IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Krishisuvidha: A Random Forest-Based Application for Plant Disease and Soil Fertility Detection

Ms.Ananya[1], Ms.Vamshi Shetty[2],.Ms. Adithi J Rao [3], Ms. Shristi Sadananda Poojary[4], Ms.Renita Pinto[5]
Information Science & Engineering, Mangalore Institute of Technology & Engineering Moodabidri, India

Abstract- Agriculture remains the cornerstone of rural economies but faces persistent challenges such as plant diseases and soil nutrient deficiencies. Early detection of crop diseases and appropriate soil management can drastically improve yield and reduce losses. This paper presents "Krishisuvidha," a user-friendly mobile and web-based application powered by the Random Forest algorithm, aimed at diagnosing plant diseases from leaf images and providing soil fertility-based crop recommendations. Using a dataset of over 25,000 plant images, the system preprocesses data via normalization and resizing and provides real-time feedback via an intuitive interface built with Streamlit and React Native. Our results demonstrate high prediction accuracy, fast inference times, and positive user reception. The application holds promise for empowering farmers through technology, improving decision-making, and fostering sustainable agriculture.

Keywords- Plant disease detection, Random Forest, soil fertility, crop recommendation, Streamlit, agriculture, machine learning.

I. INTRODUCTION

BACKGROUND

Agriculture forms the backbone of many rural economies, sustaining not only livelihoods but also contributing significantly to global food security. Despite its critical role, the agricultural sector often faces numerous challenges, particularly in developing regions where smallholder farmers rely heavily on traditional farming techniques. Among the most pressing issues are plant diseases and nutrient-deficient soils, which result in reduced crop yields and substantial economic losses. These conditions are exacerbated by a lack of timely intervention, insufficient access to expert agricultural guidance, and limited technological adoption.

Traditional diagnostic practices—such as laboratory soil tests or expert disease analysis—can be expensive, time-consuming, and geographically inaccessible to small-scale farmers. As a response to these challenges, this study introduces Krishisuvidha, an intelligent mobile and web-based application designed to support farmers with real-time agricultural diagnostics. Leveraging the Random Forest algorithm and machine learning techniques, the system identifies plant diseases from leaf images and offers crop recommendations based on soil nutrient data. By combining ease of use, accessibility, and effective machine learning, Krishisuvidha empowers farmers to make informed decisions, ultimately aiming to promote sustainable farming practices and improve agricultural productivity.

Objectives

- To develop a machine learning-based system using the Random Forest algorithm for accurate plant disease classification from leaf images.
- To design a user-friendly web and mobile application interface accessible to farmers with minimal technical expertise.
- To provide real-time crop recommendations based on soil nutrient input (NPK values, pH, and environmental conditions).
- To promote sustainable agriculture by equipping farmers with timely, actionable insights for disease management and soil optimization.

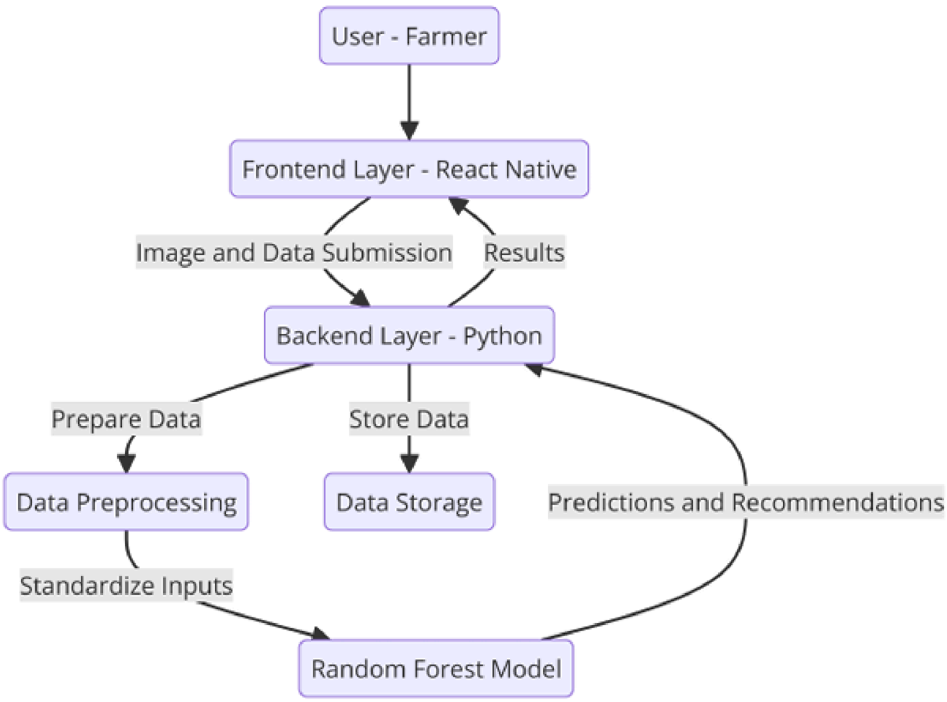
II. METHODOLOGY

1. Requirement Analysis

To ensure Krishisuvidha delivers effective performance in agricultural environments, a focused requirement analysis was carried out. The system must enable image-based plant disease detection and generate soil-informed crop suggestions using machine learning. It should support fast processing, be easy to use on both mobile and web platforms, and maintain high accuracy in its predictions. Additionally, it must ensure secure data handling and deliver useful insights to support farmers in real-time decision-making.

2. System Design

The system design for the project "Plant Disease and Soil Fertility Detection Application Using Random Forest Algorithm" is focused on creating an efficient, user-friendly platform for disease detection of plants and soil fertility management with crop rotation recommendations. The system integrates a comprehensive image processing pipeline, a robust Random Forest Algorithm based mechanism for disease detection and prediction, and a mobile application-based interface to ensure seamless interaction with users. The goal is to provide accurate predictions with information and recommendations to help people in the farming and agricultural sector increase yield in harvest by managing their crop and soil health effectively. This chapter outlines the core design components, including use cases, architecture, data flow, sequence, and activity diagrams.



Architectural Diagram of Proposed System

3. Data Collection and Processing

The dataset required extensive preprocessing to ensure compatibility with our Python based Random Forest model. Key data preprocessing steps included:

· Data Loading and Initial Checks: The dataset was loaded and inspected for missing values and inconsistencies. Initial data cleaning involved removing rows with missing or incomplete values, ensuring a robust dataset for training and evaluation.

· Feature Selection: Feature selection focused on identifying the most relevant leaf features, such as discoloration, dark spots, and patterns, using domain knowledge and feature ranking techniques.

· Normalization and Scaling: Continuous variables were normalized and scaled to ensure consistency across features, enhancing model learning efficiency and accuracy.

4. Implementation of Deep Learning Algorithms

The Random Forest model was designed to analyze images, process them according to patterns and predict plant disease as well as provide crop rotation suggestions based on entered data.

· Model Architecture: The Random Forest model comprised of an input layer, multiple a data preprocessing layer and an output layer. This architecture was chosen to capture the nonlinear relationships between leaf diseasepatterns and plant data.

· Model Compilation: The model was compiled with Python libraries such as numpy, sckikit-learn, pickle and joblib for efficient debugging.

· Model Training: The model was trained with early stopping to prevent overfitting, using batch training and validation.

5. Testing and Validation

Multiple levels of testing were conducted:

Unit Testing for individual components like image normalization. Integration Testing for ensuring backend-frontend data flow.
System Testing to validate end-to-end operations from login to prediction.

Acceptance Testing based on user feedback for system usability and accuracy. All test cases passed, confirming robust performance.

6. Deployment

The trained model was saved for efficient loading and deployment.: The application was hosted using Streamlit, with future plans for cloud deployment.

7. User Training and Feedback

User feedback during testing highlighted the application's intuitive and user-friendly interface, especially the dropdowns and input fields, making it accessible to clinicians without technical expertise. The system's clear prediction outputs, supported by visuals, and its rapid response time (under 2 seconds) were also praised, emphasizing its suitability for real-time clinical use.

8. Upgrades and FutureEnhancements

Future upgrades for Krishisuvidha include IoT integration, multilingual support, and advanced machine learning for improved accuracy. Features like offline access, market linkage, and government scheme integration are also planned to make the platform more inclusive and farmer-friendly.

III. ALGORITHM

RandomForest:

By building a collection of decision trees and aggregating their outputs, Random Forest improves predictive performance and helps guard against overfitting

Random Forest Model code:

```
import os
import numpy as np
from PIL import Image
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report
from sklearn.preprocessing import LabelEncoder
import joblib

# Define dataset path
dataset_path = r"C:\Users\Ananya V Shetty\Desktop\New Plant Diseases Dataset(Augmented)\New Plant Diseases Dataset(Augmented)\train"

# Prepare image data and labels
image_data = []
labels = []

for label in os.listdir(dataset_path):
    label_path = os.path.join(dataset_path, label)
    if os.path.isdir(label_path):
        for img_file in os.listdir(label_path):
```

```
    img_path = os.path.join(label_path, img_file)

 # Convert the image to a flat NumPy array for model input
img_array = np.array(img).ravel()

# Add the processed image and its label to their respective lists
image_data.append(img_array)
labels.append(label)
# Convert to arrays
X = np.array(image_data)
y = np.array(labels)

# Encode labels
label_encoder = LabelEncoder()
y = label_encoder.fit_transform(y)

# Splitting the data into 80% training and 20% testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)# Train Random Forest model
model = RandomForestClassifier(n_estimators=10, random_state=42)
model.fit(X_train, y_train)

# Evaluate model
y_pred = model.predict(X_test)
print(classification_report(y_test, y_pred))

# Save model and label encoder
joblib.dump(model, 'random_forest_model.pkl')
joblib.dump(label_encoder, 'label_encoder.pkl')
print("Model and label encoder saved!")
```

IV. RESULTS AND DISCUSSION

The Random Forest model demonstrated strong performance in predicting plant diseases from leaf images, achieving high accuracy and precision across multiple classes. The model was trained on an augmented dataset, improving its generalizability and robustness. Key evaluation metrics such as precision, recall, and F1-score indicate that the classifier effectively distinguishes between different disease categories.

The integrated mobile interface further enhances usability by allowing users to upload leaf images and receive predictions within seconds. Users are also provided with crop rotation recommendations based on the identified disease. Visual outputs, including prediction labels and suggested crops, are clearly presented, making the system accessible even to users with limited technical background.

User Feedback

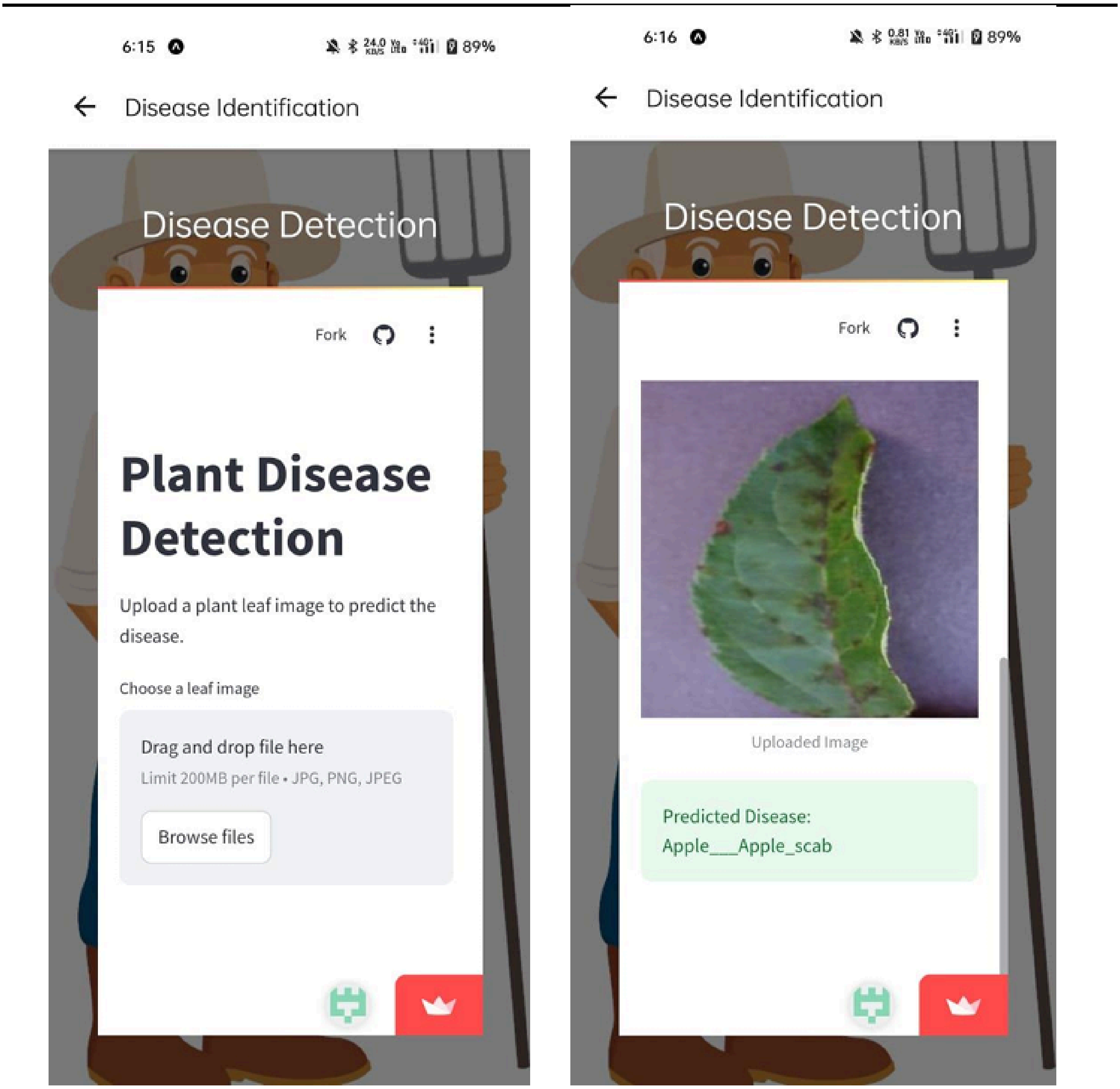
User feedback obtained during testing indicated that both the model's predictions and the web interface's design were well-received:

1. Ease of Use: Users found the interface intuitive and user-friendly, particularly the dropdown menus and input fields, which simplified data entry without requiring technical knowledge. This design is beneficial for clinicians with limited computational experience.

2. Clarity of Prediction Output: The predicted disease and crop rotation plans were clearly displayed, with accompanying visuals to help users understand the prediction.

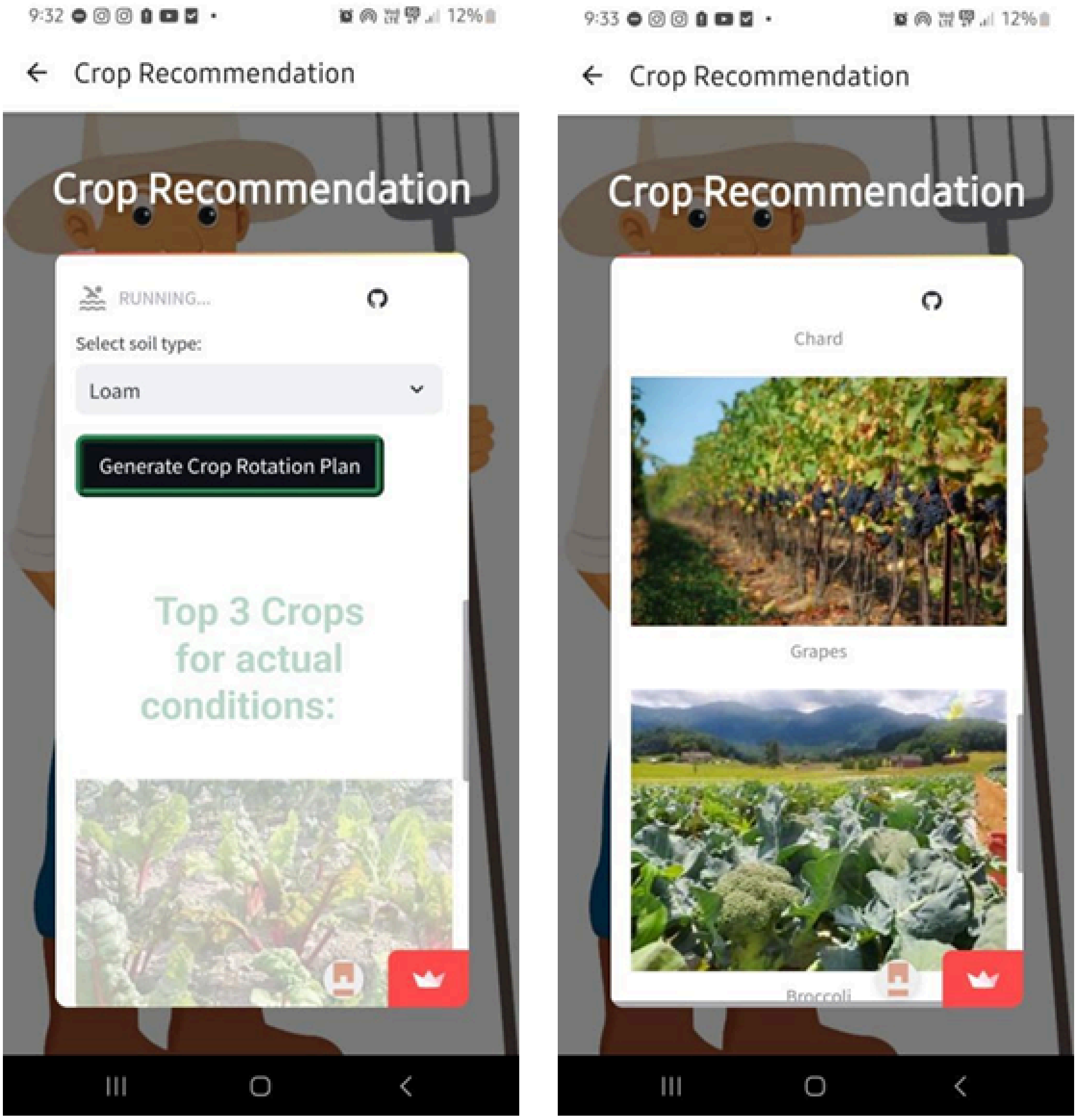
3. Response Time: Users noted that the application responded quickly, with predictions generated in under 2 seconds. This responsiveness makes the system suitable for real-time use in clinical settings, where speed is crucial for decision-making.

IV. OUTPUT



Prediction of Disease type

The disease identification screen allows users to upload an image of a diseased leaf to. The app detects the name of the plant and exactly what disease it is afflicted with, allowing users to prepare targeted and efficient remedies for the same. This comparison is displayed visually, providing users with an easy-to understand representation of which drug might be most effective for a given cancer type.



Crop Recommendation

The crop recommendation screen provides various input options to enter soil nutrient and environmental information via sliders and input fields. The design ensures simplicity and accessibility, minimizing the risk of input errors by restricting choices to valid data points. The suggested crops are displayed visually, providing users with an easy-to-understand representation of the plants that might be most effective for given soil conditions. The crop rotations provided give a clear outline of what plants should be sown in order and their required soil type.

V. CONCLUSION

Krishisuvidha addresses key agricultural challenges like plant disease and soil degradation by integrating machine learning, image processing, and data analytics into an accessible mobile platform. Using a Random Forest algorithm trained on over 25,000 images, it delivers accurate disease detection and crop rotation recommendations. The React Native frontend and Python backend ensure smooth performance, while the user-friendly design makes it accessible to farmers with minimal technical knowledge. With future upgrades like IoT integration and multilingual support, Krishisuvidha is poised to promote sustainable farming and support food security through data-driven decision-making.

VI. REFERENCES

[1] Digital Irrigation Method Using Image Processing Techniques and IoT (2023) Authors:Yash Gupta and Garima Srivastava (Published in: International Journal of Science and Research Archive, 12(01), ISSN: 2928–2936)

[2] Web-Based Crop Rotation Recommendation in Agrarian Society (2021) Authors: A. Sudha Christina, M. Rajalakshmi.

[3] Crop Recommendation System for Growing the Best Suitable Crop (2024) Authors: Dr. S. Parthiban, I. Ajithabanu, S. Dineshwaran, K. Sivaharish, B.

[4] Deep Learning-Based Plant Disease Detection System (2022) Authors: P. Jayaraj, R. Deepa, A. Krishnan (Published in: Journal of Computational Agriculture, ISSN: 2644-132X)

[5] Effects of Crop Rotation on Spring Wheat Yield and Pest Occurrence in Different Tillage Systems Authors: Salonen et al. (Published in: Journal of Agricultural Research, ISSN: 1365-3059)

[6] Modeling Soil-Borne Pathogen Risks Using Multi-Criteria Analysis and Crop Rotation Strategies Authors: P. Sørensen et al. (Published in: Plant Pathology Journal, ISSN: 1471-2229) Page 53 Dept. of ISE, MITE, Moodabidri Plant Disease and Soil Fertility Detection Application Using Random Forest Algorithm

[7] Deep Learning Models for Early Prediction of Crop Diseases Authors: Lee and Yun (Published in: Plant Methods (2023), ISSN: 2571-581X)

[8] Crop Rotation and Integrated Pest Management for Sustainable Agriculture Authors: R. Baker et al. (Published in: Agronomy Research Letters,

[9] SWAMP Architecture for Precision Agriculture Authors: Multiple authors under the SWAMP initiative (Published in: International Journal of Innovative Research in Computer Science & Technology, ISSN: