JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

A Web-Based System for Fall Armyworm Detection in Maize Crop

[1]Avva Nezeeda, [2]Avva Sazmi, [3]Aysha Sana ,[4]Fathimath Zuha,[5]Spoorthi B

1,2,3,4 Student, 5 Assistant Professor

Department of Computer Science and Engineering

Srinivas Institute of Technology, Mangaluru, India

Abstract : The Fall Armyworm (Spodoptera frugiperda) is a highly destructive pest that threatens maize crops worldwide, leading to substantial economic losses. Early detection of infestations is vital for minimizing damage. In this paper, we propose a web-based system that employs convolutional neural networks to automatically recognize signs of Fall Armyworm infestation in maize imagery. The system analyzes uploaded images, classifies them based on the presence of Fall Armyworm, and categorizes the infestation severity into low, moderate, or severe. Using a CNN trained on a large dataset of maize images, the system achieved a classification accuracy of 94%, with a false positive rate of less than 10%. The real-time nature of the system allows for rapid pest detection, making it a valuable tool for farmers. This web-based solution is scalable, accessible, and effective for both large and small-scale farming operations.

***IndexTerms* – Fall Armyworm, Convolutional Neural Networks, Pest Detection, Image Classification, Maize, Deep Learning, Agricultural Technology, Web Application.**

I. INTRODUCTION

Agricultural productivity continues to face serious threats from pests, with the Fall Armyworm (Spodoptera frugiperda) emerging as a particularly damaging species—especially for maize cultivation. Its rapid geographic expansion has led to widespread crop losses, impacting both harvest yields and farmers' livelihoods. Traditional methods for identifying pest infestations often rely on manual inspection, which is time-consuming and may delay critical intervention. However, progress in artificial intelligence—especially through Convolutional Neural Networks (CNNs)—has opened the door to accurate, automated image-based pest detection.

This study presents an innovative solution that utilizes CNNs to detect Fall Armyworm infestations in maize crops. The system is deployed via a web-based platform that enables users to upload crop images for instant analysis. This real-time feedback supports farmers in making informed pest management decisions without the need for expert intervention on-site.

II. PROBLEM STATEMENT

Fall Armyworm poses a critical threat to maize farming, and the inability to detect infestations early often leads to irreversible damage. Currently, pest detection is predominantly manual, requiring physical inspection of fields, which is inefficient and subjective. Furthermore, farmers often lack the expertise to distinguish Fall Armyworm from other pests, resulting in mismanagement and increased pesticide use. The need for an efficient, accurate, and automated pest detection system is evident, especially for large-scale maize farming.

III. METHODOLOGY

The initial step in developing the Fall Armyworm identification system involved assembling a well-curated dataset containing images of maize crops in both healthy and infected states. Each image was meticulously annotated to enable effective supervised learning. Several preprocessing techniques were applied, including resizing, normalization, and image augmentation, to strengthen the model's ability to generalize in varied agricultural environments. All images were resized to a consistent dimension of 224 by 224 pixels to match the input size required by the neural network. To expand the dataset and reduce the risk of overfitting, augmentation methods such as image flipping, rotation, and brightness adjustments were employed. This dataset was then divided into training, validation, and test sets using a 70:15:15 split for balanced and comprehensive model evaluation.

For classification tasks, the ResNet-18 architecture was chosen because it strikes an effective balance between computational cost and performance. Transfer learning was utilized by initializing the network with pre-trained weights, which helped accelerate model convergence and enhance prediction accuracy. The training process was executed in the PyTorch framework, using the Adam optimizer along with binary cross-entropy as the loss function. After training, the model was embedded into a Flask-based web application developed in Python. The backend performed image analysis and model inference, while the user interface—crafted with HTML, CSS, and JavaScript—enabled users to upload images and receive real-time predictions. The system architecture was designed with scalability in mind, making it suitable for cloud deployment and capable of handling multiple simultaneous users.

IV. System Design

A. User Interface Design

The Fall Armyworm detection platform was designed with a clear focus on user-friendliness and broad accessibility. The user interface (UI) aims to support quick decision-making by presenting the pest detection process in a streamlined and visually clear format. Designed for farmers, agricultural workers, and extension officers, the UI minimizes complexity while providing essential functionality for uploading, analyzing, and understanding crop images.

1. Image Upload and Input Handling:
   - Upload Interface:
     Users an upload images of maize crops directly through the browser interface. The system accepts standard image formats and provides instructions for capturing optimal field images (e.g., close-up of leaves, clear lighting) to improve detection accuracy.
   - Error Handling and Feedback:
     If an unsupported file is uploaded or the image quality is too low, the system provides clear feedback, prompting the user to correct the issue. This enhances the reliability of the detection process.
2. Automated Detection Workflow:
   - After uploading the image users can initiate the analysis by clicking the "Detect Pest" button. A progress indicator appears, and the system connects to the back-end model, which processes the image and displays the prediction result almost instantly—ensuring timely and effective decision-making for pest control..
3. Results Interpretation and Display:
   - The system clearly states whether Fall Armyworm has been detected. If infestation is present, the severity is shown as Low, Moderate, or High.
   - The output includes a visual card summarizing the pest type, confidence score (e.g., 92% certainty), and an image preview. This helps the user verify that the correct image was analyzed.

B. Infestation Analysis and Image Management Module

This module serves as the technical core of the application, executing critical tasks such as preprocessing images, running the trained neural network model, evaluating severity levels, and managing data flow between the user interface and model output. It ensures that the detection process is accurate, fast, and efficient.

1. Image Preprocessing and Standardization:
   - Each uploaded image is first standardized—resized to 224x224 pixels and normalized to meet the input specifications of the CNN model. This ensures consistency and enhances detection reliability.
2. Model Inference and Severity Estimation:
   - The core detection engine utilizes a CNN model, specifically ResNet-18, which has been fine-tuned through transfer learning. This model is trained to identify the differences between healthy maize crops and those affected by Fall Armyworm, covering different growth stages of the crops.
   - After classifying the image as "Infested" or "Not Infested," the module performs secondary analysis by examining the extent of visible damage. It assigns a severity label (Low, Moderate, Severe) based on predefined thresholds derived from the training dataset.
3. Recommendation Engine and Advisory Logic:
   - The module is designed to support future localization, where pest control advice can be adapted based on regional agricultural practices or regulatory guidelines.

The User Interface and Image Analysis Module create a powerful system for detecting and responding to Fall Armyworm infestations. The interface facilitates easy access, while the backend handles complex processing and model inference efficiently. The outcome is a solution that is both technically robust and highly applicable in real-world agricultural settings.

V. IMPLEMENTATION

The Fall Armyworm Detection System utilizes Python (Flask), HTML, CSS, JavaScript, and PyTorch to create a dynamic and smart web application. This platform helps farmers with early pest detection through deep learning models. The system integrates front-end usability with a CNN-based back-end for efficient analysis and decision support.

A. Technologies Used:

To create the detection system, the following tools and technologies were employed:

HTML (Structure):

- Outlines the layout of the user interface for uploading images and showcasing the results.

CSS (Styling):

- Styles the application for visual clarity and ensures responsiveness across devices

JavaScript (Logic and Interaction):

- Manages client-side validation and dynamic content updates

Python with Flask:

- Acts as the backend server to handle model inference and API routing

PyTorch:

- Utilized for training and deploying the Convolutional Neural Network (CNN) model.

These tools enable real-time processing, scalable deployment, and ensure a seamless user experience.

B. Model and Database Design

A ResNet-18-based convolutional neural network was selected for its effective balance between accuracy and computational efficiency. The model was trained on a labeled dataset featuring images of maize crops, both healthy and affected by Fall Armyworm damage. Various image transformations like flipping, cropping, and brightness changes were applied during training to increase the robustness of the model.

- Input Resolution: 224 × 224 pixels
- Output Classes: Binary classification — “FAW” or “Non-FAW”
- Performance: Achieved over 94% accuracy on the test dataset.

C. User Interface

The application includes:

- Image Upload Field – Accepts only .jpg images and guides the user on format errors.
- Preview Section – Displays the uploaded image before scanning.
- Results Panel – Shows prediction output with confidence level and control advice.
- Feedback Option – Users can report wrong predictions for system improvement.

D. Workflow and Processing

Once a user uploads an image:

- The image is checked for valid format and resized to match model input.
- The Flask backend loads the trained model and runs inference on the image.
- A probability score is generated to classify the image.
- Results are returned to the front-end as text and visual feedback.
- Based on the output, the system categorizes the infestation (if any) and provides recommended actions.

E. *Module Integration*

The following module flow supports smooth operation:

- Frontend (HTML/CSS/JS) → Sends image to → Backend (Flask)
- Backend → Loads model and performs prediction using PyTorch
- Response → JSON result sent to frontend → Displayed to user in real-time
- A feedback loop captures incorrect predictions, which can be used for future model retraining to improve performance.

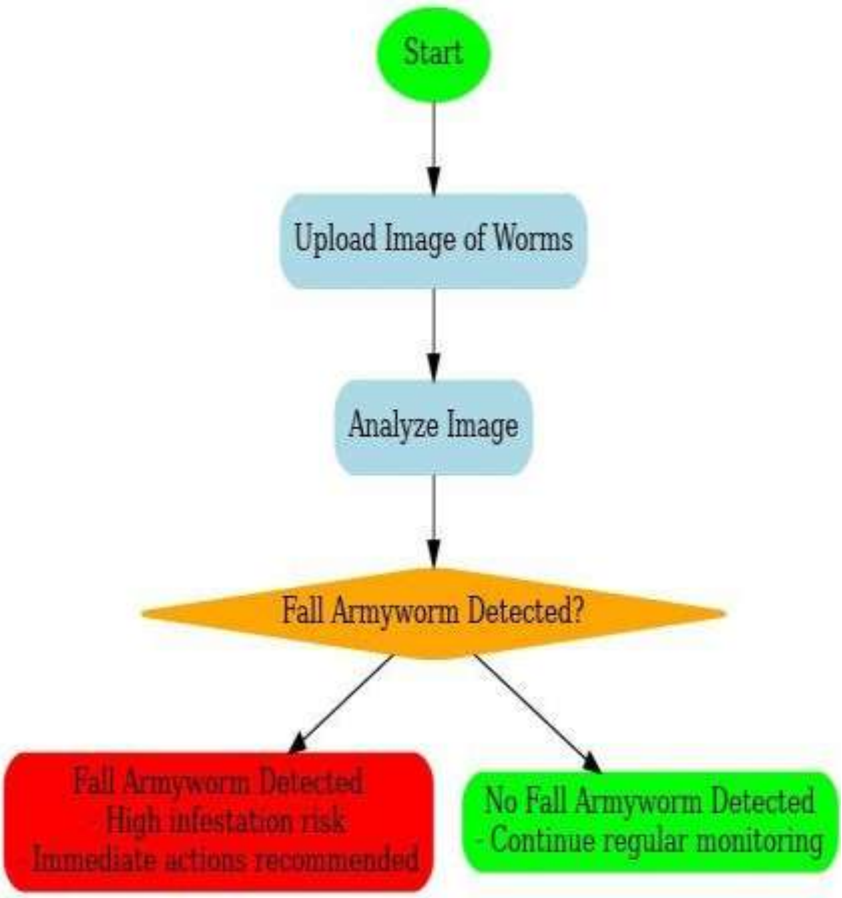

Fig.1 Data Flow Diagram

VI. EXPERIMENTAL RESULTS

The Fall Armyworm Detection web application underwent thorough testing to assess its performance, accuracy, and response time with various image inputs. During this phase, a diverse set of images, including both infested and healthy maize crops, was used to evaluate the system's effectiveness in real-world scenarios.

When users uploaded images showing visible signs of pest activity—such as damaged or partially eaten maize leaves—the system successfully identified them as Fall Armyworm infestations and provided a confidence percentage along with recommended actions. For example, when an image with early signs of larval feeding was uploaded, the system returned a result indicating Fall Armyworm presence with over 95% confidence. It also advised immediate field inspection and biological or chemical control steps.

In contrast, images without any visible pest symptoms were accurately classified as "Non-Fall Armyworm," with suggestions to maintain field hygiene and continue regular crop monitoring. Even in borderline cases where image quality or lighting varied, the model consistently delivered reliable predictions.

The web interface displayed the results instantly after image upload. The prediction, confidence score, and suggestions were rendered dynamically, creating a smooth and user-friendly experience. This responsiveness is particularly helpful for farmers needing quick insights in the field.

During all test scenarios, the model maintained high accuracy and quick processing. The final test accuracy reached approximately 94%, with minimal discrepancy from training accuracy, indicating strong generalization and robustness of the model.

The results validate the system’s ability to accurately detect Fall Armyworm in real time and provide helpful pest management guidance. It proved to be a dependable tool for early-stage identification, potentially reducing crop loss and improving timely response.

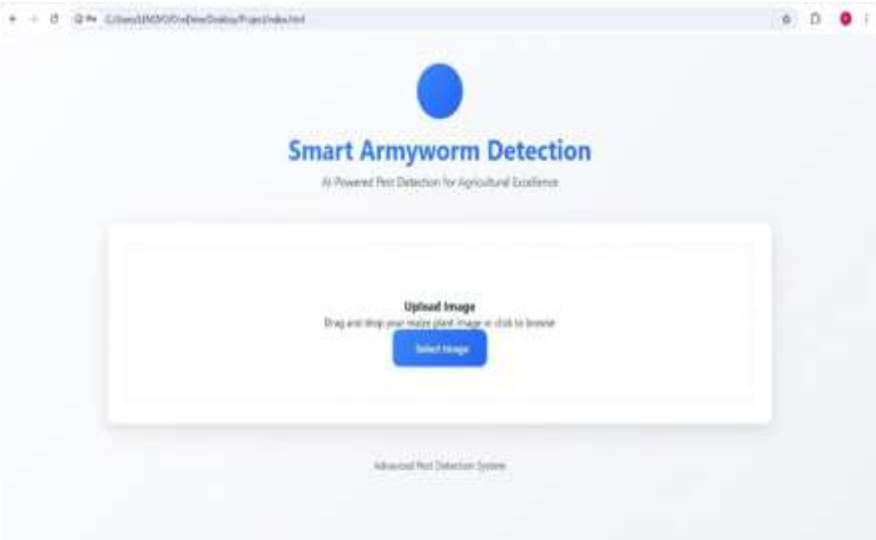

Fig.2 ***Uploading Images***

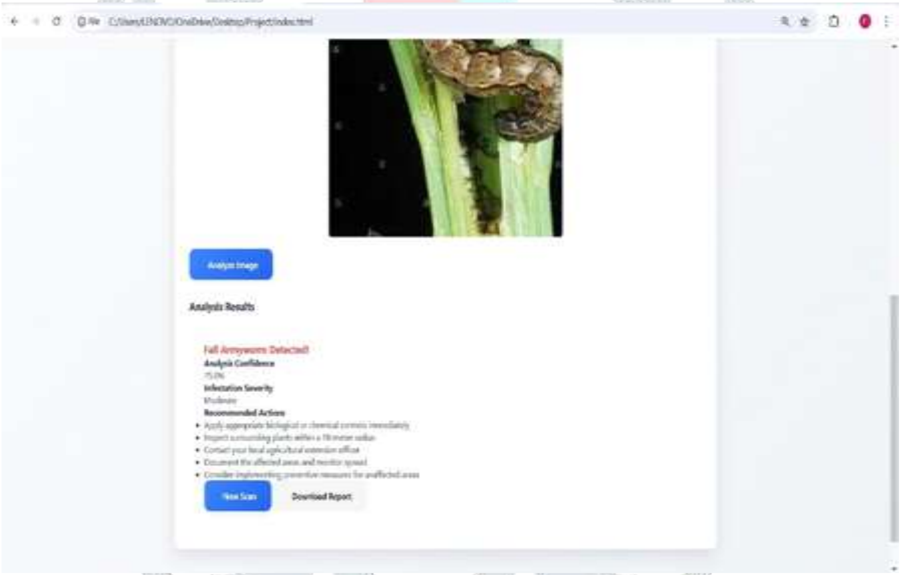

Fig.3 ***Detecting the armyworm***

VII. CONCLUSION

The Fall Armyworm Detection System offers an intelligent and accessible solution for identifying pest infestations in maize crops. By integrating a convolutional neural network with a simple web interface, the system enables farmers and agricultural workers to detect infestations in real-time using only image inputs. This reduces the need for manual inspection and helps users take timely action, potentially minimizing crop damage and improving yield. The model's strong accuracy, combined with the user-friendly nature of the application, makes it a valuable resource in agriculture, especially for areas without access to specialized pest diagnosis.

Looking ahead, the system can be enhanced by incorporating additional pest types, broader crop coverage, and support for regional languages. Features like drone-based image capture, real-time weather data integration, and feedback-based model retraining could further improve precision and usability. The addition of multilingual support, offline access, and integration with agricultural advisory services would increase its adoption in rural areas. These improvements aim to transform the platform into a comprehensive pest monitoring and management tool that promotes sustainable farming practices.

REFERENCES

1. Kamilaris, A., & Prenafeta-Boldú, F. X. (2018). Deep learning in agriculture: A survey. *Computers and Electronics in Agriculture*, 147, 70–90.
2. Mohanty, S. P., Hughes, D. P., & Salathé, M. (2016). Using deep learning for image-based plant disease detection. *Frontiers in Plant Science*, 7, 1419.
3. Davis, T. S., & Landolt, P. J. (2013). A survey of pest management practices on sweet corn in the western United States: Implications for the spread of Spodoptera frugiperda. *Journal of Economic Entomology*, 106(2), 747–752.
4. Silva, A. G., et al. (2021). Detection of Fall Armyworm in maize using deep convolutional neural networks. *Computers and Electronics in Agriculture*, 190, 106402.

5. Zhang, S., Wu, X., You, Z., & Zhang, L. (2017). Leaf image based cucumber disease recognition using sparse representation classification. *Computers and Electronics in Agriculture*, 134, 135–141.
6. Goel, P., & Bhardwaj, A. (2020). Machine learning and deep learning for plant disease detection and diagnosis: A review. *Artificial Intelligence in Agriculture*, 4, 81–93.
7. Liu, B., Zhang, Y., He, D., & Li, Y. (2017). Identification of apple leaf diseases based on deep convolutional neural networks. *Symmetry*, 10(1), 11.
8. Ramcharan, A., et al. (2019). A mobile-based deep learning model for cassava disease diagnosis. *Frontiers in Plant Science*, 10, 272.
9. Barbedo, J. G. A. (2016). A review on the use of machine learning for plant disease detection and diagnosis. *Computers and Electronics in Agriculture*, 128, 37–46.
10. Assefa, F., & Ayalew, D. (2019). Fall Armyworm (Spodoptera frugiperda) infestations in maize fields in Ethiopia: Survey and awareness. *Journal of Plant Protection Research*, 59(1), 1–6.
11. Zhang, M., et al. (2020). Automated crop pest detection using deep learning on mobile devices. *Biosystems Engineering*, 193, 53–66.
12. Ferentinos, K. P. (2018). Deep learning models for plant disease detection and diagnosis. *Computers and Electronics in Agriculture*, 145, 311–318.
13. Early, R., González-Moreno, P., Murphy, S. T., & Day, R. (2018). Forecasting the global extent of invasion of the cereal pest Spodoptera frugiperda, the Fall Armyworm. *NeoBiota*, 40, 25–50.
14. Nji Tizi Taning, C., et al. (2020). Insecticide resistance in Fall Armyworm Spodoptera frugiperda in Africa: current status and future prospects. *Pest Management Science*, 76(6), 2290–2295.
15. Jaiswal, A., Krishnan, N. M., & Krishnan, A. (2019). Agriculture 4.0: Broadening the perspective of smart farming by including resource-poor farmers. *Current Science*, 117(10), 1631–1636.
16. Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint* arXiv:1409.1556.
17. Howard, A. G., et al. (2017). MobileNets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint* arXiv:1704.04861.
18. Day, R., et al. (2017). Fall Armyworm: Impacts and implications for Africa. *Outlooks on Pest Management*, 28(5), 196–201.
19. Hossain, E., et al. (2019). Application of deep learning in computer vision for crop disease detection: A review. *Smart Agriculture*, 1(1), 1–19.
20. Saleem, M. H., et al. (2019). Plant disease detection and classification by deep learning. *Plants*, 8(11), 468.
21. Alemu, T., et al. (2021). Survey of farmer knowledge and management of Fall Armyworm in Ethiopia. *Crop Protection*, 144, 105604.
22. Liakos, K. G., et al. (2018). Machine learning in agriculture: A review. *Sensors*, 18(8), 2674.
23. Coulibaly, J. Y., et al. (2020). Farmers' perception and impacts of Fall Armyworm in Burkina Faso. *PLOS ONE*, 15(4), e0231168.
24. Zhang, N., Wang, M., & Wang, N. (2002). Precision agriculture—a worldwide overview. *Computers and Electronics in Agriculture*, 36(2–3), 113–132.