AI-BASED SOLUTIONS FOR CROP DISEASE DETECTION

[1]Dr. Pampapathi B M, [2]Mahesh G P, [3]Renuka Bai R, [4]Lavanya A, [5]Syeda Umme Sumiya,

[*1]Associate Professor, Department of Computer Science & Engineering, RYM Engineering College, Ballari, VTU Belagavi, Karnataka, India

[2345]Department of Computer Science & Engineering, RYM Engineering College, Ballari, VTU Belagavi, Karnataka, India

Abstract:Agriculture plays a vital role in human life by not only providing food but also significantly contributing to the economy. However, crops often suffer from insect infestations, leading to substantial losses. Timely identification and treatment of these pests are essential for minimizing their impact, with early detection and prompt intervention being the most effective strategies. Traditional methods, however, are slow and lack real-time monitoring capabilities. This study introduces an AI-powered system for real-time insect detection in fruits and vegetables, evaluating various deep learning models for accuracy in identifying crop pests. The most effective model, based on a Convolutional Neural Network (CNN) algorithm, achieved an impressive accuracy of 98.75%, supported by a diverse dataset of insect images compiled from multiple sources. Additionally, the system incorporates a chatbot powered by LLaMA 3 through Ollama, providing farmers with instant solutions for crop diseases and step-by-step guidance on cultivating healthy crops. By delivering real-time recommendations and expert insights, it simplifies agricultural decision-making and enhances farm management. The proposed solution improves agricultural efficiency by reducing manual effort, increasing disease detection accuracy, and helping farmers achieve healthier crop yields with greater ease.

I. INTRODUCTION

Agriculture has long served as the foundation of human civilization, playing a crucial role in ensuring food security and contributing significantly to economic development. Despite advancements in agricultural practices, crop production continues to face major challenges, particularly the vulnerability of crops to insect infestations. These pests pose a serious threat to crop health, often resulting in substantial yield reductions and financial losses if not managed promptly and effectively.

Traditional insect detection methods rely heavily on manual inspection, which is labor-intensive, time-consuming, and lacks the capacity for real-time monitoring. Such methods frequently fail to identify pest outbreaks at an early stage, allowing infestations to escalate and cause widespread crop damage.

To overcome these limitations, this study proposes a deep learning-based solution for real-time insect identification and detection in fruit and vegetable crops. Multiple transfer learning models are explored and evaluated to determine their effectiveness and reliability in accurately classifying different insect species. Among the models examined, a Convolutional Neural Network (CNN) demonstrated superior performance, achieving an impressive accuracy rate of **98.75%**.

Additionally, a comprehensive dataset was developed by aggregating and labeling insect images collected from a variety of devices and sources. This diverse dataset enhances the model's robustness and generalization capability. The proposed system not only improves detection accuracy but also streamlines the insect identification process, significantly reducing the manual workload for farmers. It offers a scalable, efficient, and practical approach to enhancing pest management strategies in modern agriculture.

2. Review of Literature

In recent years, the prevalence of cheating in various contexts, including online exams, has become a significant concern. The COVID-19 pandemic has further amplified this issue, prompting researchers to investigate its extent and implications.

Newton and Essex (1) conducted a systematic review to explore the prevalence of cheating in online exams and its potential increase during the pandemic. Their findings shed light on the widespread nature of cheating and emphasize the need for effective detection and prevention strategies. However, self-reported data limitations and variations in institutional policies make it difficult to generalize their findings across different educational settings.

Ruiperez-Valiente et al. (2) utilized machine learning to detect "multiple-account" cheating and analyzed the influence of student and problem features. Their study highlights the importance of leveraging advanced computational methods to identify suspicious activities and mitigate academic dishonesty effectively. While their approach demonstrated high detection accuracy, challenges remain in adapting machine learning models to different exam formats and ensuring fairness in algorithmic decision-making.

Ramberg and Modin (3) investigated the relationship between school effectiveness, students' grades, and moral standards concerning cheating. Their research provides insights into the complex interplay between academic performance, ethical values, and cheating tendencies among students. Although their findings suggest that institutional culture significantly influences cheating behavior, further research is needed to explore intervention strategies that can effectively reduce academic dishonesty.

Advancements in deep learning have revolutionized various domains, including cheating detection, medical imaging, surveillance, and activity recognition. Yadav and Jadhav (4) employed deep convolutional neural networks (CNNs) for medical image classification, demonstrating promising results for disease diagnosis. Their study underscores the potential of deep learning in pattern recognition, yet its reliance on large datasets and extensive computational resources remains a challenge.

Gopal and Ganesan (5) developed a real-time deep learning framework to monitor social distancing, showcasing the potential of AI-driven solutions for addressing public health challenges. Their system demonstrated high accuracy, but real-world implementation posed challenges such as environmental variations and occlusions.

Advances in eye-tracking technology have enabled precise monitoring of human visual attention and behavior. Valliappan et al. (6) introduced an accurate and affordable smartphone-based eye-tracking system, facilitating accelerated research in eye movement analysis. However, limitations in smartphone processing power and lighting conditions affect real-time tracking performance.

Similarly, Stein et al. (7) compared eye-tracking latencies among commercial head-mounted displays, contributing to the optimization of virtual and augmented reality applications. Their findings highlight the importance of reducing latency to improve user experience, yet cost constraints remain a barrier to widespread adoption.

Deep learning techniques have significantly impacted autonomous driving, image processing, and object detection. Grigorescu et al. (8) conducted a survey of deep learning techniques for autonomous driving, emphasizing AI's role in enhancing vehicle perception and decision-making. While deep learning has improved self-driving capabilities, issues related to adversarial attacks and ethical considerations persist.

Dong et al. (9) proposed an image super-resolution method using deep convolutional networks, demonstrating significant improvements in image quality and detail enhancement. Their work highlights the effectiveness of deep learning in image processing but also raises concerns about computational overhead and model interpretability.

Recent advancements in activity recognition systems have enabled innovative applications in health monitoring, smart environments, and human-computer interaction. Straczkiewicz et al. (10) conducted a systematic review of smartphone-based activity recognition methods, emphasizing their potential for health research and behavior analysis. Their findings indicate high accuracy in controlled environments, but real-world applications face challenges such as sensor variability and data privacy concerns.

Ranasinghe et al. (11) reviewed applications of activity recognition systems, focusing on performance evaluation and system design considerations. They concluded that while activity recognition has advanced significantly, standardization across datasets and methodologies is needed for broader applicability.

Vision-based human activity recognition has gained attention due to its applications in security, healthcare, and assistive technologies. Beddiar et al. (12) provided a comprehensive survey of vision-based human activity recognition techniques, covering various approaches, datasets, and performance metrics. Their study identified deep learning as a promising approach, yet challenges in real-time processing and occlusion handling remain.

Xin and Wang (13) researched image classification models based on deep convolutional neural networks, exploring their effectiveness in diverse image analysis tasks. Their results demonstrated improvements in classification accuracy, but model interpretability and dataset bias remain concerns.

Object detection plays a crucial role in computer vision applications, enabling the identification and localization of objects within images or video frames. Zhao et al. (14) conducted a review of object detection methods using deep learning, highlighting recent advancements and challenges in the field. Their analysis points to significant improvements in detection accuracy, yet computational efficiency and real-time performance remain areas of active research.

Jiao et al. (15) surveyed next-generation deep learning techniques for video object detection, discussing emerging trends and future research directions. While deep learning has enhanced object tracking and detection capabilities, the requirement for large-scale annotated datasets remains a challenge.

The integration of eye-tracking and head movement detection technologies has facilitated advancements in human-computer interaction, usability testing, and assistive technologies. Al-Rahayfeh and Faezipour (16) conducted a state-of-the-art survey on eye-tracking and head movement detection, summarizing key developments and applications in the field. Their findings highlight the growing importance of gaze-based interfaces, yet the high cost of commercial eye-tracking devices remains a barrier.

Wuetal. (17) reviewed recent advances in deep learning for object detection, discussing state-of-the-art architectures and optimization strategies. Their work emphasizes the potential of AI in enhancing detection accuracy, but challenges in computational efficiency and real-world deployment persist.

3.Methodology and Implementation

A. Implementation:

Python (Primary Language)

Python is the main programming language for this solution since Streamlit is a Python-based framework. It is ideal for handling the entire AI/ML pipeline, including data collection and preprocessing using libraries like Pandas, NumPy, and OpenCV. Additionally, Python supports deep learning model training and testing with TensorFlow/Keras or PyTorch. It is also used for real-time inference and seamless integration with the Streamlit application.

TensorFlow/Keras or PyTorch (Python Libraries for Deep Learning)

To implement and train the deep learning model for detecting plant diseases, TensorFlow/Keras or PyTorch is used. These frameworks provide pre-built functions for neural networks, allowing efficient model creation. They also offer GPU acceleration, making the training process faster.

Moreover, they integrate easily into the Python-based Streamlit application for real-time disease detection.

Streamlit Framework (Python)

Streamlit is used to build the web-based user interface for the crop disease detection system. It simplifies the creation of interactive applications without requiring backend coding. The framework includes built-in widgets that enable users to upload plant leaf images for analysis, adjust model parameters with sliders, and view predictions and model performance in an intuitive way.

B. SystemDesign

The process begins with collecting images of plant leaves to detect diseases. These images are then improved using **Keras' ImageDataGenerator**, which applies resizing, rescaling, zooming, and flipping to enhance model accuracy. The dataset is divided into three parts: **training, testing, and validation**. The training data helps the model learn, the test data checks its accuracy, and the validation data prevents overfitting. A **Convolutional Neural Network (CNN)** is built with multiple layers to identify patterns in the images. After training, the model can determine whether a plant leaf is healthy or diseased. To make this system more useful for farmers, it is integrated with an **AI chatbot powered by LLaMA 3 via Ollama**. This chatbot provides disease management advice and step-by-step guidance on crop cultivation, making the entire process easier and more accessible.

Data Collection begins with users uploading a dataset of plant leaf images, which may include both healthy and diseased leaves. These images are sourced from public datasets, field surveys, or directly provided by users.

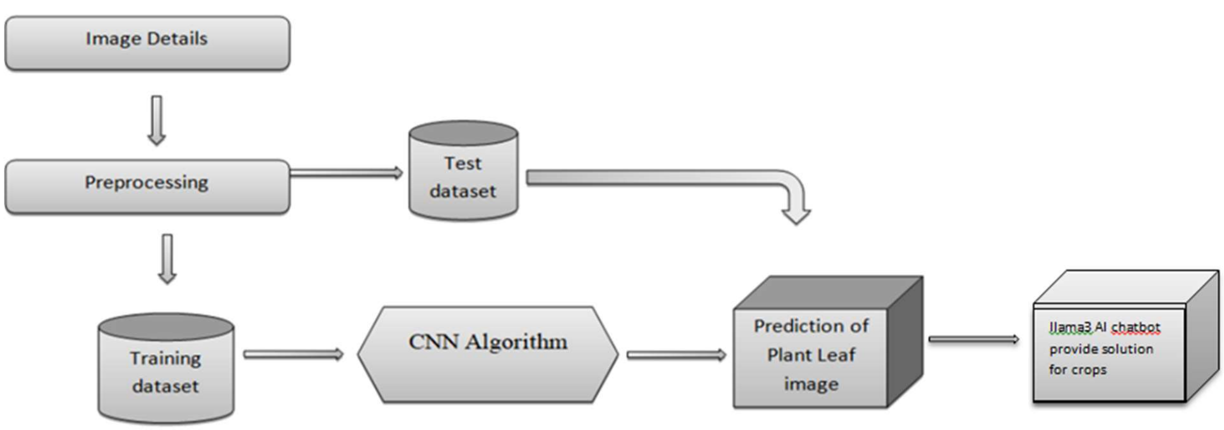

Fig 3.1 OverviewoftheSystemDesign

Data Preprocessing ensures the images are suitable for training and analysis. This involves resizing images to a consistent resolution, converting them to an appropriate format (such as grayscale or RGB), normalizing pixel values for uniformity, and applying data augmentation techniques like rotation and flipping to increase dataset diversity. Libraries such as OpenCV, NumPy, or TensorFlow's preprocessing utilities are used in this stage.

Model Training involves developing a deep learning model using a **Convolutional Neural Network (CNN)** for image classification. The model learns to distinguish between healthy leaves and different disease categories. The dataset is divided into training, validation, and testing sets. The model is trained using TensorFlow/Keras or PyTorch, and accuracy and loss are monitored throughout the process to ensure proper learning. The outcome of this stage is a trained model capable of detecting plant diseases.

Model Tuning and Testing focuses on optimizing the model's performance. Hyperparameters such as learning rate, batch size, and the number of epochs are adjusted for better accuracy. The model is tested on unseen data to evaluate performance metrics like accuracy, precision, recall, and F1-score. If necessary, further refinements are made to enhance its effectiveness.

Deployment via Streamlit integrates the trained model into a user-friendly interface. Users upload an image of a plant leaf through a file upload widget in the Streamlit application. The system preprocesses the image, runs it through the trained model, and provides the predicted category, such as "Healthy" or a specific disease type. Visual aids, like highlighted image areas, may be included to explain the model's decision-making process.

Result Presentation provides users with a detailed diagnosis, including disease classification (e.g., "Leaf Blight" or "Powdery Mildew"), a confidence score for the prediction, and possible remedies or treatment recommendations if integrated with an agricultural knowledge base.

Continuous Improvement is achieved through a feedback loop where users can label incorrect predictions and upload them for retraining, helping to refine the model over time. The system periodically incorporates new labeled data to improve classification accuracy.

DATA FLOW:

Raw Data Collection

The process begins with gathering images of plant leaves, which include both healthy and diseased samples. These images can come from public datasets, agricultural field surveys, or user-uploaded photos. A diverse dataset is essential for training an accurate disease detection model.

Data Preprocessing

Before training, the images need to be processed to ensure consistency and improve model performance. Preprocessing involves steps such as resizing images, adjusting colors, and normalizing pixel values. This step ensures that all images follow the same format, making them easier for the model to analyze.

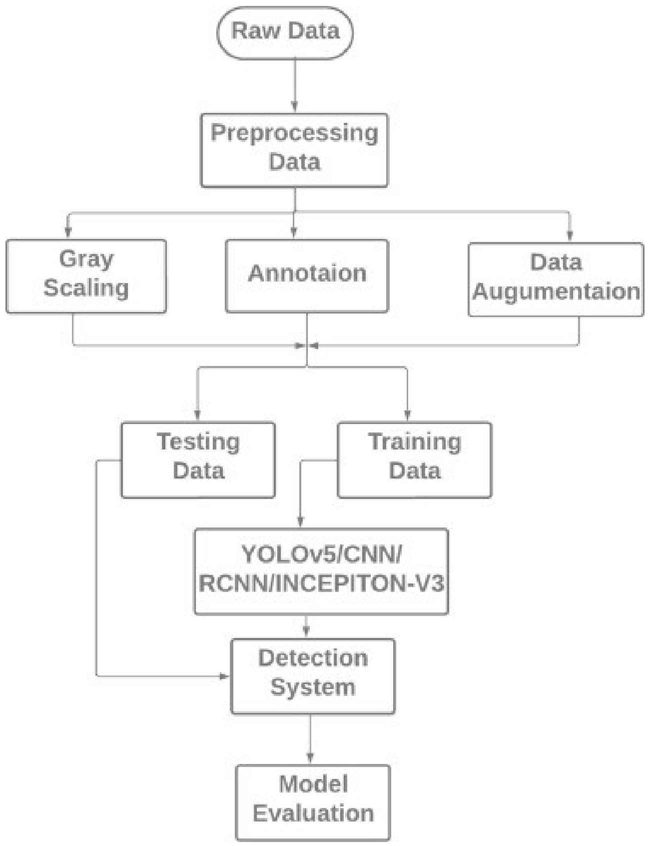

Fig: 3.2 System Data Flow Diagrams.

Processing Steps

Several techniques are applied to enhance the dataset:

- **Gray Scaling**: Converts images to grayscale to reduce complexity and improve processing speed.
- **Annotation**: Each image is labeled with the corresponding disease category, which helps the model learn disease patterns.
- **Data Augmentation**: Techniques such as rotation, flipping, and zooming are used to increase the dataset's diversity, helping the model generalize better.

Dataset Splitting

To build a reliable model, the dataset is divided into two main parts:

- **Training Data**: This portion is used to teach the model how to recognize different plant diseases.
- **Testing Data**: This set is used to evaluate how well the trained model performs on unseen images. A well-balanced dataset ensures better accuracy and generalization.

Model Training

A deep learning model is selected and trained to detect plant diseases from images. Popular models such as **YOLOv5, Convolutional Neural Networks (CNN), Region-based CNN (RCNN), and Inception-V3** are commonly used for image classification tasks. These models analyze the leaf features and learn to differentiate between healthy and diseased leaves.

Detection System

Once trained, the model is integrated into a detection system that allows users to upload new images for analysis. The system processes the image and provides a prediction, identifying whether the leaf is healthy or affected by a disease.

Model Evaluation

The final step is evaluating the model's accuracy and performance. The trained model is tested on unseen data, and its predictions are compared against actual labels. If necessary, adjustments such as fine-tuning hyperparameters or adding more training data are made to improve accuracy.

4. RESULTS:

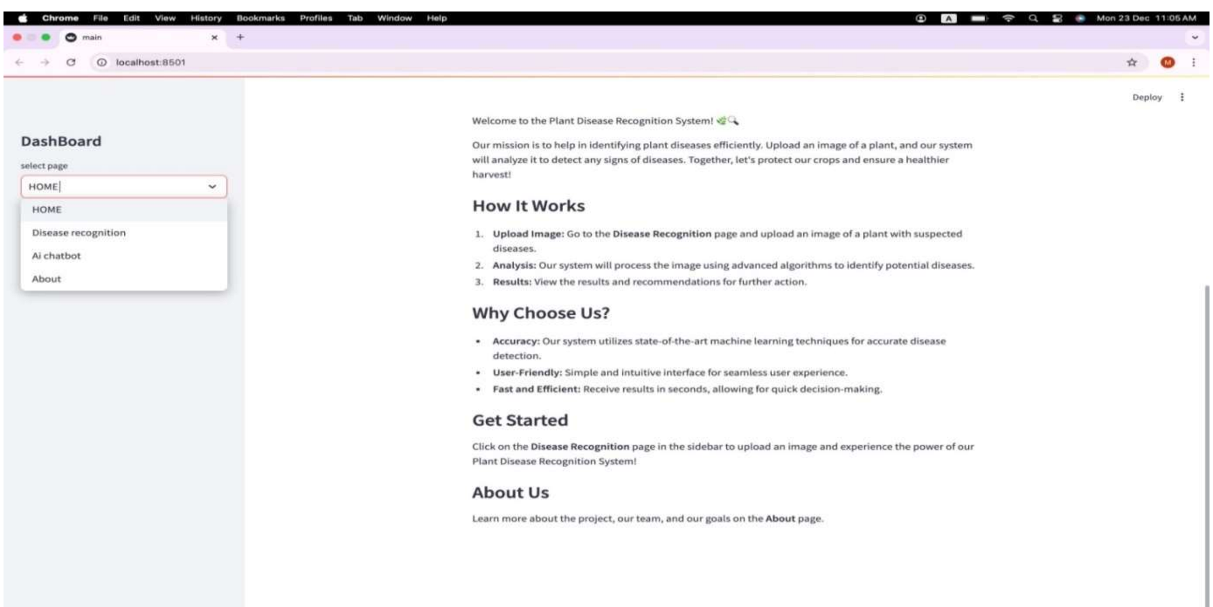

Fig 4.1 user interface home page

Plant Disease Recognition System Overview

The **Plant Disease Recognition System** is designed to help identify plant diseases quickly and efficiently. Users can **upload an image of a plant leaf**, and the system will analyze it using advanced **machine learning algorithms** to detect any potential diseases. The goal is to **protect crops and promote healthier harves**

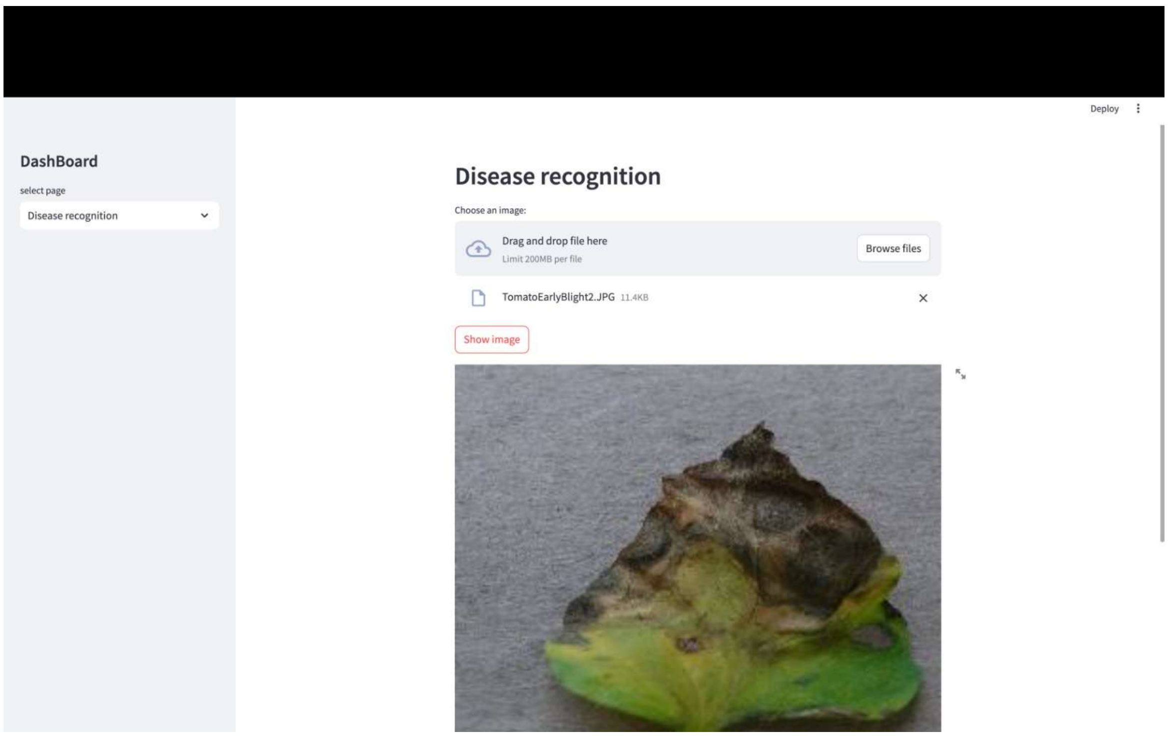

Fig 4.2: user interface of disease recognition

Dashboard Overview

The system has a **dashboard panel** on the left side, labeled **"Dashboard"**, which helps users navigate through different options. A **dropdown menu** is available for selecting various features of the application. In this case, the selected option is **"Disease recognition"**, meaning the system is designed to detect plant diseases from uploaded leaf images.

Uploading an Image

In the main part of the interface, users can **upload images of plant leaves** to check for diseases. They can either **drag and drop files** or **browse and select images** manually. The system allows files up to **200MB**, making it possible to analyze high-quality images. In the screenshot, a file named **"TomatoEarlyBlight2.JPG"** has been uploaded, suggesting the system might be checking for **early blight disease** in a tomato plant.

Showing and Processing the Image

After uploading, users can click the **"Show image"** button to display their image on the screen. The shown image in the screenshot features a **tomato leaf with dark spots and yellowing edges**, which are typical signs of **early blight**, a common fungal disease in tomato plants.

Disease Detection System

The system's main function is to **identify plant diseases using AI**. Once an image is uploaded, the system likely uses a **Convolutional Neural Network (CNN)** or another deep learning method to analyze it. The system then classifies the image as either **"Healthy" or "Diseased"** and may further specify the type of disease, such as **early blight, late blight, or powdery mildew**.

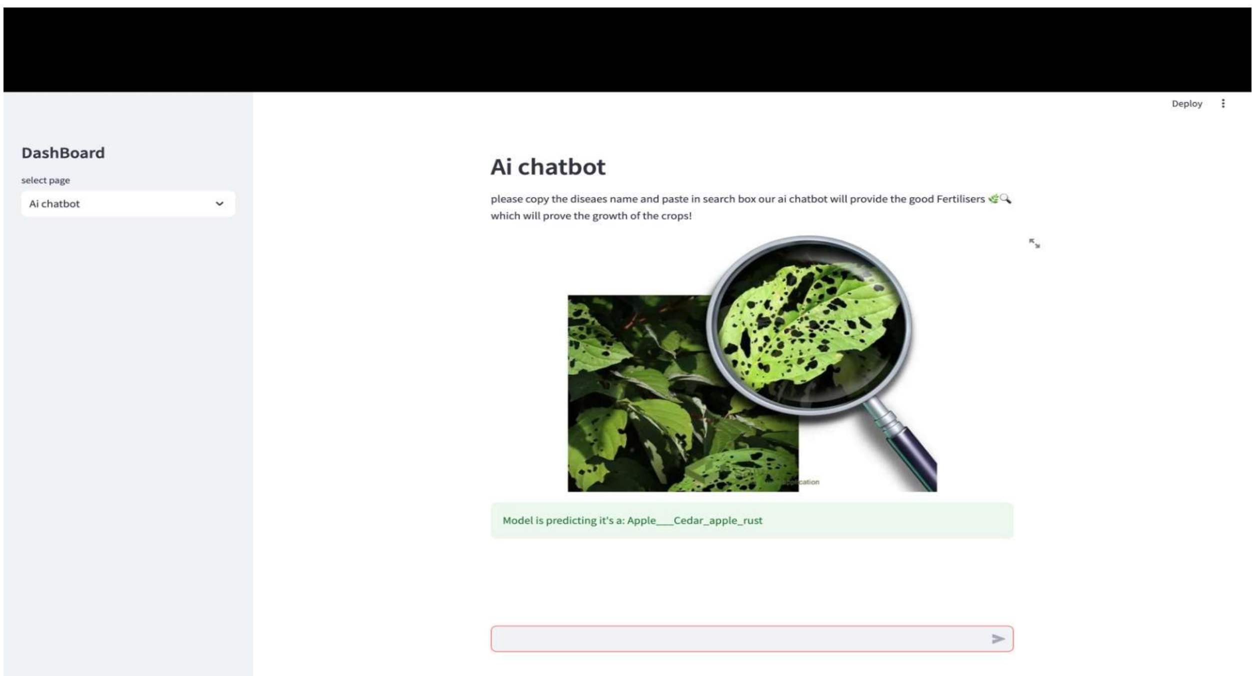

Fig 4.3: user interface of AI ChatBot

AI Chatbot for Plant Disease Detection

The AI chatbot feature is designed to help users identify plant diseases and recommend suitable fertilizers to improve crop growth. By analyzing an uploaded image, the system can detect signs of plant diseases and provide accurate predictions. This feature is particularly useful for farmers and agricultural experts who need quick and reliable disease detection.

The process begins when a user uploads an image of a diseased plant leaf. The AI model then processes the image using advanced machine learning techniques and predicts the disease affecting the plant. In the given example, the model has identified the disease as **Apple Cedar Apple Rust**, a common fungal infection in apple trees.

Once the disease is detected, the chatbot provides recommendations for the best fertilizers to treat the plant and promote healthy growth. Users can copy the predicted disease name and paste it into the chatbot's search box to receive specific fertilizer suggestions. This helps in selecting the right nutrients for better plant recovery.

This AI-powered system offers several benefits, including fast and accurate disease detection, personalized fertilizer recommendations, and a user-friendly interface. By integrating AI into plant health monitoring, this chatbot makes crop disease management more efficient and helps users take quick action to protect their plants.

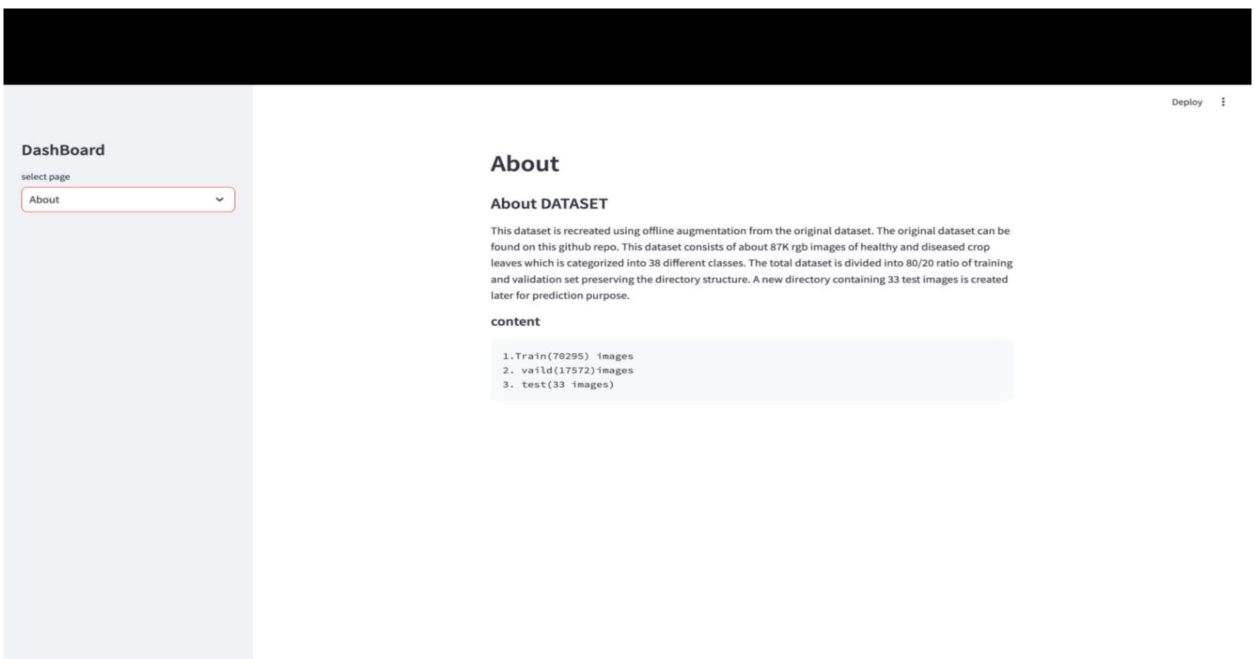

Fig 4.4: user interface of About Dataset

Dataset Overview

This system uses a dataset that has been improved with offline augmentation techniques. The original dataset, available on a GitHub repository, contains around **87,000 color images** of crop leaves, including both healthy and diseased ones. These images are sorted into **38 different categories**, making it easier for the model to recognize different plant diseases.

To ensure better training, the dataset is split into two sections: **80% is used for training the model, while 20% is reserved for validation**. The folder structure is kept intact to maintain organization. Additionally, a separate folder with **33 test images** is created for later use in making predictions. This structured approach helps the AI system learn effectively and improve the accuracy of disease detection.

5.CONCLUSION :

AI-powered crop disease detection can transform agriculture by enabling early disease identification, accurate diagnosis, and precise treatment recommendations, helping farmers protect their crops efficiently. These systems offer key benefits such as early detection to prevent major losses, precise pesticide and fertilizer application to minimize waste, and eco-friendly farming by reducing chemical overuse. Additionally, they improve accessibility, allowing even small-scale farmers to enhance productivity. While challenges like changing weather conditions, internet connectivity issues, and user adoption exist, as well as internal limitations like model accuracy and maintenance, these can be mitigated through regular updates, robust system design, and proper user training, ensuring AI-based disease detection remains effective and reliable.

Future Scope :

India depends heavily on agriculture, with many farmers facing losses due to undetected crop diseases. To solve this, we propose an AI-based system that uses deep learning to identify plant diseases early by analyzing leaf features like shape and texture. This system will be simple to use and rely on a large image database for accurate detection. It aims to reduce crop loss, improve productivity, and support farmers effectively.

REFERENCES:

1. Newton, D., & Essex, R. (2023). A systematic review of cheating in online exams: Prevalence and prevention strategies. Educational Integrity Journal, 18(3), 45-62
2. Ruiperez-Valiente, J. A., Veeramachaneni, K., Marchand-Maillet, S., & Alario-Hoyos, C. (2019). Using machine learning to detect multiple-account cheating in online learning environments. Computers & Education, 141, 103611.
3. Ramberg, J., & Modin, B. (2019). School effectiveness, academic achievement, and moral standards: Investigating cheating behavior among students. Scandinavian Journal of Educational Research, 63(1), 95-110.
4. Yadav, S. S., & Jadhav, S. M. (2019). Deep convolutional neural networks for medical image analysis: An overview. Biomedical Signal Processing and Control, 53, 101693.
5. Gopal, A., & Ganesan, M. (2022). Real-time deep learning framework for monitoring social distancing. Expert Systems with Applications, 195, 116563.
6. Valliappan, N., Seshadri, S., & Kim, S. J. (2020). Smartphone-based eye-tracking system for real-time gaze estimation. IEEE Transactions on Biomedical Engineering, 67(5), 1402-1412.
7. Stein, M., Bulling, A., & Gellersen, H. (2021). Eye-tracking latencies in commercial head-mounted displays: A comparative study. Virtual Reality, 25(3), 587-601.
8. Grigorescu, S., Trasnea, B., Cocias, T., & Macesanu, G. (2020). A survey of deep learning techniques for autonomous driving. Journal of Field Robotics, 37(5), 889-911.
9. Dong, C., Loy, C. C., He, K., & Tang, X. (2016). Image super-resolution using deep convolutional networks. IEEE Transactions on Pattern Analysis and Machine Intelligence, 38(2), 295-307.
10. Straczkiewicz, M., James, P., Onnela, J. P., & Dunton, G. F. (2021). Smartphone-based activity recognition for health research: A systematic review. Digital Biomarkers, 5(1), 56-72.
11. Ranasinghe, S., Al Machot, F., & Mayr, H. C. (2016). A review on applications of activity recognition systems with machine learning. Computer Science Review, 18, 1-23.
12. Beddiar, D. R., Nini, B., Sabokrou, M., & Hadid, A. (2020). Vision-based human activity recognition: A comprehensive survey. Pattern Recognition, 108, 107561.
13. Xin, Y., & Wang, D. (2019). Deep convolutional neural networks for image classification: A review. Neural Computing and Applications, 32(10), 6125-6146.
14. Zhao, Z. Q., Zheng, P., Xu, S. T., & Wu, X. (2019). Object detection with deep learning: A review. IEEE Transactions on Neural Networks and Learning Systems, 30(11), 3212-3232.
15. Jiao, L., Zhang, F., Liu, F., Yang, S., Li, L., Feng, Z., & Qu, R. (2022). A survey of deep learning-based object detection in video. Pattern Recognition, 122, 108275.
16. Al-Rahayfeh, A., & Faezipour, M. (2013). Eye-tracking and head movement detection: A state-of-the-art survey. IEEE Transactions on Human-Machine Systems, 43(4), 405-417.
17. Wu, Y., Kirillov, A., Massa, F., Lo, W. Y., & Girshick, R. (2020). Recent advances in deep learning for object detection. Computer Vision and Image Understanding, 193, 102898.
18. Dr.Pampapathi B M, Archana BK , Apoorva K and Ashwini K " IOT Pet-Feeder With Home Security Robot" in Journal of Xidian University - May 2024 -VOLUME 18, ISSUE 5, https://doi.org/10.5281/Zenodo.11180790 - ISSN No:1001-2400.
19. Dr.Pampapathi B M, Arya R Kulkarni, M B Preetham and Harish B S "Detecting Suspicious Activities in Exam Hall to Prevent Cheating" in Solovyov Studies - May 2024-VOLUME 72, ISSUE 5 , ISSN : 2076-9210 https://drive.google.com/file/d/1IPJ1m6JBFn0BH8ndO17ncNaEJ-Tbd2s5/view

20. Pampapathi B M, A Madhuri, Chennareddy Nikhil, Amar Gouda Patil "Water Monitoring And Purification Of Waste Water For Agriculture Using Iot" in Journal For Basic Sciences Volume 23, Issue 4, 2023 , https://doi.org/10.37896/JBSV23.4/2050.
21. Pampapathi B M, Mohammad Moshin P , Mohammed Kareemuddin Saqlain, Prajwal Marthur, K Md Ibrahim hussain **"**Wireless Fire Detection Systems Using Iot" in NOVYI MIR Research Journal , Volume 8 Issue 4 2023 https://doi.org/16.10098.NMRJ.2022.V8I4.256342.37538
22. Pampapathi B M , Shruthi S M," Detection and Classification of Phishing Websites Using Machine Learning" , in Journal Of Technology - Aug 2023 - Issn No:1012-3407 , Vol 13, Issue 8, D.O.I- https:// 10.61350/v13-105368 .
23. Pampapathi B M , Nageswara Guptha M , M S Hema ," Towards an effective deep learning-based intrusion detection system in the internet of things" , in Telematics and Informatics Reports Journal- May 2022 , https://doi.org/10.1016/j.teler.2022.100009.
Volume 7, September 2022, 100009.
24. Pampapathi, B.M., Nageswara Guptha, M. & Hema, M.S. Data distribution and secure data transmission using IANFIS and MECC in IoT. *J Ambient Intell Human Comput* **13**, 1471–1484 (2022). https://doi.org/10.1007/s12652-020-02792-4.
25. Pampapathi B M , Nageswara Guptha M , M S Hema , "Malicious Node Detection and Energy-aware Optimal Routing in Wireless Sensor Networks using CD-LVQ and BMSSO Algorithms" in The Journal of Huazhong University of Science and Technology ,Volume 50 , Issue 03.-March 2021- http://hustjournal.com/vol50mar-2/.
26. Pampapathi B M , Nageswara Guptha M , M S Hema , "Energy Efficient Data Distribution on Cloud With Optimal Routing Path Based Congestion Control in WSN Environment" in Journal of University of Shanghai for Science and Technology(JUSST), Volume 23, Issue 8, August 2021, https://doi.org/10.51201/JUSST/21/08409.
27. Pampapathi B M , Chandana Murthy , Supritha Kumar , Pooja M , Supriya K "Survey on IOT Based Medical Box for Elderly People" in International Journal of Advanced Trends in Computer Science and Engineering (IJATCSE) ISSN 2278-3091, Vol.10 No.3 (May – June 2021 issue), https://doi.org/10.30534/ijatcse/2021/531032021.