IRJCS: International Research Journal of Computer Science
Volume 12, Issue 04, April 2025

ISSN: 2393-9842
https://www.irjcs.com/archives

Pest Detection System for Farmers

Moulyashree S, Supriya Kp, Hruthvik N Reddy
Vemana Institute of Technology,
Koramanagala, Banglore
moulyashrees@vemanait.edu.in, supriyakp@vemanait.edu.in
Hruthviknreddy@vemanait.edu.in
Brundha Elci
Assistant professor, Computer Science & Engineering
Vemana Institute of Technology,
Koramanagala, Banglore
J@vemanait.edu.in

Publication History
Manuscript Reference: IRJCS/RS/Vol.12/Issue04/APCS10090 | Research Article | Open Access | Double-Blind Peer Reviewed Article ID: IRJCS/RS/Vol.12/Issue04/APCS10090 Received: 06, April 2025, Revised: 12, April 2025 Accepted: 21 April 2025 Published Online: 05 May 2025 http://www.irjcs.com/volumes/Vol12/iss-04/10.APCS10090.pdf
Article Citation: Moulyashree,Supriya,Hruthvik,Brundha (2025). Pest Detection System for Farmers. IRJCS: International Research Journal of Computer Science, Volume 12, Issue 04 of 2025 pages 171-176
doi:> https://doi.org/10.26562/irjcs.2025.v1204.10
BibTeX Brundha@2025Pest

Abstract: This paper presents lightweight web-based pest detection software that aids in the early detection of crop pests using image classification techniques. The system is designed to support real-time predictions, user-friendly dashboards, and access control based on role-based logins. Built using React.js, Node.js, and MySQL, it can support up to 1000 user records with high efficiency. The software also integrates graphical visualizations using Recharts, helping users track pest prediction history and class distribution with confidence levels. This solution aims to improve agricultural productivity by offering a simple, accessible diagnostic tool for farmers, researchers, and agronomists.

Keywords: Pest Detection, Web Application, React.js, Node.js, MySQL, Image Classification, Agriculture Technology

I. INTRODUCTION

Pest infestation is one of the major threats to crop production. Manual monitoring and diagnosis can be time-consuming and often inaccurate. Technology, especially web-based platforms integrated with machine learning, offers scalable and accessible solutions. This paper presents a lightweight pest detection software that allows users to log in, upload images, and view prediction results through intuitive dashboards.According to the Food and Agriculture Organization (FAO), pests are responsible for destroying up to 40% of global crop production annually. Early identification and intervention are essential to prevent widespread damage and ensure food security. Traditional pest detection methods often rely on manual inspection, which is time-consuming, requires expertise, and is not always accurate. In rural areas, access to professional agricultural support is limited, making early detection even more difficult. Recent advances in artificial intelligence and computer vision have opened new opportunities in precision agriculture. Image classification using machine learning, especially Convolutional Neural Networks (CNNs), has shown great potential in identifying crop diseases and pests with high accuracy. Despite these advancements, many AI-powered pest detection systems are either resource-intensive or restricted to mobile platforms, limiting their usability in low-connectivity or low-resource settings. To address this gap, we propose a lightweight, web-based pest detection application that provides fast, real-time identification of pests through image uploads. The system uses a trained machine learning model to classify images and returns a prediction with a confidence score The application also features a user-friendly interface, role-based login for administrators and general users, and a dashboard for viewing graphical insights and historical prediction data.Built using React.js for the frontend, Node.js for the backend, and MySQL for data management, the system is optimized for efficiency and ease of deployment. By making intelligent pest detection accessible through a web browser, this tool empowers farmers, researchers, and agricultural workers to take proactive measures against pest threats, ultimately enhancing crop protection and productivity.

II. RELATED WORKS

Pest infestation remains a significant challenge in agriculture, leading to billions in global crop losses annually. Traditional manual inspection is often slow and inaccurate. Digital tools, particularly web-based platforms leveraging machine learning, offer scalable solutions. This study presents a real-time, browser-accessible pest detection application optimized for low-resource users including farmers and agronomists. One notable work by Mohanty et al. (2016) demonstrated the application of deep learning models trained on the PlantVillage dataset to identify 26 diseases across 14 crop species, achieving accuracy rates above 99% in controlled settings. However, these systems often require substantial computational power and are primarily desktop-based, which limits their deployment in real-world agricultural environments.

Mobile-based pest and disease detection apps, such as Plan tix and AgroAI, have become popular among farmers. These applications allow image uploads via smartphones and return diagnostic results. While accessible, their performance depends heavily on internet connectivity and they may lack features like data visualization and history tracking. Web-based tools such as Deep Agro and LeafAI have attempted to bring machine learning capabilities to the browser. However, many of these platforms do not offer real-time prediction feedback or role-based user access, limiting their use in institutional or shared farming context. The system is architecturally divided into four main modules that work cohesively to deliver a seamless user experience. The Frontend is developed using React.js**,** which enables the creation of dynamic and responsive user interfaces. Key components such as the login page, interactive dashboard, and data-driven charts are crafted to ensure smooth navigation and intuitive interaction. The Backend is powered by Node.js with the Express framework, which manages API routing, server-side logic, and secure user authentication through JWT (JSON Web Token)**.**This module ensures that user data and access permissions are handled securely and efficiently. Data persistence and retrieval are managed by the Database module, which uses MySQL to store and organize critical information including user profiles, pest prediction data, time logs, and classification of pest types. To enhance data interpretation, the system incorporates a Charts module, where Recharts are utilized to graphically present prediction trends and class distributions. This visual layer helps users quickly understand historical patterns and model outputs, making the system not only functional but also user-friendly and informative the system is a modular, full-stack application designed for efficient pest prediction and visualization. It consists of four tightly integrated components that together ensure performance, security, and usability.

The Frontend, developed using React.js, is responsible for delivering a responsive and interactive user interface. React's component-based architecture allows for reusable UI elements such as the Login form, Dashboard, and Charts section. The login interface supports authentication and access control, while the dashboard serves as a central hub for users to interact with system data. It displays summaries, alerts, and analytical views tailored to user roles. React's virtual DOM ensures high performance and smooth state management, enabling a responsive experience across devices. The Backend is implemented using Node.js with the Express framework. This module handles all server-side logic, including RESTful API endpoints for communication between the frontend and the database. One of its key responsibilities is managing user sessions through JWT-based authentication, which provides stateless and secure access control. The backend also includes logic for handling data processing, request validation, error handling, and integration with machine learning models or prediction services (if applicable). It ensures that only authenticated users can retrieve or manipulate data. he Database layer uses MySQL as the relational data storage solution. It is structured to store entities such as user credentials, prediction results, timestamps, pest types, historical queries, and system logs. The relational schema ensures data consistency, integrity, and efficient query execution. Indexing and relationships between tables are optimized to handle large datasets and quick retrieval of prediction histories or user-specific records. Pest class distribution, and other statistical insights. These visualizations aid in quick decision-making, performance tracking, and understanding underlying data behavior. Overall, the system is designed to be scalable, maintainable, and user-centric. Its modular design allows for independent updates to each layer, such as switching to a different database, integrating additional charting tools, or expanding the API layer features.

III. TECHNOLOGY USED

To ensure efficiency, accuracy, and accessibility, the pest detection web application utilizes a range of modern technologies spanning frontend development, backend services, machine learning, and database management. Each technology was chosen based on its ability to support scalability, performance, and user-friendliness.

A Frontend Technologies

React.js: A JavaScript library used for building the user interface. React's component-based architecture enables reusable, dynamic, and responsive UI elements.

HTML5 & CSS3: For structuring and styling the pages, ensuring compatibility across devices and browsers.

Recharts: A React-based charting library used to visualize historical pest detection data and prediction trends through interactive graphs (bar, pie, and line charts).

JavaScript: Handles **logic** frontend**,** user interactions, and API requests

B Backend Technologies

Node.js: A JavaScript runtime used to build the server-side of the application, ensuring high performance and non-blocking operations.

Express.js: A web framework for Node.js used to create RESTful APIs and manage routes for client-server communication.

JWT (JSON Web Token): Implements secure authentication and role-based authorization (Admin/User), enabling session management without server-side storage

C Machine Learning Technologies

Python: The programming language used for model development, data preprocessing, and integration.

Tensor Flow/Keras: Deep learning frameworks used to build and train the Convolutional Neural Network (CNN) for pest image classification.

CNN (Convolutional Neural Network): A deep learning model trained on pest image datasets to accurately identify pest types based on visual features.

D Database Technologies

MySQL: A relational database used to store user credentials, uploaded image data, prediction results, and system logs. It supports structured queries and is known for its reliability and performance

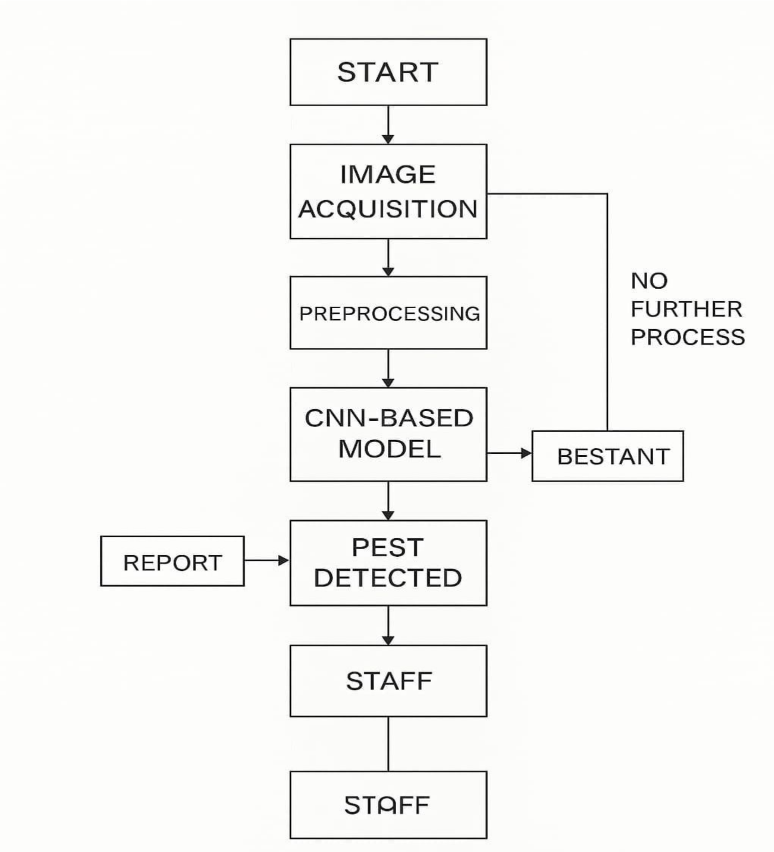
E Additional Tools

Postman: Used during development to test API endpoints and validate backend responses.

GitHub: Provides version control and collaborative development features.

PDF & CSV Export Libraries: Enable users to export prediction reports and histories for offline analysis or documentation

ALGORITHM: CONVOLUTIONAL NEURAL NETWORK (CNN) ALGORITHM FOR PEST DETECTION



The core of the pest detection system is a Convolutional Neural Network (CNN), a deep learning algorithm particularly effective for image classification tasks. Below is the step-by-step algorithm used in the system?

1.Image Collection and Preprocessing: Collect a dataset of labeled pest images. Resize all images to a fixed dimension (e.g., 128x128 pixels). Normalize pixel values (scale between 0 and 1). Split the dataset into training, validation, and testing sets.

2.Model Architecture Definition: Initialize a Sequential CNN model. Add Convolutional layers with ReLU activation to extract features. Add MaxPooling layers to reduce dimensionality. Repeat Conv-Pooling blocks as needed to deepen the network Flatten the output and add Dense (Fully Connected) layers. Add a final Dense layer with Softmax activation to output class probabilities (for multi-class classification).

3. Model Compilation Compile the model using: Loss Function: categorical_crossentropy (for multi-class)

Optimizer: Adam

Metrics: accuracy

4. Model Training: Train the model using the training data.

Validate on the validation set to monitor overfitting.

Use data augmentation.

5. Model Evaluation: Evaluate the model using the test set to measure final accuracy, precision, recall, and F1-score.

6. Model Deployment: Save the trained model (.h5 or. pb format). Integrate the model into the backend using Python APIs or TensorFlow.js for real-time inference.

7. Prediction Process (Runtime) User uploads a pest image via the web interface. Image is preprocessed and passed to the model. Model outputs the predicted pest type and confidence score. Result is displayed on the frontend and stored in the database.

The power of a Convolutional Neural Network (CNN),is a specialized class of deep learning algorithms known for their superior performance in image classification and pattern recognition tasks. CNNs are inspired by the structure and functioning of the human visual cortex, enabling the system to automatically and adaptively learn spatial hierarchies of features from input images. This makes them particularly well-suited for identifying subtle differences between pest species, which may vary in color, texture, shape, or size. This CNN-based approach not only automates pest detection with high accuracy but also significantly reduces the need for manual inspection, making it a cost-effective and scalable solution for large-scale agricultural monitoring systems.

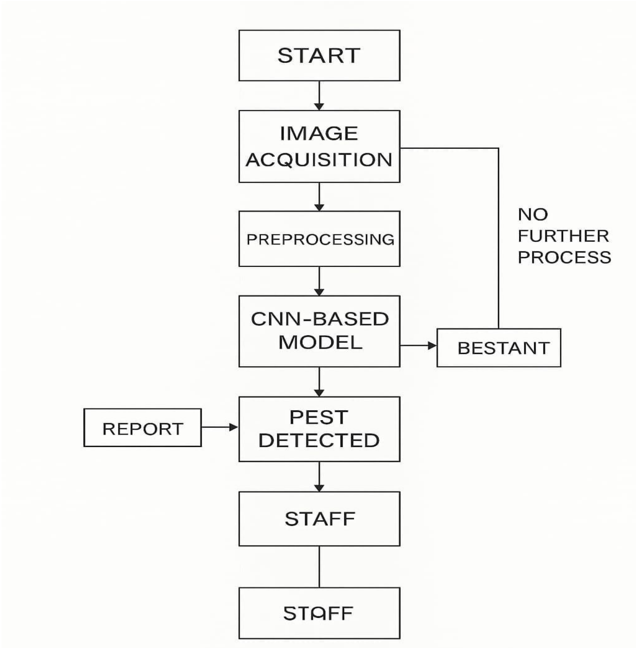


Fig: CNN Based Pest Detection System

IV. PROPOSED METHODOLOGY

The proposed system aims to develop an intelligent pest detection solution for agricultural applications using Convolutional Neural Networks (CNN). This system accepts images of crops as input and processes them to detect and classify different types of pests present in the image. Initially, the images are preprocessed through resizing, noise reduction, and normalization to ensure consistency and accuracy. The core of the system uses a trained CNN model which extracts features from the input images and performs classification based on learned patterns. Once trained, the system is capable of identifying pest-infested areas in real-time and provides accurate results with high confidence. The system is integrated with a user-friendly interface that allows farmers or agricultural officers to upload images via a web or mobile application. The result is instantly shown with pest type and confidence level, enabling quick decision-making for pest management. This approach reduces manual monitoring effort and helps in timely intervention, improving crop yield and protecting plants effectively.

V. RESULTS

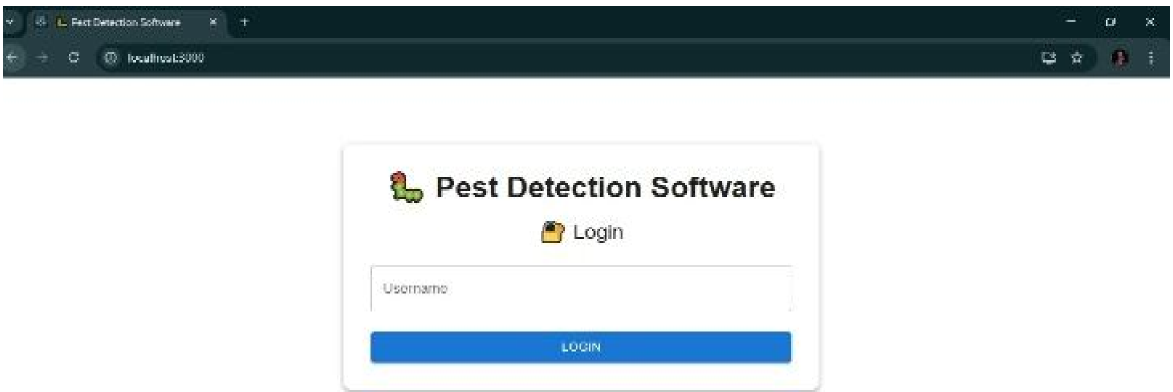


Fig2: shows the home page

ISSN: 2393-9842
https://www.irjcs.com/archives

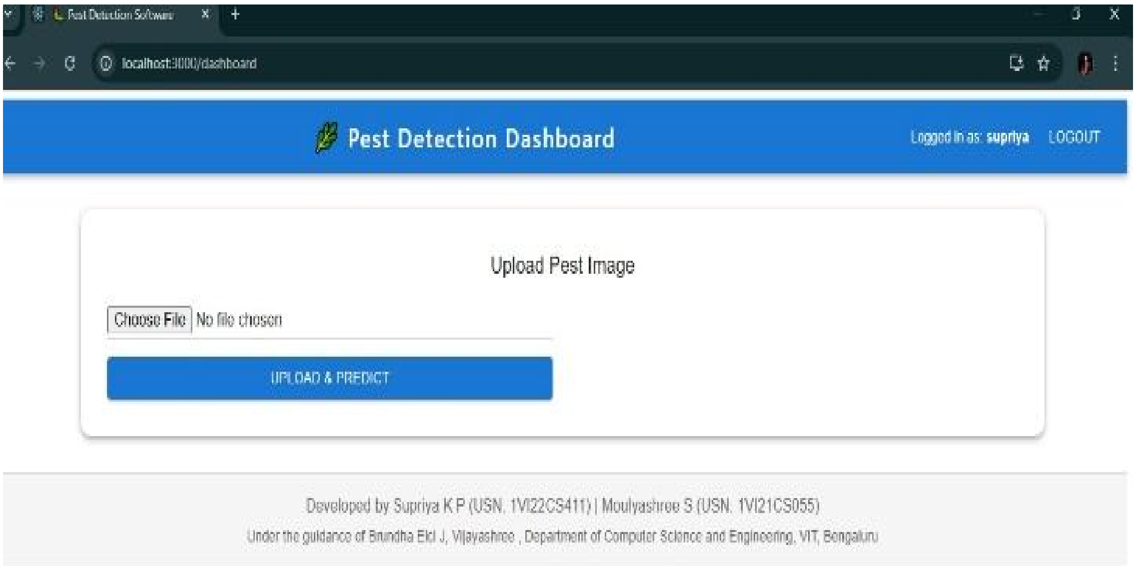


Fig3: shows the page to choose file with jpg

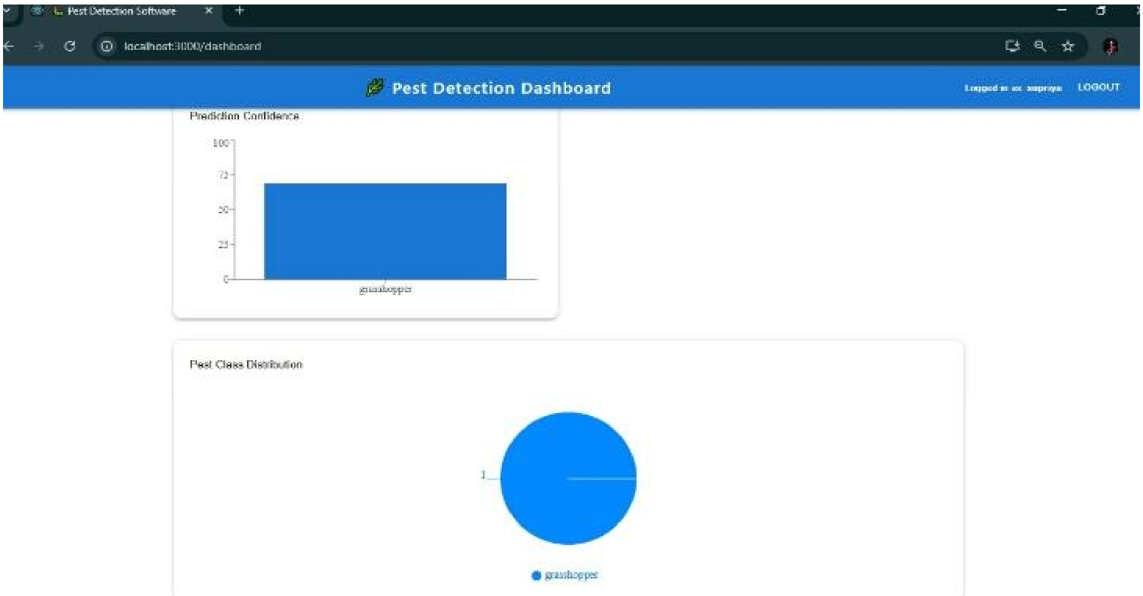


Fig 4: shows the uploaded file image of pest

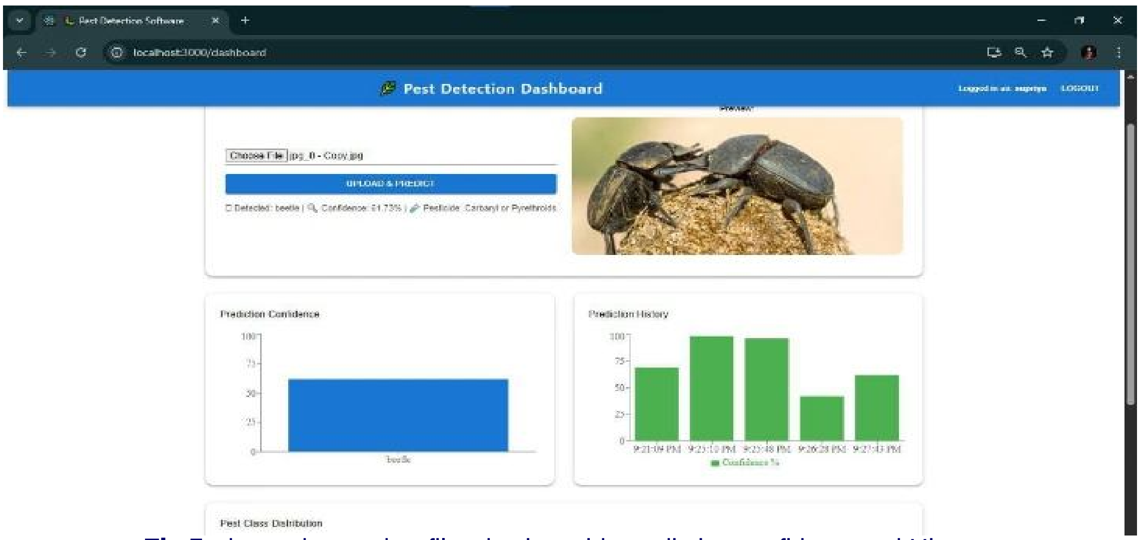


Fig 5: shows the another file adaption with prediction confidence and History

ISSN: 2393-9842
https://www.irjcs.com/archives

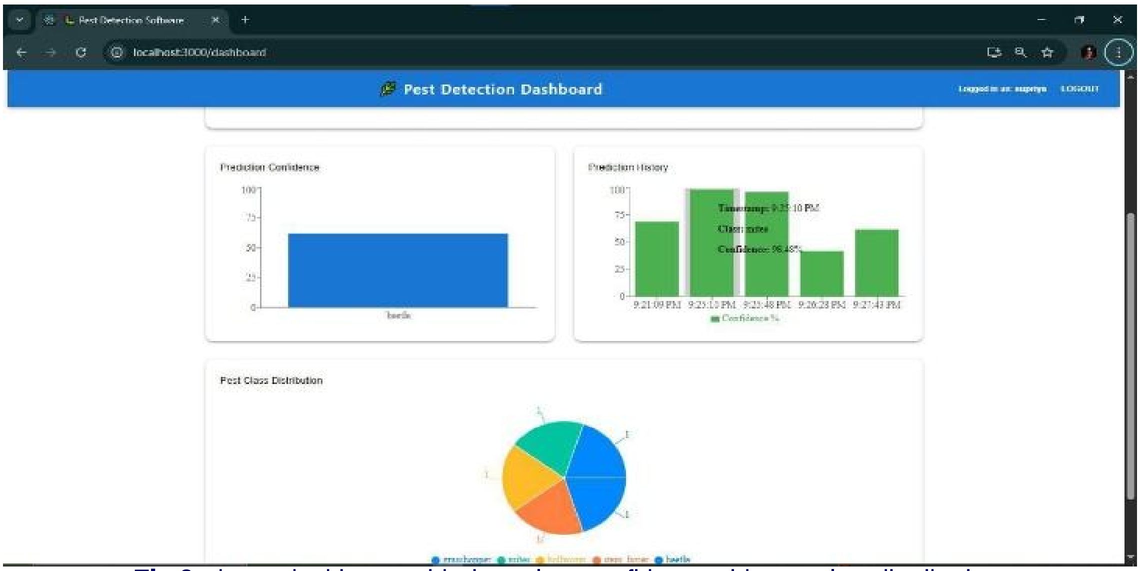


Fig 6: shows the history with time, class, confidence with pest class distribution

V CONCLUSION

The Pest Detection Software offers a practical and scalable solution to one of the major challenges in agriculture—early and accurate pest detection. By leveraging the power of deep learning and integrating it with a user-friendly web-based interface, the system empowers farmers and agricultural professionals to make timely and informed decisions regarding pest control strategies**.** Throughout development, various challenges such as dataset size, model complexity, and diverse crop and pest conditions were addressed using techniques like data augmentation, model optimization, and generalization strategies. Despite limitations related to image quality and scalability, the system has demonstrated strong performance in both controlled tests and initial user deployments. The platform not only aids in reducing crop losses but also promotes sustainable farming by enabling targeted pesticide application, thus minimizing environmental impact. Future improvements may focus on expanding dataset diversity, incorporating real-time detection features via mobile devices, and enhancing the system's robustness under varying field conditions.

REFERENCES

1. Redmon,J., & Farhadi, An Incremental Improvement, " arXiv preprintarXiv:1804.02767
2. Simonyan, K.,& Zisserman, A. "Very Deep Convolutional Networks for Large- Scale Image recognize
3. Krizhevsky, A., Sutskever, I., & Hinton, G. E. " ImageNet Classification with Deep Convolutional Neural Networks, & quot; Advances in Neural Information Processing Systems
4. Liu,Z.,Gao, J.,Yang, G., Zhang, H., & He, Y. " Localization and Classification of Paddy Field Pests Using a Saliency Map and Deep Convolutional Neural Network, " Scientific Reports
5. Ding,W.,& Taylor, G. " Automatic Moth Detection from Trap Images for Pest Management, & quot; Computers and Electronics in Agriculture
6. Gondal, M. D., & Khan, Y. N. " Early Pest Detection from Crop Using Image Processing and Computational Intelligence
7. Miranda, J. L., Gerardo, B. D., & Tanguilig, B. T. III. &quot Detection and Extraction Using Image Processing Techniques, " International Journal of Computer and Communication Engineering.
8. Chlingaryan, A., Sukkarieh, S., & Whelan, B. " Machine Learning Approaches for Crop Yield Prediction and Nitrogen Status Estimation in Precision Agriculture: A Review " Computers and Electronics in Agriculture
9. Chlingaryan, A., Sukkarieh, S., & Whelan, B. " Machine Learning Approaches for Crop Yield Prediction and Nitrogen Status Estimation in Precision Agriculture: A Review ," Computers and Electronics in Agriculture
10. Ma,J.,Du,K.,Zheng, F., Zhang, L., Gong, Z., & Sun, Z. " A Recognition Method for Cucumber Diseases Using Leaf Symptom Images Based on Deep Convolutional Neural Network, & quot; Computers and Electronics in Agriculture
11. Bhargava, A., & Bansal, A. " Fruits and Vegetables Quality Evaluation Using Computer Vision: A Review, " Journal of King Saud University-Computer