized as a heatmap, showcasing the model's effectiveness in plant disease identification.

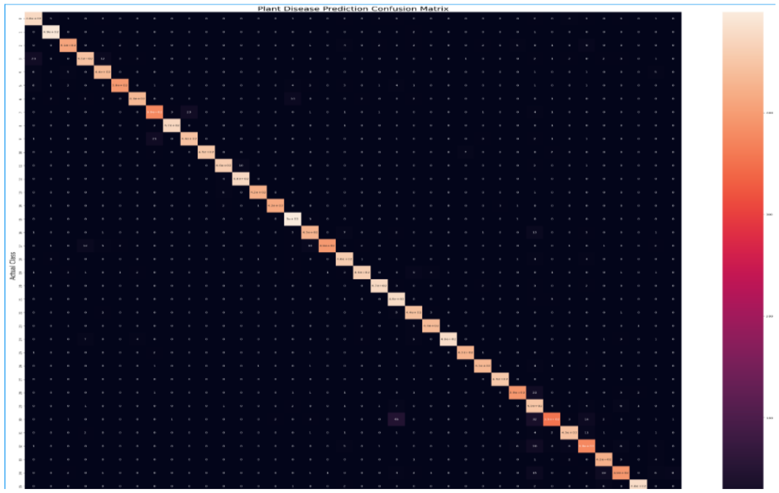


VII. MODEL PREDICTION

The trained 14-DCNN model architecture and weights were utilized to detect plant diseases from input images. Real-time plant disease images were fed into the 14-DCNN model, which successfully identified both the plant species and the associated disease. The predictions were then visualized using the Matplotlib library for better clarity.

During the testing phase, a sample plant image was loaded and pre-processed using the OpenCV library. The image was resized to 128x128 pixels to meet the model's input size requirements. The image was then passed through the trained 14-DCNN model to generate predictions. The predicted disease name was displayed on the image, accompanied by a title indicating the detected disease.

To further enhance its applicability, the 14-DCNN model was converted into a TensorFlow Lite (tflite) format using the TensorFlow Lite converter, employing a latency optimization approach. This optimized tflite file can be deployed on mobile and embedded devices, enabling real-time predictions on resource-constrained devices and broadening the model's utility in practical, on-the-go applications.



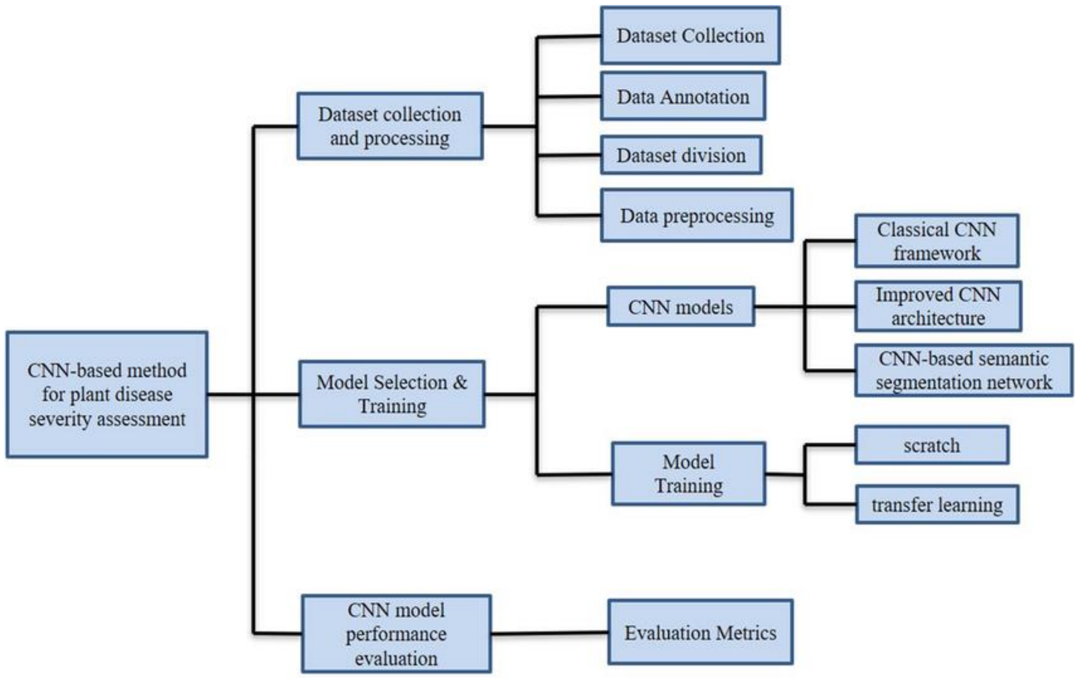
VIII. CONCLUSION

In this study, a novel deep convolutional neural network (DCNN) model was developed to identify plant leaf diseases from images. The 14-DCNN model was specifically trained to recognize and classify a wide range of plant diseases, leveraging real-time predictions from input images. To enhance the dataset for training and validation, data augmentation techniques were applied, expanding the dataset to a total of 147,500 images. This was achieved by incorporating three augmentation methods: NST, DCGAN, and BIM. Each class, containing both original and augmented images, included 2,500 images.

The 14-DCNN architecture consists of five convolutional layers and five max-pooling layers. Hyperparameter optimization was conducted using a random search approach with a coarse-to-fine technique to identify the optimal configuration for the model. The model was trained using a large dataset, comprising 139,000 images for both training and validation. After training, the model demonstrated outstanding performance metrics, achieving a classification accuracy of 99.985%, along with excellent precision (99.7999%), recall (99.7966%), and an F1 score of 99.7968% on the training set.

The trained 14-DCNN model successfully predicted plant diseases from leaf images, with predictions visualized using the matplotlib library, displaying both the model's output and the identified plant disease name. To facilitate deployment on mobile and embedded devices for real-time predictions, the model was converted to TensorFlow Lite (tflite) format using TensorFlow's conversion tools.

Looking ahead, the research will be extended by increasing the dataset with new plant disease classes and more images. Future work will also focus on enhancing the DCNN architecture by adding additional convolutional layers to further improve performance. Moreover, there are plans to expand the model's functionality to estimate disease severity and detect diseases in other plant parts, such as flowers, fruits, and stems, using similar deep learning techniques.



IX. REFERENCES

[1] Tirkey D, Singh KK, Tripathi S. Performance analysis of AI-based solutions for crop disease identification detection, and classification. Smart Agric Technol. 2023. https://doi.org/10.1016/j.atech.2023.100238.

[2] Ramanjot, et al. Plant disease detection and classification: a systematic literature review". Sensors. 2023. https://doi.org/10.3390/s23104769.

[3] Bock C. H., Poole G. H., Parker P. E., and Gottwald T. R., Plant disease severity estimated visually, by digital photography and image analysis, and by hyperspectral imaging, *Critical Reviews in Plant Sciences*. (2010) **29**, no. 2, 59–107, 2-s2.0-77950942373, **https://doi.org/10.1080/07352681003617285**.

[4] Heltin Genitha C, Dhinesh E, Jagan A. Detection of leaf disease using principal component analysis and linear support vector machine. Adv Comput ICoAC Proc Int Conf. 2019. https://doi.org/10.1109/ICoAC48765.2019.246866.

[5] P. S. Gupta, P. Hans, and S. Chand. 2020. Classification Of Plant Leaf Diseases Using Machine Learning And Image Preprocessing Techniques.

[6] Sardogan M, Tuncer A, Ozen Y. Plant leaf disease detection and classification based on CNN with LVQ algorithm. Comput Sci Eng Conf. 2018. https://doi.org/10.1109/UBMK.2018.8566635.

[7] Uchida S, Ide S, Iwana BK, Zhu A. A further step to perfect accuracy by training CNN with larger data. Int Conf Front Handwrit Recognit. 2016. https://doi.org/10.1109/ICFHR.2016.0082.

[8] Rumpf, T.; Mahlein, A.-K.; Steiner, U.; Oerke, E.-C.; Dehne, H.-W.; Plümer, L. Early detection and classification of plant diseases with Support Vector Machines based on hyperspectral reflectance. *Comput. Electron. Agric.* **2010**, *74*, 91–99. [Google Scholar] [CrossRef]

[9] Wetterich, C.B.; Neves, R.F.D.O.; Belasque, J.; Ehsani, R.; Marcassa, L.G. Detection of Huanglongbing in Florida using fluorescence imaging spectroscopy and machine-learning methods. *Appl. Opt.* **2017**, *56*, 15–23.

[10] Mokhtar, U.; Ali, M.A.; Hassanien, A.E.; Hefny, H. Identifying Two of Tomatoes Leaf Viruses Using Support Vector Machine. In *Information Systems Design and Intelligent Applications*; Mandal, J.K., Satapathy, S.C., Sanyal, M.K., Sarkar, P.P., Mukhopadhyay, A., Eds.; Springer India: New Delhi, India, 2015; pp. 771–782. [Google Scholar]

[11] Bharate, A.A.; Shirdhonkar, M.S. A review on plant disease detection using image processing. In Proceedings of the 2017 International Conference on Intelligent Sustainable Systems (ICISS), Palladam, India, 7–8 December 2017; pp. 103–109. [Google Scholar]

[12] Grinblat, G.L.; Uzal, L.C.; Larese, M.G.; Granitto, P.M. Deep learning for plant identification using vein morphological patterns. *Comput. Electron. Agric.* **2016**, *127*, 418–424. [Google Scholar] [CrossRef] [Green Version]

[13] Fuentes, A.; Yoon, S.; Kim, S.C.; Park, D.S. A Robust Deep-Learning-Based Detector for Real-Time Tomato Plant Diseases and Pests Recognition. *Sensors* **2017**, *17*, 2022. [Google Scholar] [CrossRef] [PubMed] [Green Version]

[14] Shijie, J.; Ping, W.; Peiyi, J.; Siping, H. Research on data augmentation for image classification based on convolution neural networks. In Proceedings of the 2017 Chinese Automation Congress (CAC), Jinan, China, 20–22 October 2017; pp. 4165–4170. [Google Scholar]

[15] Pandian, J.A.; Geetharamani, G.; Annette, B. Data augmentation on plant leaf disease image dataset using image manipulation and deep learning techniques. In Proceedings of the 2019 IEEE 9th International Conference on Advanced Computing (IACC), Tiruchirappalli, India, 13–14 December 2019; pp. 199–204. [Google Scholar]

[16] Shorten, C.; Khoshgoftaar, T.M. A survey on Image Data Augmentation for Deep Learning. *J. Big Data* **2019**, *6*, 60. [Google Scholar] [CrossRef]

[17] Lu, C.-Y.; Rustia, D.J.A.; Lin, T.-T. Generative Adversarial Network Based Image Augmentation for Insect Pest Classification Enhancement. *IFAC-PapersOnLine* **2019**, *52*, 1–5. [Google Scholar] [CrossRef]

[18] Bang, S.; Baek, F.; Park, S.; Kim, W.; Kim, H. Image augmentation to improve construction resource detection using generative adversarial networks, cut-and-paste and image transformation techniques. *Autom. Constr.* **2020**, *115*, 103198. [Google Scholar] [CrossRef]

[19] https://www.kaggle.com/datasets/vipoooool/new-plant-diseases-dataset