International Journal of Science, Strategic Management and Technology
Volume **02** Issue **11** November-2026 | ISSN: **3108-1762 (Online)** | Impact Factor: 3.8
An International, Peer-Reviewed, Open Access Scholarly Journal Indexed in recognized academic databases

AI Smart Agriculture System

Pradeep Gowda
Head of the Department
Dept. of CSE
Sri Revana Siddeshwara Institute of Technology, Chikkajala, Banglore 562157

Ushashree B N
Assistant Professor
Dept. of CSE
Sri Revana Siddeshwara Institute of Technology, Chikkajala, Banglore 562157
Ushashreebn@gmail.com

G Harshavardhan
Dept. of CSE
Sri Revana Siddeshwara Institute of Technology, Chikkajala, Banglore 562157
gharshavardhangouda@gmail.com

Manu B
Dept. of CSE
Sri Revana Siddeshwara Institute of Technology, Chikkajala, Banglore 562157
14manu444@gmail.com

Raj Kumar AB
Dept. of CSE
Sri Revana Siddeshwara Institute of Technology, Chikkajala, Banglore 562157
rajkumarballolli4622@gmail.com

Siddhu G
Dept. of CSE
Sri Revana Siddeshwara Institute of Technology, Chikkajala, Banglore 562157
siddugowda760@gmail.com)

https://doi.org/10.55041/ijsmt.v2i11.024

Cite this Article: Gowda, P., N, U. B., Harshavardhan, G., B, M., AB, R. K. & G, S. (2026). AI Smart Agriculture System. International Journal of Science, Strategic Management and Technology, *02*(11), 1-9. https://doi.org/10.55041/ijsmt.v2i11.024

License: This article is published under the Creative Commons Attribution 4.0 International License (CC BY 4.0), permitting use, distribution, and reproduction in any medium, provided the original author(s) and source are properly credited.

Abstract— This paper presents an AI-based application that brings several common agricultural support tasks into one place. The system combines plant disease identification from images, crop prediction from soil and weather-related values, fertilizer suggestions, soil fertility assessment and a question-answering assistant. It is built with Python and Stream lit. A MobileNetV2-based convolutional neural network is used for plant-image classification, while a Random Forest model predicts suitable crops from nitrogen, phosphorus, potassium, temperature, humidity, pH and rainfall values. Fertilizer and soil-fertility outputs are generated using predefined rules. An Ollama-powered assistant provides general responses to farming questions. The project report records approximate accuracies of85–90% for disease detection and 88–92% for crop prediction. These figures are presented as project-reported results and need further testing with varied field data. The application is intended to support farmers' decisions, not replace expert advice or laboratory soil analysis

.Index Terms— Smart agriculture, plant disease detection,MobileNetV2, convolutional neural network, Random Forest, crop prediction, fertilizer recommendation, soil fertility, Stream lit..

I. INTRODUCTION

Background and Motivation:. Agriculture is affected by many factors, including soil condition rainfall, temperature, water availability, pests and plant diseases

.Farmers often have to consider several of these factors at the same time, and useful information may be spread across different tools or sources. This can make it difficult to identify a problem early or compare possible crop choices.

Project Objective and Scope: Plant diseases can reduce both crop yield and quality. A visual check may not always be enough to distinguish diseases with similar symptoms. Fertilizer decisions also depend on nutrient levels in the soil; applying too much or too little can increase costs and affect soil health. Crop selection, meanwhile, is influenced by nutrients and conditions

Contribution: Recent developments in machine learning and computer vision offer practical ways to analyse plant images and agricultural data. This project brings several such functions together in a single Stream lit application. Its purpose is to provide accessible decision-support in formation while leaving final farming decisions with the user and qualified agricultural professionals.

.

II. PROBLEM DEFINITION & OBJECTIVES

Farmers may experience delays in identifying plant diseases, uncertainty when choosing crops, and difficulty deciding how much fertilizer to apply. These tasks are often handled using separate tools. The project addresses this issue by developing one application that combines image-based disease detection, crop prediction, fertilizer recommendations, soil-fertility analysis and general agricultural guidance.

The main objectives are to support early identification of possible plant diseases; suggest crops using soil and environmental inputs ; provide fertilizer guidance based on nutrient values; classify soil fertility from NPK readings; and make general farming information easier to access through a simple interface.

© Author(s). This work is peer-reviewed, openly published, and permanently archived
This article is openly accessible and reusable with proper attribution.

III. PROPOSED SYSTEM

The application contains five main modules. The disease-detection module analyses an uploaded plant image using a MobileNetV2-basedCNN. The crop-prediction module uses a Random Forest classifier .Fertilizer suggestions are produced through predefined nutrient-related rules, while the soil-analysis module interprets nitrogen, phosphorus and potassium values. A conversational assistant, powered by Ollama, responds to general farming questions Stream lit connects these functions through a common user interface.

IV. METHODOLOGY

A. Plant Disease Detection

The user uploads a plant image in JPG or PNG format. The image is resized to 224 × 224 pixels, converted to RGB and normalized so that pixel values fall between 0 and 1. A MobileNetV2-based CNN then extracts image features and predicts a possible disease category with a confidence score.

The project report identifies the PlantVillage dataset as the source used for disease-detection work.

B. Crop Prediction

The crop-prediction module uses seven input features: nitrogen, phosphorus, potassium, temperature, humidity, pH and rainfall. A Random Forest classifier combines the predictions of multiple decision trees and returns the crop class receiving the most votes. There port identifies crop_recommendation.csv as the dataset used for this module.

C. Fertilizer and Soil Analysis

The fertilizer module compares nutrient values with predefined conditions and provides a corresponding recommendation. The soil-analysis module uses NPK values to assign a fertility category. Since these outputs depend on the rules and thresholds configured in the application, they should be treated as preliminary guidance rather than a substitute for laboratory testing or advice from an agronomist.

D. Agricultural Assistant

The assistant uses Ollama to respond to general questions about farming practices, crop selection and disease prevention. Its responses are intended to provide accessible information. They may not account for a particular farm's location, soil history, crop variety or current field conditions.

E. Application Workflow

A user opens the Stream lit application and selects the required function. For disease detection, the user uploads an image. For crop prediction and soil-related functions, the user enters the requested numerical values. The relevant model or rule-based module processes the input and displays its result. The project also includes SQLite for local data storage where database persistence is configured.

V. IMPLEMENTATION

The system is developed in Python, with Streamlit used to build the interface. TensorFlow/Keras supports the deep-learning component, Scikit-learn is used for the Random Forest classifier, and image processing. SQLite3 provides local database functionality, while Ollama powers the conversational assistant.

The modules are kept separate so that each function can be maintained or improved without redesigning the entire application.

VI. TESTING AND PERFORMANCE

The project report describes tests covering plant-image upload and disease prediction, crop prediction from numerical inputs, fertilizer recommendations, soil-fertility classification and chatbot responses. It reports disease-detection accuracy of approximately 85–90% and crop-prediction accuracy of approximately 88–92%. Image processing is reported to take around 2–3 seconds, while crop prediction takes less than one second.

These figures should be interpreted with care. The report does not provide a detailed train-test split, confusion matrix or independent field-validation results for the stated accuracy ranges. Actual performance may change with image quality, dataset coverage, crop variety, input accuracy and growing conditions. Further testing with data collected from different farms and regions is therefore necessary.

© Author(s). This work is peer-reviewed, openly published, and permanently archived
This article is openly accessible and reusable with proper attribution.

VII. RESULTS AND DISCUSSION

The implemented application brings the five agricultural support functions into one interface. Users can upload a plant image or enter soil and environmental values, then view the prediction or recommendation produced by the selected module. The project report records successful operation of the listed test cases and describes the interface as simple and responsive.

The main contribution is the integration of several functions in a single software platform without requiring dedicated IoT hardware. At the same time, the reliability of the recommendations depends on the quality of the underlying models, datasets and rules. The reported accuracy figures are preliminary project results and should not be interpreted as guaranteed performance in real field conditions.

VIII. CONCLUSION AND FUTURE WORK

This project presents an integrated smart-agriculture application that combines plant disease detection, crop prediction, fertilizer recommendation, soil-fertility analysis and AI-based agricultural assistant. By bringing these functions together in a Streamlit interface, the system offers users a convenient starting point for exploring plant and soil information.

By bringing these functions together in a Streamlit interface, the system offers users a convenient starting point for exploring plant and soil information. The reported implementation demonstrates how deep learning, machine learning and rule-based methods can be used within one application.

Future improvements identified in the project include connecting IoT sensors for real-time soil and environmental readings, adding weather prediction, expanding the datasets, improving model evaluation, deploying the application securely on a cloud platform, and developing a mobile version with regional-language support. Testing with real farm data and review by agricultural experts will be important before the system is used to guide field-level decisions

X. REFERENCES

[1] S. P. Mohanty, D. P. Hughes, and M. Salathé, “Using Deep Learning for Image-Based PlantDisease Detection,” Frontiers in Plant Science, vol. 7, Art.no. 1419, 2016. doi: 10.3389/fpls.2016.01419.

[2] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen,“MobileNetV2: Inverted Residuals and Linear Bottlenecks,” in Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018,pp. 4510–4520. doi: 10.1109/CVPR.2018.00474.

[3] L. Breiman, “Random Forests,” Machine Learning, vol. 45, pp. 5–32, 2001.doi: 10.1023/A:1010933404324.

[4] “Crop Recommendation System using Machine Learning,” International Journal of Engineering Research & Technology (IJERT). Available: https://www.ijert.org/crop-recommendation-system-using-machine-learning.

[5] TensorFlow, “MobileNet models.” Available: https://www.tensorflow.org/api_docs/python/tf/keras/applications/MobileNetV2.

[6] Scikit-learn developers, “Random Forest Classifier.” Available: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html.

[7] Stream lit, “Streamlit documentation.” Available: https://docs.streamlit.io/.

[8] Ollama, “Ollama documentation.” Available: https://docs.ollama.com/.

[9] Python Software Foundation, “Python 3 Documentation.” Available: https://docs.python.org/3/.

[10] Python Software Foundation, “sqlite3 — DB-API 2.0 interface for SQLite databases.” Available: https://docs.python.org/3/library/sqlite3.html.

[11] PlantVillage, “PlantVillage Dataset.” Available: https://github.com/spMohanty/PlantVillage-Dataset.

© Author(s). This work is peer-reviewed, openly published, and permanently archived
This article is openly accessible and reusable with proper attribution.