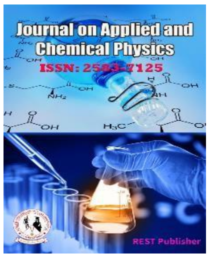
Journal on Applied and Chemical Physics

Vol: 4(4), December 2025

REST Publisher; ISSN: 2583-7125

Website: https://restpublisher.com/journals/jacp/

DOI: https://doi.org/10.46632/jacp/4/4/4



Farm Ease App Crop Information and Disease Prediction Using Machine Learning For Farmer Info

K. Nandini, C. Prakash Narayanan, K. Sharmila, *M. Archana Devi

P.S.V College of Engineering and Technology, Krishnagiri, Tamil Nadu, India.

*Corresponding author Email: deviarchu92@gmail.com

Abstract: *Crop disease prediction remains a major challenge in precision agriculture, and numerous methods have been developed and evaluated to tackle this problem. Because plant diseases are affected by factors such as climate, weather conditions, soil characteristics, fertilization practices, and seed varieties, accurate prediction requires the integration of multiple datasets. This demonstrates that plant disease prediction is a complex process involving several interconnected stages rather than a simple, direct task. Farmers are central to the agricultural ecosystem, and agriculture plays a vital role in national development by contributing significantly to a country's Gross Domestic Product (GDP). The overall performance of agriculture largely depends on farmers who cultivate and manage crops. To assist them, a real-time plant disease prediction prototype was developed using the Python programming language, integrating hybrid machine learning techniques with data analysis. Agricultural productivity is closely tied to economic growth, and due to the widespread occurrence of plant diseases, early detection is essential for sustaining the agricultural sector. If plant diseases are not addressed in a timely manner, they can severely affect crop quality and yield. This research proposes an automated image segmentation–based approach for detecting and classifying plant leaf diseases and provides a review of various disease classification techniques. Image segmentation, which is a crucial step in leaf disease detection, is carried out using a genetic algorithm.*

Index Terms: *Crop disease forecasting, precision agriculture, hybrid machine learning, data analysis, image segmentation, agricultural productivity, disease classification techniques, real-time prediction algorithm*

1. INTRODUCTION

The agricultural system is supported by farmers. As is well known, agriculture is an integral part of a country's development. Agriculture is crucial to India's economy and job market. One of the common problems faced by Indian farmers is the failure to select crops suitable for their soil. Another widespread problem Indian farmers are suffering because they have failed to protect their crops. plants from diseases in a timely manner. Many studies are being conducted to develop an accurate and effective crop prediction model. One of the techniques used in these types of studies is assembling. By using the prediction of this assembling technique, we can improve yield efficiency. When we need a high-performance prediction model, we use hybrid models with higher accuracy. Above all, the most effective model is determined through proper data cleaning. In this project, we used Python for both the backend and the frontend. This project provides a machine learning model in the form of a Jupiter Notebook. The business logic, implemented in Python, uses hybrid ensemble learning techniques to predict diseases in plants or their leaves. The proposed system integrates data obtained by clicking 2000 photographs and, by applying machine learning algorithms, predicts the most suitable crops according to the current environmental conditions. This provides young farmers with opportunities for various crops that can be cultivated. Machine learning refers to creating computer programs that can access data and learn automatically.

2. OBJECTIVE

Plant diseases have become a major problem, as they can cause a significant reduction in both the quality and quantity of agricultural products. Automatic detection of plant diseases is an essential research topic because it can be useful in monitoring vast crop fields and can help in automatically identifying disease symptoms on plant leaves as soon as they appear. The proposed system is a software solution for the automatic detection and classification of plant leaf diseases. This project consists of four main steps: first, a color transformation system is developed for the input RGB image; then, green pixels are masked and removed using a specific threshold value; subsequently, a segmentation process is performed; structural statistics for the useful segments are calculated; and finally, the extracted features are passed through a classifier.

A method for detecting and classifying leaf diseases based on concealing and removing green pixels, applying a specific threshold to extract the affected area, and calculating texture-based statistics using MATLAB for disease assessment.

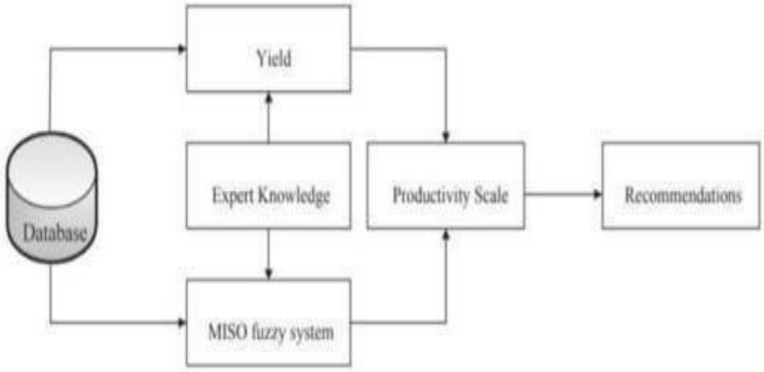


FIGURE 1. Fuzzy Model

3. LITERATURE REVIEW

This section provides a comprehensive review of current research on disease detection methods, covering publications up to 2024. The authors collected relevant research on machine learning (ML) and deep learning (DL) techniques for detecting and classifying plant diseases by searching several academic databases such as Scopus, ScienceDirect, Google Scholar, and Web of Science. To identify relevant research, they used keywords including machine learning, deep learning, classification, disease detection, healthy plant, and diseased plant. Following the initial search, they reviewed the bibliographies of selected articles to identify additional relevant works that might have been missed. The abstract of each research paper was reviewed by all co-authors to assess whether it met the inclusion criteria. This study excluded research that did not focus on machine learning algorithms. For example, documents primarily dealing with data collection methods such as unmanned aerial vehicles (UAVs) or unmanned ground vehicles (UGVs) instead of algorithm development were excluded. Additionally, publications in languages other than English, technical reports, review papers, and graduate theses were excluded. Finally, the authors selected 79 research papers: 65 focusing on classification techniques and 14 on object detection methods.

4. METHODOLOGY

System Analysis: Computer analysis involves Fact-finding and interpretation, problem identification, and data collection are all used to recommend improvements in a computer system. This is a problem-solving technique that involves extensive collaboration between computer users and computer developers. Computer analysis is concerned with understanding and integrating various factors to determine the optimal, or at least a satisfactory, solution or course of action. Since the current crop disease forecasting and detection system aims to meet the users' needs, significant importance was given to the system analysis phase. The initial study was conducted extensively, and several fact-finding techniques such as interviews, document searches, observation, and comparison were used to arrive at the best possible solution. The current process for this specific type was studied and analyzed in depth. All forms and other printed or unprinted formats for data collection were thoroughly checked, and the findings were compared. The flow of the same process in various other organizations was considered, and comparative studies were conducted, which greatly aided the system specification phase. Detailed observations were made to understand the difficulties in the process and the delays in obtaining results. A thorough study was conducted to gain a better understanding of the system.

Study Proposal: The research proposal was put forward only after a thorough analysis. This is because the current system used in the departments and its functioning within the department were extensively studied, and discussions were also held with department staff to identify which tasks could be automated.

Existing System: At present, plant disease detection is primarily carried out through direct visual inspection by experts, who identify and diagnose diseases based on their observations. This approach requires a large team of specialists and continuous monitoring of crops, making it highly expensive, especially for large-scale farms. Moreover, in many countries, farmers may lack access to agricultural experts or may not even be aware of such services. As a result, seeking expert assistance becomes both time-consuming and costly.
In this context, the proposed technique offers an effective solution for monitoring extensive crop areas. The existing system employs a fuzzy logic model combined with expert knowledge to predict the most suitable crop for a given agricultural land. Factors such as land characteristics, weather conditions, wind patterns, and expert experience are represented using fuzzy sets. The knowledge and experience of agricultural experts are used to formulate the final decision rules, while fuzzy logic processes multiple input variables to generate accurate and reliable outcomes.

Current System Architecture: The current system is widely used in the control mechanisms of various applications. It operates using fuzzy logic, which functions in three main stages: russification, rule evaluation, and defuzzification. After integrating the rules, three commonly used fuzzy inference methods (fuzzy controllers) can be employed: Mamdani, Sugeno, and Tsukamoto systems.

- A typical fuzzy rule with multiple inputs and one output can be expressed as follows:
- If Input1 is A1 and Input2 is A2, then Output is Y1; If Input1 is B1 and Input2 is B2, then Output is Y2.
- Here, A1, A2, B1, B2, Y1, and Y2 represent fuzzy sets, while Input1, Input2, and Output are fuzzy variables.
- The degree to which each rule is activated depends on the truth value of its associated conditions.

Disadvantages: Fuzzy logic-based systems require extensive hardware testing for verification and validation. Designing accurate fuzzy rules and membership functions is challenging, and fuzzy logic is often confused with probability theory due to similar terminology. Additionally, the rules in a fuzzy control system need to be continuously updated. Such systems cannot provide precise yield estimations, and developing the complete logic for a fuzzy expert system is a complex and time-consuming task.

Proposed Method: The proposed approach utilizes a hybrid integration technique to predict optimal crop yield. Instead of relying on multiple parameters, this system focuses on key factors essential for crop growth, such as soil fertility indicated by NPK values, pH level, temperature, average rainfall, and humidity. Based on these parameters, the system recommends the most suitable crop for a given soil type. As part of this work, a comparative analysis is conducted using popular classification algorithms to predict the most suitable crop. The algorithms considered include Logistic Regression, Decision Tree, and Random Forest, which helps in selecting the most accurate and efficient model for crop recommendation.

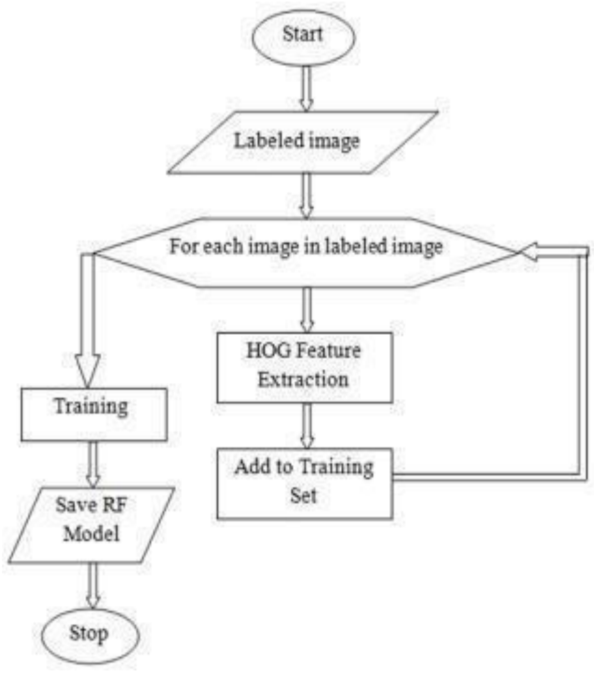


FIGURE 2. proposed System Forest classification, SVM, and Naive Bayes classification.

Advantages: The benefits derived from the new system are divided into three categories.

Direct Savings: When this plan is implemented, the time taken to develop a solution to address user problems will be reduced. Certain specific costs, such as those for postage and stationery, are reduced through improved processes. Such savings can be easily calculated or measured.

Measurable Benefits: When this project spawns as a product it will give the organization profit without further costs invested in the project.

Intangible Benefits: These are positive effects that are not easily quantifiable in monetary terms. The authors acknowledge that while many system implementations are clearly valuable, expressing their effects in financial terms is challenging. A common example is "better information"—which intuitively seems like a qualitative benefit. However, this section confidently argues that even these intangible benefits can often be broken down into measurable components and assigned monetary values. By thoughtfully examining what "better information" actually means in each specific context, the authors reinforce their belief that what initially seems immeasurable can often be quantified?

Return on Investment: This is the ultimate financial test for the viability of any system. Once both costs and benefits have been analysed, decision-makers must determine whether this investment makes business sense. This section emphasizes that to justify the system design and development costs, the system must generate sufficient returns. This is particularly relevant when a system or tool has the potential to be a standalone commercial product—whether it's a utility application or an entirely new software package. The underlying principle is straightforward: the financial benefits must outweigh the development costs. The authors note that their analysis addresses the financial and economic questions raised during the initial investigation phase, and they propose a structured approach to evaluating whether the project is a sound business proposition for the organization.

Feasibility Study: If a project is not feasible operationally, economically, and practically, it will not be considered a suitable product or automated system in all respects. The main activities in a feasibility study are as follows:
• Identifying the key characteristics of the system
• Determining the key output requirements, including response times
• Determining the types and estimated volumes of data
• Considering possible alternatives to meet user needs
• Studying other systems that meet similar needs

Operational Feasibility: Following are some points underlining the operational feasibility of the system. Some of the members in the organization though initially suppressive, started giving co-operation and support to the entire development phases. The different product report designers though functional at present, in practice requires a lot of decorations, incorporations and reformations. The clear advantage of using better user/developer friendly software increases the volume of package in the market. The customer support team was involved with the system right from the beginning and was always in touch with latest developments. The proposed system makes a best effect to satisfy the requirements of the end user/developer, keeping in mind certain infrastructure constraints. Even superficial, minor issues can escalate into a major problem later in the development cycle, so every possible aspect of functional feasibility was verified.

Technical Feasibility: Even if a project is desirable from many angles, it may sometimes be technically impossible. It is a regrettable truth that many desirable projects are handed over to the development team without any consideration of the ability to complete them. Being a developer to this project, some of the system-level software earlier Developed by us removes the infeasible solution to the technical part of it. The time is taken into consideration in which simultaneously more than one.

Project is assigned to me makes some difficulties to the full work schedule, sometimes the lack of technical knowledge for the innovation concepts from the project demand takes more time than the schedule. The technical issues generally raised during the investigation are discussed below:

1. This system has the capability to meet the requirements of the proposed system. It can also fulfil additional needs.
2. The proposed technology can satisfy the user/developer in a new dimension.
3. The proposed system will answer all type of enquires coming from customers for the organization.
4. This system is inherently very open and can be easily expanded to meet new needs that may arise in the future.

Financial and Economic Feasibility

Cost-Benefit Analysis: Costs are incurred in various areas. It is important to consider all the consequences of introducing new systems. Therefore, a cost analysis should include the following areas:

System Personnel: This includes not only the salaries of team members and other overhead costs, but also necessary specialized equipment, additional office space, computer test runs, and visits to other organizations. Even the cost of the feasibility study is included as an expense here.

User Personnel: This includes the time users spend providing the information required for the study, the time needed to implement potential solutions, and the time required to maintain the system after implementation. User costs are not significantly involved in this system.

Behavioural Feasibility: People are initially defiant to innovations and need enough training and advertisements that would result in lot of expenditure that is an additional expenditure for the organization. As the developers practice using the system, the in-flow of monetary benefits will be more advertisements to the organization, and the operation of the proposed system suffices the users to carry out their operations through the proposed system. The main objective of this feasibility study is examine operational, technical, economic viability. developing analysis and visualization software for scanning ports. This study includes the following:

- The concerned staffs were involved in planning the project.
- It eases the monitoring of scanning the ports in different ways.
- From the server as well as from the client to server.

5. ARCHITECTURE

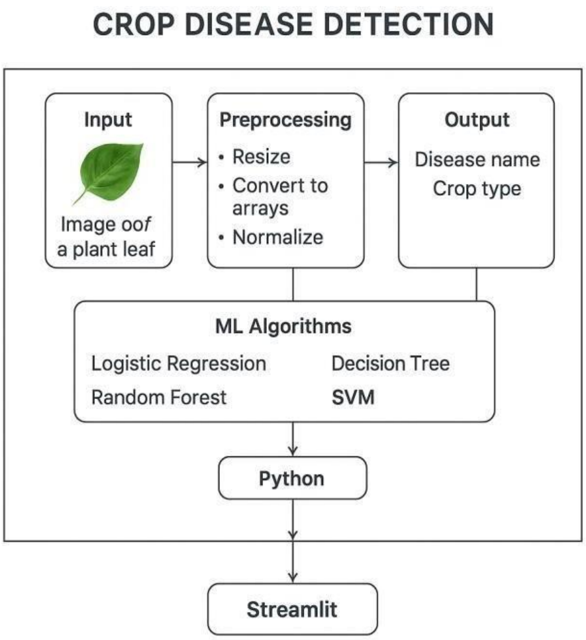


FIGURE 3. Architecture of Crop Disease Diagram

Description: This diagram illustrates a machine learning-based system developed for detecting plant diseases from images of leaves. The process begins with capturing an image of a plant leaf; this serves as the raw input for disease detection. Before being fed into the machine learning model, several pre-processing steps are performed to ensure image consistency and improve model performance. These steps include resizing the image to a standard dimension, converting the image into numerical arrays of pixel values, and normalizing the data to enable faster learning and better accuracy. The pre-processed images are then used to train and test the system using various machine learning algorithms such as Logistic Regression, Decision Trees, Random Forests, and Support Vector Machines. These algorithms classify the input images by learning patterns from the training dataset. As output, the model identifies the specific disease affecting the crop and also determines the crop type to which the leaf belongs. The entire system is implemented using the Python programming language, which supports image pre-processing, model training, and prediction. Additionally, Streamlet, a Python-based web framework, is used to create an interactive user interface that allows users to easily upload leaf images and receive disease predictions.

6. SYSTEM IMPLEMENTATION

System Modules

Image Acquisition and Pre-processing: This group collects photographs of plant leaves and prepares them for analysis. The dataset is obtained from sources like Kaggle, containing both healthy and diseased leaf images of various crops (e.g., tomato, potato, corn). Each image undergoes preprocessing using OpenCV to ensure uniformity and better learning efficiency. Functions Performed Are Loading input image (from dataset or user upload), Resizing images to a standard size, converting images into numerical arrays, explores broad concepts in evolutionary computation and their potential applications in agricultural disease diagnosis. The focus of this research is on the DiaMOS plant dataset, which represents a systematic approach to documenting plant health throughout the growing season. Specifically, the dataset captures images of a pear tree from February to July, providing a total of 3,505 images categorized as either fruits (499 images) or leaves (3,006 images). The fruit documentation tracks four distinct growth stages fruit set, stone fruit, fruit development, and ripening. Meanwhile, the leaf images are categorized into four stress levels leaf spot, leaf curl, snail damage, and healthy leaves. This comprehensive temporal and categorical coverage make the dataset particularly valuable for training classification models using machine learning and deep learning. In agriculture, especially for economies heavily reliant on agricultural productivity like India, the research this emphasizes the importance Teachers cite example little leaf disease in pine trees across the Southern United States, which kills infected trees within six years, illustrating how early automated detection can prevent catastrophic losses. Traditional visual inspection methods are labor-intensive, limited, and prone to inaccuracies, whereas automated image-based detection systems offer faster, more consistent, and scalable solutions. The authors emphasize that machine vision can enable automated process control, inspection, and guidance systems, which could revolutionize agricultural monitoring. By analyzing visual cues on plant leaves, these systems can provide cost-effective disease identification without requiring specialized human expertise at every location. Interestingly, the text shifts to a discussion of evolutionary computation principles, tracing their origins back to I. Rechenberg in the 1960s. The authors provide vivid examples of natural selection to illustrate adaptive algorithms: soldier ants that infiltrate unnamed colonies and respond to chemical signals, and rat snakes that have evolved different colors in different geographical regions to optimize camouflage and hunting efficiency in diverse environments ranging from urban areas to mountains and coastal regions. These biological examples serve as conceptual foundations for genetic algorithms, which are presented as optimization tools that mimic natural selection. Genetic algorithms operate by maintaining a population of candidate solutions, selecting the fittest individuals, and using them to generate offspring for subsequent generations. The expectation is that through this iterative process of selection and recombination, each new population will show improvement over the previous one, eventually converging on an optimal or near-optimal solution. The connection between evolutionary computation and plant disease detection suggests that genetic algorithms or similar evolutionary approaches this can be used to improve machine learning models. used for disease classification. This could involve evolving neural network architectures, optimizing feature selection, or fine-tuning classification parameters to improve detection accuracy on datasets like the DiaMOS plant dataset. Bio-inspiration from adaptive systems in nature provides a theoretical framework and computational methodology to address the practical agricultural challenge of automated disease detection.

- The genetic algorithm efficiently optimizes two variables.
- It searches from a large sample set of the cost surface.
- Many variables can be processed simultaneously.
- It can optimize variables with highly complex cost surfaces.
- It provides multiple optimal solutions, not just a single solution.

Feature Extraction and Model Training: In this module, the pre-processed images are converted into meaningful features that can be interpreted by machine learning algorithms. The system is trained using multiple machine learning models to learn patterns that distinguish healthy leaves from diseased ones. The key operations include extracting relevant features such as colour, texture, and shape, training different models, evaluating and comparing their accuracies, and finally selecting the best-performing model. Based on the evaluation results, Logistic Regression achieved the highest performance with an accuracy of 97% and was chosen as the final model.

Exploratory Data Analysis:

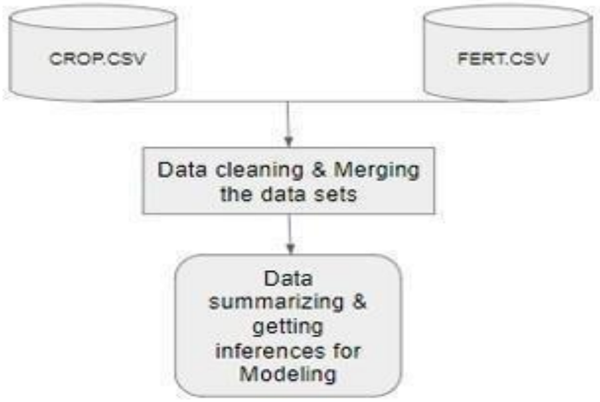


FIGURE 4. Exploratory Data Analysis

Advanced mathematical techniques form the foundation of machine learning; however, an equally important, but often underestimated, component of any data science project is exploratory data analysis (EDA). Although it is not always used extensively, EDA is a proven approach that helps to quickly and meaningfully understand new datasets. Insights can be efficiently gained by comparing the results of different analysis techniques across datasets. The primary objective of this phase is to explore the dataset, identify missing values and anomalies, and interpret underlying patterns using visual and quantitative methods. This analysis helps to determine logical next steps, potential problems, and areas that require further investigation. Key steps in data analysis include identifying variables and data types, performing correlation analysis, conducting descriptive statistics, applying variable transformations, and handling missing values.

Disease Prediction and Output Generation: This module manages the core prediction functionality of the system. Once the machine learning model is trained, it is able to detect diseases from new leaf images uploaded by users. The system predicts both the crop type and the specific disease, which are provided as the final output. The operations performed in this module include accepting a new leaf image from the user, applying pre-processing steps such as resizing and normalization, feeding the processed image to the trained model, generating predictions (for example, "Tomato leaf – bacterial spot"), and clearly displaying the results to the user.

User Interface and Deployment: This module focuses on creating a user-friendly interface using Streamlet, which ensures that the system is easily accessible to end users such as farmers and researchers. It also supports backend integration with SQL Server to store uploaded images and prediction history. The main functions of this module include creating an interactive graphical user interface for image upload and displaying results, integrating a machine learning model with Streamlet for real-time predictions, connecting to SQL Server for data storage, and visualizing prediction results and model accuracy

7. ALGORITHMS

Logistic Regression: Logistic regression is one of the commonly used classification techniques. It is designed to predict a binary outcome based on a set of independent variables and is categorized as a linear model. Linear models include both linear regression and logistic regression. While linear regression is applied to regression problems where the target variable is continuous or numerical, logistic regression, despite being linear in nature, is used for classification tasks. In particular, binary logistic regression classifies data into one of two possible classes. It models the relationship between the input variables and the probability of a class using an S-shaped (sigmoid) curve, which maps the output to values between 0 and 1. Based on this probability, the model internally determines the final class prediction.

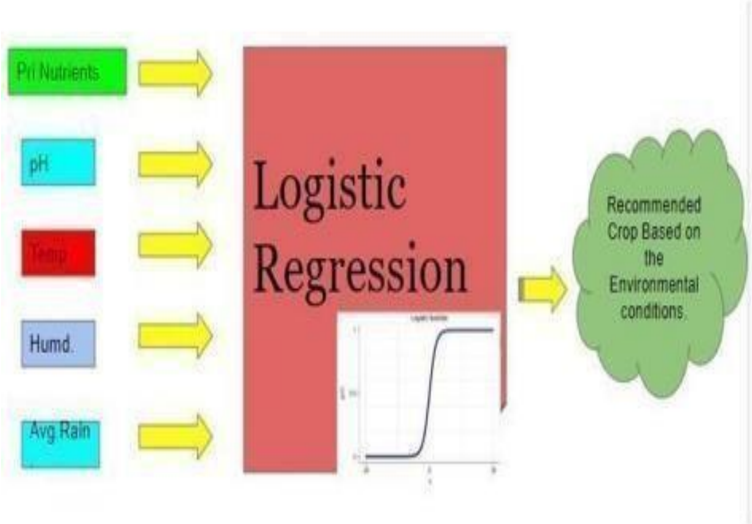


FIGURE 5. Logistic Regression

Justification: In our problem, the features include primary nutrients, pH, temperature, humidity, and average rainfall, and our target is the label representing the name of the crop. All of these will be provided as input to our logistic regression model, and our model will be trained based on the previous dataset. Since our data follows a normal distribution, we have achieved an accuracy of 97.00% using the logistic regression model.

Decisions Tree Model:

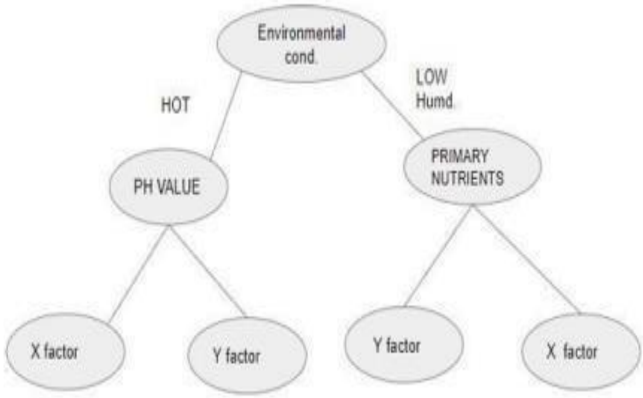


FIGURE 6. Decision tree model

In real life, trees are often used as metaphors and have inspired many concepts, including machine learning techniques for classification and regression. Decision trees are a popular machine learning model that can achieve high accuracy on various tasks while being relatively easy to understand. What distinguishes decision trees from many other models is the interpretability of their knowledge representation. Once a decision tree is trained, its learned knowledge is explicitly organized in a hierarchical structure, making the decision process clear and visually intuitive. This structured representation allows even non-experts to understand how decisions are made. The process of building a decision tree is called induction, which involves defining hierarchical decision boundaries based on the training data. However, due to their training nature, decision trees are prone to overfitting, where the model learns noise instead of general patterns. To address this issue, pruning is used, which removes unnecessary branches from the tree. Pruning reduces model complexity, helps prevent overfitting, and further improves interpretability.

At a high level, decision tree induction follows four main steps:

- Start with a training dataset containing feature variables and the corresponding classification or regression outcomes.

- Identify the best attribute for splitting the data, which determines how the dataset is partitioned.
- Create a tree node based on this split, where each node represents a decision point centred around a specific attribute.
- Recursively repeat the process on each data subset, continuing until maximum accuracy is achieved with a minimum number of splits or nodes.

A random forest, as the name suggests, is an ensemble model consisting of multiple individual decision trees working together. In random forests, each tree in the forest generates a class prediction, and the final output of the model is determined by majority voting among all the trees. By combining the predictions of multiple trees, random forest models improve accuracy, reduce overfitting, and provide more robust predictions compared to a single decision tree.

8. EXPERIMENTAL RESULT

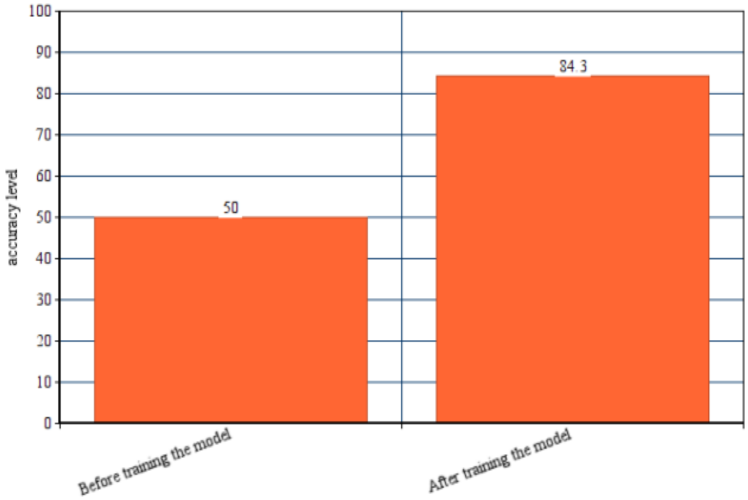


FIGURE 7. Random Forest Classification

The bar chart shows the improvement in model performance for crop disease detection before and after training. Prior to training, the model achieves an accuracy of 50%, which is equivalent to random guessing. This indicates that without learning from labelled crop images, the model is unable to effectively distinguish between healthy and diseased plants. After training, the accuracy increases significantly to 84.3%. This improvement demonstrates that the model has successfully learned important visual features from the training dataset, such as disease-related patterns on leaves, including spots, discoloration, and texture variations. As a result, the trained model can classify crop images into healthy and diseased categories with much higher accuracy. Overall, the graph highlights the importance of machine learning–based training in crop disease detection systems, as proper training enables the model to generalize from data and provide reliable predictions that support early disease detection and better crop management.

9. SYSTEM TESTING

Testing is a vital phase in the software development life cycle. A test environment consists of a set of input data used to evaluate how well a system performs its intended functions. The primary objective of testing is to identify defects; therefore, test data is carefully designed to ensure that the system processes inputs correctly. Software testing plays a key role in quality assurance by evaluating the system's specifications, design, and implementation. As software systems become more complex and the cost of failures increases, thorough planning and systematic testing have become increasingly important.

Testing Objectives:

Effective testing helps identify defects in the software and verifies whether the system functions according to its specifications. It also ensures that performance requirements are satisfied and that the software behaves as expected under different conditions.

Program Testing:

Programs are generally tested in three main ways:

1. Correctness testing
2. Performance testing
3. Computational complexity testing

Correctness testing verifies whether the program performs the intended task accurately. This is often challenging, particularly for large and complex systems. Performance testing focuses on improving execution speed and reducing memory usage, often involving optimization of the program implementation. Computational complexity testing involves empirical analysis of an algorithm's efficiency or comparison of multiple algorithms that solve the same problem.

Test Documentation

A comprehensive testing plan should include:

- Preventive measures
- Random testing strategies
- Testing of all program components
- Test data preparation
- Troubleshooting procedures
- Time allocation for testing
- Retesting plans

Software development involves continuous activities where human errors are inevitable. Since complete accuracy in development and communication is difficult to achieve, quality assurance practices are essential. Testing involves executing the program with the goal of detecting defects and serves as the final evaluation of specifications, design, and code. Due to the unique challenges involved in software testing, the proposed system is tested at multiple levels before being approved for user acceptance testing. A test is considered successful when it reveals previously undiscovered errors, thereby contributing to the overall reliability and quality of the system.

Testing Objectives: Testing is the process of executing a program with the primary aim of identifying errors. An effective testing strategy increases the likelihood of discovering previously undetected defects. A test is considered successful when it reveals an error that was not identified earlier.

Testing Principles

- Tests should be well planned in advance and aligned with end-user requirements.
- Testing should begin on a small scale and gradually expand to larger and more complex scenarios.
- Exhaustive testing of all possible cases is not feasible.
- To maximize effectiveness, testing should ideally be carried out by an independent third party.
- The main goal of test case design is to create test cases that have a high probability of uncovering defects.
- To achieve this objective, different test case design approaches are required.
- Software testing is broadly classified into **White-Box testing** and **Black-Box testing**.

White-Box and Black-Box Testing

White-Box Testing focuses on the internal structure and logic of the program. Test cases are designed to ensure that every statement is executed at least once and that all logical paths and conditions are thoroughly tested. This approach verifies the internal working of the software.

Black-Box Testing, on the other hand, ignores the internal implementation and concentrates on validating the functional requirements of the system. It emphasizes input and output behaviour, where test cases are generated by partitioning input and output domains to achieve comprehensive coverage.

Acceptance Testing: Acceptance testing is performed to confirm that the system is ready for deployment and operational use. This phase begins after system development is complete and aims to assure end users that the software meets all specified requirements. It includes functional, performance, and stress testing to validate system behaviour under real-world conditions. Several tools support acceptance testing. A test coverage analyser tracks which control paths are executed by test cases, while a timing analyser (profiler**)** measures execution time to identify performance bottlenecks. Additionally, code standard checkers and static analysis tools are used to ensure compliance with coding standards and best practices. Test cases are designed to ensure that all program statements are executed at least once and that all logical conditions are tested for both true and false outcomes. Using white-box testing techniques, software engineers can develop test cases that:

- Verify both true and false outcomes of logical decisions
- Test loops at boundary conditions as well as within normal operating ranges
- Validate the accuracy and integrity of internal data structures

This systematic approach ensures reliability, correctness, and readiness of the software for end-user deployment.

10. CONCLUSION

This project utilizes machine learning techniques to evaluate outcomes using classification algorithms such as Random Forest, Logistic Regression, Decision Tree, Support Vector Machine (SVM), and Naive Bayes. Except for SVM, the remaining algorithms in the proposed model demonstrate improved performance in crop disease and yield prediction. The classification results exhibit enhanced dataset performance, leading to the conclusion that the proposed model is more efficient than current approaches for crop disease detection. Implementing this method can significantly improve agricultural practices by helping farmers reduce crop losses and increase productivity, thereby strengthening national agricultural resources. Furthermore, this model can be extended by integrating it with other agricultural sectors such as horticulture and sericulture, contributing to increased overall agricultural production.

Future Enhancements: To further improve the predictive accuracy and scalability of the proposed approach, several enhancements can be considered. Since the current system provides a general framework, future work will focus on applying this method to datasets collected from various countries to ensure broader applicability and robustness. Additionally, incorporating multi-class classification in the prediction phase will significantly improve crop yield estimation and disease detection accuracy. This extension represents a promising direction for future research and development.

REFERENCES

[1] Emmanuel Moupojou, "FieldPlant: A Dataset of Field Plant Images for Plant Disease Detection and Classification With Deep Learning" Research Article in IEEE Access,2023.

[2] Kizito Nkurikiyeyezu, Jean Pierre Nyakuri, Celestin Nkundineza and Omar Gatera," State-of-the-Art Deep Learning Algorithms for of Things-Based Detection of Crop Pests and Diseases: A Comprehensive Review",2024.

[3] K. Suganthi, R. Karthik and Armaano Ajay " A Deep Learning Approach for Crop Disease and Pest Classification Using Swin Transformer and Dual- Attention Multi-Scale Fusion Network", 2024.

[4] Lijuan Zheng, Ei Chen and Jiping Xiong "Algorithm for Crop Disease Detection Based on Channel Attention Mechanism and Lightweight Up- Sampling Operator", 2024.

[5] Nikolaos Ploskas, Vasileios Balafas and Emmanouil Karantoumanis " Machine Learning and Deep Learning for Plant Disease Classification and Detection", 2023

[6] Sitelbanat Abdelbagi, "Internet of Things Assisted Plant Disease Detection and Crop Management Using Deep Learning for Sustainable Agriculture" Research Article in IEEE Access, 2025.

[7] Shujuan Zhang, Lili Li, and Bin Wang, " Plant Disease Detection and Classification by Deep Learning_A Review", 2021.

