International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 09 | Sept - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

Pest Detection in Crops Using Image Processing and Deep Learning

Sandhya M N [1], Basavanna M [2], Poornima B H [3]
Department of Studies in Computer Applications (MCA), Davangere University, Shivagangothri, Davanagere-577007
Department of Studies in Computer Applications (MCA), Davangere University, Shivagangothri, Davanagere-577007
Department of Studies in Computer Science, Davangere University, Shivagangothri, Davanagere-577007

ABSTRACT

One of the biggest problems in agriculture is pest infestation, which lowers productivity and causes losses. Conventional pest detection methods are not appropriate for large-scale farming because they are laborious, inaccurate, and expensive. This project uses deep learning and image processing techniques to develop an automated earlier pest type detection system. CNN and ResNet models are used to preprocessed pictures of maize and rice leaves. According to experimental data, the CNN model outperformed the ResNet model in terms of pest classification, with a higher accuracy of 98.7% and a smaller loss. The method encourages sustainable agriculture, lessens the overuse of pesticides, and facilitates early pest identification with preventative measures included.

Keywords: Pest Detection, Image Processing, Deep Learning, Pest Classification, CNN, Resnet

I. INTRODUCTION

Pests are one of the most serious threats to agriculture, it causes large-scale damages to crop and reducing both yield and quality. Farmers often struggle with early identification of pests because traditional methods such as manual field inspection, sticky traps, or consultation by expertise are time-consuming, labour-intensive, and sometimes inaccurate.in many cases farmers apply pesticides excessively, to prevent the losses, which not only increases costs but also harms the environment and food safety

By using advanced technologies, image processing and deep learning have emerged as good tools for automating pest detection. By capturing images of crop leaves and analyzing them through machine learning algorithms, it is easier to detect pest presence and classify pest types quickly and accurately. This paper focuses on detecting different pest types in maize and rice crops, which are staple foods for a large population. Images of healthy and pest-affected leaves are collected, preprocessed, and used to train CNN and ResNet models for classification. These models shows that CNN achieves better accuracy and lower loss, making it more suitable for practical agricultural applications.

Traditional methods like checking fields by hand or using traps takes more time, efforts, and are not always correct. To avoid leaf damage, farmers often spray too many pesticides, which can harm the soil, crop, and environment also. With new technologies like image processing and deep learning, pests can be identified from crop leaves images quickly and accurately. This helps farmers take early action and protect their crops in a safer and smarter way.

The rest of the paper is organized as follows: Section II presents the related work and reviews existing methods for pest detection in agriculture. Section III Methodology for the proposed research. Section IV describes the Technology Used, while Section V discusses the Results. Section VI presents conclusion of the paper. Finally, Section VII includes References.

Problem Statement

Pests in crops like maize and rice cause major yield losses and economic damage to farmers. Traditional detection methods are slow, labor-intensive, and often inaccurate for large-scale farming.Overuse of pesticides as a preventive measure raises costs and harms the environment.
An automated system using image processing and deep learning can improve accuracy and efficiency. This project compares CNN and ResNet models to identify the most effective solution for pest detection.

Objectives

1. To implement an automated solution for pest detection and classification in maize and rice leaves using deep learning-based image analysis.

2. To preprocess crop leaf images through resizing, normalization to improve dataset quality and model training.

3. To design and implement CNN and ResNet models, and compare their performance based on accuracy and loss.

4. To support early pest identification, reduce excessive pesticide usage, and promote sustainable agricultural practices.

Sample Images

In this Paper, various sample images of crop leaves were collected from Kaggle repositories to analyze pest infestation using image processing.

This Paper consists of **9 Pest Classification** includes:

The datasets includes both healthy and pest-affected leaves captured under natural field condition. The application focuses on detecting common agricultural pests such as Aphid, Armyworm, Beetle, Bollworm, Mites, Grasshopper, Sawfly, Stem Borer, and also classifies Healthy Leaves. The images were preprocessed using techniques like resizing and noise removal to improve clarity and emphasize the affected regions.. Then images are uploaded and system predicts the pest type output.

Fig.1 Sample Input Images of Pest Classification

For this paper, a dataset of crop leaf images was used, consisting of both healthy and pest-infected samples collected from publicly available repositories such as PlantVillage,Kaggle [9] https://www.kaggle.com/datasets/simranvolunesia/pest-dataset

and field-captured images. The dataset included multiple pest types (e.g., whiteflies, aphids, bollworms) to ensure diversity. All images were preprocessed through resizing, normalization, noise reduction, and augmentation (rotation, flipping, scaling) to improve model robustness.

II.RELATED WORK

[1] Chiranjeevi et al. 2023, designed a deep learning approach utilizing data from citizen-science insect collections. Their CNN-based model achieved >96% accuracy, but the limitation was that it included many non-agricultural insects and lacked focus on crop-specific pests.

[2] White et al. 2023, released the BOLLWM dataset for bollworm pest detection in Indian cotton fields. The contribution was dataset creation, but no baseline accuracy was provided; the limitation was the absence of benchmark detection results.

[3] Nagaraj et al. 2023, applied CNNs for pest detection, reporting 93.8% classification accuracy on their dataset. The limitation was lack of testing in real-field conditions.

[4] A. Padma Priya 2024, “Pest Detection and Prevention for Agricultural Crops Using YOLOv8 Algorithm,” in Proc Conf. Intelligent Systems for Cybersecurity (ISCS ’24), May 2024.Method: YOLOv8 real-time detector demo. Reported performance: not publicly reported in proceedings abstract. Limitation: concept/demo stage; metrics not published.

[5] Krishnateja et al. 2025, employed a Convolutional Neural Network (CNN)-based model for agricultural pest detection, achieving a reported accuracy of 96% in classifying pest images. The method was integrated into a web-based application for smart farming. Method: CNN deployed via Django web app for real-time detection. Limitation: small datasets / lab conditions only.

[6] Pazhanivelan et al. 2025, proposed a multi-class deep CNN for pest and disease detection. The model achieved 96–97% accuracy in field tests, but required large annotated datasets and was sensitive to occlusion and low-light.

[7] K. Bhavadharni.2025, "Pest Detection on Plants Using Image Processing," Method/metric: Image preprocessing plus ML classifier; authors report positive classification results (no detailed mAP). Limitation: small-scale evaluation; uncertain generalization to diverse crops/environments.

[8] Pramod K. Singh et al. (IIIT-Allahabad), 2025. "Real-Time Crop Disease and Pest Detection Method/metric: CVGG-16 CNN with federated learning and IoT; reported 97.25% detection. Using Federated Learning: CVGG-16," IIIT-A projet report/news. success in maize/potato tests. Limitation: requires robust connectivity and compute resources for federated setup.

III. METHODOLOGY

Paper methodology involves collecting and organizing maize and rice crop leaf images for the dataset, including both healthy and pest-affected samples.The dataset is preprocessed using resizing, normalization, and augmentation to ensure high-quality inputs and better generalization for the CNN. Convolutional Neural Network (CNN) and Res Net models are designed and trained on the dataset to classify different pest types. Finally, the models are evaluated using performance metrics like accuracy and loss, with CNN showing better results compared to Res Net.

1. Requirement Analysis

In the requirement analysis phase, all the functional and non-functional requirements of the system are collected from stakeholders such as clients, users, and domain experts. These requirements are documented in a detailed Software Requirement Specification (SRS) document, which serves as the foundation for the entire project.
The main focus here is to understand what the system is supposed to do without worrying about how it will be implemented.

2. System Design

Once the requirements are clearly defined, the system design phase begins. This phase translates the SRS into a blueprint for development.
The design is usually divided into High-Level Design (HLD), which defines the system architecture, modules, and data flow, and Low-Level Design (LLD), which specifies detailed module logic, database design, and interface structures. The purpose of this phase is to decide how the system will be built and to prepare guidelines for developers.

3. Implementation (Coding)

In this phase, the actual development of the software takes place. Developers write code for each module according to the design specifications. Suitable programming languages, tools, and frameworks are chosen to implement the system.
The goal here is to convert the planned design into executable programs, with each module developed and tested individually to ensure correctness.

4. Integration and Testing

After coding, all the developed modules are combined into a complete system in the integration phase. Once integrated, the system undergoes thorough testing to check for defects, ensure functionality, and verify that the system meets the requirements specified in the SRS. Different testing levels such as unit testing, integration testing, system testing, and acceptance testing are performed. The primary objective here is to deliver a bug-free and reliable product.

5. Deployment

In the deployment phase, the fully tested software is installed and delivered to the customer for real-world use. Depending on the project, deployment may be carried out in stages such as beta release, pilot implementation, or a full release. Training may also be provided to end-users at this stage. The purpose of deployment is to make the system operational in its intended environment.

6. Maintenance

Maintenance activities are classified into corrective (fixing errors), adaptive (modifying the system for new environments), and perfective (improving performance). This phase ensures the system continues to function smoothly and remains effective over time.

System Architecture

The system architecture shown in Fig.2 follows a pipeline where crop leaf images are collected, preprocessed, and then passed into deep learning models (CNN and ResNet) for pest type classification.The output provides accurate detection of pest typre as a results, allowing farmers to implement pest management measures promptly.

The proposed methodology uses Convolutional Neural Networks (CNN) to automatically detect and classify pest types from crop leaf images. The process involves collecting and preprocessing maize and rice leaf datasets, applying resizing, normalization, and augmentation, and then training CNN and ResNet models for comparison.

ARCHITECTURE DIAGRAM

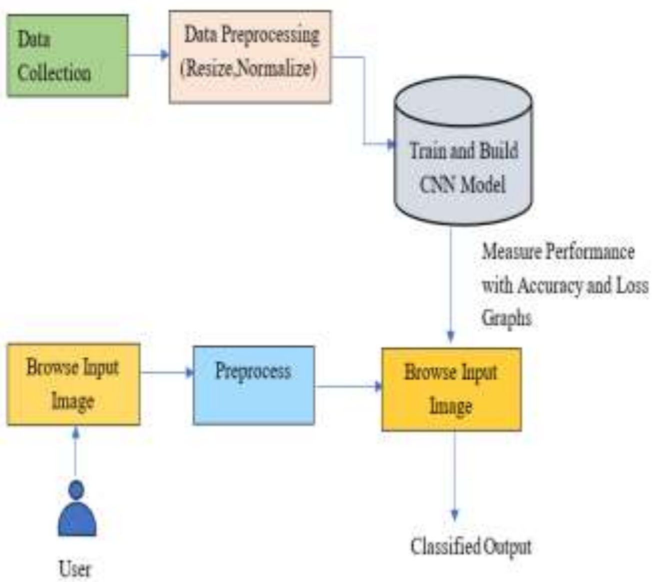

Fig.2 System Architecture

Work Flow Explanation

This paper is designed to build an automated system capable of detecting pests on crop leaves using image processing and Convolutional Neural Networks (CNNs), a key approach in deep learning. The process is divided into sequential stages, from problem identification to evaluation and potential deployment.

This paper develops an automated system to detect pests on crop leaves using CNNs.
A dataset of healthy and pest-infected leaves is collected from online sources and custom images.
Images are preprocessed by resizing, removing noise, normalizing colors, and applying augmentation.

The CNN automatically learns features like color, texture, and shape for classification.A model such as MobileNet is trained using optimization techniques like Adam or SGD.The trained model is tested on unseen data and evaluated with accuracy, precision, recall, and confusion matrix.Finally, results are analyzed, and errors are studied to improve the system's performance.

Algorithm: CNN algorithm for pest detection

Input: Crop leaf image dataset D= {x1, x2, ..., xn} with labels Y= {y1, y2, ..., yn}

Output: Classified pest type (e.g., Whitefly, Aphid, Bollworm, Healthy).

1.Collect dataset D containing pest-infected and healthy leaf images

2.Perform image preprocessing:

* Resize all images to fixed size m x n.

* Normalize pixel values: x'=x/255

3.Initialize CNN model with Convolution, pooling and fully connected layers.

* Forward Propagation shown in eq (1)

$y_{i,j} = \Sigma_m \Sigma_n x_{i+m.j+n.}{}^{km,n}$

Where x = input image, k= kernal /filter

4. Softmax Classification:

$$P(y=i|x) = \frac{e^{zi}}{\Sigma_j e^{zj}} \qquad \text{eq (2)}$$

Where P(y=i|x) = probability that image belongs to class i.

Activation (ReLU):

$$F(z) = \max(0,z) \qquad \text{eq (3)}$$

4.Pooling (e.g., Max Pooling):

$$y_{i,j} = max_{m,n}(x_{i+m,j+n}) \qquad \text{eq (4)}$$

Loss Function (categorical cross-entropy):

$$L = -\sum_{i=1}^{c} y_i \log(\hat{y}_i) \qquad \text{eq (5)}$$

Where C = number of pest classes, where y_i = true label $\hat{y}_i$ = predicted probability.

Update weights using Backpropagation and Adam optimizer:

$$\omega \leftarrow \omega - \eta.\frac{\partial L}{\partial w} \qquad \text{eq (6)}$$

Repeat training until convergence.

5.Classify test images into pest types (aphid, armyworm, beetle)

Performance Evaluation Metrics

To measure model performance:

1.Accuracy

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad \text{eq (7)}$$

2. Precision

$$\text{Precision} = \frac{TP}{TP+FP} \quad \text{eq (8)}$$

Where:

- TP = True Positive (correctly detected pest)
- TN = True Negative (correctly detected healthy leaf)
- FP = False Positive (wrongly predicted pest)
- FN = False Negative (missed pest)

IV. TECHONOLOGY USED

Development Environment: Spyder IDE

Spyder (Scientific Python Development Environment) is a free IDE included with Anaconda, offering code editing, interactive testing, debugging, and inspection tools. After installing Anaconda, Spyder can be launched on Windows, macOS, or Linux via the command line or through Anaconda Navigator by clicking the Spyder icon. For detailed guidance, users can refer to the official Spyder website or documentation.

Anaconda Terminal: it is just like command prompt, but it makes sure that you are able to use anaconda and conda commands from the prompt, without having to change directories or your path. These locations contain commands and scripts that you can run.

Matplotlib

Matplotlib was used to visualize training and testing results of CNN and ResNet models. Graphs like accuracy, loss curves, and comparison bar charts were plotted using this library. These visualizations helped in analyzing which model performed better for pest detection.

Python Programming

Python was the main programming language used to implement this project. I t provided libraries like TensorFlow, Keras, OpenCV, and NumPy for deep learning and image processing. Using Python made it easier to train CNN and Res Net models and evaluate their accuracy.

Libraries Used

TensorFlow: TensorFlow is an open-source deep learning framework developed by Google that provides a comprehensive ecosystem for building, training, and deploying machine learning and neural network models. It supports both CPU and GPU computations, making it efficient for handling large-scale data.

Keras: Keras is a high-level deep learning API running on top of TensorFlow, designed to simplify the process of developing neural networks. It provides user-friendly interfaces, modular components, and pre-built layers, which make model design and experimentation faster and more intuitive.

Flask: Flask is a lightweight and flexible Python web framework that is widely used for developing web applications and RESTful APIs. In machine learning projects, Flask is commonly used to deploy trained models, allowing users to interact with the system through a web-based interface.

OpenCV: OpenCV (Open Source Computer Vision Library) is an open-source library used for real-time computer vision and image processing. It provides efficient tools for tasks such as image enhancement, object detection, and feature extraction, making it essential in medical imaging applications like MRI scan analysis.

V. RESULTS

The CNN and ResNet models were successfully trained on maize and rice crop leaf datasets, achieving high accuracy in pest type classification. CNN outperformed ResNet with slightly better accuracy and lower loss. The system generated accuracy and loss graphs, validating the effectiveness of the models.
Fig.3 shows that bar graph for this pest detection system, CNN proved to be more efficient and reliable than ResNet, achieving higher accuracy and lower loss.

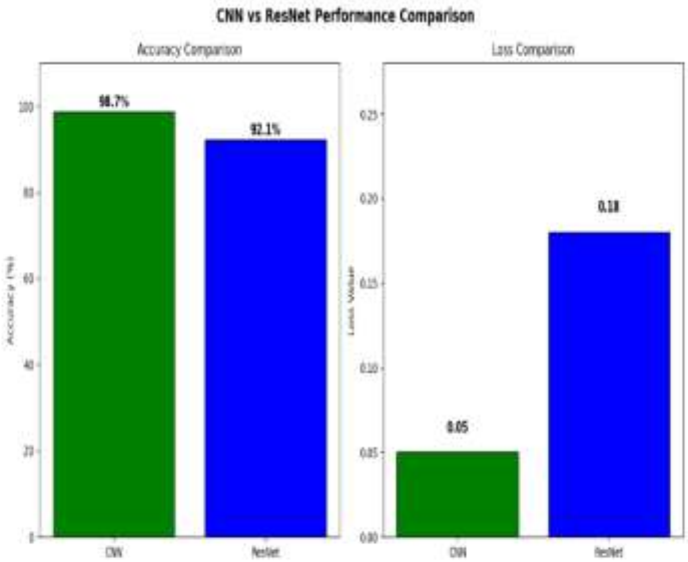

Fig.3 Performance Graph Between CNN and ResNet

Table.1 Comparison table between CNN and ResNet based on the accuracy and loss values

Models	Accuracy (%)	Loss Value
CNN	98.7	0.05
ResNet	92.1	0.18

The performance comparison between CNN and ResNet clearly shows in Table.3 that CNN achieved superior results for pest detection. The CNN model reached an accuracy of 98.7% with a very low loss value of 0.05, whereas the ResNet model attained 92.1% accuracy with a higher loss of 0.18.

This indicates that CNN not only generalizes better but also minimizes classification errors more effectively than ResNet in this dataset. Hence, CNN proves to be a more efficient and reliable model for accurate pest type detection in crop.

Snapshots:

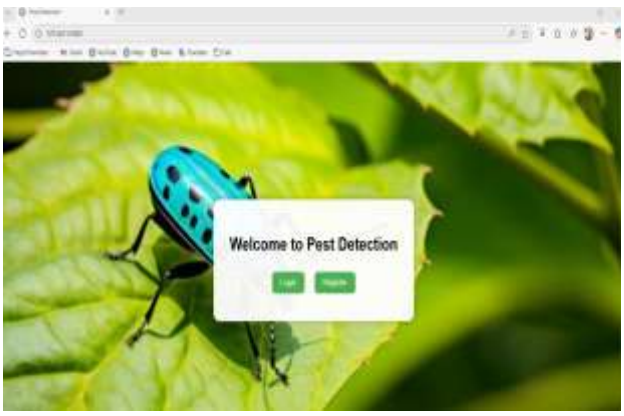

Fig.1 Home Page

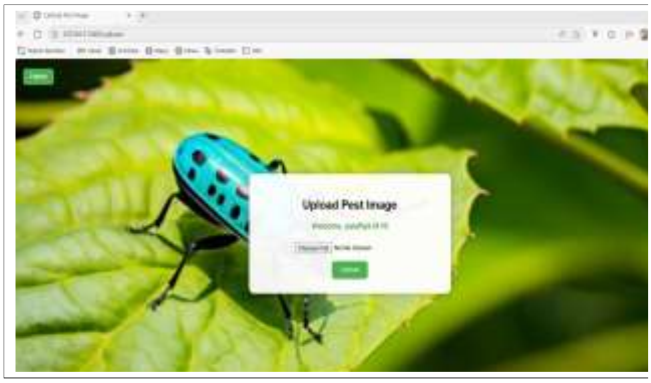

Fig.2 Upload Page

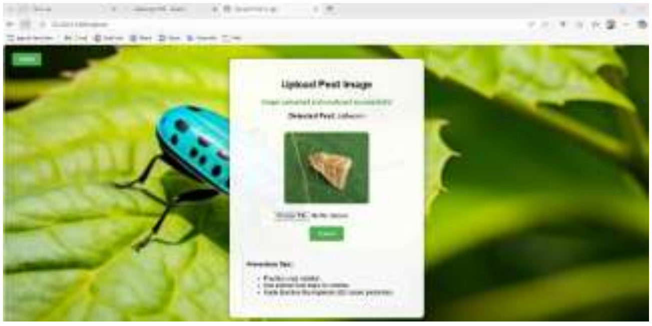

Fig.3 Prediction Page (Bollworm Pest)

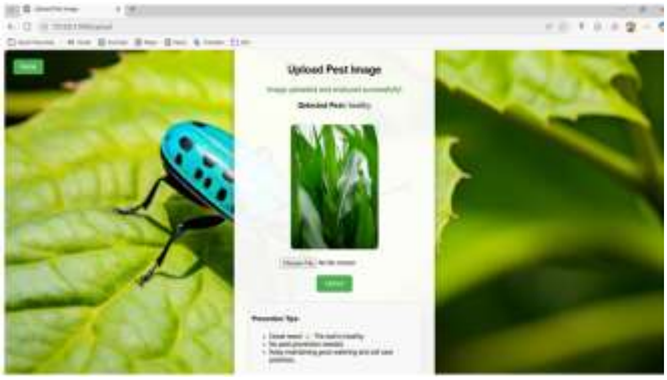

Fig.4 Healthy Leaves

The system provides a secure Login/Register page for user authentication, where new users can register and existing users can log in. It focuses on detecting common crop pests such as Aphid, Armyworm, Beetle, Bollworm, Mites, Grasshopper, Sawfly, Stem Borer, and also identifies healthy leaves.

On the Upload page, users can select and preview crop leaf images from their device, and the Predict page shows the model's result with the predicted pest or healthy condition along with confidence scores. To support farmers, the system also provides useful pest prevention tips based on the detected class for effective crop management.

VI. CONCLUSION

This paper focused on developing an automated pest detection system for maize and rice crops using image processing and deep learning techniques. Dataset of healthy and pest-affected leaf images was preprocessed through resizing, normalization, and augmentation. CNN and ResNet models were trained and compared, with CNN achieving higher accuracy 98.7% and lower loss.

The system generated accuracy and loss graphs, validating its effectiveness in pest classification. In this project we are also providing a some prevention tips for users. This work demonstrates how deep learning can support farmers in early pest identification, reduce excessive pesticide use, and promote sustainable agriculture

V. REFERENCE

[1] S. Chiranjeevi et al, "Deep Learning Powered Real-Time Identification of Insects Using Citizen Science Data," arxiv preprintar Xiv:2306.02507, 2023. Available: https://arxiv.org/abs/2306.02507

[2] J. White, C. Agrawal, A. Ojha, A. Agnihotri, M. Sharma, and J. Doshi, "BOLLWM: A Real-World Dataset for Bollworm Pest Monitoring from Cotton Fields in India," *arXiv*, Apr. 2023. [Online]. Available: https://arxiv.org/abs/2304.00763

[3] G. Nagaraj, D. Sungeetha, M. Tiwari, V. Ahuja, A. K. Varma, and P. Agarwal, "Advancements in Plant Pests Detection: Leveraging Convolutional Neural Networks for Smart Agriculture," Engineering Proceedings, vol.59, no.1, Art.201, 2023. Available: https://www.mdpi.com/2673-4591/59/1/201

[4] A. Padma Priya, C. V. V. Sudhakar, A. Pavan Kumar, and Y. Y. Reddy, "Pest Detection and Prevention for Agricultural Crops Using YOLOv8 Algorithm,"in Proc. Int.Conf. Intelligent Systems for Cybersecurity(ISCS24), 2024. Available: https://www.researchgate.net/publication/382235982

[5] I. Krishnateja, G. V. Gopi, G. Malini, B. L. V. Vamshi Krishna, and B. Kejiya, "Agricultural Pest Detection Using Convolutional Neural Networks: A Smart Farming Solution," Int. J. Adv. Comput.Eng.Commun.Technol.,vol.14,no.1,pp.39–48,2025.Available: https://journals.mriindia.com/index.php/ijacect/article/view/169

[6] R. Pazhanivelan, K. P. Ragunath, N. S. Sudarmanian, S. Satheesh, and P. Shanmugapriya, "Deep Learning-Based Multi-Class Pest and Disease Detection in Agricultural Fields," J. Sci. Res.Rep.,vol.31,no.1,pp.538–546,Jan.2025.Available: https://doi.org/10.9734/jsrr/2025/v31i12797

[7] K. Bhavadharni and K. Banuroopa, "Pest Detection on Plants Using Image Processing," Int. J. Sci. Res. Comput. Sci. Eng. Inf.Technol., vol.11, no. 2, pp.1782–1786, 2025. Available: https://www.researchgate.net/publication/390064173

[8] P. K. Singh et al, "Real-Time Crop Disease and Pest Detection Using Federated Learning and IoT: The CVGG-16 Model," IIIT-Allahabad Research Project News Release, 2025. Available:https://timesofindia.indiatimes.com/city/allahabad/iiit-a-researchers-develop-ai-tech-for-real-time-crop-disease-detection-in-indian-farms/articleshow/123370013.cms

[9] Datasets: https://www.kaggle.com/datasets/simranvolunesia/pest-dataset