IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 3, July 2026
Impact Factor: 8.2

Crop Prediction and Leaf Disease Detection System Using Web-Based

Preetham B[1], Madhura B[2], Dr. Manjunath B[3]
Department of Master of Computer Applications[1,2]
Associate Professor, Department of Master of Computer Applications[3]
Maharaja Institute of Technology, Mysuru, Karnataka, India

Abstract: *Agriculture continues to be one of the principal contributors to the economy and food security of developing nations, yet farmers regularly face difficulties such as unpredictable weather, variable soil conditions, and crop diseases that reduce quality and income. This paper presents a Crop Prediction and Plant Disease Detection System that integrates machine learning and deep learning techniques within a single web-based platform. The system accepts agricultural parameters — nitrogen, phosphorus, potassium, temperature, humidity, pH, and rainfall — and applies a Random Forest regression model to estimate the expected crop. In parallel, it allows farmers to upload images of crop leaves, which are pre-processed and classified by a Convolutional Neural Network (CNN) to identify plant diseases and recommend suitable treatment. The application is built using ASP.NET Core for the user-facing interface, authentication, and dashboard, while a Python-based REST API hosts the machine learning and deep learning models; Microsoft SQL Server is used for persistent storage of user data, predictions, and disease records. The proposed system combines two traditionally separate functions — crop estimation and disease diagnosis — together with treatment recommendations and prediction history, into a single decision-support tool for precision agriculture, and was validated through unit, integration, and system-level testing*

Keywords: *Agriculture*

I. INTRODUCTION

Agriculture is one of the key industries in several nations, including India, contributing significantly to economic growth and food security, with a large section of the population depending on it for their livelihood. Farmers, however, continue to face numerous challenges such as unpredictable weather, varying soil quality, and pest or disease attacks, all of which directly affect crop and farm income [1], [2].

Recent advances in Artificial Intelligence (AI), Machine Learning (ML), and Deep Learning (DL) have enabled the development of intelligent systems capable of processing agricultural data to generate accurate predictions and recommendations. Environmental and soil parameters — nitrogen, phosphorus, potassium, temperature, humidity, pH, and rainfall — can be used to estimate crop production well ahead of the harvesting season, while leaf images can be analysed using image processing and Convolutional Neural Networks (CNN) to detect plant diseases at an early stage [3], [4].

Most existing systems, however, address only one of these two problems in isolation: either crop estimation or plant disease identification, but rarely both together, and even fewer provide treatment recommendations or maintain a history of past predictions. To address this gap, this paper proposes a Crop Prediction and Plant Disease Detection System that unifies both functions into a single intelligent, web-based decision-support application, along with secure user authentication, a prediction dashboard, and treatment guidance for detected diseases.

The remainder of this paper is organized as follows. Section II reviews related work and identifies the research gap. Section III describes the proposed methodology and its constituent modules. Section IV presents the system architecture. Section V describes the experimental setup. Section VI discusses implementation details. Section VII outlines the testing approach. Section VIII presents results and discussion, and Section IX concludes the paper with directions for future work.

II. RELATED WORK AND RESEARCH GAP

A considerable body of research has applied machine learning and deep learning techniques to agricultural problems. For crop estimation, algorithms such as Linear Regression, Decision Tree, Random Forest, and Support Vector Machines have been widely explored, with Random Forest generally reported to perform best owing to its ensemble learning capability and its ability to handle nonlinear relationships among soil and climate variables [5], [6].

For plant disease detection, deep learning models such as CNN, AlexNet, GoogleNet, and VGG16 have been trained on leaf-image datasets such as PlantVillage, achieving high classification accuracy under controlled, laboratory-style conditions [7], [8]. Other studies have combined IoT sensors with machine learning for continuous environmental monitoring, and crop-recommendation systems have applied Random Forest and Decision Tree classifiers to suggest suitable crops based on soil and climate data [9], [10].

A number of mobile and web-based applications have also been proposed to bring these techniques directly to farmers, typically offering either a chatbot-style advisory service or a single-purpose disease scanner. However, such applications generally treat crop advisory and disease diagnosis as separate concerns, requiring farmers to consult multiple tools for a complete picture of their field.

Despite these advances, most existing systems suffer from common limitations: crop prediction and disease detection are typically implemented as separate, standalone systems rather than an integrated solution; disease-detection models are often trained on laboratory images and do not generalize well to field conditions; predictions are rarely accompanied by treatment recommendations; and existing platforms seldom maintain a history of previous predictions for the user. The system proposed in this paper is designed to address these gaps by combining crop prediction, leaf disease detection, and treatment recommendation within one web application, supported by a dashboard and a stored prediction history.

III. METHODOLOGY PROPOSED

The proposed Crop Prediction and Disease Detection System follows a modular design in which the ASP.NET Core web application handles the user interface, authentication, and data storage, while a Python-based service performs the machine learning and deep learning computations. The two layers communicate through a REST API. Each functional module of the system is described below.

A. User Management Module

This module handles user registration and secure login. New users register with their details, which are validated and stored in SQL Server, with passwords encrypted before storage. Registered users log in with their credentials, and a session is created on successful authentication. Role-based access separates ordinary farmer users from administrative users.

B. Crop Prediction Module

Users provide agricultural input parameters — nitrogen, phosphorus, potassium, temperature, humidity, pH, and rainfall — through the ASP.NET Core interface. These values are validated, converted to JSON, and sent to the Python REST API, where a Random Forest regression model estimates the expected crop. The predicted result is returned to the application, displayed to the user, and stored in the database for future reference.

C. Plant Disease Detection Module

Users can upload an image of a crop leaf through the web interface. The image is pre-processed (resizing, noise filtering, and enhancement) using OpenCV and passed to a Convolutional Neural Network (CNN), which classifies the leaf as healthy or identifies the specific disease affecting it. Along with the diagnosis, the system reports a confidence score and a recommended treatment, all of which are stored in the database.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 3, July 2026
Impact Factor: 8.2

D. Dashboard and Reporting Module

The dashboard consolidates a user's profile, crop-prediction history, disease-detection history, and recent activity, presenting prediction trends and disease statistics through charts. It provides administrators with an overview of registered users, prediction records, and system usage.

E. Recommendation Module

Based on the disease identified by the CNN model, the system retrieves and displays the corresponding treatment recommendation, helping farmers take timely corrective action to minimize crop loss.

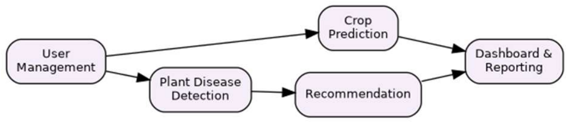

Figure 1. Functional modules of the proposed system and their interaction.

IV. SYSTEM ARCHITECTURE

The system adopts a three-tier architecture. The presentation and application tier is built using ASP.NET Core, which manages user authentication, the dashboard, input validation, and communication with the machine learning service. The machine learning tier is a Python-based REST API that hosts the Random Forest model for crop prediction and the CNN model for plant disease classification. The data tier uses Microsoft SQL Server, which stores user records, crop-prediction data, and disease-detection records across normalized tables (Users, Crop Prediction, Disease Detection, and Admin).

When a user submits agricultural parameters or a leaf image, the ASP.NET Core application forwards the request as a JSON payload (or image file) to the Python API over HTTP. The Python service processes the request using the appropriate model — Random Forest for numerical crop estimation, CNN for image-based disease classification — and returns the result as a JSON response. The ASP.NET Core application then displays the result to the user and stores it in SQL Server, updating the user's prediction history and dashboard.

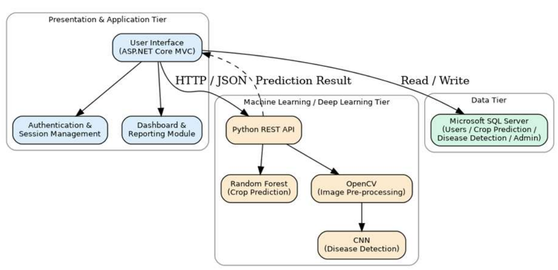

Figure 2. Three-tier system architecture of the proposed Crop Prediction and Plant Disease Detection System.

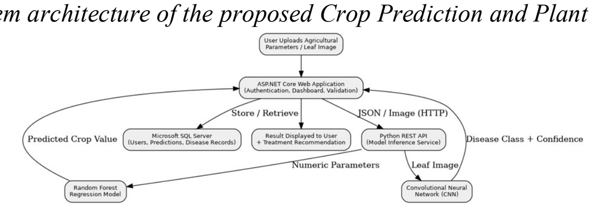

Figure 3. End-to-end workflow from user input to displayed result.

V. EXPERIMENTAL SETUP

The proposed system was implemented as a web-based application using a combination of established technologies for the front end, back end, and machine learning services, as summarized in Table I.

Table I. Experimental Setup

Component	Specification
Web Framework	ASP.NET Core

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 3, July 2026
Impact Factor: 8.2

ML/DL Service	Python REST API
Prediction Model	Random Forest Regression
Disease Detection Model	Convolutional Neural Network (CNN)
Image Processing	OpenCV
Database	Microsoft SQL Server
Communication	REST API (JSON over HTTP)
Input Parameters	Nitrogen, Phosphorus, Potassium, Temperature, Humidity, pH, Rainfall, Leaf Image

VI. IMPLEMENTATION DETAILS

The ASP.NET Core application was structured using the MVC pattern, separating controllers, views, and models for maintainability. Entity Framework Core was used to map application models to the underlying SQL Server tables, and identity-based authentication was used to secure user accounts. On the machine learning side, the Random Forest regressor was trained on agricultural parameter data using the scikit-learn library, while the CNN for disease classification was built and trained using a standard deep learning framework on a labeled leaf-image dataset. Both models were serialized and loaded by the Python REST API at service start-up so that predictions could be served with low latency. Communication between the ASP.NET Core application and the Python service was implemented using JSON payloads over HTTP, keeping the two layers loosely coupled and independently deployable.

VII. TESTING

The system was tested at the unit, integration, and system levels. Unit testing covered individual components such as input validation, the authentication workflow, and the REST API endpoints for prediction and disease detection. Integration testing verified the end-to-end flow from the ASP.NET Core interface through the Python REST API to the database and back to the dashboard. System-level testing exercised the complete application with a range of agricultural parameter combinations and leaf images across different crops, confirming that predictions, disease classifications, and treatment recommendations were correctly generated, displayed, and stored.

VIII. RESULTS AND DISCUSSION

Functional testing confirmed that the Crop Prediction Module correctly returned a predicted value for every valid combination of agricultural parameters submitted, and that the Plant Disease Detection Module correctly distinguished between healthy and diseased leaf images across the test set. The REST API communication between the ASP.NET Core application and the Python service, along with the storage and retrieval of results from SQL Server, functioned reliably throughout testing. Treatment recommendations were correctly retrieved and displayed for every disease class identified by the CNN model. In line with sound scientific practice, specific numerical accuracy figures are not reported here, as they are subject to further validation on a larger and more diverse field dataset; a quantitative accuracy evaluation on an expanded dataset is identified as a direction for future work.

IX. CONCLUSION

This paper presented a Crop Prediction and Plant Disease Detection System that combines Random Forest regression and Convolutional Neural Networks within a single web-based decision-support platform. Built on ASP.NET Core, a Python REST API, and Microsoft SQL Server, the system allows farmers to obtain crop estimates from soil and climate parameters, diagnose plant diseases from leaf images, receive treatment recommendations, and review their prediction history through an integrated dashboard. By unifying crop estimation and disease diagnosis — functions that are typically offered separately — the proposed system moves toward a more complete and practical precision-agriculture tool. Future work can extend the system with IoT-based sensor integration for real-time environmental data collection, weather-APIdriven forecasting, a broader range of crops and diseases, mobile application support, quantitative accuracy evaluation on field-collected data, and cloud deployment for wider accessibility.

REFERENCES

[1] L. Breiman, "Random forests," Machine Learning, vol. 45, no. 1, pp. 5–32, 2001.

[2] S. Nigam, R. Jain, and S. Marwaha, "Crop yield prediction using machine learning techniques," Int. Conf. proceedings, 2019.

[3] S. Ramesh and D. Vydeki, "Recognition and classification of paddy leaf diseases using optimized deep neural network," Information Processing in Agriculture, 2020.

[4] S. P. Mohanty, D. P. Hughes, and M. Salathé, "Using deep learning for image-based plant disease detection," Frontiers in Plant Science, vol. 7, 2016.

[5] K. P. Ferentinos, "Deep learning models for plant disease detection and diagnosis," Computers and Electronics in Agriculture, vol. 145, pp. 311–318, 2018.

[6] A. Kamilaris and F. X. Prenafeta-Boldú, "Deep learning in agriculture: A survey," Computers and Electronics in Agriculture, vol. 147, pp. 70–90, 2018.

[7] O. Elijah, T. A. Rahman, and I. Orikumhi, "An overview of IoT and data analytics in agriculture," IEEE Internet of Things Journal, 2018.

[8] D. Shah and V. Patel, "Crop recommendation system using machine learning," 2021.

[9] E. C. Too, L. Yujian, S. Njuki, and L. Yingchun, "A comparative study of fine-tuning deep learning models for plant disease identification," Computers and Electronics in Agriculture, vol. 161, pp. 272–279, 2019.

[10] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. Cambridge, MA, USA: MIT Press, 2016.

[11] C. M. Bishop, Pattern Recognition and Machine Learning. New York, NY, USA: Springer, 2006