I. INTRODUCTION

The problem of plant disease outbreaks poses a significant threat to crop yield, food security, and agricultural sustainability. Early detection and accurate prediction of plant diseases are crucial for effective crop management and timely interventions. In this project, we aim to address this problem by developing a plant disease prediction system using machine learning, specifically Convolutional Neural Networks (CNN).

Our approach involves training a CNN model on a large dataset of plant images, encompassing both healthy plants and plants affected by various diseases. The model will learn to extract relevant features and patterns from the images, enabling it to make predictions about the presence or absence of diseases in unseen plant images. By leveraging the power of deep learning and image analysis techniques, we strive to achieve high accuracy in detecting and classifying plant diseases.

Our work aligns with existing research in the field of plant disease detection and prediction. Previous studies have demonstrated the effectiveness of CNNs in image classification tasks, including plant disease diagnosis. However, we aim to contribute by utilizing a more extensive dataset, incorporating a wide range of plant species and disease types, to improve the model's robustness and generalization capabilities.

The results of our project will include a trained CNN model capable of predicting plant diseases from input images. We anticipate providing an evaluation of the model's performance, including accuracy, precision, recall, and F1 score metrics. Additionally, we plan to present insights into the learned features and patterns that contribute to disease identification. The outcomes of this project will aid farmers, agricultural experts, and researchers in the early detection and management of plant diseases, ultimately reducing crop losses and ensuring sustainable agricultural practices.

In conclusion, our project aims to tackle the critical problem of plant disease outbreaks through machine learning techniques. By leveraging CNNs and a comprehensive dataset, we anticipate improving the accuracy and efficiency of plant disease prediction. Our research contributes to the growing body of work in the field of precision agriculture and can have a significant impact on global food production and agricultural sustainability.

1.1 MOTIVATION

- **Early detection**: Plant disease prediction enables early detection, allowing proactive measures to minimize crop losses.
- **Timely intervention**: Accurate predictions prompt prompt interventions for effective treatments and disease management.
- **Improved crop management**: Disease prediction provides insights for optimizing irrigation, fertilization, and integrated pest management strategies.
- **Enhanced resource allocation**: Targeted interventions minimize chemical inputs, reducing environmental impact while maximizing control measures' effectiveness.
- **Sustainable agriculture**: Disease prediction promotes environmentally friendly practices by reducing reliance on broad-spectrum pesticides.
- **Increased crop yield and food security**: Minimizing crop losses improves yield, ensuring food security and stable agricultural supply.
- **Knowledge sharing and collaboration**: Prediction models foster information exchange and community-driven disease prevention efforts.
- **Advancement of precision agriculture**: Plant disease prediction contributes to the field's technology-driven approach for optimizing farming practices.
- **Research and innovation**: Ongoing development improves accuracy through advancements in machine learning and image analysis techniques.

1.2 SCOPE OF THE PROJECT

- Develop a machine learning model using Convolutional Neural Networks (CNN) for plant disease prediction.
- Train the model using a comprehensive dataset of plant images, encompassing various plant species and disease types.
- Implement image preprocessing techniques to enhance the quality and consistency of input images for better prediction accuracy.
- Evaluate the performance of the model using metrics such as accuracy, precision, recall, and F1 score, and compare it with existing approaches for plant disease prediction.

1.3 PROBLEM STATEMENT

To develop a machine learning model for the accurate and automated detection of plant diseases in leaves, using image processing techniques and a large dataset of labeled images. The proposed system aims to provide farmers with a quick and reliable diagnosis of plant diseases, facilitating timely management and control measures to improve crop yield and quality.

II. LITERATURE SURVEY

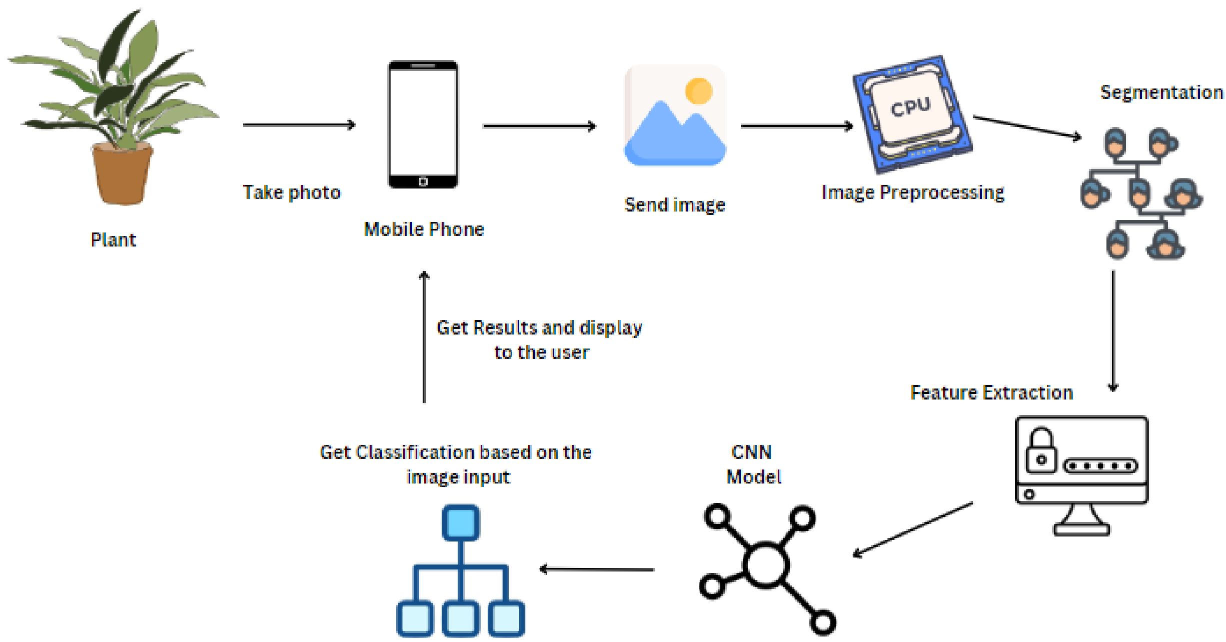
Title	Year	Journal	Technique	Limitations	Future
Plants Diseases Prediction Framework: A Image-Based System Using Deep Learning	Aug 2022	Researchgate	CNN	There is a requirement of a huge training dataset. Leads to high computational cost. Training process can be slow in the absence of a GPU.	Image augmentation can also be applied to the dataset to increase its size and compare the results between all classifiers as well as The agricultural department seeks to automate the process of recognizing high-yield crops (real-time)
Using Deep Learning for Image-Based Plant Disease Detection	Sept 2016	Methods article	Deep learning architecture like Alexnet and Google Net	When tested on a set of images taken under conditions different from the images used for training, the model's accuracy is reduced substantially, to just above 31%.	Supplement the existing models.
Deep feature-based plant disease identification using machine learning classifier	Nov 2022	MICS	Deep CNN	It will not produce accurate results if the images are taken from different angles as well as with different backgrounds. Small data will result in a high difference in training and testing accuracy then the model will suffer from	Includes collecting the images from different geographical areas, different cultivation conditions, and image quality to increase the disease class and dataset size.

				an overfitting problem.	
Plant Disease Detection using Machine Learning	July 2020	IRJET	Transfer learning and convolutional neural networks	Currently the system is trained using Plant Village dataset, the model is trained to detect only 26 types of plant diseases.	Proposes to train the system with much more data of various other plants and diseases to further increase the scope of the system and build accuracy
A Novel Hybrid Severity Prediction Model for Blast Paddy Disease Using Machine Learning	Jan 2023	MPDI	Extract the features from the images using the CNN approach and classifies the images using the SVM approach	Detects only 4 stages of the severity of a disease	
Deep Learning for Image-Based Cassava Disease Detection	Oct 2017	Front Plant Sci	Transfer learning from a deep convolutional neural network (CNN) model	Various changes in the background had little effect on the accuracy	Work to validate the method in the field with mobile devices has begun through work with TensorFlow Android Inference Interface.
Machine Learning for Plant Disease Incidence and Severity Measurements from Leaf Images	2016	IEEE	Three classifiers (LinearSVC, KNN, and Extra Trees) were trained using scikit-learn toolbox on leaf images representing different diseases. Performance was evaluated using 10-fold cross validation.	No clear explanation for why the ORB features perform unusually well.	Includes expanding the dataset used to train the machine learning algorithms,, and investigating the feasibility of integrating the diagnostic system with other agricultural technologies, such as drones or sensors.

Automation and integration of growth monitoring in plants (with disease prediction) and crop prediction	2021	Materials Today	CNN	It can predict 38 Different classes	
Disease Detection of Pomegranate Plant Using Image Processing	May 2016	IEEE	SVM and K-means Clustering	Various changes in the background had little effect on the accuracy	This method is planned to apply to different species related to Pomegranate plant
Intelligent Plant Disease Identification System using Machine Learning	14 Nov 2020	MDPI	Support Vector Machine and Extreme Learning Machine used as classifiers to classify healthy and diseased plants	ELM is an effective method to classify, but utilizes too much time to compute	Integrating real-time hardware with ELM to detect more plant diseases
Crop Disease Detection Using Deep Learning	2018	IEEE	Proposes a deep learning-based model which is trained using a public dataset of healthy and diseased crop leaves.	Convolutional Neural Network (CNN) for image detection and object detection.	Computational cost is high, and not a variety of species of crops covered.

III. DESIGN

3.1 HIGH LEVEL DESIGN



The user captures an image of a plant using their mobile phone's camera. The captured image undergoes preprocessing to enhance its quality and prepare it for analysis. The preprocessed image is divided into meaningful regions by Segmentation, isolating the specific area affected by the disease. Relevant features are extracted from the segmented regions using Feature Extraction to capture distinctive information. A trained Convolutional Neural Network (CNN) model processes the extracted features to predict the most probable disease affecting the plant. The CNN model assigns a disease class label to the input image. The predicted disease classification and relevant information are displayed on the user's mobile phone, helping them make informed decisions for plant care and disease management.

3.2 DETAILED DESIGN

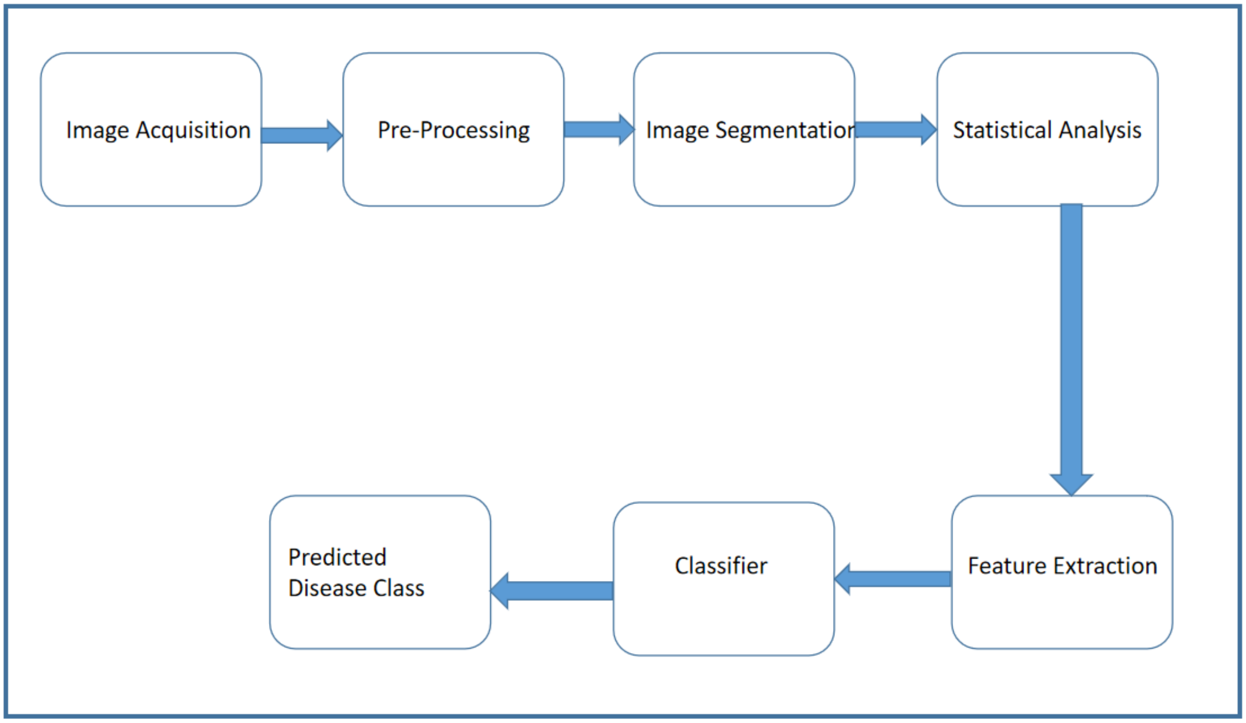
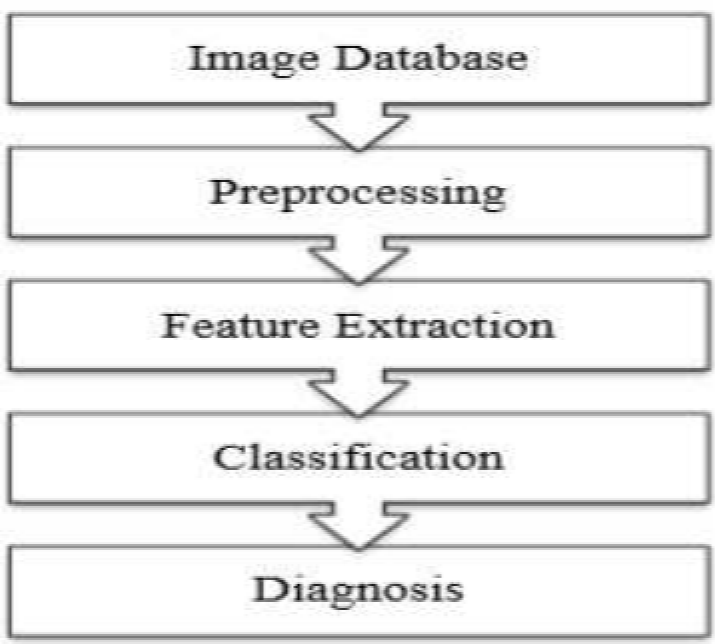


Fig. 3.1 Block diagram depicting the flow of the plant disease detection [1]



Plant disease detection using machine learning involves creating a model that can accurately classify images of plants as either healthy or diseased. The workflow for this process typically involves the following steps:

1. **Image database**: A large database of images of plants, both healthy and diseased, is collected and curated for use in the training and testing of the machine learning model.

2. **Preprocessing images**: The images are preprocessed to enhance their quality and remove any noise or artifacts that may negatively impact the model's accuracy. This can involve techniques such as resizing, cropping, and adjusting color levels.

3. **Feature extraction**: The preprocessed images are then analyzed to extract features that can be used to distinguish between healthy and diseased plants. This can involve identifying patterns or structures within the images, such as the presence of lesions or discoloration.

4. **Classification**: The extracted features are used to train a classification algorithm, such as a support vector machine or convolutional neural network, to distinguish between healthy and diseased plants.

5. **Diagnosis**: Once the model is trained, it can be used to diagnose plant diseases by analyzing new images of plants. The model will classify the new images as either healthy or diseased, allowing for early detection and treatment of plant diseases.

3.3 SEQUENCE DIAGRAM

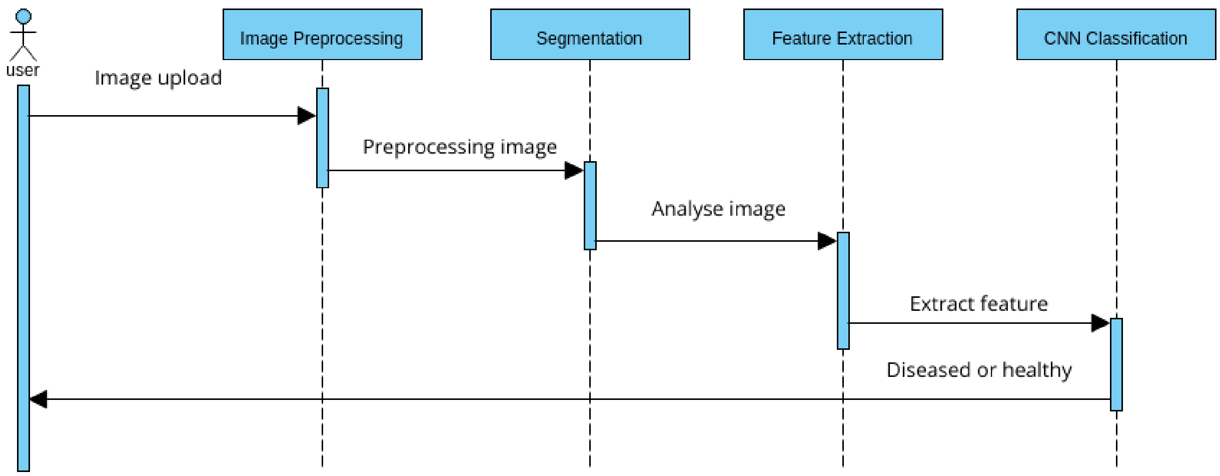


Fig.3.1 Sequence diagram[2] showing the sequence of tasks

- **User**: Initiates the sequence by uploading an image for plant disease prediction.
- **Image Processing**: The uploaded image is received by the Image Processing lifeline, which performs preprocessing operations to enhance the quality of the image for further analysis.
- **Segmentation**: The preprocessed image is passed to the Segmentation lifeline, which analyzes the image to identify and segment regions of interest related to plant diseases.
- **Feature Extraction**: The segmented image is then forwarded to the Feature Extraction lifeline, where relevant features are extracted from the segmented regions to capture disease-specific characteristics.
- **CNN Classification**: The extracted features are passed to the CNN Classification lifeline, where a Convolutional Neural Network model is utilized to classify the image as either diseased or healthy based on the learned patterns and features.
- **Result**: The classification result is sent back to the User lifeline, indicating whether the uploaded plant image is predicted as diseased or healthy.

3.4 USE CASE DIAGRAM

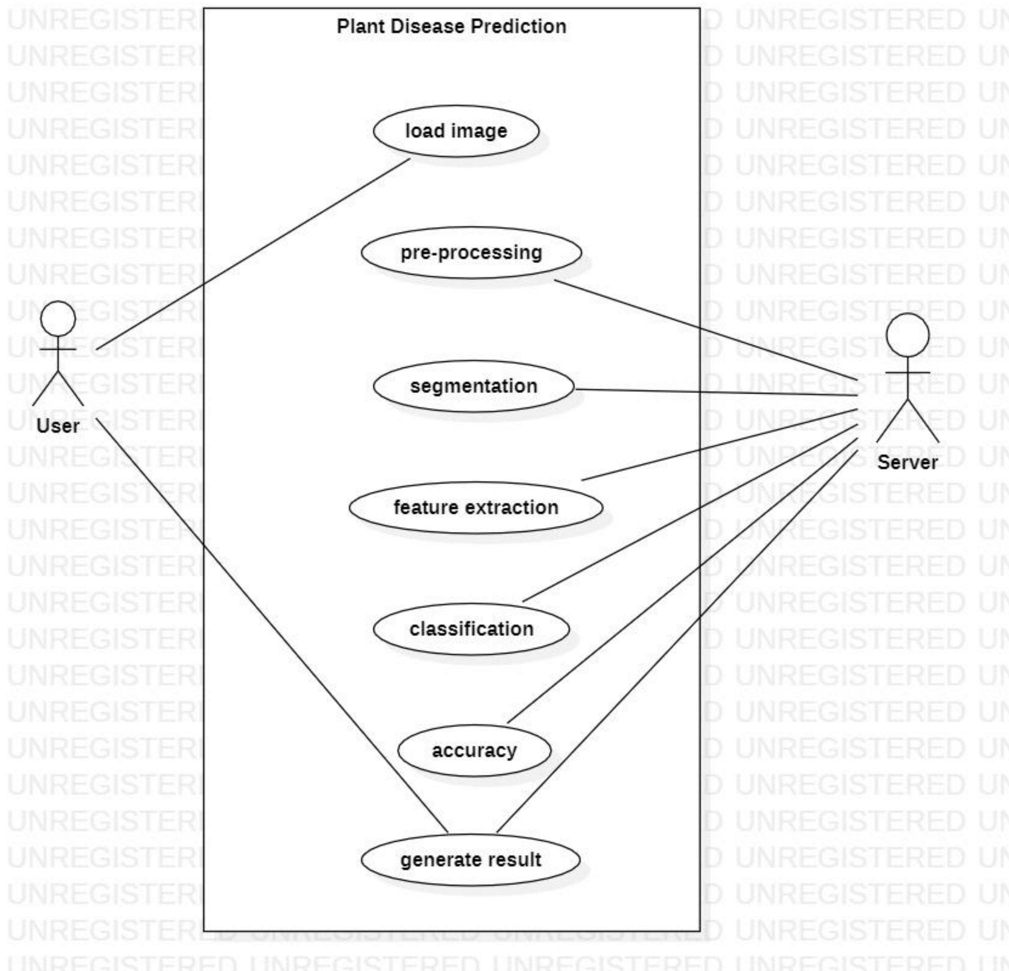


Fig.3.2 Use case diagram[1] describing different use cases possible in the model

Actors involved in the Plant Leaf disease detection model are:

1. **User**: The "User" actor in the plant disease detection model represents an individual who interacts with the system to utilize its functionality for detecting plant diseases. The user can be a farmer, a researcher, or any person involved in plant health management

2. **System**: The "System" actor in the plant disease detection model represents the intelligent software system responsible for detecting plant diseases from input images. It encompasses various modules, algorithms, and functionalities designed to perform accurate disease detection.

The use cases present in this model are:

1. **Load Image**: It represents the functionality of the user loading an image of a plant into the system for disease detection. The user provides the image as input to the system.

2. **Image Preprocessing**: It represents the server's functionality of preprocessing the loaded image. Image preprocessing involves enhancing the quality of the image, removing noise, and preparing it for further analysis.

3. **Segmentation**: It represents the server's functionality of segmenting the preprocessed image. Segmentation involves dividing the image into meaningful regions, such as isolating the specific area affected by the disease.

4. **Feature Extraction**: It represents the server's functionality of extracting relevant features from the segmented image regions. Feature extraction involves identifying distinctive attributes or patterns that can be used for disease detection.

5. **Classification**: It represents the server's functionality of performing classification on the extracted features. Classification involves using machine learning or deep learning algorithms to predict the disease based on the learned patterns from the input features.

6. **Accuracy**: It represents the server's functionality of evaluating the accuracy of the disease classification. Accuracy assessment involves comparing the predicted disease with ground truth labels or expert knowledge to measure the model's performance.

7. **Generate Result**: It represents the functionality of generating the disease detection result based on the classification and accuracy assessment. The result includes information such as the detected disease, its severity, and any additional recommendations. The result is displayed or provided to the user for further actions or decision-making.

IV. IMPLEMENTATION

The steps involved in the implementation of a CNN model for leaf disease detection:

1. Model Initialization: The code initializes a sequential model. This model is a linear stack of layers.

2. Convolutional Layers: The model adds a series of convolutional layers with various filter sizes, activations, and padding. These layers are responsible for learning and extracting features from the input leaf images.

3. Max Pooling Layers: After each pair of convolutional layers, a max pooling layer is added. These layers reduce the spatial dimensions of the feature maps, helping to extract the most important information while reducing computational complexity.

4. Flatten Layer: Following the convolutional and pooling layers, a flatten layer is added to convert the 2D feature maps into a 1D feature vector, preparing it for the fully connected layers.

5. Fully Connected Layers: The model adds a dense layer with 1568 units and a ReLU activation. A dropout layer is then added to mitigate overfitting.

6. Output Layer: Finally, the model adds a dense output layer with 38 units (assuming there are 38 classes of leaf diseases to be classified) and a softmax activation, which produces the probabilities for each class.

7. Model Compilation: The model is compiled using the Adam optimizer with a learning rate of 0.0001. The loss function is set to "sparse_categorical_crossentropy" since the labels are integers, and the metrics are defined as accuracy.

8. Model Summary: Prints a summary of the model's architecture, displaying the number of parameters and the shape of each layer's output.

Why this implementation may be better than existing solutions:

1. Depth and Complexity: The model's deep architecture allows it to capture intricate patterns and features from leaf images, potentially leading to improved detection accuracy.

2. Localization: CNNs excel at spatial localization, enabling identification of specific disease regions within a leaf. This information is valuable for targeted treatment or management strategies.

3. Transfer Learning: The implementation can leverage pre-trained models for improved performance by fine-tuning on the leaf disease dataset, benefiting from learned features from diverse images.

4. Regularization Techniques: The inclusion of dropout helps prevent overfitting, improving generalization and robustness of the model.

5. Optimization: The model utilizes the Adam optimizer with adaptive learning rate, leading to more efficient optimization and faster convergence.

4.1 PROPOSED METHODOLOGY

Here is a proposed methodology for implementing leaf disease detection using the CNN model:

1. Dataset Preparation: Collect a diverse dataset of leaf images with healthy and diseased leaves, organized into class-labeled directories.

2. Data Loading and Preprocessing: Load the images and normalize pixel values with `Rescaling`.

3. Model Creation: Build a CNN model, adding convolutional, max pooling, flatten, and fully connected layers. Include dropout for regularization and a softmax output layer.

4. Model Compilation and Training: Compile the model with an optimizer, loss function, and evaluation metrics. Split the dataset into training and validation sets. Train the model using `fit`, monitoring validation accuracy.

5. Model Evaluation: Evaluate the trained model on a separate test dataset using evaluation metrics and visualize performance with a confusion matrix.

6. Model Optimization: Fine-tune the model's architecture and hyperparameters. Explore techniques like transfer learning and experiment with configurations, such as layer adjustments, filter sizes, and data augmentation.

7. Deployment and Integration: Save the trained model for future use. Integrate the model into an application or system for automated leaf disease detection. Monitor and update the model periodically with new data.

4.2 ALGORITHM USED FOR IMPLEMENTATION

The algorithm used for the implementation can be outlined as follows:

1. Import necessary libraries.
2. Define the CNN model architecture using Keras.
3. Compile the model with optimizer, learning rate, loss function, and evaluation metrics.
4. Load and preprocess the leaf image dataset.
5. Split the dataset into training and validation sets.
6. Train the model on the training dataset for a specified number of epochs.
7. Evaluate the model using evaluation metrics and visualize its performance with a confusion matrix.
8. Optimize the model by adjusting its architecture and hyperparameters.
9. Save the trained model for future use.
10. Integrate the model into an application or system for leaf disease detection.
11. Continuously monitor and update the model's performance with new data.

4.3 TOOLS & TECHNOLOGIES USED

These tools and technologies provide the necessary functionality to implement and evaluate the leaf disease detection system using a CNN model.

Python: The programming language used for implementing the code.

TensorFlow: An open-source deep learning framework used for building and training neural networks.

Keras: A high-level neural networks API that runs on top of TensorFlow, used for simplifying the process of building and training deep learning models.

NumPy: A library for numerical operations in Python, used for various mathematical computations.

Matplotlib: A plotting library used for visualizations, such as displaying the model's summary.

scikit-learn: A machine learning library in Python, used for evaluation metrics and visualizing the model's performance.

4.4 TESTING

```
Train Accuracy  : 98.07 %
Test Accuracy   : 96.88 %
Precision Score : 96.88 %
Recall Score    : 96.88 %
```

In a machine learning model, several evaluation metrics are commonly used to assess the performance of the model. Let's discuss four important metrics: train accuracy, test accuracy, precision score, and recall score.

1. **Train Accuracy**:

Train accuracy measures the accuracy of the model's predictions on the training data. It represents how well the model fits the training data and indicates the model's ability to learn patterns and make correct predictions on the data it has been trained on.

2. **Test Accuracy**:

Test accuracy measures the accuracy of the model's predictions on the test data, which consists of unseen examples. It evaluates how well the model generalizes to new, unseen data.

3. **Precision Score**:

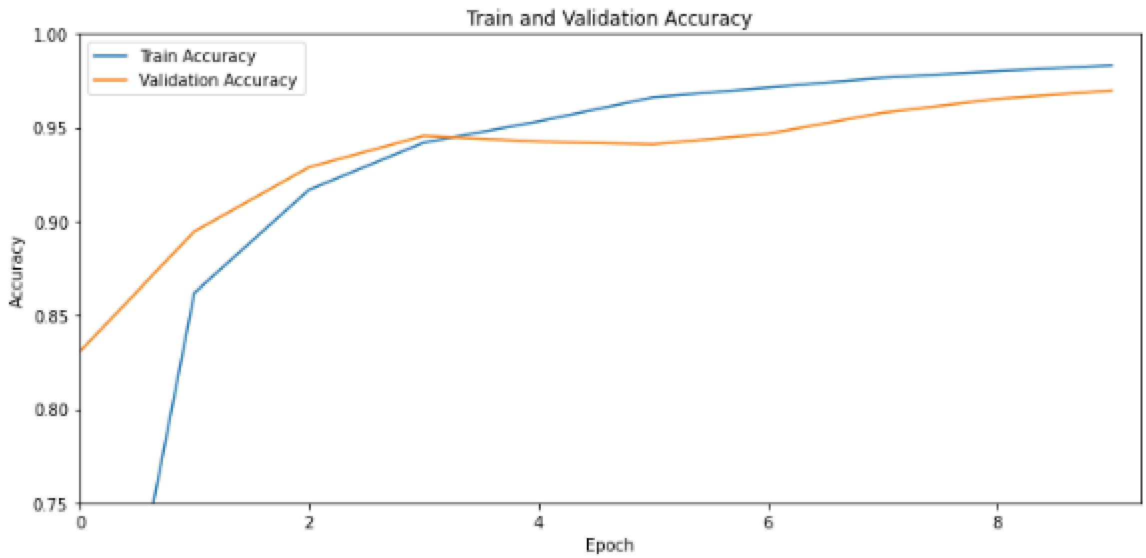
Precision score is a metric used to evaluate the quality of positive predictions made by the model. It is calculated as the ratio of true positives (correctly predicted positive instances) to the sum of true positives and false positives (incorrectly predicted positive instances). Precision measures how many of the positive predictions made by the model are actually correct.

4. **Recall Score**:

Recall score, also known as sensitivity or true positive rate, measures the model's ability to correctly identify positive instances. It is calculated as the ratio of true positives to the sum of true positives and false negatives (positive instances incorrectly classified as negatives). Recall quantifies how well the model captures all the positive instances in the dataset. A high recall score indicates a low false negative rate, implying that the model can identify most of the positive instances correctly.

V. RESULTS & DISCUSSIONS

Accuracy vs. Epochs graph



In this graph, accuracy refers to the proportion of correctly classified instances by the model. It is a commonly used evaluation metric to assess the overall performance of a classification model.

The epochs represent complete iterations through the training dataset during model training. The accuracy vs. epochs graph plots the accuracy values on the y-axis against the number of epochs on the x-axis.

Analyzing the accuracy vs. epochs graph helps understand how the model's accuracy evolves over the course of training. Initially, the accuracy may be low as the model makes random predictions. However, as the training progresses, the model learns and adjusts its parameters to improve its predictions, resulting in an increase in accuracy.

VI. CONCLUSION & FUTURE WORK

The plant leaf disease detection model utilizing a Convolutional Neural Network (CNN) yielded significant results and conclusions. The CNN-based model achieved high accuracy in accurately detecting and classifying plant leaf diseases. It showcased robust performance in distinguishing between healthy and diseased leaves, while also accurately identifying the specific type of disease present. Notably, the model demonstrated early disease detection capabilities, allowing for proactive intervention and treatment. This early detection can effectively prevent the further spread of diseases, minimize crop damage, and improve overall crop yield. The model's efficiency and speed were also noteworthy, providing rapid and efficient diagnosis based on leaf images. Its automated nature greatly reduces the time and effort typically required for manual observation and expert analysis, enabling farmers to make timely decisions and take appropriate actions. Importantly, the model's generalizability across different plant species and disease types highlights its versatility and applicability in diverse agricultural settings. Overall, these results underscore the effectiveness of CNNs in plant leaf disease detection, offering practical benefits and paving the way for future research and applications in the field.

While the plant leaf disease detection model using CNNs has proven to be effective, these are the future enhancements considered:

1. **Collaborating with Agricultural labs in Bangalore**: The proposed approach involves integrating specific weather and soil condition data obtained from agricultural labs in Bangalore. By incorporating this localized information, the model and application will provide accurate and relevant insights for farmers in Bangalore, enabling them to effectively monitor and combat plant diseases. This research aims to address the unique challenges faced by farmers in the Bangalore region and contribute to the advancement of precision agriculture practices in the area.

2. **User Interface Integration**: Integration of the plant leaf disease detection model with a Flutter application to enhance the user interface (UI) experience. By combining the robust disease detection capabilities of the model with the flexibility and user-friendly nature of Flutter, the application can provide an intuitive and interactive platform for farmers in Bangalore. This integration will enable farmers to easily capture and upload leaf images, receive real-time disease diagnosis results, and access actionable recommendations directly through the application.

3. **Collaboration and Data Sharing:** Encouraging collaboration and data sharing among researchers, farmers, and agricultural organizations can lead to larger and more diverse datasets. Sharing annotated datasets and collaborating on model development can accelerate research progress and facilitate the development of more effective plant leaf disease detection models.

4. **Dataset Expansion:** Continuously expanding and diversifying the dataset can improve the model's ability to detect a wider range of plant diseases and variations. Including images of diseased leaves at different stages of infection and from various sources can enhance the model's generalization capabilities.

VII. REFERENCES

1. Priti Badar, Suchitra.C "Disease Detection of Pomegranate plant using Image Processing"
   IJCSMC, Vol. 5, Issue. 5, May 2016

2. Amrj Bagha (Visual Paradigm). "Plant Disease Detection Sequence diagram" [Online]: https://online.visual-paradigm.com/community/share/plant-disease-detection-sequence-diagram-19z8kb2mem

3. Madhu Kirola, Neha Singh, Kapil Joshi, Sumit Chaudhary "Plants Diseases Prediction Framework: A Image-Based System Using Deep Learning"
   ResearchGate, Conference Paper, August 2022

4. Sharada P Mohanty, David P Hughes, Marcel Salathe "Using Deep Learning for Image-Based Plant Disease Detection"
   Frontiers in Plant Science, Vol-16, 22 September 2016

5. Shweta Lamba, Vinay Kukreja, Anupam Baliyan, Shalli Rani, Syed Hassan Ahmed "A Novel Hybrid Severity Prediction Model for Blast Paddy Disease Using Machine Learning"
   MDPI Journal, Published, 12th January 2023.

6. Paramasivam Alagumariappan, Najumnissa Jamal Dewan, Gughan Narasimhan Muthukrishnan, Bhaskar K. Bojji Raju, Ramzan Ali Arshad Bilal and Vijayalakshmi Sankaran "Intelligent Plant Disease Identification System Using Machine Learning"
   7th International Electronic Conference on Sensors and Applications, 14th November 2020

7. Amanda Ramcharan, Kelsee Baranowski, Peter McCloskey, Babuali Ahmed, Janes Legg, David P. Hughes "Deep Learning for Image-Based Cassava Disease Detection"
   Frontiers in Plant Science, Sec. Technical Advances in Plant Science, Volume - 8(2017)

8. Kishan Das Menon H, Dipali Mishra, Deepa D "Automation and integration of growth monitoring in plants (with disease prediction) and crop prediction"
Volume 43, Part 6 (2021)

9. Sk Mahmudal Hassan, Arnab Kumar Maji "Deep feature-based plant disease identification using machine learning classifier"
Springer Publications, 22 November 2022

10. Jayswal, H. S. and Chaudhari, J. P. (2020). , "Plant Leaf Disease Detection and Classification using Conventional Machine Learning and Deep Learning". International Journal on Emerging Technologies, 11(3): 1094–1102.

11. Noor, T.H.; Noor, A.; Elmezain, M., "Poisonous Plants Species Prediction Using a Convolutional Neural Network and Support Vector Machine Hybrid Model".
Electronics 2022, 11, 3690: https://doi.org/10.3390/electronics11223690

12. G. Owomugisha and E. Mwebaze, "Machine Learning for Plant Disease Incidence and Severity Measurements from Leaf Images," 2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA), Anaheim, CA, USA, 2016, pp. 158-163, doi: 10.1109/ICMLA.2016.0034.

13. Ms. Nilam Bhise, Ms. Shreya Kathet, Mast. Sagar Jaiswar, Prof. Amarja Adgaonkar, "Plant Disease Detection using Machine Learning", 2020 IRJET, ISO 9001:2008 Certified Journal, Volume: 07 Issue: 07, 07 July 2020

14. S. Ramesh et al., "Plant Disease Detection Using Machine Learning," 2018 International Conference on Design Innovations for 3Cs Compute Communicate Control (ICDI3C), Bangalore, India, 2018, pp. 41-45, doi: 10.1109/ICDI3C.2018.00017.

15. O. Kulkarni, "Crop Disease Detection Using Deep Learning," 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA), Pune, India, 2018, pp. 1-4, doi: 10.1109/ICCUBEA.2018.8697390.