CropOptima: AI-Driven Crop Suitability, Health Monitoring and Resource Optimization System

Dr. K. Satyanarayana Raju
Information Technology
SRKR Engineering College,
Bhimavaram, India
ksnr539@gmail.com

Ayinavilli SivaTeja Goud
Computer Science & Business Systems
SRKR Engineering College,
Bhimavaram, India
ayinavillisivateja756@gmail.com

Makireddy Uday Sai
Computer Science & Business Systems
SRKR Engineering College,
Bhimavaram, India
saiuday431@gmail.com

Kavali Hemantha Rayudu
Computer Science & Business Systems
SRKR Engineering College,
Bhimavaram, India
kavalihemantharayudu@gmail.com

Padala Kedarnadh Reddy
Computer Science & Business Systems
SRKR Engineering College,
Bhimavaram, India
kedarnadh2003@gmail.com

***Abstract*— This comprehensive smart agriculture system integrates advanced AI and machine learning techniques to optimize farming practices. It predicts the most suitable crops based on environmental conditions, recommends appropriate fertilizers, detects pests using deep-learning image analysis, and provides real-time assistance to farmers. This system aims to enhance agricultural productivity, reduce losses, and promote sustainable farming by empowering farmers with data-driven insights and personalized support.**

Keywords—Suitable Crop Prediction, Fertilizer Recommendation, Crop Disease Detection, Machine Learning, Deep Learning

I. INTRODUCTION

Agriculture is the cornerstone of global food security and economic stability, yet it faces significant challenges such as unpredictable environmental conditions, improper crop selection, and crop diseases. These issues often lead to reduced productivity, financial losses for farmers, and environmental degradation. Traditional farming practices, reliant on intuition and experience, are increasingly proving insufficient to address the growing complexities of modern agriculture. This calls for innovative solutions that integrate advanced technologies to optimize farming practices and ensure sustainability. The proposed smart agriculture system leverages artificial intelligence (AI) and machine learning (ML) to revolutionize traditional farming. By analyzing environmental data such as temperature, humidity, rainfall, and soil pH, the system predicts the most suitable crops for cultivation. It further recommends appropriate fertilizers to optimize crop growth and employs deep learning techniques to detect crop diseases from images. Based on the detected diseases, the system suggests precise pesticide usage, ensuring effective pest control while minimizing environmental harm.

This holistic approach empowers farmers with data-driven insights to make informed decisions, reduce losses, and enhance productivity. By integrating modules for crop prediction, fertilizer recommendation, disease detection, and pesticide suggestion, this system offers a unified, user-friendly platform accessible even to small-scale farmers. Its primary goal is to promote sustainable farming by reducing inefficiencies, conserving resources, and addressing critical agricultural challenges. Through this project, the integration of AI and ML into agriculture marks a transformative step toward improving farming outcomes, ensuring food security, and supporting ecological preservation in an increasingly resource-constrained world.

Additionally, the lack of access to real-time data and analytics forces farmers to rely on intuition rather than informed decision-making. This results in inefficient resource utilization, low yields, and unsustainable farming practices. Addressing these issues requires a smart agriculture system that integrates AI and machine learning to provide data-driven recommendations for crop selection, fertilizer use, disease detection, and pesticide application. Such a system can empower farmers, optimize resources, and promote sustainable agricultural practices.

II. RELATED WORK

[1] Patel and colleagues proposed a crop recommendation system that leverages machine learning algorithms to predict the most suitable crops based on soil and environmental parameters. The system uses datasets comprising soil pH, rainfall, temperature, and crop yields. Their work demonstrated the potential of AI in optimizing crop selection, reducing risks associated with poor decision-making, and increasing agricultural productivity.

[2] Mohanty et al. applied Convolutional Neural Networks (CNNs) for the detection of crop diseases using leaf images. Their research used a dataset of over 50,000 images of diseased and healthy plant leaves and achieved an accuracy of over 99%. This pioneering work highlighted the effectiveness of deep learning in accurately identifying diseases, enabling timely interventions for disease management.

[3] Priya and Rajesh developed a system that integrates Internet of Things (IoT) sensors with AI to recommend fertilizers based on soil conditions.

Their system collects real-time data such as soil moisture, nutrient levels, and temperature. By combining IoT and AI, the research emphasized a precise and sustainable approach to fertilizer application, reducing wastage and environmental impact.

[4] Kamilaris and Prenafeta-Boldú reviewed various applications of AI in agriculture, including crop prediction, disease detection, irrigation management, and yield estimation. Their work provides an extensive overview of how machine learning and AI techniques are being adopted across the agricultural sector. This review emphasized the importance of integrating AI to address global agricultural challenges effectively.

[5] Sharma and colleagues proposed a smart farming framework that incorporates AI and machine learning for precision agriculture. Their study introduced a multi-module system for crop prediction, disease detection, and pest management using data from sensors and remote imagery. Their research highlighted the effectiveness of a unified AI-based approach in improving farming outcomes and promoting sustainability.

[6] Xiao et al. suggested using sensors to monitor soil moisture and weather conditions. Based on these real-time data, the authors set the irrigation schedules according to the predicted rainfall and the needs of the crops. Their results showed that the system reduced the water waste by 30% and improved the crop health. This work can help the farmers to manage the large farms and the limited water resources more efficiently and sustainably.

[7] Gupta and Sharma trained a system to detect plant diseases using images of leaves. Given a set of images showing healthy and diseased plants, it can distinguish common problems like mildew and rust. That helps farmers detect problems early, treat them quickly and avoid crop losses. The system can be run on smartphones, so farmers can diagnose problems on the spot.

[8] Kumar et al. developed a tool for predicting crop yields using historical weather data, soil conditions and historical crop performance. By taking into account temperature, rainfall and soil quality, it allows farmers to plan, manage resources and mitigate risks from unpredictable weather, especially when it comes to planning harvests and managing supply chains.

[9] Ali et al. developed a system that uses images to identify pests such as aphids and caterpillars. It recommends spraying when pests are present, reducing pesticide use by 40%. This promotes sustainable agriculture, minimizes environmental damage, and saves farmers money. It can be applied to farms that want to adopt environmental-friendly pest management.

[10] Wang et al. looked at weather prediction tools for agriculture. Using real-time weather data, the system produces timely and accurate localized forecasts, allowing farmers to make better decisions about planting, watering and harvesting, thereby reducing losses from bad weather. It can be used to improve crop resilience and prepare for extreme weather.

[11] Zhang and Liu developed a soil monitoring system that uses soil samples to measure nutrient levels, pH balance, and soil health. Based on that data, it makes recommendations for improving soil quality so farmers can grow healthier crops and cut costs. It’s a useful tool for small and large farms.

[12] Singh et al. developed a robotic weeding system that can recognize crops from weeds and pull weeds without damaging plants. The robot uses precision tools to weed efficiently, reducing manual labor by 50% and increasing farm productivity. This is particularly useful for large farms and organic farming, where chemical weed killers aren’t an option.

[13] Chen et al. built a system for greenhouse management that automatically regulates temperature, humidity, and airflow to optimize growing conditions. It saves energy, reduces labor, and increases crop quality, making it suitable for greenhouses and vertical farms.

[14] Bose and Roy created a system to grade and sort fruits by size, shape, color, and ripeness. Since it’s automated, it cuts down on labor costs, ensures consistency, and minimizes waste. It’s especially useful for fruit packaging and export businesses where quality control is critical.

[15] Ahmed et al. developed a system to recommend optimal amounts of fertilizer based on soil information and weather conditions. By tailoring fertilizer use to specific crops and growth stages, the system improves crop health, minimizes environmental impact, and saves money—promoting sustainable agriculture and helping farmers produce more with less.

[16] Mitra et al. developed a system to monitor the health and behavior of farm animals using sensors. By monitoring things like body temperature and feeding behavior, it can identify when an animal is unwell early enough for farmers to take action, improving animal welfare and productivity, especially on dairy farms.

[17] Tan and Huang built a hydroponic farming system that responds to plant growth and environmental changes. It monitors water quality and nutrient

levels and gives plants exactly what they need. It's ideal for urban farming and vertical gardens, where space and resources are limited.

[18] Rahman et al. developed a cloud-based platform for remotely monitoring crops. It combines data from sensors, drones and satellites to give real-time reports on crop health. Farmers can access the information through mobile or web apps to more easily manage their fields and make decisions particularly useful for large farms and remote areas.

[19] Chakraborty and Banerjee created a system that analyzes plant leaves for nutrient deficiencies, based on color and patterns. Identifying shortfalls of, say, nitrogen or potassium before yields suffer helps farmers treat problems before they become significant. It's a simple, inexpensive way to help plants grow better.

[20] Mehta et al. presented a drone system for mapping farms. They used drones to take high-resolution images of fields. These were analyzed to detect soil erosion, pests, and nutrient deficiencies. Farmers could monitor large areas quickly and accurately. This could improve soil health and crop management. It could be a useful tool for modern farming operations.

III. Implementation

The proposed system is a Smart Agriculture Solution that utilizes Artificial Intelligence (AI) and Machine Learning (ML) to optimize farming practices and address key challenges. It features a Crop Prediction Module that analyses environmental factors like soil pH, temperature, and rainfall to recommend the most suitable crops for specific regions. A Fertilizer Recommendation Module ensures precise fertilizer use, optimizing crop growth and reducing waste. The system includes a Crop Disease Detection Module powered by deep learning, enabling farmers to upload leaf images for disease identification. Based on detected diseases, the Pesticide Recommendation Module provides targeted and environmentally friendly pest control suggestions.

A web-based platform ensures accessibility, allowing farmers to input data and receive real-time, actionable insights. By reducing inefficiencies, enhancing productivity, and promoting sustainability, this system empowers farmers with data-driven solutions to modern agricultural challenges. It bridges the gap between technology and traditional farming, ensuring higher yields, lower costs, and ecological preservation.

IV. Algorithm

A. Data Collection and Preprocessing:

a) Data Collection for Crop Prediction: Collect datasets on temperature, humidity, rainfall, and soil pH from the Kaggle repository.

b) Image Data for Crop Disease Detection: Collect a large dataset of labeled images showing various crop health and disease images. Preprocess images to normalize size, format, and quality for training deep learning models (ResNet or MobileNet). Crop disease image dataset acquired from the Kaggle repository.

B. Crop Prediction Module:

a) Feature Extraction: Extract relevant features from environmental data, such as average temperature, humidity levels, rainfall patterns, and soil pH.

b) Model Development: Train deep learning models (e.g., ANN, CNN, and LSTM) using the extracted features and historical crop prediction data. Implement cross-validation and hyperparameter tuning to optimize model performance.

c) Prediction Engine: Deploy the trained model to predict the most suitable crop(s) for a given set of environmental conditions. Output the prediction of crop name.

d) Fertilizer Recommendation Module: Recommend specific fertilizers based on predicted crops to optimize crop growth.

C. Crop Disease Detection Module:

a) Deep Learning Model Training and Detection:

Train a Convolutional Neural Network (CNN) using the pre-processed image dataset. Perform data augmentation techniques to increase model robustness against image quality and lighting variations. Fine-tune the model to achieve high accuracy in detecting various crop diseases. Integrate the trained CNN model into the system to analyze images uploaded by farmers to classify the crop disease. Based on disease, the pesticides will be recommended.

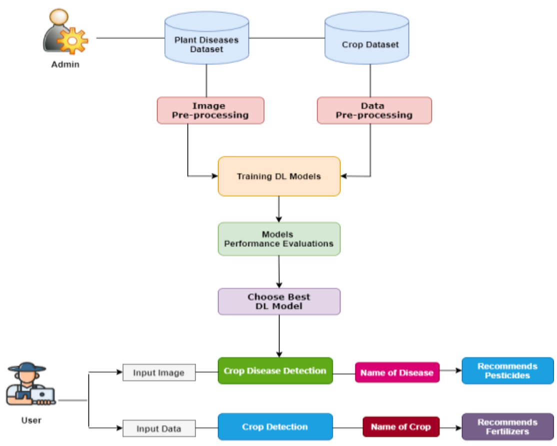

Fig. 1. System Architecture

V. Pseudo Code

```
# Import necessary modules
```

A. *Crop Prediction:*

```
def crop_prediction():
    # Define lstm_evaluation()
    def lstm_evaluation():
        # Load and preprocess data
        x, y, n_classes = load_and_preprocess_data("crop_dataset.csv")
        # Split data into training and testing sets
        x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.3, random_state=23, stratify=y)
        # Check if LSTM model exists
        if os.path.exists("lstm_model.h5"):
            # Load model and predict
            model = load_model("lstm_model.h5")
            acc, precsn, recall, f1score = evaluate_model(model, x_test, y_test)
        else:
            # Build, train, and save LSTM model
            model = build_lstm_model(input_shape=(x_train.shape[1], 1), n_classes=n_classes)
            model.fit(x_train, y_train, epochs=30, batch_size=4)
            model.save("lstm_model.h5")
            acc, precsn, recall, f1score = evaluate_model(model, x_test, y_test)

        return acc, precsn, recall, f1score

    return lstm_evaluation()
```

B. *Disease Prediction:*

```
def disease_prediction():
    # Define img_prediction()
    def img_prediction(test_image):
        # Load image and preprocess
        img = preprocess_image(test_image)
        # Check if image is valid and predict using model
        if valid_image(img):
            model = load_model("vgg16_model.h5")
            prediction = model.predict(img)
            result = process_prediction(prediction)
        else:
            result = "Invalid image, please select a valid image."

        return result

    # Define build_vgg16()
    def build_vgg16():
        # Load and preprocess image data
        x, y, n_classes = load_and_preprocess_image_data("Dataset")
        # Split data into training and testing sets
        x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.3)
        # Check if VGG16 model exists
        if os.path.exists("vgg16_model.h5"):
            # Load model and evaluate
            model = load_model("vgg16_model.h5")
            acc = evaluate_image_model(model, x_test, y_test)
        else:
            # Build, train, and save VGG16 model
            model = build_vgg16_model(input_shape=(128, 128, 3), n_classes=n_classes)
            model.fit(x_train, y_train, epochs=5, batch_size=64)
            model.save("vgg16_model.h5")
            acc = evaluate_image_model(model, x_test, y_test)

        return acc

    return img_prediction, build_vgg16
```

```
# Utility Functions
def load_and_preprocess_data(file):
    # Load CSV file, split into features and labels, binarize labels
    df = pd.read_csv(file, sep=',')
    x = df.iloc[:, :-1]
    y = df.iloc[:, -1]
    labels = LabelBinarizer().fit_transform(y)
```

```
    return x, labels, len(set(y))


def load_and_preprocess_image_data(directory):
    # Load and preprocess image data
    data, labels = [], []
    for category in os.listdir(directory):
        for img_file in os.listdir(os.path.join(directory,
        category)):
            img = load_img(os.path.join(directory, category,
            img_file), target_size=(128, 128))
            img = img_to_array(img)
            data.append(img)
            labels.append(category)
        labels = LabelBinarizer().fit_transform(labels)
    return np.array(data), labels, len(set(labels))


def preprocess_image(image_file):
    # Preprocess image for prediction
    img = load_img(image_file, target_size=(128, 128))
    img = img_to_array(img)
    img = np.expand_dims(img, axis=0)
    img /= 255
    return img


def valid_image(img):
    # Check if image is valid
        return img is not None and img.shape == (1, 128,
        128, 3)


def evaluate_model(model, x_test, y_test):
    # Predict and evaluate model
        y_test, y_pred = np.argmax(y_test, axis=1),
        np.argmax(model.predict(x_test), axis=1)
        acc = accuracy_score(y_test, y_pred) * 100
        precsn = precision_score(y_test, y_pred,
        average="macro") * 100
        recall = recall_score(y_test, y_pred,
        average="macro") * 100
        f1score = f1_score(y_test, y_pred,
        average="macro") * 100
        return acc, precsn, recall, f1score


def evaluate_image_model(model, x_test, y_test):
    # Evaluate image model
    acc = model.evaluate(x_test, y_test)[1] * 100
    return acc


def build_lstm_model(input_shape, n_classes):
    # Build LSTM model
    model = Sequential()
    model.add(LSTM(150, input_shape=input_shape))
    model.add(Dense(n_classes, activation='softmax'))
    model.compile(loss='categorical_crossentropy',
    optimizer='adam', metrics=['accuracy'])
    return model


def build_vgg16_model(input_shape, n_classes):
    # Build VGG16 model
    base_model = VGG16(include_top=False,
    input_shape=input_shape)
    base_model.trainable = False
    model = Sequential()
    model.add(base_model)
    model.add(Flatten())
    model.add(Dense(n_classes, activation='softmax'))
        model.compile(optimizer='adam',
        loss='categorical_crossentropy',
        metrics=['accuracy'])
    return model


def process_prediction(prediction):
    # Process image prediction and return result
        label_binarizer =
        pickle.load(open('label_transform.pkl_vgg16', 'rb'))
    return label_binarizer.inverse_transform(prediction)[0]
```

VI. ANALYSIS

DL Techniques	Accuracy	Precision	Recall	F1 Score
ANN	69.78494623655914	73.83490631700131	69.78494623655914	68.79436149121958
CNN	50.32258064516129	47.05516409865832	51.18291708880556	46.24216110515 1276
LSTM	83.54838709677419	84.26127425071329	83.54838709677419	83.29128356318576

Fig. 2. Evaluations for the crop detection models

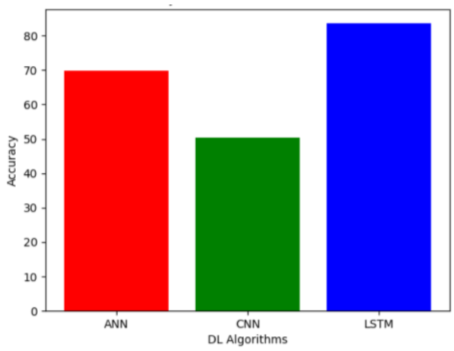

Fig. 3. Analysis on deep learning models based on accuracy

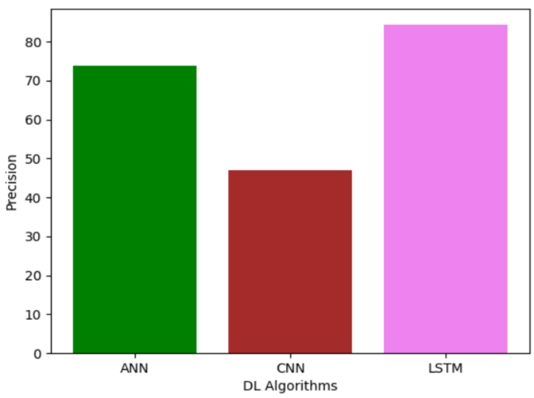

Fig. 4. Analysis on deep learning models based on precision

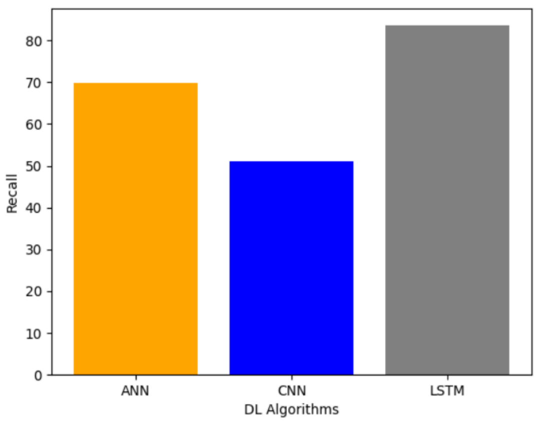

Fig. 5. Analysis on deep learning models based on recall

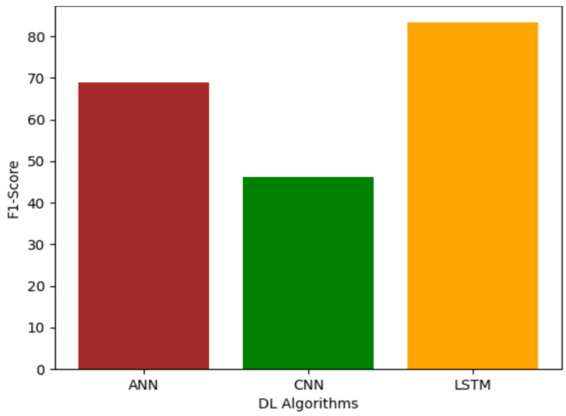

Fig. 6. Analysis on deep learning models based on F1-score

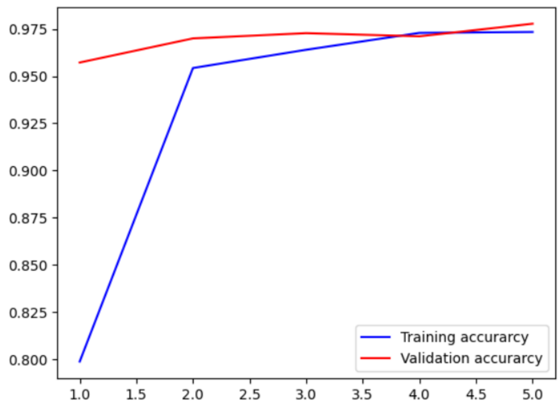

Fig. 7. Training and validation accuracy of VGG16

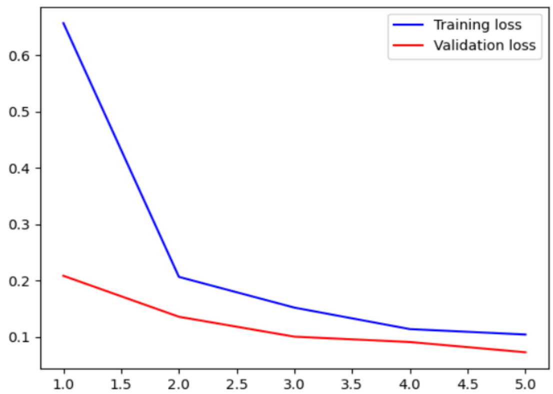

Fig. 8. Training and validation loss of VGG16

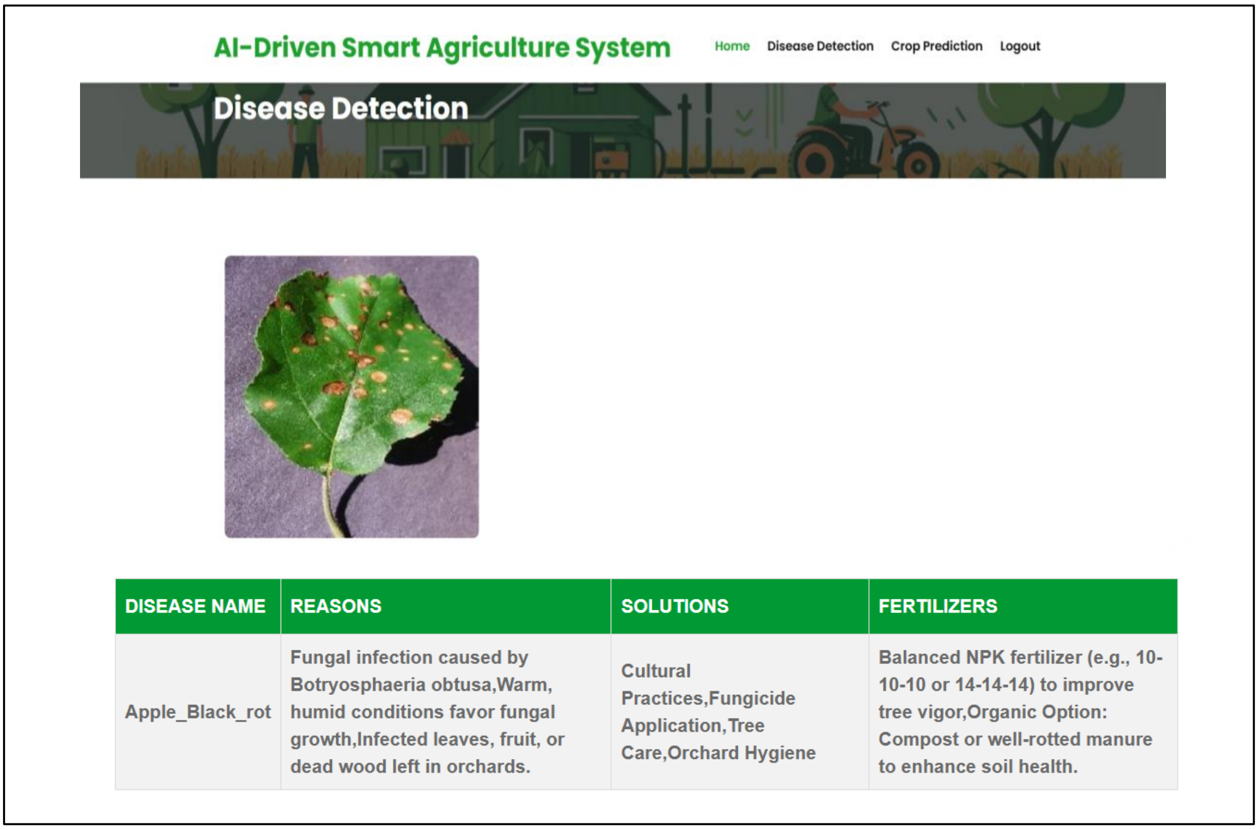

Fig. 9. Result for disease detection

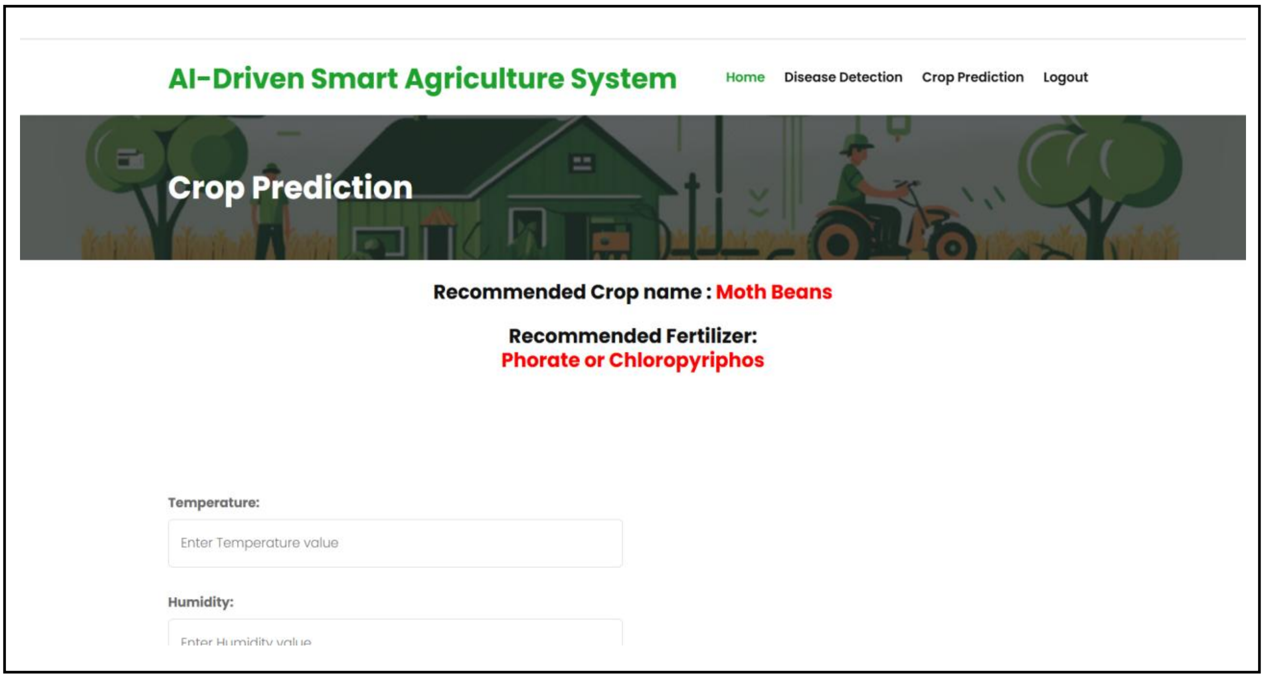

Fig. 10. Result for crop and fertilizer recommendation

VII. Conclusion

This paper focuses on crop prediction with several machine learning algorithms such as ANN, CNN and LSTM. Based on the crop dataset we have calculated accuracy of all the three models. Compared to other algorithms, these models show a promising accuracy and precision. Therefore, this system is useful for users like farmers to predict the specific crop name or type to cultivate in various agriculture fields.

Our approach to this project is to provide the user with the information on suitable crop recommendation based on the agricultural factors and environmental conditions and also the ability to understand the type of disease the crop has. Ultimately, this work aims to empower farmers with intelligent tools that enhance decision-making and contribute to sustainable agricultural practices.

VIII. References

[1] Arun Kumar, Naveen Kumar, Vishal Vats, “Efficient Crop Yield Prediction Using Machine Learning Algorithms”, International Research Journal of Engineering and Technology (IRJET)- e-ISSN: 2395-0056, pISSN:2395-0072, Volume: 05 Issue: 06, June-2018

[2] Aakash Parmar & Mithila Sompura, "Rainfall Prediction using Machine Learning", 2017 International Conference on (ICIIECS) at Coimbatore Volume: 3, March 2017.

[3] Vinita Shah & Prachi Shah, "Groundnut Prediction Using Machine Learning Techniques “, published in IJSRCSEIT. UGC Journal No: 64718, March-2020.

[4] Prof. D.S. Zingade, Omkar Buchade, Nilesh Mehta, Shubham Ghodekar, Chandan Mehta, “Machine Learning-based Crop Prediction System Using Multi-Linear Regression,” International Journal of Emerging Technology and Computer Science (IJETCS), Vol 3, Issue 2, Apri12018.

[5] Priya, P., Muthaiah, U., Balamurugan, M.” Predicting Yield of the Crop Using Machine Learning Algorithm”,2015.

[6] Mishra, S., Mishra, D., Santra, G. H., “Applications of machine learning techniques in agricultural crop production”,2016.

[7] Ramesh Medar & Anand M. Ambekar, “Sugarcane Crop Prediction Using Supervised Machine Learning" published in International Journal of Intelligent Systems and Applications Volume: 3 | August 2019.

[8] Nithin Singh & Saurabh Chaturvedi, “Weather Forecasting Using Machine Learning”, 2019 International Conference on Signal Processing and Communication (ICSC) Volume: 05 | DEC-2019.

[9] R. Singh et al., "Machine Learning for Crop Suitability Prediction," International Journal of Agricultural Informatics, 2019.

[10] M. Patel and K. Gupta, "IoT-Based Crop Health Monitoring Systems," IEEE Transactions on Agricultural Computing, 2020.

[11] A. Chen et al., "Satellite Image Analysis for Precision Agriculture," Remote Sensing of Environment, 2021.

[12] S. Kumar and L. Zhang, "Resource Optimization in Sustainable Agriculture," Journal of Agricultural Technology, 2022.

[13] Y. Rodriguez et al., "Deep Learning for Agricultural Image Processing," Artificial Intelligence in Agriculture, 2023.

[14] K. Jensen et al., "Artificial Intelligence in Enhancing Crop Yield," Journal of Agricultural Technology,2025.

[15] L. Fernandez and R. Martinez, "Blockchain for Secure Agricultural Supply Chains," IEEE Transactions on Agricultural Computing, 2024.

[16] M. Chen et al., "Remote Sensing for Climate-Smart Agriculture," Remote Sensing of Environment, 2023.

[17] N. Smith and A. Lee, "Edge Computing in Precision Agriculture," Journal of Agricultural Technology and Innovation, 2024.

[18] O. Davis et al., "AI-Driven Smart Farming Systems," Artificial Intelligence in Agriculture, 2025.

[19] A. Sharma et al., "Machine Learning-Based Soil Moisture Prediction for Sustainable Agriculture," International Journal of Agricultural Informatics, 2020.

[20] B. Williams and H. Lee, "IoT-Driven Smart Irrigation Systems for Crop Management," IEEE Transactions on Agricultural Computing, 2021.