IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

AI Based Plant Disease Detection And Recommendation Using Deep Learning Techniques Inceptionv3 And Mobilenetv2

Author:
[1]Guduru Ramakrishna Reddy,
M.Tech. Scholar,
Department of Computer Science,
School of Engineering,
Malla Reddy University,
Hyderabad, Telangana, India

Under the esteemed guidance of:
[2]Dr. G. Jose Moses,
Professor,
Department of Computer Science,
School of Engineering,
Malla Reddy University,
Hyderabad, Telangana, India

Abstract: Agriculture is a primary occupation in India, and it loses 35% of productivity due to plant diseases. Plants play a crucial role in the agriculture sector, exerting a profound influence on a nation's economy and environmental equilibrium. Similar to human health, plants can be affected by diseases caused by pathogens like viruses and bacteria. Effective plant care necessitates diligent observation, accurate disease identification, and appropriate management strategies. Early detection and prevention techniques for plant diseases are highly beneficial to recover plants quickly and maintain their productivity. Currently, the methods employed for identifying plant diseases involve either human-based or lab-based techniques. This research paper presents an artificial intelligence (AI) model that identifies plant diseases, provides explanation of detected disease, and suggests appropriate recommendation. The project "AI based Plant Disease Detection and Recommendation using Deep Learning" is to improve agricultural output by offering an effective method for detecting plant diseases and actionable recommendation. This platform offers two main modules aimed to enhance productivity and sustainability.

This platform is developed using python for backend processing, flask as the web framework, and html, CSS, and JavaScript for the frontend. This system leverages state-of-the-art deep learning techniques to detect and classify plant diseases with high accuracy using plant leaves. InceptionV3 and MobileNetV2 are two distinct deep learning architectures, employed for image classification and object detection tasks. InceptionV3 is for its high accuracy, while MobileNetV2 is for its efficiency and suitability for resource-constrained environments like mobile devices. InceptionV3 achieved a training accuracy of 93.88% and validation accuracy of 90.05%, while MobileNetV2 outperformed with a training accuracy of 98.30% and validation accuracy of 94.38%. The dataset includes 70,000+ images covering 14 plants including apple, blueberry, cherry, corn (maize), grape, orange, peach, pepper, potato, raspberry, soybean, squash, strawberry, and tomato and classified into 38 distinct classes, including both healthy and diseased categories. This project is a highly valuable contribution to farmers and agricultural users for an efficient, user-friendly, and comprehensive tool for plant disease management, thereby encouraging sustainable farming practices.

Key Words – Artificial Intelligence, Machine Learning, Deep Learning, Plant Disease Detection, Convolutional Neural Network (CNN), Fertilizer Recommendation.

1. INTRODUCTION

1.1. Introduction to Plant Disease

Plant diseases are major challenge in agriculture, ultimately impacting crop yield and quality worldwide. Leaves are vital organs of plants, responsible for photosynthesis and overall plant health. When leaves are affected by diseases, it can disrupt these processes, leading to stunted growth, reduced productivity, and, in severe cases, total crop failure. Plant leaf diseases are caused by various pathogens, including fungi, bacteria, viruses, and pests, as well as non-pathogenic factors like nutritional deficiencies and adverse environmental conditions. These plant diseases exhibit a range of visible symptoms such as spots, discoloration, blights, mildew, lesions, and wilting, which differ based on the plant and pathogen. The traditional approach to managing plant diseases involves manual inspection by agricultural experts, which is time-consuming, labor-intensive, and prone to human error, in large-scale farming. Lack of timely intervention often results in the spread of diseases, causing economic losses and threatening food security. With the advancement of technology, automated plant disease detection systems using machine learning have emerged as effective solutions. These systems leverage image-based analysis to identify and classify diseases in plant leaves with high accuracy, enabling early detection and timely management. By integrating plant disease detection with actionable insights such as fertilizer or pesticide recommendations, these systems help optimize resource utilization, reduce chemical use, and improve agricultural productivity. In this context, the study and development of efficient systems for plant leaf disease detection have gained significant importance, contributing to sustainable agriculture and addressing the global demand for food security.

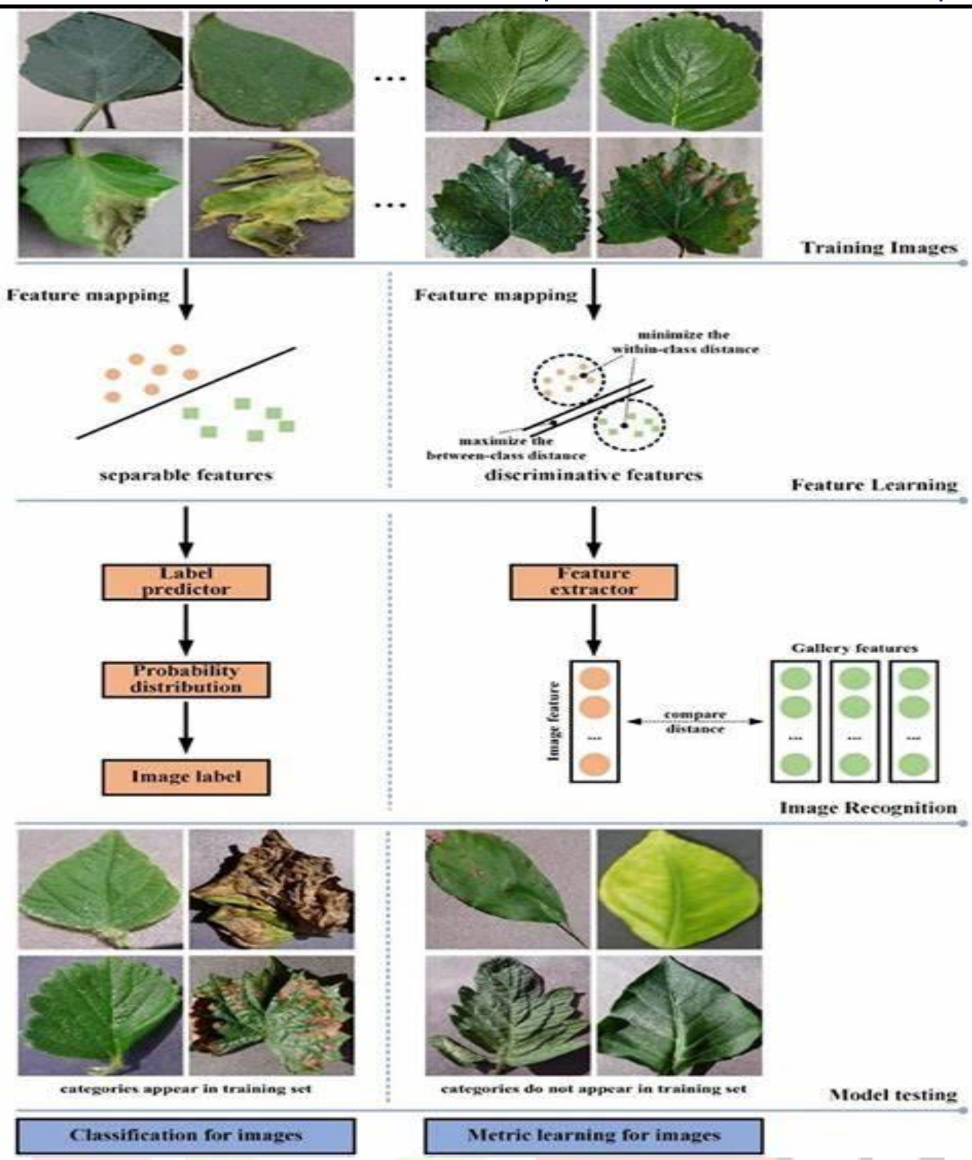

Figure: Image based Plant Disease Classification and Detection

1.2. Deep Learning

Deep learning allows computational models that are composed of multiple processing layers to learn representations of data with multiple levels of abstraction. These methods have dramatically improved the state-of-the-art in speech recognition, visual object recognition, object detection and many other domains such as drug discovery and genomics. Deep learning discovers intricate structure in large data sets by using the back propagation algorithm to indicate how a machine should change its internal parameters that are used to compute the representation in each layer from the representation in the previous layer. Deep convolutional nets have brought about breakthroughs in processing images, video, speech and audio, whereas recurrent nets have shone light on sequential data such as text and speech.

Applications of Deep Learning:

Deep learning has made significant contributions to a wide range of domains, driving transformative advancements in areas such as computer vision, natural language processing, speech recognition, and robotics. In computer vision, deep learning models have achieved remarkable performance in image classification, object detection, and image segmentation tasks. In natural language processing, deep learning has revolutionized machine translation, sentiment analysis, text generation, and question-answering systems. Deep learning also plays a crucial role in speech recognition systems, enabling accurate voice-controlled interfaces and transcription services. Furthermore, deep learning techniques have been instrumental in enhancing autonomous systems, such as self-driving cars and robotic control.

1.2.1. Supervised Learning:

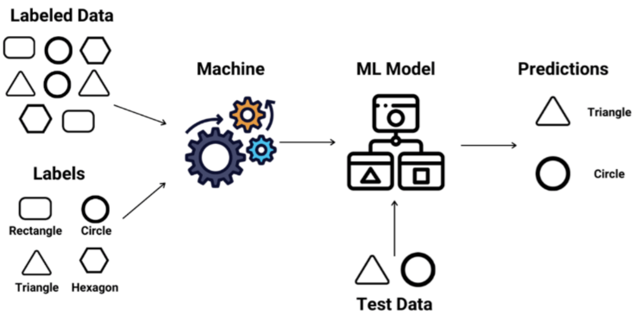

Figure: Basic Structure of Supervised Learning

The most common form of machine learning, deep or not, is supervised learning. Imagine that we want to build a system that can classify images as containing, say, a house, a car, a person or a pet. We first collect a large data set of images of houses, cars, people and pets, each labelled with its category.

1.2.2. *Unsupervised Learning*

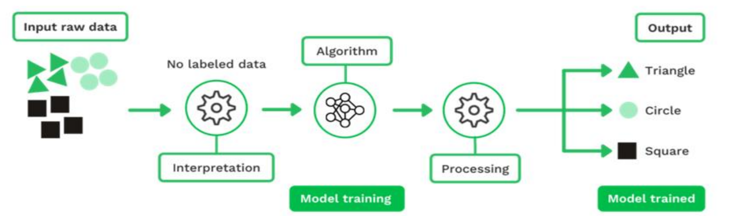

Figure: Basic Structure of Supervised Learning

In unsupervised learning, an algorithm explores input data without being given an explicit output variable (e.g., explores customer demographic data to identify patterns). You can use it when you do not know how to classify the data, and you want the algorithm to find patterns and classify the data for you.

1.2.3. Convolutional neural networks

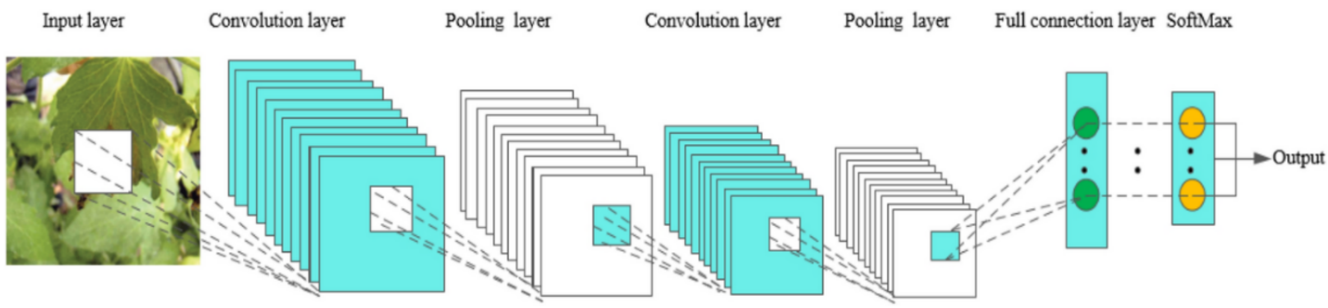

Figure: The basic structure of CNN

ConvNets are designed to process data that come in the form of multiple arrays, for example a colour image composed of three 2D arrays containing pixel intensities in the three colour channels. Many data modalities are in the form of multiple arrays: 1D for signals and sequences, including language; 2D for images or audio spectrograms; and 3D for video or volumetric images. There are four key ideas behind

ConvNets that take advantage of the properties of natural signals: local connections, shared weights, pooling and the use of many layers. The architecture of a typical ConvNet is structured as a series of stages. The first few stages are composed of two types of layers: convolutional layers and pooling layers. Units in a convolutional layer are organized in feature maps, within which each unit is connected to local patches in the feature maps of the previous layer through a set of weights called a filter bank. The result of this local weighted sum is then passed through a non-linearity such as a ReLU. All units in a feature map share the same filter bank. Different feature maps in a layer use different filter banks. The reason for this architecture is twofold. First, in array data such as images, local groups of values are often highly correlated, forming distinctive local motifs that are easily detected. Second, the local statistics of images and other signals are invariant to location. In other words, if a motif can appear in one part of the image, it could appear anywhere, hence the idea of units at different locations sharing the same weights and detecting the same pattern in different parts of the array. Mathematically, the filtering operation performed by a feature map is a discrete convolution, hence the name. Although the role of the convolutional layer is to detect local conjunctions of features from the previous layer, the role of the pooling layer is to merge semantically similar features into one. Because the relative positions of the features forming a motif can vary somewhat, reliably detecting the motif can be done by coarse-graining the position of each feature. A typical pooling unit computes the maximum of a local patch of units in one feature map (or in a few feature maps). Neighbouring pooling units take input from patches that are shifted by more than one row or column, thereby reducing the dimension of the representation and creating an invariance to small shifts and distortions. Two or three stages of convolution, non-linearity and pooling are stacked, followed by more convolutional and fully-connected layers. Backpropagating gradients through a ConvNet is as simple as through a regular deep network, allowing all the weights in all the filter banks to be trained. Deep neural networks exploit the property that many natural signals are compositional hierarchies, in which higher-level features are obtained by composing lower-level ones. In images, local combinations of edges form motifs, motifs assemble into parts, and parts form objects. Similar hierarchies exist in speech and text from sounds to phones, phonemes, syllables, words and sentences. The pooling allows representations to vary very little when elements in the previous layer vary in position and appearance. The convolutional and pooling layers in ConvNets are directly inspired by the classic notions of simple cells and complex cells in visual neuroscience, and the overall architecture is reminiscent of the LGN–V1–V2–V4–IT hierarchy in the visual cortex ventral pathway. When ConvNet models and monkeys are shown the same picture, the activations of high-level units in the ConvNet explains half of the variance of random sets of 160 neurons in the monkey’s inferotemporal cortex. ConvNets have their roots in the neocognitron46, the architecture of which was somewhat similar, but did not have an end-to-end supervised-learning algorithm such as backpropagation. A primitive 1D ConvNet called a time-delay neural net was used for the recognition of phonemes and simple words.

2. LITERATURE SURVEY

2.1. Image-based disease diagnosing and predicting of the crops through the deep learning mechanism [1]

AUTHORS: H. Park, J.-S. Eun, and S.-H. Kim

The crop productivity depends on environmental factors or product resources, such as temperature, humidity, labor and electrical costs. However, above all, crop disease is the crucial factor and causes 20-30% reduction of the productivity in case of its infection. Thus, the disease of the crop is the important factor affecting the productivity of the crops. Therefore, the farmer concentrates on the cause of the disease in the crops during its growth, but it is not easy to recognize the disease on the spot. Until now, they just relied on the opinion of the experts or their own experiences when the disease is doubtful. However, it triggers a decrease in productivity as no taking appropriate action and time. In this paper, to address this problem the mechanism, which dynamically analyses the images of the disease, is provided. The mechanism performs the diagnosing and predicting of the disease with data set of images using deep learning. Thus, it encourages increasing of the productivity through the fast recognition of disease and the consequent action.

2.2. CNN intelligent early warning for apple skin lesion image acquired by infrared video sensors [2]

AUTHORS: C. W. H. Tan and W. X. Zhao

Video sensors and agricultural IoT (internet of things) have been widely used in the informationalized orchards. In order to realize intelligent-unattended early warning for disease-pest, this paper presents convolutional neural network (CNN) early warning for apple skin lesion image, which is real-time acquired by infrared video sensor. More specifically, as to skin lesion image, a suite of processing methods is devised to simulate the disturbance of variable orientation and light condition which occurs in orchards. It designs a method to recognize apple pathologic images based on CNN, and formulates a self-adaptive momentum rule to update CNN parameters. For example, a series of experiments are carried out on the recognition of fruit lesion image of apple trees for early warning. The results demonstrate that compared with the shallow learning algorithms and other involved, well-known deep learning methods, the recognition accuracy of the proposal is up to 96.08%, with a fairly quick convergence, and it also presents satisfying smoothness and stableness after convergence.

2.3. Basic study of automated diagnosis of viral plant diseases using convolutional neural networks [3]

AUTHORS: Y. Kawasaki, H. Uga, S. Kagiwada, and H. Iyatomi

Detecting plant diseases is usually difficult without an experts' knowledge. Therefore, fast and accurate automated diagnostic methods are highly desired in agricultural fields. Several studies on automated plant disease diagnosis have been conducted using machine learning methods. However, with these methods, it can be difficult to detect regions of interest, (ROIs) and to design and implement efficient parameters. In this study, we present a novel plant disease detection system based on convolutional neural networks (CNN). Using only training images, CNN can automatically acquire the requisite features for classification, and achieve high classification performance. We used a total of 800 cucumber leaf images to train CNN using our innovative techniques. Under the 4-fold cross-validation strategy, the proposed CNN-based system (which also extends the training dataset by generating additional images) achieves an average accuracy of 94.9 % in classifying cucumbers into two typical disease classes and a non-diseased class.

2.4. Deep neural networks-based recognition of plant diseases by leaf image classification [4]

AUTHORS: S. Sladojevic, M. Arsenovic, A. Anderla, D. Culibrk, and D. Stefanovic

The latest generation of convolutional neural networks (CNNs) has achieved impressive results in the field of image classification. This paper is concerned with a new approach to the development of plant disease recognition model, based on leaf image classification, by the use of deep convolutional networks. Novel way of training and the methodology used facilitate a quick and easy system implementation in practice. The developed model is able to recognize 13 different types of plant diseases out of healthy leaves, with the ability to distinguish plant leaves from their surroundings. All essential steps required for implementing this disease recognition model are fully described throughout the paper, starting from gathering images in order to create a database, assessed by agricultural experts. The experimental results on the developed model achieved precision between 91% and 98%, for separate class tests, on average 96.3%.

2.5. Automatic and reliable leaf disease detection using deep learning techniques [5]

AUTHORS: M. E. H. Chowdhury, T. Rahman, A. Khandakar, M. A. Ayari, A. U. Khan, M. S. Khan, N. Al-Emadi, M. B. I. Reaz, M. T. Islam, and S. H. M. Ali

Plants are a major source of food for the world population. Plant diseases contribute to production loss, which can be tackled with continuous monitoring. Manual plant disease monitoring is both laborious and error-prone. Early detection of plant diseases using computer vision and artificial intelligence (AI) can help to reduce the adverse effects of diseases and also overcome the shortcomings of continuous human monitoring. In this work, we propose the use of a deep learning architecture based on a recent convolutional neural network called EfficientNet on 18,161 plain and segmented tomato leaf images to classify tomato diseases. The performance of two segmentation models i.e., U-net and Modified U-net, for the segmentation of leaves is reported. The comparative performance of the models for binary classification (healthy and unhealthy leaves), six-class classification (healthy and various groups of diseased leaves), and ten-class classification (healthy and various types of unhealthy leaves) are also reported. The modified U-net segmentation model showed accuracy, IoU, and Dice score of 98.66%, 98.5%, and 98.73%, respectively, for the segmentation of leaf images. EfficientNet-B7 showed superior performance for the binary classification and six-class classification using segmented images with an accuracy of 99.95% and 99.12%, respectively. Finally, EfficientNet-B4 achieved an accuracy of 99.89% for ten-class classification using segmented images.

3. SYSTEM ANALYSIS

This chapter discusses the theories and some background study on Plant Disease Detection and Recommendation system.

3.1. EXISTING SYSTEM

The existing system for plant disease detection employs an advanced ensemble learning classifier that integrates four deep learning models: VGG16, VGG19, ResNet101 V2, and InceptionV3. This approach is designed to enhance the reliability and robustness of plant disease identification by leveraging the strengths of each model through a soft voting technique, which combines the individual outputs for optimal classification performance. The methodology involves several critical steps:

- **Data Collection:** The system uses the Plant Village dataset, a comprehensive collection comprising 54,305 images of both healthy and diseased plant leaves.
- **Data Preprocessing:** Images undergo preprocessing to ensure uniformity and compatibility with the deep learning models.
- **Model Training and Testing:** Each of the four models is independently trained and tested on the dataset.
- **Ensemble Learning:** The predictions from the individual models are aggregated using soft and hard voting techniques to maximize accuracy.
- **Explainable AI (XAI):** To make the predictions interpretable, the system integrates LIME (Local Interpretable Model-Agnostic Explanations), which highlights the key features influencing the model's decision-making process.

The ensemble learning framework demonstrates robust performance, achieving an overall accuracy of over 93%, showcasing its effectiveness in detecting a wide range of plant leaf diseases. By incorporating Explainable AI, the system not only ensures accurate disease classification but also enhances user confidence by providing transparent insights into the prediction process.

Disadvantages of Existing System:

- Complexity of Ensemble Learning: Integrating multiple deep learning models (VGG16, VGG19, ResNet101 V2, and InceptionV3) increases the computational complexity, leading to higher resource requirements during both training and inference.
- Dependence on Large Datasets: The system relies on the Plant Village dataset, which, while comprehensive, may not cover all real-world variations, such as environmental effects or uncommon diseases, limiting its generalizability.
- High Computational Cost: The use of sophisticated models and soft voting techniques demands significant computational power, making it less suitable for deployment on low-resource systems like mobile devices or edge computing platforms.
- Limited Real-Time Application: Due to the system's reliance on Explainable AI (LIME) and ensemble learning, it may experience latency issues, hindering its ability to provide instant predictions in real-time scenarios.
- Lack of Fertilizer Recommendations: While the system provides accurate disease classification, it does not include features such as tailored fertilizer recommendations, which could further aid users in addressing identified plant health issues.

- Interpretability Challenges in LIME: Although LIME enhances model transpar-ency, its explanations are localized and may not always represent the global behavior of the model, potentially leading to misinterpretation of results.
- These limitations highlight areas for improvement, paving the way for systems with enhanced efficiency, scalability, and broader functionality.

3.2. PROPOSED SYSTEM

- The proposed system for Plant Disease Detection and Fertilizer Recommendation introduces a streamlined and efficient approach to enhance plant health management. The system is developed using Python as the backend programming language, Flask as the web framework, and HTML, CSS, and JavaScript for the frontend.
- The proposed system employs two advanced deep learning architectures, InceptionV3 and MobileNetV2, for plant leaf disease detection and classification. The models achieve high classification accuracies, with InceptionV3 attaining 93.88% training accuracy and 90.05% validation accuracy, while MobileNetV2 achieves 98.30% training accuracy and 94.38% validation accuracy.
- The proposed models are trained and tested on a dataset comprising 70,295 images from 14 different plant types with 38 distinct classes (healthy and diseased categories). The dataset encompasses a wide variety of fruits and vegetables, including apple, blueberry, cherry, Corn (maize), grape, orange, peach, pepper, potato, raspberry, Soybean, squash, strawberry, and tomato, totaling 14 different plants. The 38 classes are: Apple scab, Apple Black rot, Apple Cedar apple rust, Apple healthy, Blueberry healthy, Cherry (including sour) Powdery mildew, Cherry (including sour) healthy, Corn (maize) Cercospora leaf spot Gray leaf spot, Corn (maize) Common rust, Corn (maize) Northern Leaf Blight, Corn (maize) healthy, Grape Black rot, Grape Esca (Black Measles), Grape Leaf blight (Isariopsis Leaf Spot), Grape healthy, Orange Haunglongbing (Citrus greening), Peach Bacterial spot, Peach healthy, Pepper bell Bacterial spot, Pepper bell healthy, Potato Early blight, Potato Late blight, Potato healthy, Raspberry healthy, Soybean healthy, Squash Powdery mildew, Strawberry Leaf scorch, Strawberry healthy, Tomato Bacterial spot, Tomato Early blight, Tomato Late blight, Tomato Leaf Mold, Tomato Septoria leaf spot, Tomato Spider mites Two-spotted spider mite, Tomato Target Spot, Tomato Yellow Leaf Curl Virus, Tomato mosaic virus, Tomato healthy.
- The system is designed to predict the type of disease present on a plant leaf and classify it accordingly. Additionally, it provides a fertilizer recommendation tailored to the identified disease and plant type, offering actionable insights to address plant health issues.
- To ensure a comprehensive user experience, the system includes visualizations such as:
- Model Accuracy and Loss Charts: These illustrate the performance of the models during training and validation phases.
- Class Distribution Graph: This provides insights into the dataset composition, showcasing the distribution of healthy and diseased classes.

- The proposed system combines high-performance deep learning models with a user-friendly interface and actionable recommendations, catering to the practical needs of farmers and agricultural stakeholders.

Advantages of Proposed System

The proposed system for Plant Disease Detection and Recommendation offers several advantages that enhance its utility and effectiveness:

- **High Accuracy:** The use of advanced deep learning models, InceptionV3 and MobileNetV2, ensures superior accuracy in detecting and classifying plant diseases, with MobileNetV2 achieving an impressive validation accuracy of 94.38%.
- **Recommendations:** In addition to disease detection, the system provides tailored fertilizer recommendations based on the predicted disease and plant type, enabling users to take precise and informed corrective actions.
- **Scalability:** The system supports the classification of 38 distinct classes across 14 different plants, making it applicable to a wide range of agricultural scenarios.
- **User-Friendly Interface:** Developed using Flask with HTML, CSS, and JavaScript, the system offers an intuitive and visually appealing web interface, ensuring ease of use for farmers and agricultural professionals.
- **Comprehensive Visualizations:** The inclusion of model accuracy and loss charts and a class distribution graph provides users with insights into the system's performance and the dataset composition, enhancing transparency and understanding.
- **Efficiency Improvement:** By focusing on two lightweight yet powerful deep learning architectures, the system ensures faster inference times and lower computational overhead compared to ensemble learning approaches.
- **Real World Applicability:** The system's ability to predict diseases in plant leaves and recommend solutions addresses real-world challenges faced by farmers, contributing to improved crop health and yield.
- **Broader Dataset Coverage:** With a dataset of 70,295 images, the system leverages a larger and more diverse dataset compared to its predecessor, increasing its robustness and reliability in various conditions.

4. SYSTEM DESIGN AND IMPLEMENTATION

4.1. System Requirements

Hardware Requirements:

- System : i5 Processor.
- Hard Disk : 500 GB.
- Monitor : 15'' LED
- Input Devices : Keyboard, Mouse
- Ram : 8 GB

Software Requirements:

- Operating system : Windows 10 / 11
- Coding Language : Python
- Python Modules : flask, markup, tensorflow, nuppy, disease, matplotlib, sklearn
- Web Framework : Flask
- Frontend : HTML, CSS, JavaScript

4.2. System Architecture

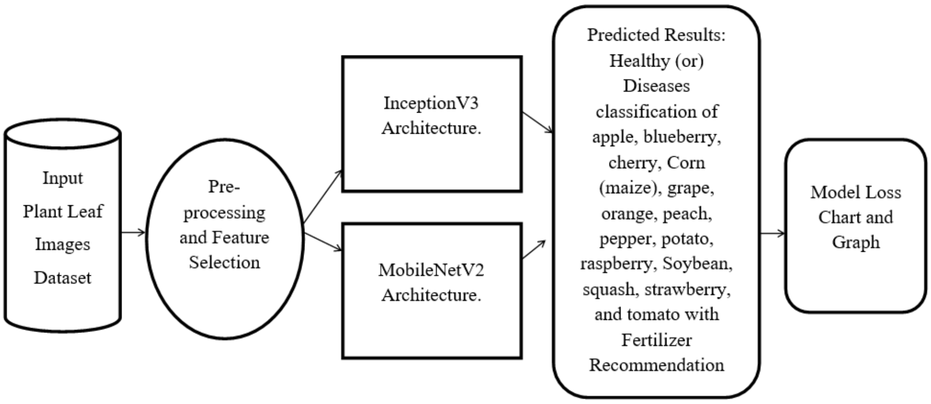

Figure: AI based Plant Disease Detection and Recommendation System Overview Architecture

4.3. System Implementation

4.3.1. Data Collection

- In the first module of the Plant Disease Detection using Deep Learning and Fertilizer Recommendation, we make the data collection process. This is the first real step towards the real development of a deep learning model, collecting data. This is a critical step that will cascade in how good the model will be, the more and better data that we get; the better our model will perform
- There are several techniques to collect the data, like web scraping, manual interventions. The dataset is located in the model folder. The dataset is referred from the popular dataset repository called kaggle. The following is the link of the dataset:
- Kaggle Dataset Link:
  https://www.kaggle.com/datasets/jayaprakashpondy/plant-leaves

4.3.2. Dataset

- In this module, Setup two main directories: Training and Validation. Training, and validation directories, create subdirectories is 38 class labels.
- The dataset encompasses 70000+ images with a wide variety of fruits and vegetables, including apple, blueberry, cherry, Corn (maize), grape, orange, peach, pepper, potato, raspberry, Soybean, squash, strawberry, and tomato, totalling 14 different plants. The 38 classes are: Apple scab, Apple Black rot, Apple Cedar apple rust, Apple healthy, Blueberry healthy, Cherry (including sour) Powdery mildew, Cherry (including sour) healthy, Corn (maize) Cercosporin leaf spot Gray leaf

spot, Corn (maize) Common rust, Corn (maize) Northern Leaf Blight, Corn (maize) healthy, Grape Black rot, Grape Esca (Black Measles), Grape Leaf blight (Isariopsis Leaf Spot), Grape healthy, Orange Huanglongbing (Citrus greening), Peach Bacterial spot, Peach healthy, Pepper bell Bacterial spot, Pepper bell healthy, Potato Early blight, Potato Late blight, Potato healthy, Raspberry healthy, Soybean healthy, Squash Powdery mildew, Strawberry Leaf scorch, Strawberry healthy, Tomato Bacterial spot, Tomato Early blight, Tomato Late blight, Tomato Leaf Mold, Tomato Septoria leaf spot, Tomato Spider mites Two-spotted spider mite, Tomato Target Spot, Tomato Yellow Leaf Curl Virus, Tomato mosaic virus, Tomato healthy.

4.3.3. Data Preparation

- During the data preparation stage, it is crucial to preprocess the data to ensure it is suitable for training. This involves tasks such as resizing images to a standard size, normalizing pixel values, and encoding labels if necessary. To achieve this, the Image Data Generator from Keras can be utilized. For instance, to resize images to a standard size of 224x224 pixels, the Target size parameter can be set to (img_height, img_width) = (224, 224).
- Additionally, pixel values can be normalized by setting the rescale parameter to 1./255, which scales the pixel values to be between 0 and 1. Furthermore, data augmentation techniques such as random shear and zoom can be applied to enhance the training data. By leveraging these techniques, the data can be effectively pre-processed to improve the performance of the deep learning model.

4.3.4. Feature Extraction

- For models like MobileNetV2 and InceptionV3, which come pre-trained with feature extraction layers, explicit feature extraction may not always be necessary. By setting trainable = False, we freeze these pre-trained layers, allowing them to retain their learned representations while preventing further updates during training.
- In the context of MobileNetV2 and InceptionV3, setting trainable = False ensures that the weights of the feature extraction layers remain fixed during training. This approach is commonly used in transfer learning scenarios, where the pre-trained model is fine-tuned on a new dataset for a specific task, such as image classification or object detection.
- By adopting this strategy, we strike a balance between leveraging powerful pre-trained representations and adapting the model to our specific dataset, ultimately improving both training efficiency and model performance.

4.3.5. Splitting the dataset

- Divide your dataset into training and validation to evaluate your model's performance. Typically, you might use an 80-20 split, but this can vary based on your dataset size and specific requirements.

4.3.6. Model Selection

- The training module is responsible for training the deep learning models using the preprocessed data. It implements two popular architectures: MobileNetV2 and InceptionV3.
  - ***MobileNetV2***

  - MobileNetV2 is a lightweight and efficient convolutional neural network architecture designed for mobile and embedded devices. It uses depth-wise separable convolutions to build deep neural networks while maintaining a small model size and low computational complexity. MobileNetV2 consists of an initial fully convolutional layer, followed by a series of inverted residual blocks with linear bottlenecks. These blocks use depth-wise convolutions to filter features and 1x1 convolutions to combine features. The network ends with a final convolution layer and global average pooling.

- ***InceptionV3***

  - InceptionV3 is a deep convolutional neural network by Google, designed to balance high accuracy with computational efficiency. It utilizes "Inception modules," which combine multiple filter sizes (1x1, 3x3, 5x5) and pooling in parallel, enabling the model to capture multi-scale features. This architecture reduces computational costs by replacing large filters with smaller, factorized ones (e.g., replacing 5x5 filters with two 3x3 filters). Auxiliary classifiers also act as regularizers to stabilize training. With 48 layers, InceptionV3 achieves strong performance in tasks like image classification and object detection. Its pretrained weights on ImageNet make it effective for transfer learning, offering a practical balance of depth and efficiency.

4.3.7. Training the Model

- To train the models, the training module first loads the pre-trained weights for MobileNetV2 and InceptionV3 from the dataset. It then adds a global average pooling layer and a fully connected layer with the appropriate number of classes for the specific task. The base layers of the pre-trained models are frozen by setting their trainable flag to False, allowing only the added layers to be trained.
- The training process involves optimizing the model parameters using a suitable optimization algorithm, such as Adam or SGD, and a loss function appropriate for the task (e.g., categorical cross-entropy for classification). The training data is fed to the model in batches, and the gradients are computed and used to update the model weights. The training process continues for a specified number of epochs or until a certain performance metric is achieved on a validation set.
- After training, the module saves the trained models for future use in the prediction and evaluation modules. The saved models can be loaded and used for inference on new data or fine-tuned on additional datasets if needed.

4.3.8. Analyze and Prediction

- Once training is complete, analyze the training process (e.g., loss curves) and make predictions on your validation set to assess the model's performance.

4.3.9. Accuracy on test set

- Once the model is trained, it needs to be evaluated for its performance. This module involves splitting the dataset into training and testing subsets and assessing the model's accuracy, precision, recall, and F1-score.

- The MobileNetV2 architecture achieves a Training accuracy of 98.30% and validation accuracy of 94.38%. The InceptionV3 architecture attains a Training accuracy of 93.88% and validation accuracy of 90.05%.

4.3.10. Saving the Trained Model

- Once you're confident enough to take your trained and tested model into the production-ready environment, the first step is to save it into an .h5 or .pkl file using a library like pickle.
- Make sure you have pickle installed in your environment.

4.3.11. Prediction Module

- Develop a prediction module to make predictions using the trained MobileNetV2 and InceptionV3 model. This module should take input images; preprocess them as necessary, and output predictions.

4.3.12. Model Evaluation Module

- This module evaluates the performance of the trained models using the testing dataset. It calculates accuracy metrics and other performance indicators to assess model effectiveness.
- Model Accuracy and Loss Charts: These illustrate the performance of the models during training and validation phases.
- Class Distribution Graph: This provides insights into the dataset composition, showcasing the distribution of healthy and diseased classes.

5. SIMULATION RESULTS

5.1. Web output for AI based Plant Disease Detection and Recommendation system

5.1.1. *Home Page*

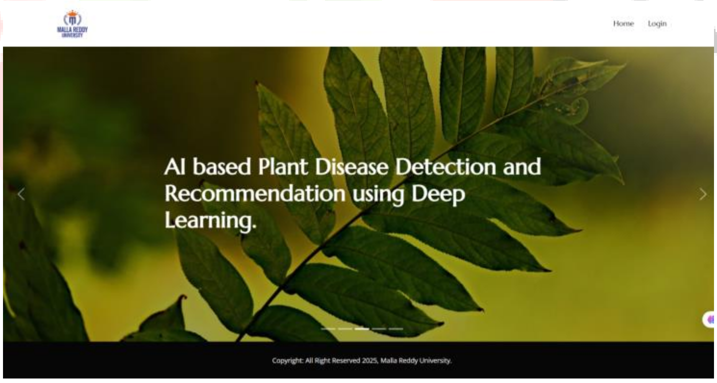

Figure: Plant Disease Detection and Recommendation Home Page

5.1.2. *Login Page*

Login as an admin for testing with Username as 'admin' and Password as 'admin'.

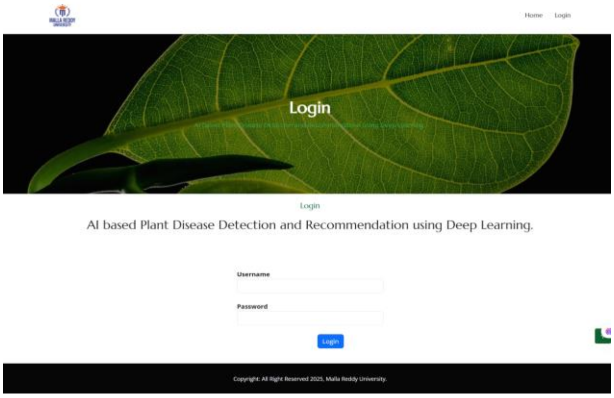

Figure: Plant Disease Detection and Recommendation Login Page

5.1.3. *Input Preview Page*

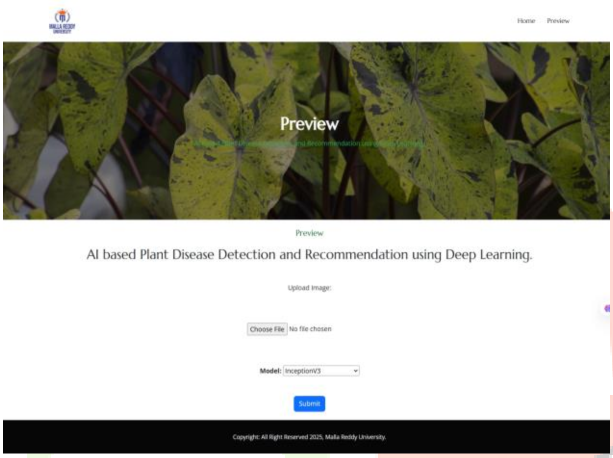

Figure: Plant Disease Detection and Recommendation Input Page

Provide Input Details.

Upload Image for disease detection and select model to perform disease classification, identification, explanation and recommendation.

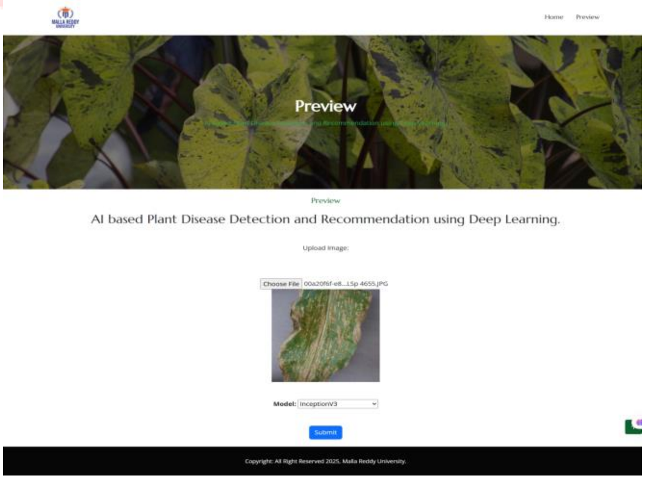

Figure: Plant Disease Detection and Recommendation Input Preview Page

Click on Submit for results.

5.1.4. *Output Preview Page*

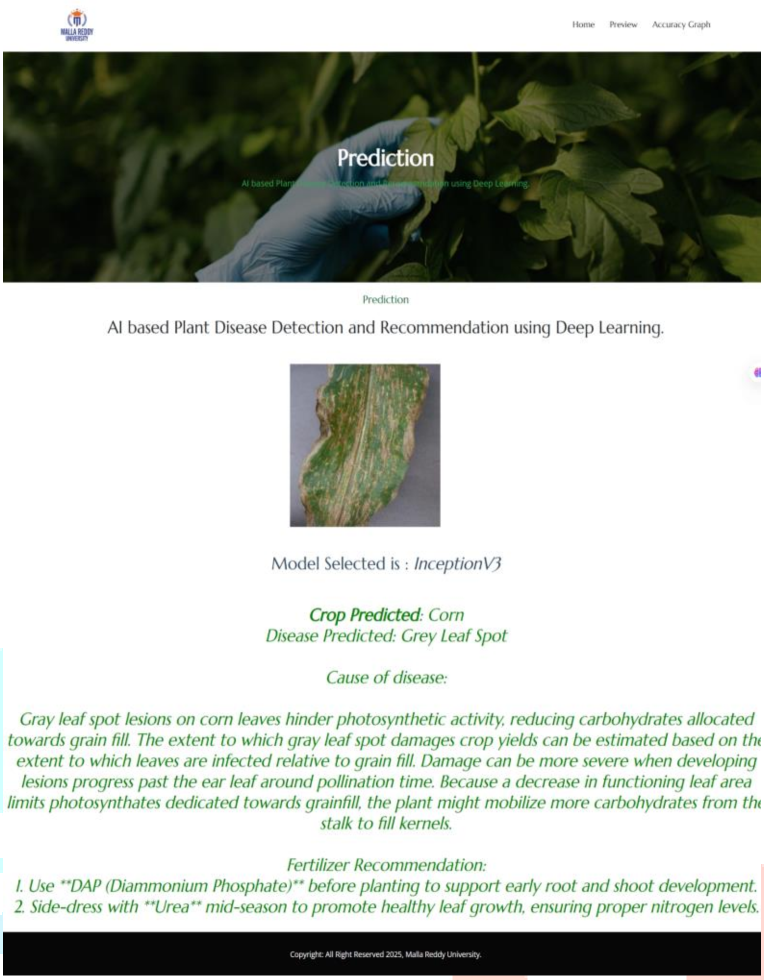

Figure 1: Plant Disease Detection and Recommendation Result Page

5.1.5. *Dataset usage chart*

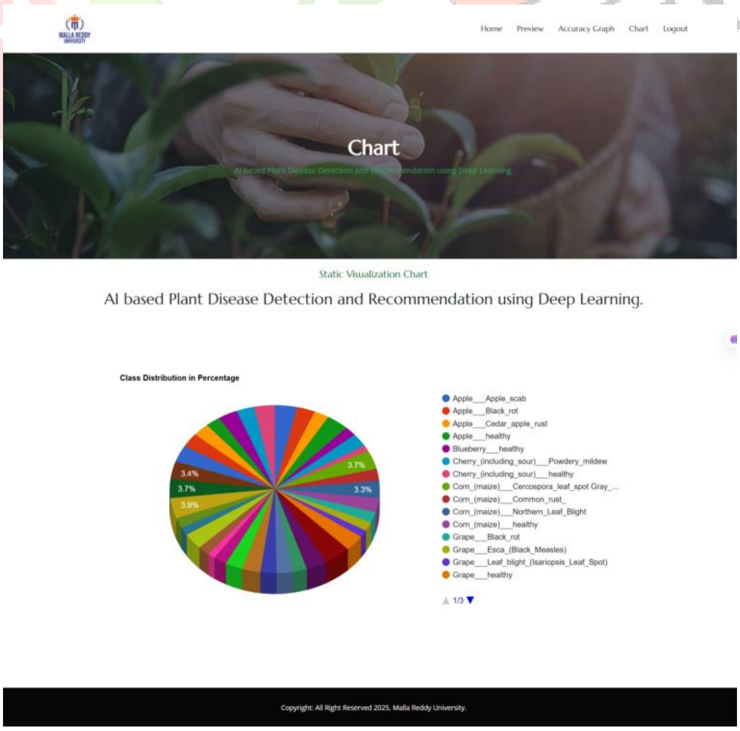

Figure: Plant Disease Detection and Recommendation Datasets usage chart

5.1.6. *Test Accuracy Graph Page*

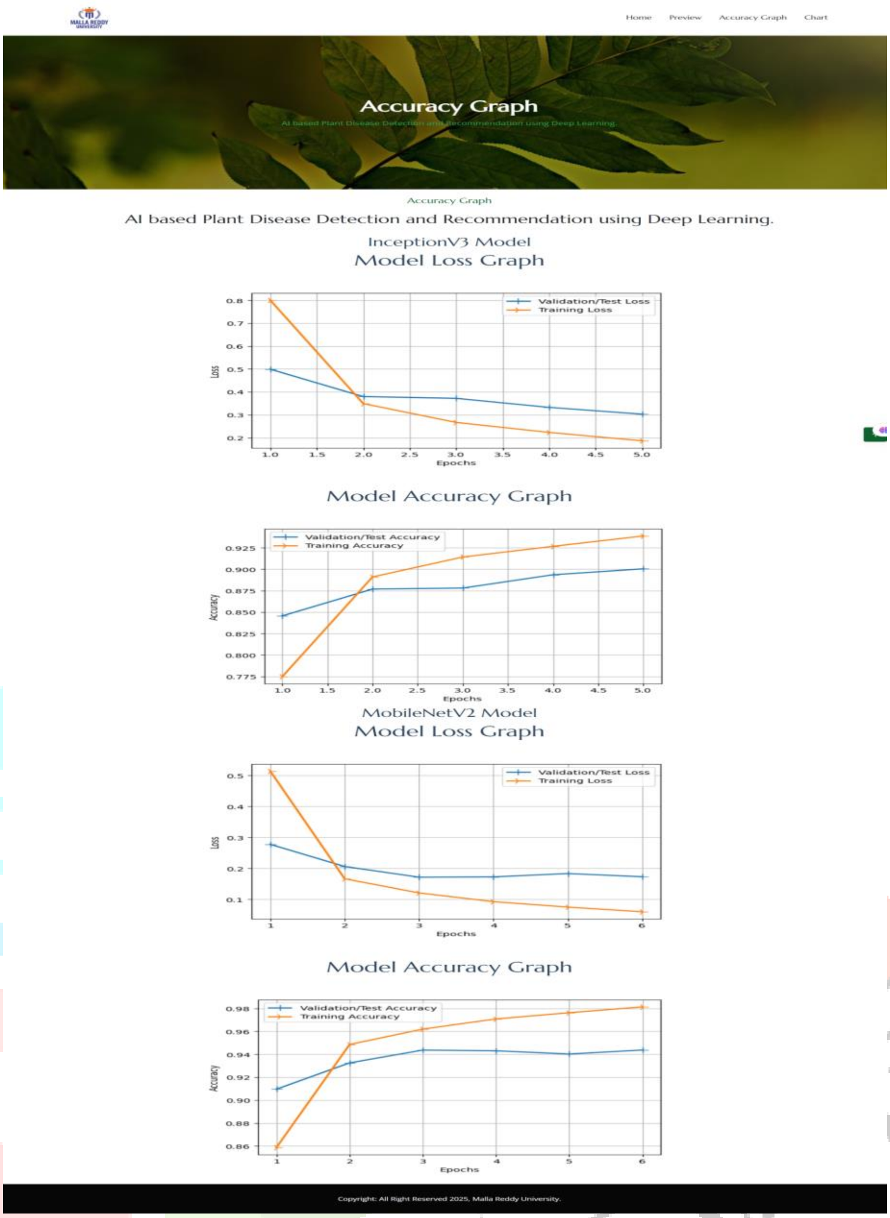

Figure 2: Plant Disease Detection and Recommendation Model Loss and Accuracy graph

5.1.7. *Performance Metrics: Comparing InceptionV3, MobileNetV2 with Other Models*

Model	Accuracy	Parameters	Size (MB)	Inference Time
AlexNet	83.5%	60M	227	15ms
VGG16	90.1%	138M	528	22ms
ResNet50	91.2%	25.6M	98	17ms
MobileNetV2	98.3%	3.5M	14	8ms
InceptionV3	93.8%	23.8M	92	16ms

Table: Model performance metrics comparison

The comparison reveals the fundamental trade-off between accuracy and efficiency. InceptionV3 excels in raw performance metrics, achieving nearly 99% accuracy and precision. However, MobileNetV2 provides a compelling alternative with only a modest reduction in accuracy while offering 3.4x faster inference speed.

Most significantly, MobileNetV2's substantially smaller model size (14MB vs. 92MB) and reduced memory requirements make it vastly more suitable for deployment on entry-level smartphones common in rural India. This enables practical field use without requiring high-end devices or consistent internet connectivity.

CONCLUSION AND FUTURE SCOPE

5.2. Conclusion

An AI based Plant Disease Detection and Recommendation System successfully addresses the critical challenge of identifying plant diseases and providing actionable insights to improve agricultural productivity. By leveraging cutting-edge deep learning models, namely InceptionV3 and MobileNetV2, the system achieves high accuracy in detecting and classifying plant leaf diseases across a dataset comprising 70,000 images representing 38 distinct classes from 14 plant types. This ensures a comprehensive solution capable of diagnosing a wide range of diseases with remarkable precision.

The integration of a fertilizer recommendation module further enhances the system's practicality, enabling users to take immediate and appropriate measures to mitigate plant health issues. This feature bridges the gap between disease detection and actionable agricultural interventions, making the system highly relevant for real-world applications. The system's performance is visualized through model accuracy and loss charts, providing transparency into the training and validation processes, and a class distribution graph offers insights into the dataset composition. These features improve user understanding and confidence in the system's capabilities. Developed with a user-friendly web interface using Flask, HTML, CSS, and JavaScript, the platform ensures accessibility and ease of use for a wide audience, including farmers, agricultural experts, and researchers. The efficient use of computational resources, particularly with the lightweight MobileNetV2 architecture, ensures the system's adaptability for deployment in diverse environments, including resource-constrained settings.

In conclusion, this project represents a significant step forward in leveraging technology to enhance agricultural practices. By combining accurate disease detection, tailored fertilizer recommendations, and an intuitive interface, the system empowers users to make informed decisions that contribute to healthier crops and improved yields.

5.3. Future Scope

The Plant Disease Detection and Fertilizer Recommendation System has demonstrated its effectiveness and utility in addressing agricultural challenges, but several enhancements and extensions can be considered to broaden its impact and functionality:

- **Incorporating More Plant Types and Diseases:** Expanding the dataset to include additional plant species and rare or region-specific diseases can make the system more versatile and applicable to diverse agricultural scenarios worldwide.
- **Integration with IoT Devices:** The system can be integrated with IoT-enabled agricultural sensors and drones to automate real-time disease detection in large farming areas, providing timely interventions.

- **Mobile Application Development:** Developing a mobile application version of the system will increase accessibility for farmers in remote areas, enabling them to use the system directly from their smartphones.
- **Multi-Language Support:** Adding support for multiple languages can cater to farmers from different linguistic backgrounds, enhancing usability and adoption in diverse regions.
- **Dynamic Environmental Factor Analysis:** Incorporating weather conditions, soil quality, and other environmental factors into the system can provide more precise recommendations for disease management and fertilizer application.
- **Self-Learning Capability:** Implementing reinforcement learning or continuous learning techniques will enable the system to improve its accuracy over time by learning from user feedback and new data.
- **Crop Yield Prediction:** Extending the system to predict the impact of detected diseases on crop yield will provide farmers with a better understanding of potential losses and guide them in proactive planning.
- **Integration with Agricultural Market Data:** Linking the system with real-time market data for fertilizers and pesticides can help users make cost-effective decisions while addressing plant health issues.

By addressing these aspects, the system can evolve into a more robust, scalable, and comprehensive agricultural tool, driving innovation and sustainability in modern farming practices.

REFERENCES

[1] H. Park, J.-S. Eun, and S.-H. Kim, ‘‘Image-based disease diagnosing and predicting of the crops through the deep learning mechanism,’’ in Proc. Int. Conf. Inf. Commun. Technol. Converg. (ICTC), Oct. 2017, pp. 129–131.

[2] C. W. H. Tan and W. X. Zhao, ‘‘CNN intelligent early warning for apple skin lesion image acquired by infrared video sensors,’’ High Technol. Lett., vol. 10, no. 1, pp. 67–74, 2016.

[3] Y. Kawasaki, H. Uga, S. Kagiwada, and H. Iyatomi, ‘‘Basic study of automated diagnosis of viral plant diseases using convolutional neural networks,’’ in Proc. Int. Symp. Vis. Comput. Cham, Switzerland: Springer, 2015, pp. 638–645.

[4] S. Sladojevic, M. Arsenovic, A. Anderla, D. Culibrk, and D. Stefanovic, ‘‘Deep neural networks-based recognition of plant diseases by leaf image classification,’’ Comput. Intell. Neurosci., vol. 2016, pp. 1–11, Mar. 2016.

[5] M. E. H. Chowdhury, T. Rahman, A. Khandakar, M. A. Ayari, A. U. Khan, M. S. Khan, N. Al-Emadi, M. B. I. Reaz, M. T. Islam, and S. H. M. Ali, ‘‘Automatic and reliable leaf disease detection using deep learning techniques,’’ AgriEngineering, vol. 3, no. 2, pp. 294–312, May 2021.

[6] M. Chohan, A. Khan, R. Chohan, S. H. Katpar, and M. S. Mahar, ‘‘Plant disease detection using deep learning,’’ Int. J. Recent Technol. Eng., vol. 9, no. 1, pp. 909–914, May 2020.

[7] G. Shrestha, Deepsikha, M. Das, and N. Dey, ‘‘Plant disease detection using CNN,’’ in Proc. IEEE Appl. Signal Process. Conf. (ASPCON), Oct. 2020, pp. 109–113.

[8] L. S. P. Annabel, T. Annapoorani, and P. Deepalakshmi, ‘‘Machine learning for plant leaf disease detection and classification—A review,’’ in Proc. Int. Conf. Commun. Signal Process. (ICCSP), Apr. 2019, pp. 0538–0542.

[9] B. Dey, J. Ferdous, and R. Ahmed, ''Machine learning based recommendation of agricultural and horticultural crop farming in India under the regime of NPK, soil pH and three climatic variables,'' Heliyon, vol. 10, no. 3, Feb. 2024, Art. no. e25112.

[10] L. C. Ngugi, M. Abelwahab, and M. Abo-Zahhad, ''Recent advances in image processing techniques for automated leaf pest and disease recognition—A review,'' Inf. Process. Agricult., vol. 8, no. 1, pp. 27–51, Mar. 2021.

[11] Y. M. Oo and N. C. Htun, ''Plant leaf disease detection and classification using image processing,'' Int. J. Res. Eng., vol. 5, no. 9, pp. 516–523, Nov. 2018.

[12] N. Karthikeyan, M. Anjana, S. Anusha, R. Divya, and A. Vinod, ''Leaf disease detection using image processing,'' Int. J. Innov. Sci. Eng. Technol, vol. 7, pp. 60–65, 2020.

[13] B. Dey, R. Ahmed, J. Ferdous, M. M. U. Haque, R. Khatun, F. E. Hasan, and S. N. Uddin, ''Automated plant species identification from the stomata images using deep neural network: A study of selected mangrove and freshwater swamp forest tree species of Bangladesh,'' Ecol. Informat., vol. 75, Jul. 2023, Art. no. 102128. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1574954123001577

[14] M. A. Ganaie, M. Hu, A. Malik, M. Tanveer, and P. Suganthan, ''Ensemble deep learning: A review,'' Eng. Appl. Artif. Intell., vol. 115, Sep. 2022, p. 105151.

[15] S. P. Mohanty, D. P. Hughes, and M. Salathé, ''Using deep learning for image-based plant disease detection,'' Frontiers Plant Sci., vol. 7, p. 1419, Sep. 2016.

[16] Y. Djenouri, A. Belhadi, A. Yazidi, G. Srivastava, and J. C. Lin, ''Artificial intelligence of medical things for disease detection using ensemble deep learning and attention mechanism,'' Expert Syst., vol. 41, no. 6, Jun. 2024, Art. no. e13093.

[17] B. Dey, M. M. U. Haque, R. Khatun, and R. Ahmed, ''Comparative performance of four CNN-based deep learning variants in detecting Hispa pest, two fungal diseases, and NPK deficiency symptoms of Rice (Oryza sativa),'' Comput. Electron. Agricult., vol. 202, Nov. 2022, Art. no. 107340. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0168169922006494

[18] Kaggle Dataset: https://www.kaggle.com/datasets/jayaprakashpondy/plant-leaves