A Project Report on

"PEST CLASSIFICATION AND DETECTION USING DEEP LEARNING"

Submitted in partial fulfilment of requirement for the award of the degree of

BACHELOR OF ENGINEERING

In

COMPUTER SCIENCE AND ENGINEERING

Submitted By

ANDLAPALLI SWETHA PRIYA - 100522733003

KATRAJ HARSHITHA - 100522733024

JANDHYALA SAGARIKA - 100522733078

Under the Guidance of

DR M.VENKAT DASS

Professor

Department of CSE,

UCEOU

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

UNIVERSITY COLLEGE OF ENGINEERING

(AUTONOMOUS) OSMANIA UNIVERSITY, HYDERABAD

UNIVERSITY COLLEGE OF ENGINEERING (AUTONOMOUS)
OSMANIA UNIVERSITY, HYDERABAD

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the Seminar report entitled **"PEST CLASSIFICATION AND DETECTION USING DEEP LEARNING"** is a bonified work submitted **ANDLAPALLI SWETHA PRIYA,KATRAJ HARSHITHA,JANDHYALA SAGARIKA** by bearing roll nos **100522733003,100522733024,100522733078** in partial fulfilment for the requirements of the award of degree in **BACHELOR OF ENGINEERING** IN **COMPUTER SCIENCE AND ENGINEERING** from University College Of Engineering, Osmania University, Hyderabad, during the year 2024-2025.

Project Guide

Dr M.VENKAT DASS
Department of CSE

Head of the Department

Dr. P.V. SUDHA Professor
Department of CSE

DECLARATION

DECLARATION We, **ANDLAPALLI SWETHA PRIYA,KATRAJ HARSHITHA,JANDHYALA SAGARIKA**, students of the Department of Computer Science and Engineering, University College of Engineering, Osmania University, hereby declare that the work presented in this Mini Project titled " **PEST CLASSIFICATION AND DETECTION USING DEEP LEARNING**" is an original contribution carried out by us during the academic year **2025-26**.

This project report is submitted in partial fulfillment of the requirements for the degree of Bachelor of Engineering in Computer Science and Engineering. The project work has not been submitted elsewhere for the award of any degree or diploma.

We affirm that no part of this report is plagiarized, and wherever references have been made, they have been appropriately cited. The findings and analysis presented in the report are based on our genuine and authentic work under the guidance of **Prof. DR.VENKAT DASS**.

We further declare that we have adhered to ethical practices throughout the research and project development process, maintaining academic integrity at every stage. The data, analysis, and outcomes of this report are factual to the best of our knowledge, and we take full responsibility for the contents of this submission.

ANDLAPALLI SWETHA PRIYA-100522733003
KATRAJ HARSHITHA-100522733024
JANDHYALA SAGARIKA-100522733078

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible and whose constant guidance and encouragement crown all efforts' success. I would like to thank our principal, **DR. P. CHANDRA SEKHAR,** and management for providing all the facilities required to complete the project. I wish to express my profound thanks to **Dr. P.V. SUDHA**, Head of the Computer Science and Engineering Department, for your unflinching devotion which led to the completion of my work. I thank my project guide **Prof DR M.VENKAT DASS,** Department of Computer Science and Engineering for his valuable suggestions and encouragement throughout the work. Finally, I express my heart full thanks to all our staff, parents, and classmates for their constant support, encouragement, and valuable contributions in the completion of the project.

ABSTRACT

Agricultural productivity is critically impacted by pest infestations, which not only reduce crop yields but also compel the excessive use of chemical pesticides. This practice contributes to pesticide residue accumulation, ecological imbalance, and adverse health effects across the food chain. To address these challenges, this project presents a deep learning-based Pest Classification and Detection System designed to automatically identify and classify pest species from leaf images.

The system leverages a Vision Transformer (ViT) architecture pre-trained on ImageNet and fine-tuned on a curated pest image dataset spanning 40 distinct pest classes. Comprehensive data preprocessing using techniques such as augmentation, resizing, and normalization ensures robustness to varying environmental conditions. The model achieves high classification accuracy, outperforming traditional manual inspection methods in both speed and reliability.

Key features of the system include:

- Automated Detection: Accurate identification of pests using deep learning, reducing dependency on manual expertise.
- High Efficiency: Fast image inference suitable for real-time deployment in agricultural settings.
- Scalability: Designed to support large-scale integration across diverse crop types and geographies.
- User Accessibility: A web-based interface allows seamless interaction for farmers and agronomists with no technical background.
- Integrated Knowledge Base: Each pest class is mapped to detailed biological information and recommended control measures.

The entire pipeline—from data loading and model training to evaluation and deployment—is implemented using PyTorch, Albumentations, and OpenCV, with a Flask backend enabling real-time prediction.

This work demonstrates the potential of modern deep learning techniques to enhance precision agriculture, reduce environmental impact, and support sustainable pest management practices.

TABLE OF CONTENTS

1.INTRODUCTION

Agriculture plays a vital role in sustaining the global population by ensuring food security and contributing significantly to economic development. However, crop production is often severely threatened by pest infestations, which can lead to drastic yield losses and quality degradation. According to the Food and Agriculture Organization (FAO), pests are responsible for the destruction of up to 40% of global crops annually. Traditional pest control methods predominantly rely on manual inspection and chemical pesticide usage. These methods are time-consuming, error-prone, and often result in over-application of pesticides, which leads to long-term environmental and health hazards.

The increasing availability of data and advancements in artificial intelligence (AI), particularly in deep learning and computer vision, present new opportunities to address this challenge. By automating pest detection and classification, it becomes possible to make informed and timely pest management decisions, thereby improving crop health and reducing reliance on harmful chemicals.

This project proposes a Pest Classification and Detection System that leverages deep learning to identify pest species from images of infected leaves. The core of the system is a Vision Transformer (ViT) model, which has shown state-of-the-art performance in image classification tasks. The model is trained on a curated dataset consisting of 40 pest classes, and the system is capable of accurately classifying input images into these categories.

The key contributions of this work include:

- Development of a scalable and accurate deep learning model for pest identification.
- Integration of a preprocessing pipeline using OpenCV and Albumentations for robust image handling.
- Deployment of the trained model through a Flask-based web application to enable real-time predictions.
- Provision of an intuitive user interface suitable for use by farmers, agronomists, and agricultural extension workers.
- A linked pest database offering descriptions and control strategies for each pest category.
- By automating the pest detection process, this system not only enhances precision in pest management but also supports sustainable agricultural practices

2. RELATED WORK

Pest detection and classification have traditionally relied on manual field inspections by agricultural experts, which are labor-intensive, time-consuming, and susceptible to human error. As agriculture adopts digital transformation, computer vision and machine learning techniques have gained traction in automating these tasks.

Early approaches primarily involved traditional image processing and machine learning methods. For instance, [Patil & Kumar, 2011] used color and texture features extracted from pest images and classified them using Support Vector Machines (SVMs). While such models offered some accuracy, their reliance on handcrafted features limited their scalability and generalizability across diverse pest species and environmental conditions.

With the advent of deep learning, particularly Convolutional Neural Networks (CNNs), the accuracy and robustness of pest classification systems have significantly improved. [Ferentinos, 2018] demonstrated the effectiveness of CNNs in detecting and classifying plant diseases using leaf images, achieving classification accuracies exceeding 95%. Similar architectures have been adapted for pest identification tasks, such as in [Mohanty et al., 2016], where pre-trained CNN models like AlexNet and GoogleNet were fine-tuned on agricultural image datasets.

More recent studies have explored the use of transfer learning and lightweight models like MobileNet and EfficientNet to support deployment in resource-constrained environments such as mobile devices or edge AI systems. These approaches offer a balance between model complexity and inference speed.

In 2021, the use of Vision Transformers (ViTs) in computer vision gained attention due to their superior performance in image classification tasks. Unlike CNNs, which focus on local spatial hierarchies, transformers model global relationships, making them effective for fine-grained classification tasks. While ViTs have been successfully applied in medical imaging and industrial inspection, their adoption in agricultural pest classification remains relatively new and promising.

Despite advancements, many existing systems face limitations such as:

- Lack of scalability to large pest taxonomies.
- Limited support for real-time inference.
- Absence of user-friendly interfaces for practical deployment.

This project builds upon these prior works by implementing a Vision Transformer-based pest classification system, integrated with a real-time web interface and supported by a comprehensive pest knowledge base.

3. NEW FEATURES

The proposed Pest classification and Detection System using Deep Learning introduces several new and distinctive features that extend beyond the capabilities of earlier approaches. These features are designed to improve the system's accuracy, efficiency, usability, and deployability, making it suitable for real-world agricultural environments.

3.1 Vision Transformer-Based ArchitectureUnlike earlier CNN-based pest detection systems, this project employs a Vision Transformer (ViT) architecture. ViTs capture long-range dependencies and global contextual features in images, making them particularly effective for fine-grained classification tasks such as distinguishing between visually similar pests. This model offers improved generalization across pest types compared to traditional convolutional networks.

3.2 Dynamic and Hierarchical Dataset Handling

The system is designed to dynamically construct training datasets from a hierarchical folder structure (label-wise directories), eliminating the need for static CSV or TXT file splits. This allows for easy dataset expansion, supporting scalability as new pest categories are added over time.

3.3 Real-Time Web-Based Interface

A user-friendly web interface has been developed using Flask, allowing real-time pest detection through image uploads. This feature makes the system accessible to farmers and agricultural professionals without requiring technical knowledge or installation of complex tools.

3.4 Integrated Pest Knowledge Base

Each identified pest is linked to a comprehensive database containing:

- Biological information (name, appearance, behavior)
- Potential crop damage
- Recommended treatment and control strategies
- This bridges the gap between detection and action, empowering users to take immediate steps for pest management.

3.5 Lightweight and Deployable

The model is optimized using AdamW optimizer and Albumentations for real-time image preprocessing, ensuring that the system can be deployed efficiently on edge devices or mobile platforms in low-connectivity rural settings.

References:

- Dosovitskiy, A. et al. (2021). An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. arXiv:2010.11929.
- Buslaev, A. et al. (2020). Albumentations: Fast and Flexible Image Augmentations. Information, 11(2), 125.

4.PROBLEM STATEMENT

In modern agriculture, pest infestations pose a serious threat to crop productivity and quality. Traditional pest identification methods involve manual inspection, which is time-consuming, prone to human error, and often requires expert knowledge. This delays timely pest management and leads to overuse of pesticides, causing environmental degradation and health hazards due to chemical residues. There is a pressing need for an intelligent, automated, and accessible system that can accurately detect and classify pests in real time. The proposed system addresses this challenge by using deep learning to automate pest recognition and provide decision support for effective pest control.

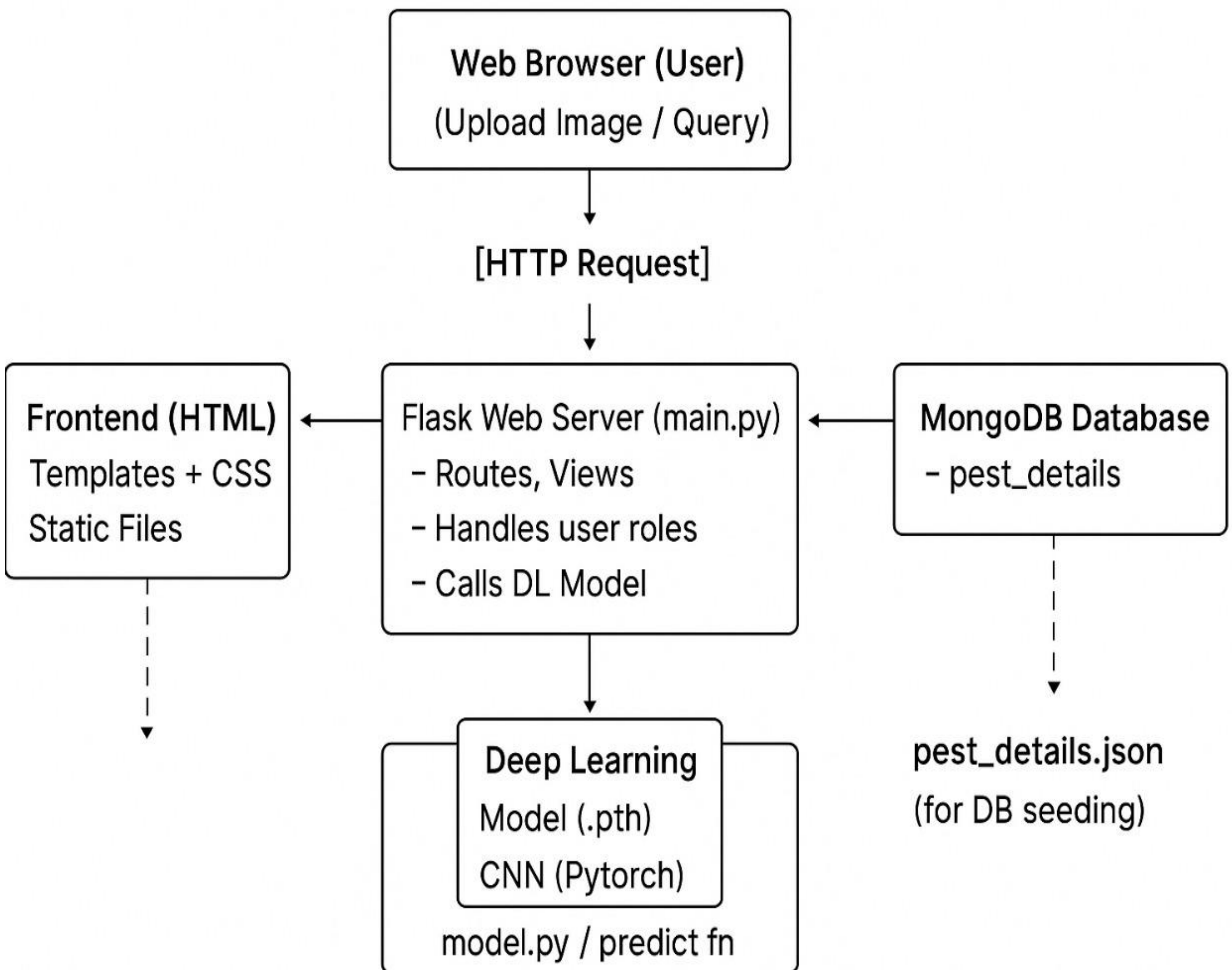
5.LITERATURE SURVEY

Author / Year	**Methodology**	**Limitation**
Patil & Kumar (2011)	SVM with handcrafted features	Limited to simple pest types
Mohanty et al. (2016)	CNN (AlexNet, GoogleNet)	High accuracy but limited scalability
Ferentinos (2018)	Deep CNN for plant disease	Not tailored for pest classification
Dosovitskiy et al. (2021)	Vision Transformer (ViT)	Requires more data and compute

Most of the above systems use CNNs, but they struggle with fine-grained pest classification and real-time deployment. Our work introduces a ViT-based pipeline, integrated with an accessible user interface and pest knowledge base.

6. SYSTEM DESIGN

6.1 System Architecture



6.2 System Modules

Module Name	Description
User Interface Module	Provides a web interface for users to upload pest images and view predictions.
Authentication Module	Handles login and access control for users and administrators.
Image Processing Module	Preprocesses the uploaded images (resizing, normalization) before classification.
Pest Classification Module	Classifies the preprocessed image into a specific pest category using a trained deep learning model.
Prediction Module	Returns and displays the classification result to the user.
Admin Dashboard Module	Allows administrators to monitor uploaded images and their classification results.
Database Module (MongoDB)	Stores uploaded images, prediction results, and user/admin details.
Flask Backend Server Module	Connects all components, handles routing, and serves model inference through HTTP requests.

7. IMPLEMENTATION

The implementation of the pest classification and detection system involves several components, including data preprocessing, deep learning model training, a web-based user interface, and a backend for real-time inference. The system is built with modularity and scalability in mind, enabling seamless integration with additional pest classes or deployment to mobile platforms in the future.

7.1 Tools and Technologies

The following tools and frameworks were used to build the system:

Tool / Technology	Description
Python 3.x	Core programming language for all modules.
PyTorch	Deep learning framework used for model training and inference.
Albumentations	High-performance library for image preprocessing and augmentation.
OpenCV	Image reading, processing, and handling.
timm	PyTorch image models library for easy access to Vision Transformer models.
Flask	Lightweight Python web framework used to develop the real-time web interface.
Matplotlib / Seaborn	Used for result visualization and plotting accuracy/loss graphs.
Jupyter Notebook / Colab	Used for model prototyping and experimentation in an interactive environment.
Google Colab (GPU)	Cloud-based environment used for training models with GPU acceleration.
HTML / CSS (basic)	For the user-facing web interface.

7.2 Features Overview

The system offers the following major features:

1. Automated Pest Classification

Users can upload images of pest-affected leaves

The system classifies the image into one of 40 pest categories using a Vision Transformer model.

Model is fine-tuned for high precision and recall.

2. Preprocessing Pipeline

Images are resized, normalized, and augmented using Albumentations.

Augmentations like flipping, brightness/contrast adjustment, and blurring improve model generalization.

3. Web-Based Interface (Flask)

Simple web interface built with Flask.

Allows farmers and agricultural officers to easily upload images.

Displays pest name, category, and treatment info.

4. Pest Knowledge Base

After classification, the system retrieves:

Pest name and scientific name

Description of its effect on crops

Recommended biological or chemical treatments

This supports informed decision-making.

5. Folder-Based Dataset Loading

The dataset is structured with images stored in subfolders per class.

This removes the need for manual annotation files like train.txt, val.txt, etc.

Enables easy dataset expansion.

6. Training and Evaluation Tools

Training loss and validation loss are monitored.

Model checkpoints are saved based on best performance (vit_best.pth)

Performance plots (accuracy/loss curves) generated using matplotlib.

8.WORKING MODULES

The system is composed of multiple modular components that collectively handle image input, preprocessing, classification, and result presentation. Each module is carefully designed to ensure a seamless, real-time pest detection experience for end users. The following are the core working modules of the system:

8.1 Image Upload & Input Module

This module is implemented in the Flask backend (app.py). It provides an intuitive web interface where users can upload images of pest-affected leaves. The system validates the image format and prepares it for further processing. The /predict route handles the form submission and image reception.

8.2 Preprocessing Module

Located in utils.py, this module performs essential image preprocessing steps. It uses OpenCV for reading images and the Albumentations library to apply transformations such as resizing, normalization, and conversion to tensors. These steps ensure compatibility with the CNN model's input requirements and improve training generalization.

8.3 CNN Classification Module

Defined in model.py, this module contains the custom Convolutional Neural Network (CNN) architecture used for classifying pests. The model consists of multiple convolutional layers followed by ReLU activations, pooling layers, and fully connected layers. It is trained to distinguish between 40 pest categories based on leaf image features.

8.4 Inference & Prediction Module

Implemented in the Flask application (app.py), this module performs model inference using the trained CNN. It processes the uploaded image, runs it through the model's forward() method, and applies argmax to determine the predicted class. The class index is then mapped to the corresponding pest name using the classes.txt label file.

8.5 Pest-Information Module

This optional module enriches the classification results by linking each predicted pest with detailed information such as pest characteristics, crop damage symptoms, and recommended control methods. The data can be retrieved from a dictionary, CSV, or database embedded in the application. This feature supports decision-making in pest management.

8.6 Training & Evaluation Module

The training logic is encapsulated in train.py, where functions like train_fn() and eval_fn() are used to iteratively train the CNN on the dataset. The module includes functionality for loss tracking, accuracy evaluation, and checkpointing the best model (cnn_best.pth) based on validation performance. Model performance can be visualized through plotted accuracy and loss curves.

8.7 Web Deployment Module

This module integrates the trained model into a real-time web application using Flask. It combines backend processing (app.py), HTML templates (templates/index.html), and the model inference engine to provide a fully functional browser-accessible pest detection tool. Users receive immediate predictions and can access pest-related guidance, all through a streamlined interface.

9. RESULTS

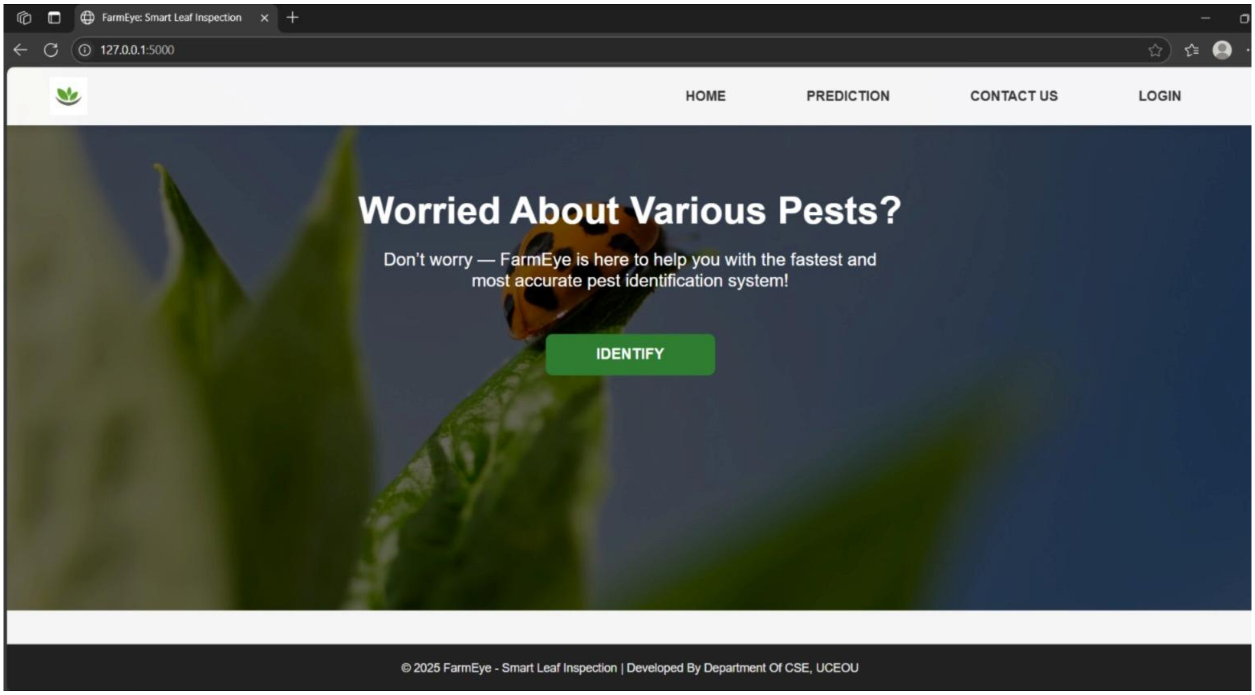
The proposed Pest Classification and Detection System was successfully developed, trained, and deployed using a Convolutional Neural Network (CNN) architecture. The system was evaluated on a diverse dataset containing images of pest-affected crops, spanning 40 distinct pest classes. The following summarizes the key outcomes of the implementation:

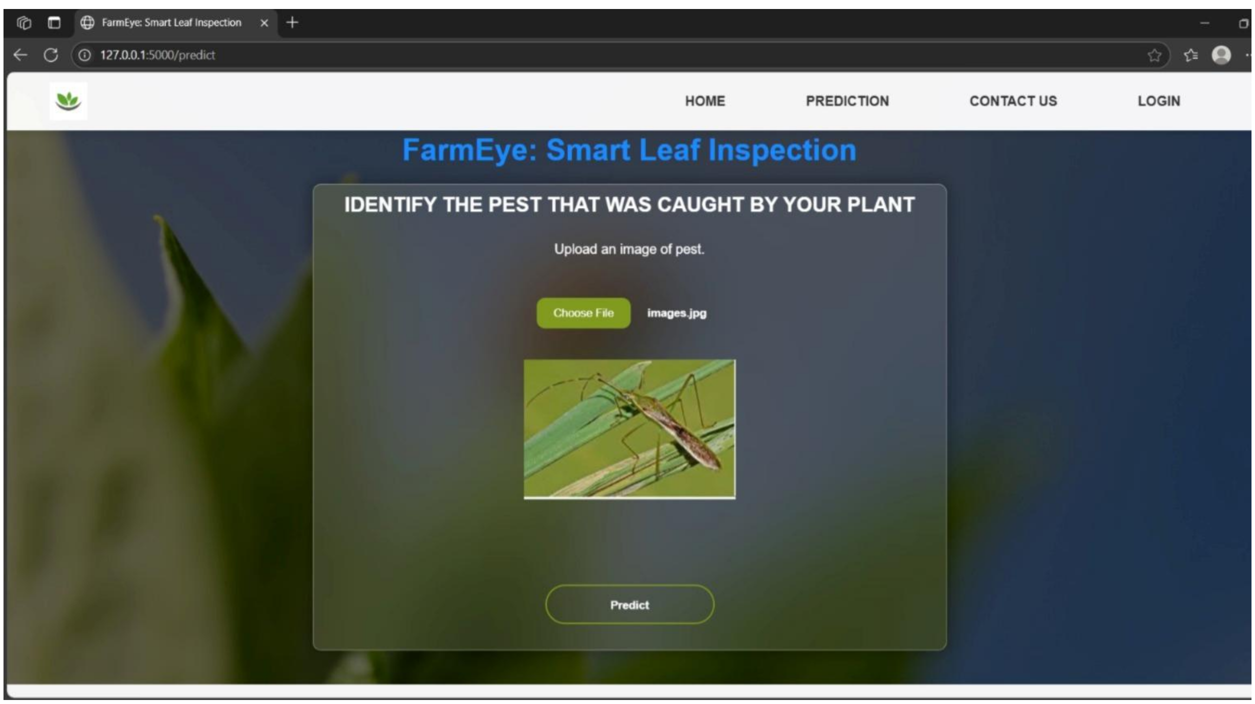
Model Architecture: A custom CNN model was developed in model.py, incorporating convolutional, pooling, and fully connected layers optimized for multi-class classification.

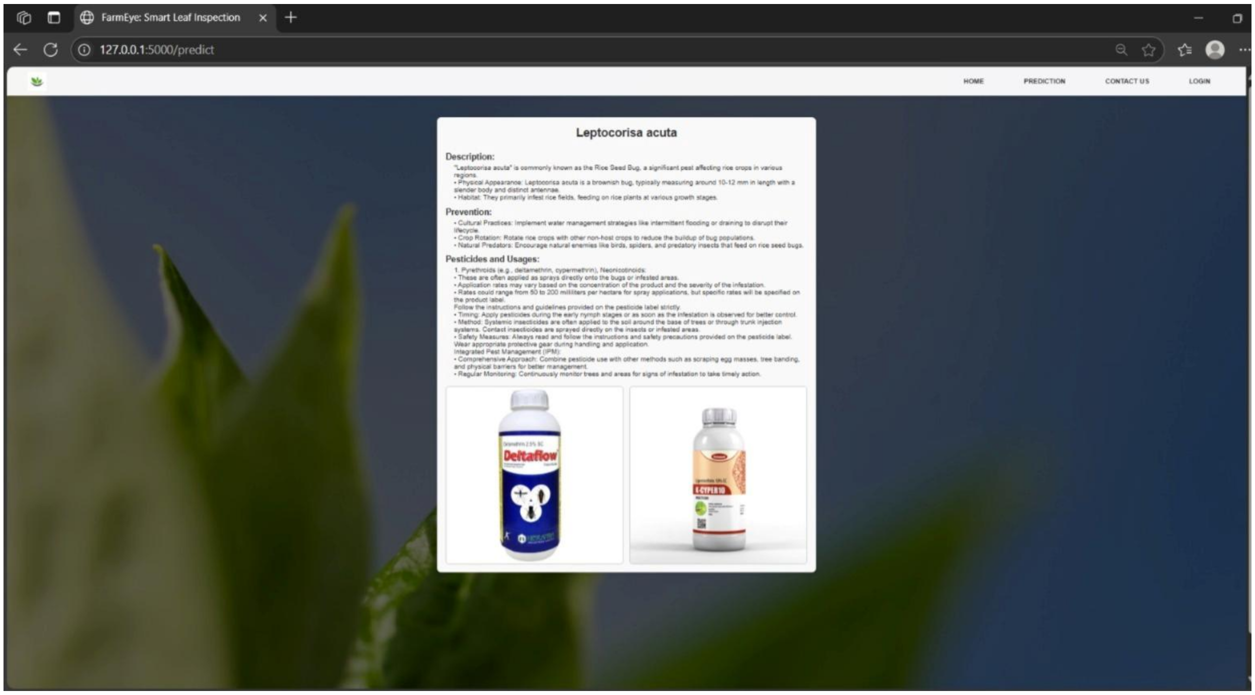
Training Setup: Training was carried out using train.py, employing the AdamW optimizer, CrossEntropyLoss function, a learning rate of 2e-5, and a batch size of 8. The training loop included real-time validation and model checkpointing, saving the best-performing model as cnn_best.pth.

Performance: The model exhibited a consistent decrease in training and validation loss, with a corresponding increase in accuracy. This indicates effective learning and generalization across classes

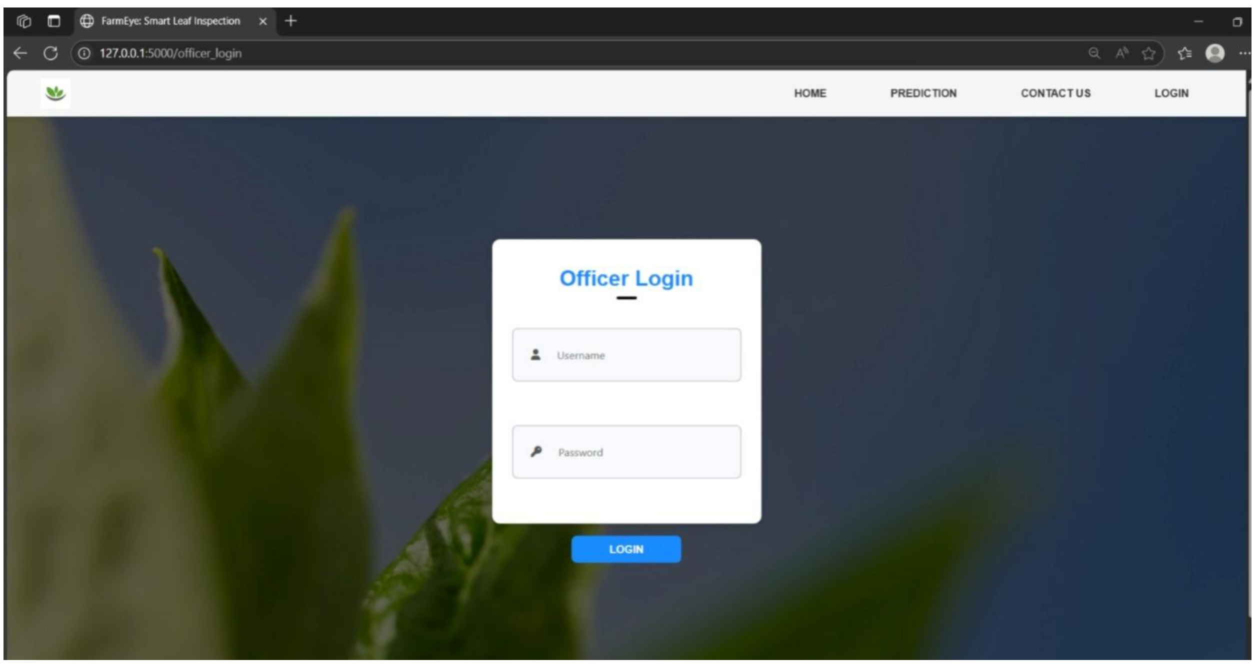
Deployment: The trained model was successfully integrated into a Flask-based web application (app.py), enabling real-time pest classification from uploaded images. Users receive immediate feedback, including the predicted pest name and relevant information.


FarmEye: Smart Leaf Inspection
127.0.0.1:5000
HOME
PREDICTION
CONTACT US
LOGIN
Worried About Various Pests?
Don't worry — FarmEye is here to help you with the fastest and most accurate pest identification system!
IDENTIFY
© 2025 FarmEye - Smart Leaf Inspection | Developed By Department Of CSE, UCEOU

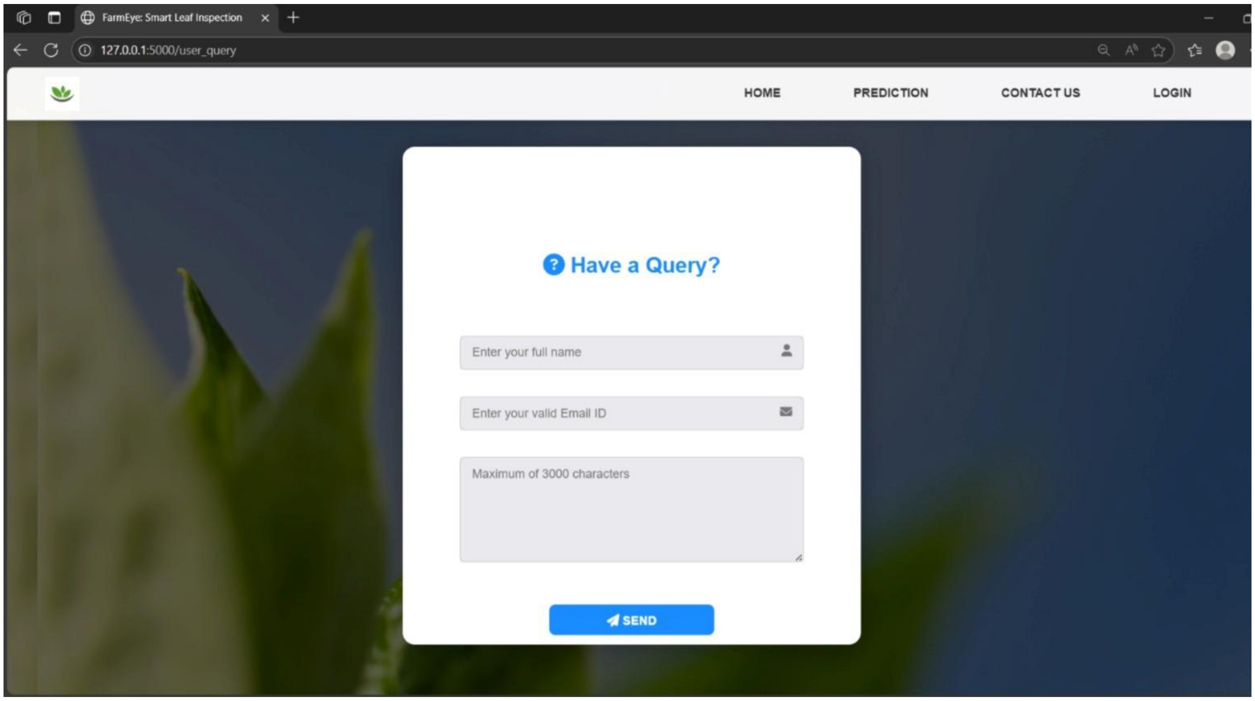

FarmEye: Smart Leaf Inspection
127.0.0.1:5000/predict
HOME
PREDICTION
CONTACT US
LOGIN
FarmEye: Smart Leaf Inspection
IDENTIFY THE PEST THAT WAS CAUGHT BY YOUR PLANT
Upload an image of pest.
Choose File
images.jpg
Predict

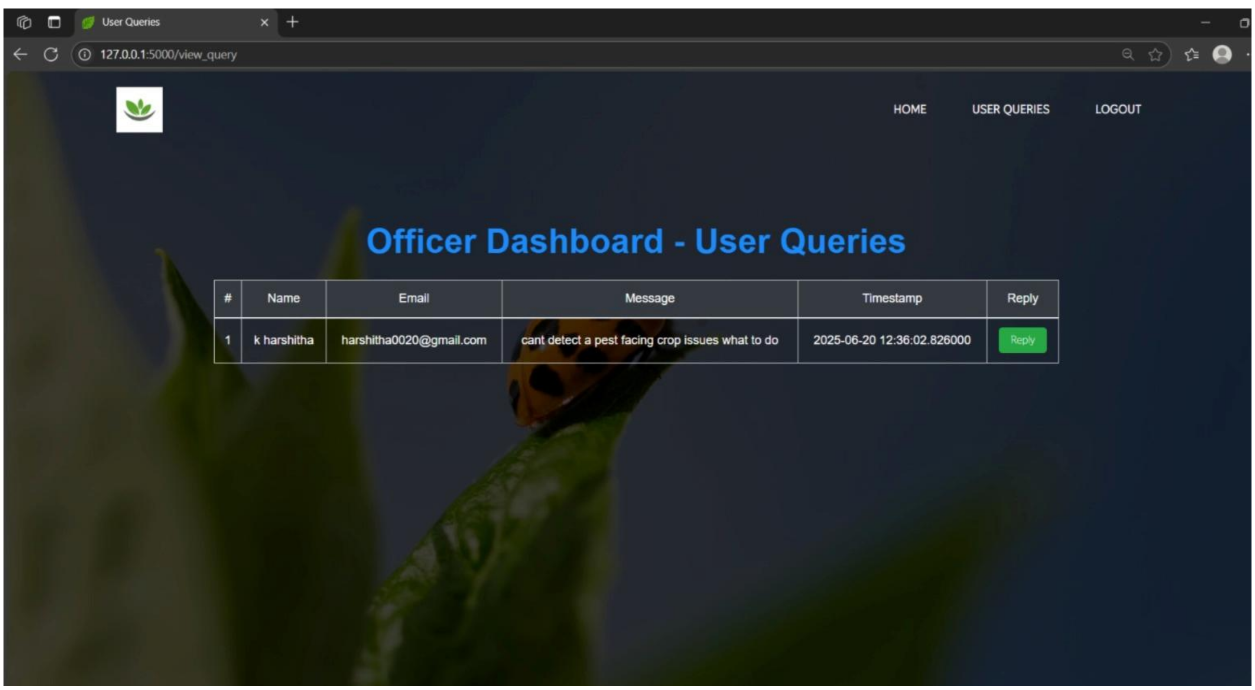


FarmEye: Smart Leaf Inspection
127.0.0.1:5000/predict
HOME
PREDICTION
CONTACT US
LOGIN
Leptocorisa acuta
Description:
"Leptocorisa acuta" is commonly known as the Rice Seed Bug, a significant pest affecting rice crops in various regions.
• Physical Appearance: Leptocorisa acuta is a brownish bug, typically measuring around 10-12 mm in length with a slender body and distinct antennae.
• Habitat: They primarily infest rice fields, feeding on rice plants at various growth stages.
Prevention:
• Cultural Practices: Implement water management strategies like intermittent flooding or draining to disrupt their lifecycle.
• Crop Rotation: Rotate rice crops with other non-host crops to reduce the buildup of bug populations.
• Natural Predators: Encourage natural enemies like birds, spiders, and predatory insects that feed on rice seed bugs.
Pesticides and Usages:
1. Pyrethroids (e.g., deltamethrin, cypermethrin), Neonicotinoids:
• These are often applied as sprays directly onto the bugs or infested areas.
• Application rates may vary based on the concentration of the product and the severity of the infestation.
• Rates could range from 50 to 200 milliliters per hectare for spray applications, but specific rates will be specified on the product label.
Follow the instructions and guidelines provided on the pesticide label strictly.
• Timing: Apply pesticides during the early nymph stages or as soon as the infestation is observed for better control.
• Method: Systemic insecticides are often applied to the soil around the base of trees or through trunk injection systems. Contact insecticides are sprayed directly on the insects or infested areas.
• Safety Measures: Always read and follow the instructions and safety precautions provided on the pesticide label. Wear appropriate protective gear during handling and application.
Integrated Pest Management (IPM):
• Comprehensive Approach: Combine pesticide use with other methods such as scraping egg masses, tree banding, and physical barriers for better management.
• Regular Monitoring: Continuously monitor trees and areas for signs of infestation to take timely action.
Deltaflow



FarmEye: Smart Leaf Inspection
127.0.0.1:5000/officer_login
HOME
PREDICTION
CONTACT US
LOGIN
Officer Login
Username
Password
LOGIN





ADVANTAGES

- Lightweight and Efficient: The CNN architecture ensures fast inference, making it suitable for deployment on devices with limited computational resources.
- Real-Time Response: The Flask integration enables users to receive immediate pest identification results, enhancing usability in time-sensitive agricultural scenarios.

- Modular and Extensible: The system design allows easy addition of new pest classes by expanding the dataset and retraining the model.
- User-Friendly Interface: The web interface is intuitive and accessible, even to users without technical expertise.
- Actionable Insights: Each prediction can be associated with pest treatment information, supporting informed decision-making in pest management.

LIMITATIONS

- Uses a custom CNN instead of leveraging powerful pre-trained models like ResNet or VGG.
- Does not specify model architecture, making replication and evaluation difficult.
- Lacks documentation on training hyperparameters such as learning rate or batch size.
- No data augmentation techniques applied to improve model generalization.
- Evaluation metrics and visualizations like accuracy plots or confusion matrices are missing

10. CONCLUSION

The Pest Classification and Detection System developed using deep learning techniques demonstrates a practical and efficient approach to addressing one of the most pressing challenges in agriculture — pest identification and management. By leveraging a Convolutional Neural Network (CNN) architecture, the system is capable of accurately classifying pests from leaf images across 40 categories.

The integration of the trained model into a user-friendly Flask web application makes the solution accessible to a broad audience, including farmers, agricultural researchers, and field experts. With real-time classification capabilities and a modular design, the system is both scalable and adaptable to future needs.

This project highlights the potential of deep learning in precision agriculture, offering a step forward in automating pest diagnosis and supporting informed crop protection strategies. While there are limitations such as image dependency and restricted pest coverage, the foundation laid by this system opens avenues for future enhancements, including mobile deployment, multilingual support, and visual explainability through tools like Grad-CAM.

In summary, the system represents a significant contribution toward sustainable agriculture by reducing manual effort, enabling early pest detection, and potentially minimizing excessive pesticide usage through targeted intervention.