International Journal of Information Technology & Computer Engineering

ISSN 2347–3657

Volume 13, Issue 3, 2025

Agro Care

B Anitha, **Guggilla Rithika, Lankala Ruchitha, Gunishetty SaiNamitha**

[1]Assistant Professor, Department of Information Technology, Bhoj Reddy Engineering College for Women

[2.3.4]B,tech students, Department of Information Technology, Bhoj Reddy Engineering College for Women

ABSTRACT

Agriculture is the backbone of many economies and a crucial source of livelihood for millions. However, crop production is often threatened by various plant diseases that can lead to significant losses in yield and quality. Early detection and accurate diagnosis of these diseases are essential for effective crop management and sustainable farming practices.AgroCare is an AI-powered web application developed to assist in the automatic detection of plant diseases using deep learning techniques, particularly Convolutional Neural Networks (CNN). The system enables users to upload images of plant leaves, which are then analyzed by a trained CNN model to identify the presence and type of disease. Once diagnosed, AgroCare also provides suggestions for appropriate treatments or supplements to help manage the disease effectively.Built using Python, Flask, and deep learning frameworks like PyTorch or TensorFlow, AgroCare offers a fast, accurate, and user-friendly solution for farmers and agricultural professionals. By automating the disease detection process, AgroCare aims to support smarter agricultural practices, reduce reliance on manual inspection, and improve crop health through timely intervention.

Keywords : CNN(Convolutional Neural Networks), Image Classification, Plant Disease Detection,Flask Web Application

1. INTRODUCTION

Agriculture plays a vital role in the survival and development of human civilization. However, plant diseases remain a major threat to crop productivity and food security worldwide[1]. These diseases are caused by various factors such as fungi, bacteria, viruses, and environmental stress, and can spread rapidly if not detected and treated in time. Traditional disease diagnosis methods often require expert knowledge, lab facilities, and manual inspection, which may not always be feasible for farmers, especially in remote areas[2]. To address these challenges, deep learning-based image recognition techniques have emerged as a powerful solution.

In this project, we propose **AgroCare**, an AI-based web application designed to automatically detect plant diseases from leaf images using Convolutional Neural Networks (CNN)[3]. By leveraging a large dataset of plant disease images, the CNN model is trained to accurately classify different types of plant infections[4]. The web interface allows users to upload leaf images and instantly receive the disease name along with recommended treatment suggestions. This system provides a scalable, efficient, and accessible alternative to manual diagnosis[5]. AgroCare aims to empower farmers and agricultural workers with timely and accurate information to take preventive or corrective action, thereby reducing crop losses and improving yield quality.

Existing System:

The current approaches to identifying plant diseases largely depend on manual observation and expert consultation. Farmers or agricultural specialists visually inspect plants for signs of disease, which is often time-consuming and prone to human error. In certain cases, laboratory analysis is used to confirm the disease, which further delays timely intervention. These traditional methods lack precision, are not scalable for large-scale farming, and often result in delayed detection, leading to reduced crop yield and increased financial losses.

Proposed System:

The proposed system introduces an automated plant disease detection model using deep learning, specifically Convolutional Neural Networks (CNN). The model is trained on a large dataset of plant leaf images (PlantVillage) to accurately classify diseases across various plant species. A web-based interface built with Flask allows users to upload images of leaves and instantly receive predictions about the presence and type of disease. This system enables early detection, improves diagnosis accuracy, and supports farmers in taking timely corrective actions. The model can also be further fine-tuned to expand its capabilities across additional crops and diseases.

2. RELATED WORK

Survey:

While previous studies have explored the use of UAV-based imagery, RGB images, and spectral data for plant disease detection, there has been limited investigation into the comparative performance of object detection and classification models for accurate early detection. Mohanty et al.[1] created models with AlexNet and GoogLeNet architectures, using RGB images of plant leaves. Their work achieved an accuracy of 99.35% with the GoogLeNet model on the PlantVillage dataset. For example, using a LeNet architecture, Amara et al.[2] detected banana leaf diseases in three categories and achieved an accuracy ranging from 92% to 99%. Although the initial LeNet model performed well, the research highlights the importance of image preprocessing to improve CNN model accuracy. In contrast, Oppenheim and Shani[3] utilized a VGG-based CNN to classify potato leaf diseases, achieving a peak

accuracy of 96% when using 90% of the data for training, indicating the effectiveness of CNN-based approaches in real-time applications. Additionally, Fuentes et al.[4] describe how object detection techniques using CNN filter banks achieved a 96% recognition rate in tomato disease detection, demonstrating the potential of object detection methods in precision agriculture. However, the study did not incorporate ensemble methods for enhanced performance. Therefore, in order to better classify and localize diseases in crop images, Zhang et al.[5] used an improved Faster RCNN architecture combined with k- means clustering, achieving an mAP of 98.54% for tomato diseases. Instead of employing ResNet models for classification only, Chen et al.[6] enhanced the YOLOv5 model's neck module, SE module, and loss function to identify rubber tree diseases, reaching an accuracy of 70%. In their attempt to compare classification and object detection frameworks, the authors of the IEEE paper [7]highlighted the gaps in robust dataset availability and proposed extensive computational analysis using YOLOv5 and ResNet50 on the PlantDoc dataset. To categorize plant leaves as diseased or healthy, they tested 18 classification algorithms and five object detection algorithms, noting that YOLOv5 performed best for object detection tasks. However, due to challenges in generalizing these models to real-world, noisy environments, the need for lightweight, memory-efficient models remains an open research area.[8]

3. REQUIREMENT ANALYSIS

Functional Requirements:
A model is required which is used to detect the plant disease from the uploaded leaf image. Python library like pandas is required for managing datasets and reading supplement info. Libraries like OpenCV are used to handle and preprocess uploaded images.A CNN model is used for detecting features and classifying diseases from the image. PyTorch framework is used to build and train this CNN model. Flask is used for building the web interface through which image is uploaded.Functional requirements for building AgroCare using deep learning and Flask include:

- User Management
- Image Upload
- Disease Classification
- Supplement Recommendation
- Result Display
- Model Retraining
- Secure Login System
- Admin Dashboard
- Mobile and Web Access
- Profile Management

Non-Functional Requirements:
Image must be preprocessed before use — it must be resized and normalized before feeding into the model. Uploaded files should be validated for format and size. If preprocessing is skipped, the predictions will be inconsistent.Libraries like OpenCV and torchvision help achieve preprocessing. The system's performance, reliability, and security are maintained throughout the application. Non-functional aspects are focused on speed, uptime, and safe data storage.
Scalability:

- Data Handling: The system must support the expansion of plant image data storage and processing as the number of users grows.
- User Load: It should efficiently manage a growing number of simultaneous users, including farmers and agricultural experts accessing the platform.

Performance:

- Response Time: The system should offer fast processing times for image classification and result display, ensuring quick feedback to the users.
- Throughput: Capable of handling large volumes of uploaded images without significant performance degradation during peak usage.

Reliability:

- Uptime: High system availability with minimal downtime to ensure continuous disease monitoring and user access.
- Error Handling: Robust mechanisms to detect, log, and recover from system failures or input errors.

2.2 Software Requirements:

2.3

Operating Systems

: Windows & macOS
Front-End

: HTML ,CSS,JS
Web Framework

: Flask using Python
Environment

: Venv

2.4 Hardware Requirements:
Processor

: I3 / Intel Processor
Ram

: 8GB.
Hard Disk

: 1TB

4-DESIGN

System Architecture:

International Journal of Information Technology & Computer Engineering
ISSN 2347–3657
Volume 13, Issue 3, 2025

It describes the structure and behavior of technology infrastructure of an enterprise, solution or system. In other words, System architecture can be described as the flow of application which is represented below in the pictorial form.The purpose of system architecture activities is to define a comprehensive solution based on principles, concepts, and properties logically related to and consistent with each other. The solution architecture has features, properties, and characteristics which satisfy, as far as possible, The problem or opportunity expressed by a set of system requirements (traceable to mission/business and stake holders requirements).

System architecture is abstract, conceptualization-oriented, global, and focused to achieve the mission and life cycle concepts of the system. It also focuses on high-level structure in systems and system elements. It addresses the architectural principles, concepts, properties, and characteristics of the system-of-interest. It may also applied to more than one system, in some cases forming the common structure, pattern, and set of requirements for classes or families of similar or related systems.

The SEBoK considers systems engineering to cover all aspects of the creation of a system, including system architecture.

The majority of interpretations of system architecture are based on the fairly intangible notion of structure (i.e relationships between elements). Some authors limit the types of structure considered to be architectural: for example, restricting themselves to functional and physical structure. Recent practice has extended consideration to include behavioral, temporal and other dimensions of structure.

ISO/IEC/IEEE 42010 Systems and Software Engineering – Architecture Description (ISO 2011) provides a useful description of the architecture considering the stakeholder concerns, architecture viewpoints, architecture views, architecture models, architecture descriptions, and architecting throughout the life cycle.

A discussion of the features of systems architectures can be found in(Maier and Rechtin 2009). An attempt to develop and apply a systematic approach to characterizing architecture belief systems in systems engineering has been described by the INCOSE UK Architecture Working Group (Wilkinson et al.2010, Wilkinson 2010).

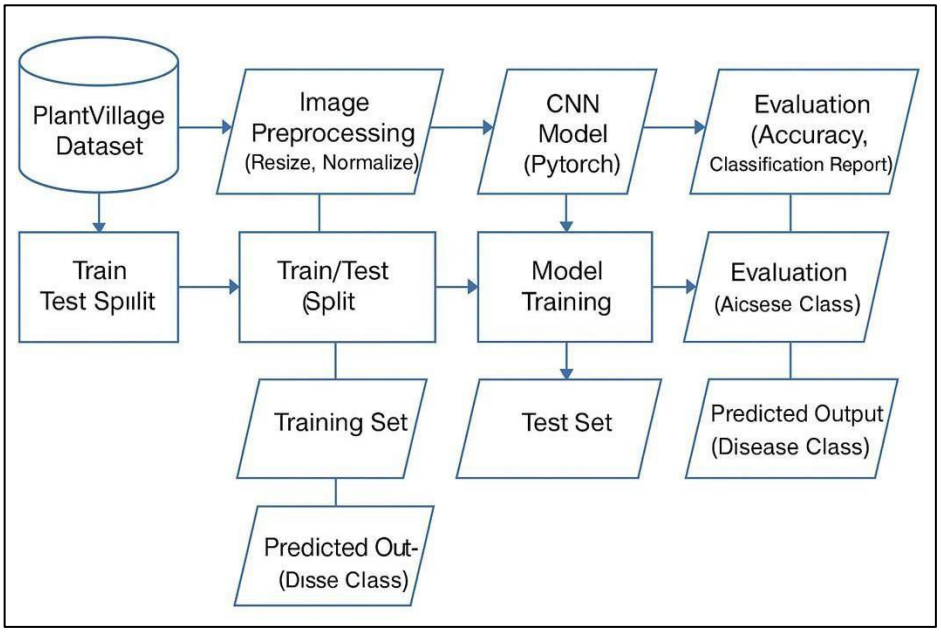


Fig. 4.1.1.1 System Architecture

Technical Architecture:

Technical Architecture refers to the structural process of designing and building system's architecture with focus on the users and sponsors view of the environment. Technology architecture associates application components from application architecture with technology components representing software and hardware components. Its components are generally acquired in the market place and can be assembled and configured to constitute the enterprise's technological infrastructure. A technical architecture diagram provide a bird's eye view of the infrastructure of our project. The diagram illustrates how components in a system interact with one another in the large scale of things. Technical Architecture (TA) is a form of IT architecture that ids used to design computer system. It involves the development of a technical blueprint with regard to the arrangement, interaction, and interdependence of all elements so that system-relevant requirements are met.

AgroCareDesign

Throughout the past decade, architecture has become a broadly used term in the context of information

International Journal of Information Technology & Computer Engineering
ISSN 2347–3657
Volume 13, Issue 3, 2025

technology. This doesn't come as a surprise considering how most companies had to redesign their IT landscape to adopt digital trends like cloud computing software as service (SaaS). This digital transition required not only skilled developing teams but first and foremost IT architects. In their roles as IT strategists and planners, they map out a target architecture and make sure that all IT decisions align with business goals and requirements.

But IT architecture encompasses a variety o different roles and disciplines that are sometimes difficult to tell apart. This is largely due to highly dynamic nature of IT, its widespread adoption throughout all industries and business that have developed their own practices. In general, there's differentiation between enterprise architecture, solution architecture and technology architecture. In order to understand what technology architecture means, it's helpful to examine the term architecture on its own.

At its core, the term architecture describes the formation of a structure by strategically assembling single components. In this process of assembling, the architect has to adhere to certain rules or requirements like legal constraints, financial constraints or scientific laws. In the world technology architecture design, the focus lies on technology limitations, meaning that a technology architect makes sure that a new application is compatible with the existing technology at a company by specifying things like the communications network or hardware that it uses.

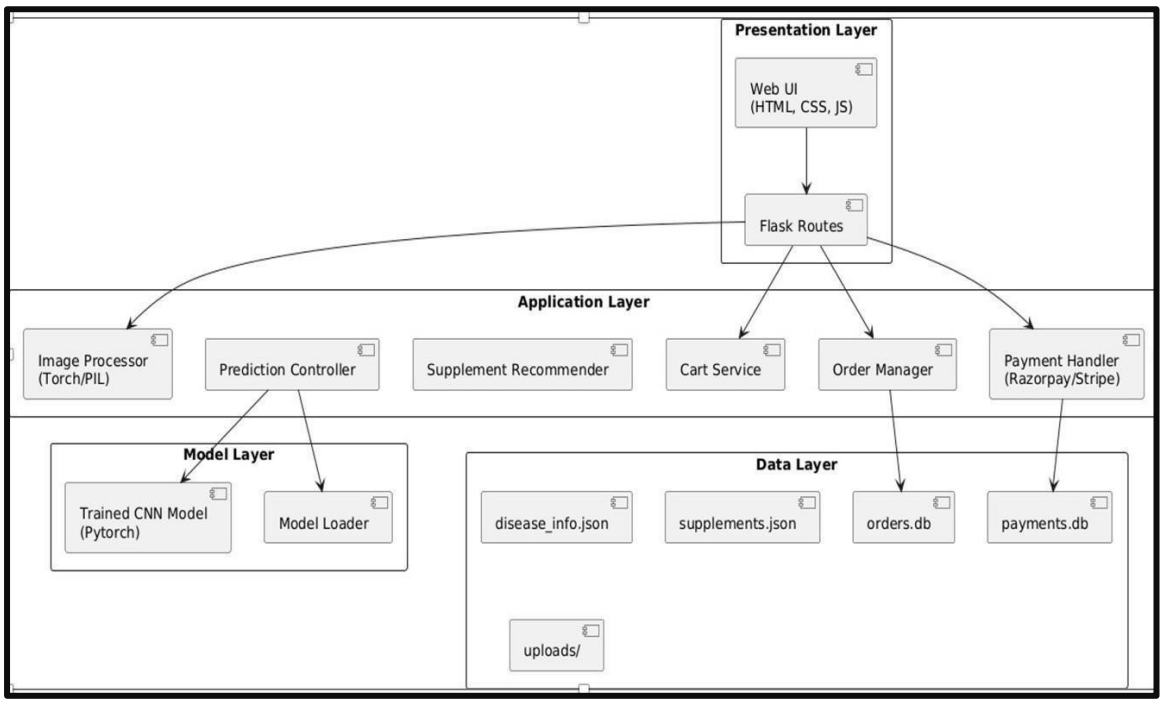


Fig. 4.1.2.1 Technical Architecture

5-IMPLEMENTATION

2.5 Libraries

- *Pandas:*
  Pandas is a data manipulation library that offers data structures like DataFrames for handling tabular data. It is used here to load and manage the supplementary dataset containing disease information, treatments, and supplements. Pandas makes it easy to query and retrieve relevant disease details after classification.
- *Numpy:*
  NumPy is a fundamental Python library used for numerical computing. It provides support for large, multi-dimensional arrays and matrices, along with a collection of mathematical functions to operate on these arrays efficiently. In this project, NumPy helps handle image data as arrays and perform numerical operations during preprocessing and analysis.
- *OpenCv:*
  OpenCV (Open Source Computer Vision Library) is used for image processing tasks. It provides tools to read, resize, clean, and manipulate images. In the plant disease detection system, OpenCV is used to preprocess uploaded leaf images, such as resizing them to a fixed size and enhancing image quality for better model input.
- *Pytorch:*
  Pytorch is a popular deep learning framework used to build and train neural networks.It provides flexible tools fro defining CNN architectures , automatic differentiation, and GPU acceleration.In this project , Pytorch is the core library used to implement the

International Journal of Information Technology & Computer Engineering

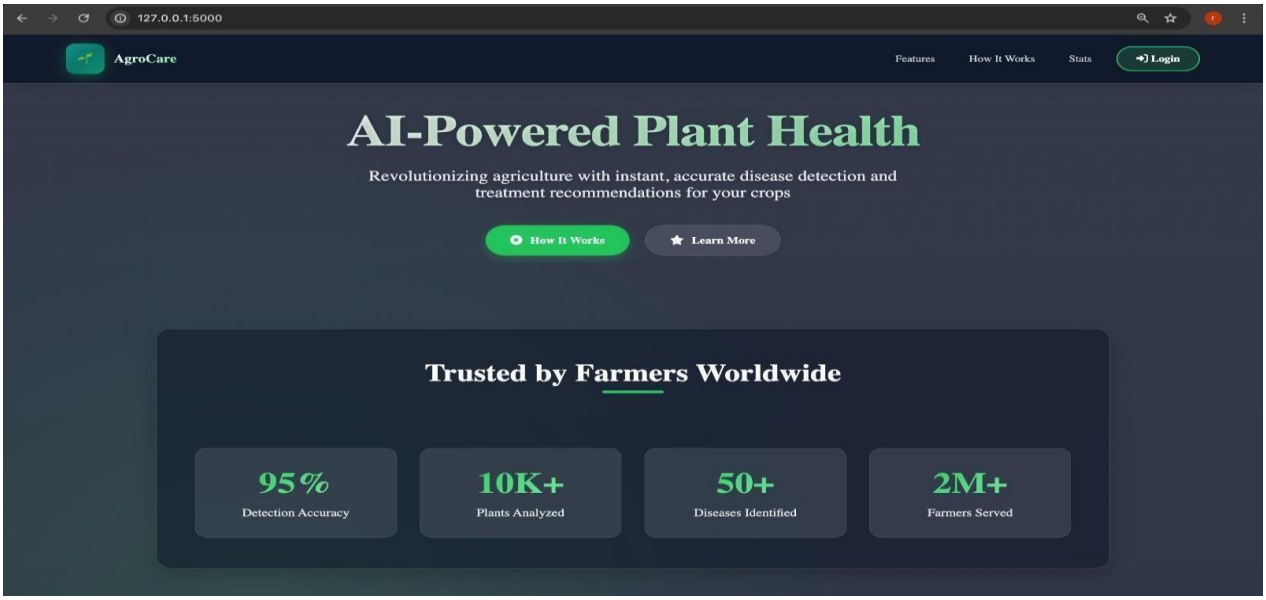
ISSN 2347–3657

Volume 13, Issue 3, 2025

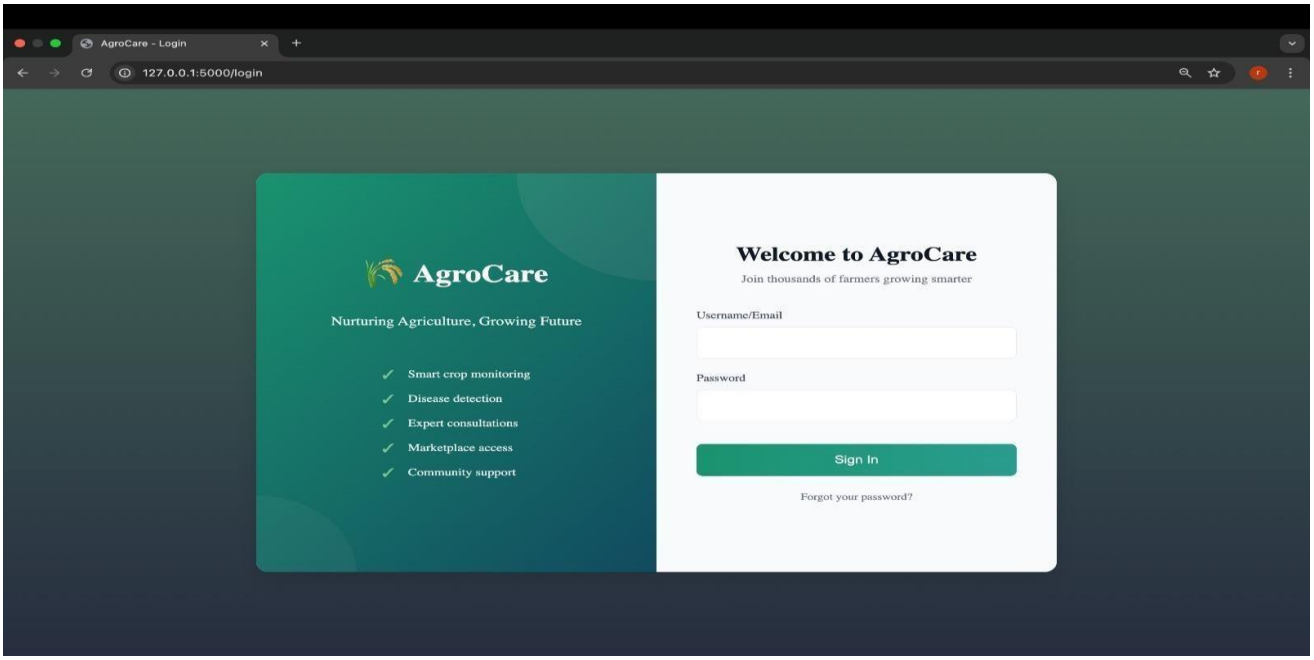
CNN model , perform training , validation and prediction.

- *PIL(Python Imaging Library)/Pillow:*
  PIL or its modern fork Pillow is used for image processing in Python. It allows opening, manipulating, and saving image files in various formats. In this project, it helps convert uploaded images into formats compatible with PyTorch tensors and assists in any additional image preprocessing required..
- *Flask:*
  Flask is a lightweight Python web framework used to build the web interface for the plant disease detection system. It handles user requests, such as image uploads, and passes these images to the CNN model for prediction. Flask then fetches the relevant disease information and displays the results back to the user through dynamic web pages.Its simplicity makes it easy to integrate with HTML, CSS, and JavaScript for a smooth user experience. Flask is ideal for small to medium projects like this because it allows quick development and easy deployment on local servers or cloud platforms.

6 SCREENSHOTS



Screenshot 6.1 Agrocare Landing Page

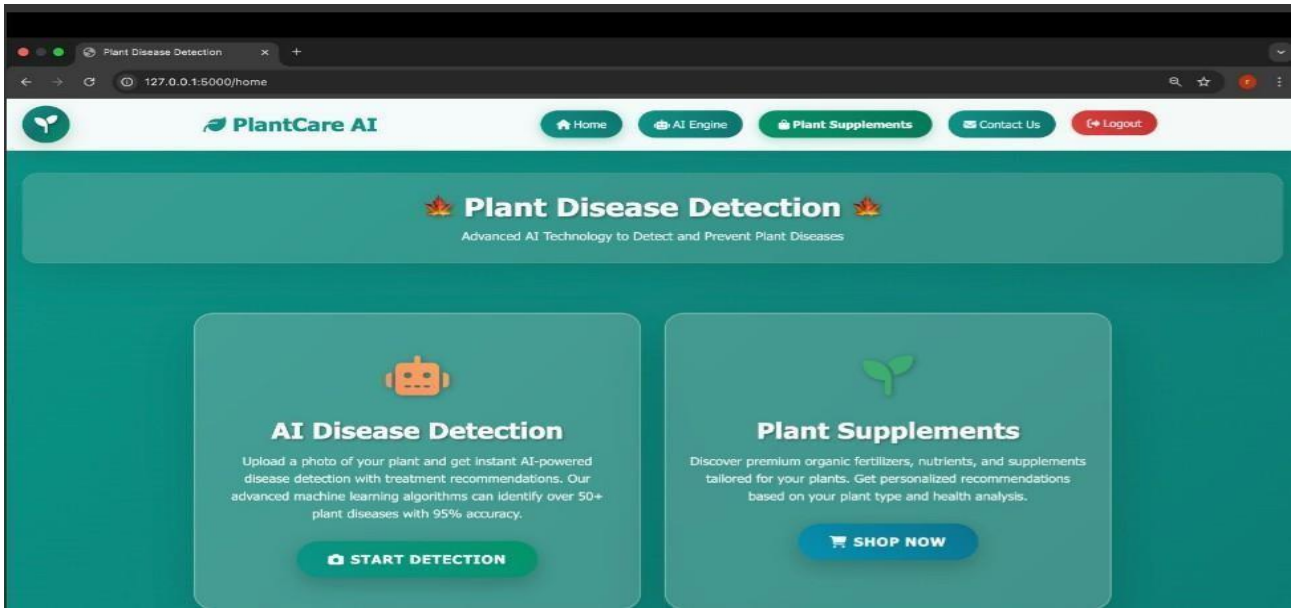


International Journal of Information Technology & Computer Engineering

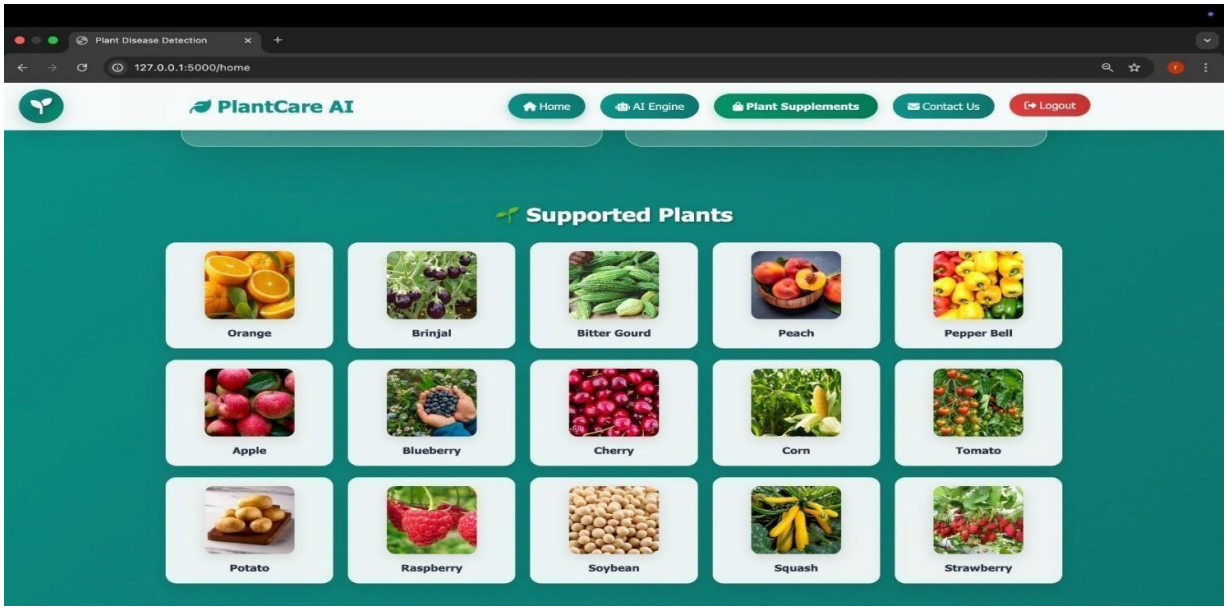
ISSN 2347–3657

Volume 13, Issue 3, 2025

Screenshot 6.2 Agrocare Login Page



Screenshot 6.3 Agrocare Home Page

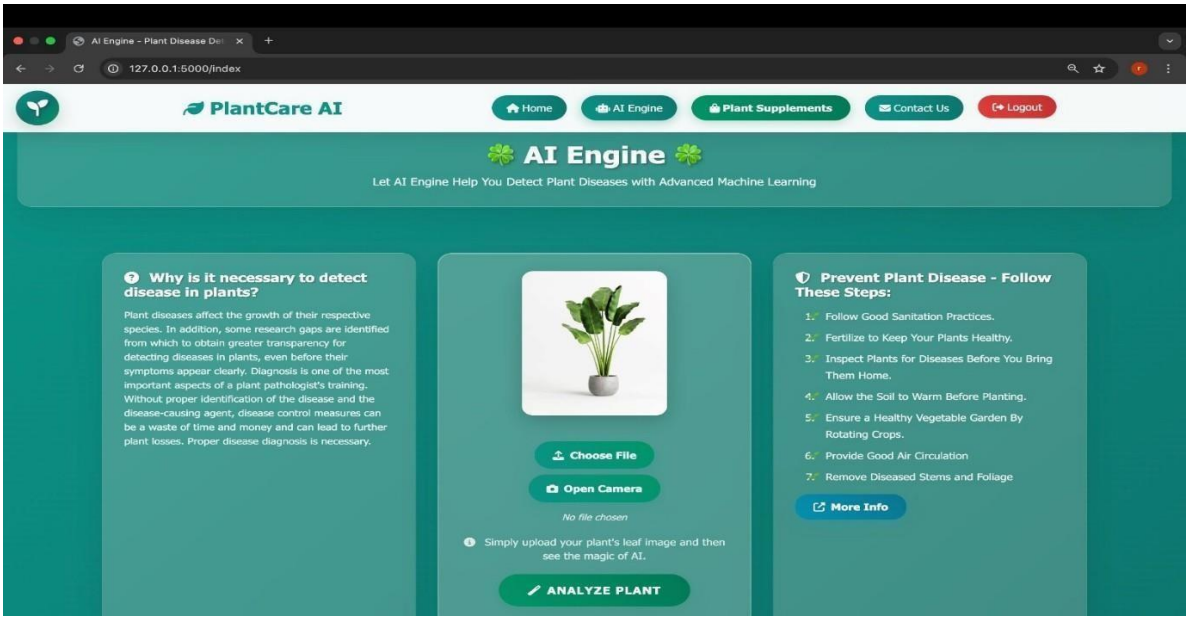


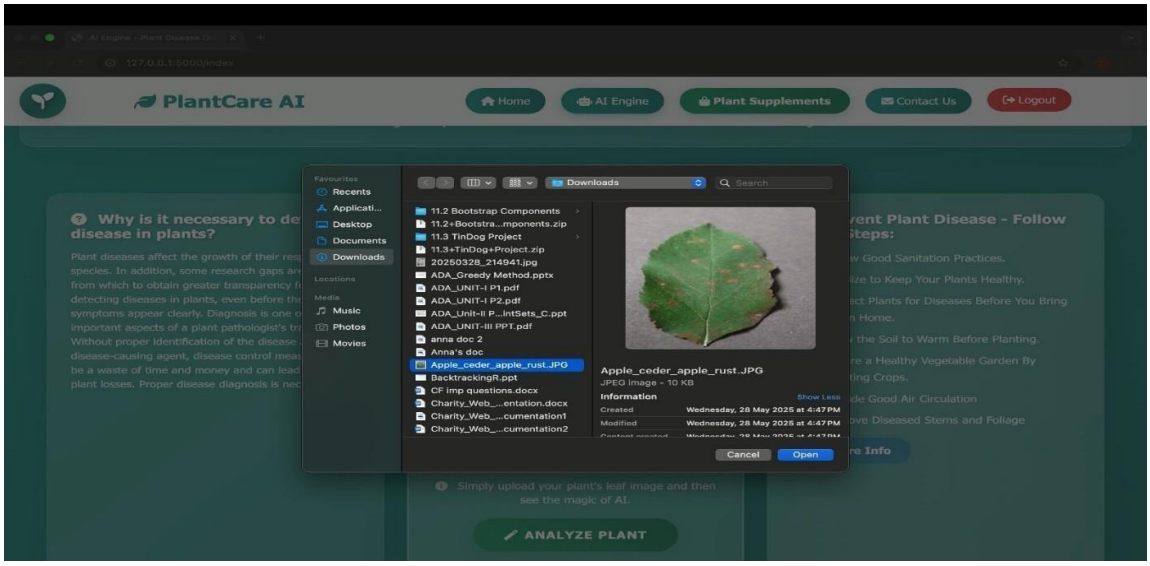
Screenshot 6.4 Agrocare Prediction Page

International Journal of Information Technology & Computer Engineering

ISSN 2347–3657

Volume 13, Issue 3, 2025



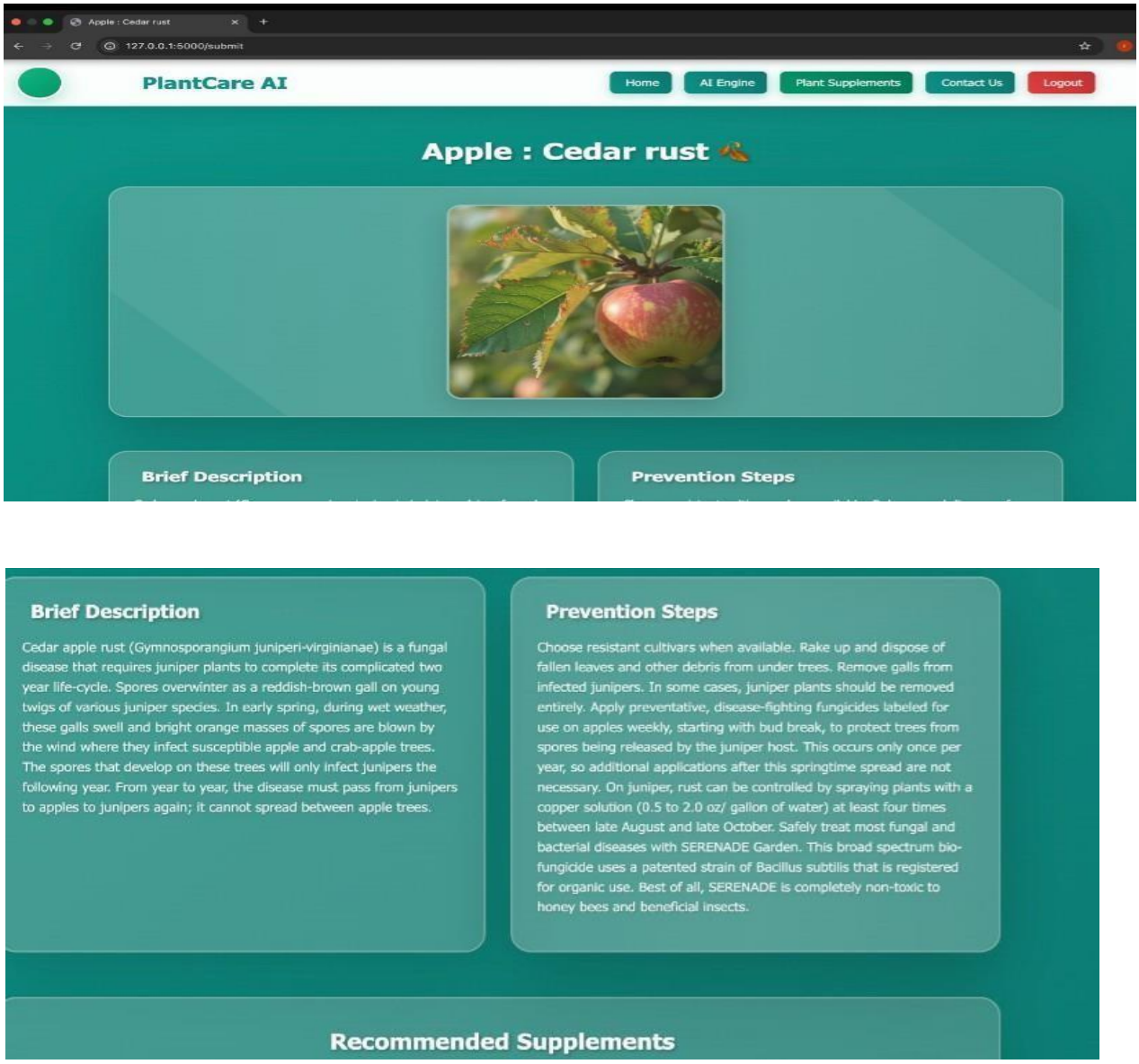


Screenshot 6.5Agrocare Detection Page

International Journal of Information Technology & Computer Engineering

ISSN 2347–3657

Volume 13, Issue 3, 2025



Screenshot 6.6 Agrocare Supplements Page

7-CONCLUSION

Using plant leaf image data, our deep learning-based Plant Disease Detection system—powered by a CNN model in PyTorch—achieves reliable disease classification performance across multiple crop types. The model is integrated into a Flask-based web application that allows users to upload plant images and receive real-time disease predictions with supplement recommendations. The system provides a non-invasive, user-friendly, and scalable solution to assist farmers and agriculturists in managing plant health proactively.

Our findings indicate that image-based detection offers an accurate and rapid approach for identifying plant diseases.

REFERENCES

[1] S. P. Mohanty, D. P. Hughes and M. Salathé,"Using Deep Learning for Image-Based Plant Disease Detection,"*Frontiers in Plant Science*, vol. 7, p. 1419, 2016. [Online]. Available: https://doi.org/10.3389/fpls.2016.01419

[2] K. P. Ferentinos, "Deep learning models for plant disease detection and diagnosis," *Computers and Electronics in Agriculture*, vol. 145, pp. 311–318, 2018. [Online].Available: https://doi.org/10.1016/j.compag.2018.01.009

[3] V. Singh and A. K. Misra, "Detection of plant leaf diseases using image segmentation and soft computing techniques," *Information Processing in Agriculture*, vol. 4, no. 1, pp. 41– 49, 2017. [Online]. Available: https://doi.org/10.1016/j.inpa.2016.10.005

[4] Q. Zhou, Z. Zhang, X. Liu, and Y. Wang, "Intelligent Agriculture Platform for Plant Disease

Detection and Treatment Recommendation Based on IoT and Machine Learning,"
IEEE Access, vol. 8, pp. 153495–153505, 2020. [Online].
Available:
https://doi.org/10.1109/ACCESS.2020.3015477

[5] S. Sladojevic, M. Arsenovic, A. Anderla, D. Culibrk, and D. Stefanovic, "Deep Neural Networks Based Recognition of Plant Diseases by
Leaf Image Classification," *Computational Intelligence and Neuroscience*, vol. 2016, Article ID 3289801, 11 pages. [Online]. Available: https://doi.org/10.1155/2016/3289801

[6] J. Too, L. Yujian, S. Njuki, and L. Yingchun, "Automated Plant Disease Detection Using Deep Learning: A Review of Models,Dataset AndChallenges,"**IEEE Access**, vol. 8, pp. 200586–200606, 2020.[Online].
Available:
https://doi.org/10.1109/ACCESS.2020.3039624