International Journal of Environmental Sciences
ISSN: 2229-7359
Vol. 11 No. 6s, 2025
https://www.theaspd.com/ijes.php

Edge-Deployed Visual Pest Detection System For Real-Time Crop Protection

Dr. Annapoorna B R[1*], Dr. D R Ramesh Babu[2]
[1*]*Assistant Professor, Department of Computer Science and Engineering Dayananda Sagar College of Engineering Bengaluru, India, br.anu243@gmail.com*
[2]*HOD & Vice Principal, Department of Computer Science and Engineering, Dayananda Sagar College of Engineering Bengaluru, India, bobrammysore@gmail.com*

Abstract– *Early and precise identification of agricultural pests is essential to avoid crop damage and maximize agricultural output. An edge-deployable pest detection system based on convolutional neural networks (CNNs) and deep learning models for real-time monitoring and pest infestation early detection is the premise of this research. Natively designed for deployment in field conditions, the system runs on low-power edge hardware like Raspberry Pi and NVIDIA Jetson Nano, carrying out on-site image processing independent of cloud connectivity. Embedded camera-captured images of crop leaves are processed locally to identify and classify pest species. Moreover, the system includes an alert mechanism in real-time through visual displays, or smartphone push notifications to alert farmers in due time, thus minimizing yield loss. Systematic tests on public and curated pest image datasets showed high precision, low latency, and effectiveness in terms of resource utilization. This study highlights the opportunity of integrating deep learning and edge computing technologies toward intelligent agriculture, especially for remote or resource-poor locations.*
Keywords: *convolutional neural networks (CNNs), deep learning models, agricultural pests, early detection*

I. INTRODUCTION

The pests are one of the greatest agricultural threats to world agriculture, often leading to high losses in yield and economy. Conventional pest identification mostly depends on visual examination, which, apart from being time-consuming, is also labor-intensive and subject to human error. As the agricultural industry is more and more accepting of precision farming, more and more pressure has been put on smart, real- time, and highly accurate pest monitoring solutions that are cost-effective and scalable for enabling sustainable crop management. Recent developments in computer vision and deep learning have enabled automated pest classification based on images with promising results. However, most of the current models require a lot of computational resources, which restricts their deployment in rural agricultural areas where high-performance hardware and reliable internet connectivity are scarce. To address these challenges, we suggest an edge-deployable visual alert and pest detection system that embeds AI-based pest recognition into the field domain itself, thus promoting accessibility and responsiveness. The system employs lightweight convolutional neural network (CNN) models that are optimized for running on embedded platforms like Raspberry Pi and NVIDIA Jetson Nano. The system takes crop leaf images through attached cameras and processes them on-site to detect and classify pests. Crop productivity is severely jeopardized by pest infestations that can lead to high yield losses and economic losses for farmers. Conventional pest identification techniques are highly dependent on visual inspection, which is time-consuming, labor-intensive, and subject to errors. With the development of artificial intelligence (AI) and deep learning, scientists have started using these technologies to create intelligent, autonomous pest detection systems with a view to improving crop protection and encouraging sustainable agriculture. Artificial intelligence-based pest detection systems use image processing, machine learning, and deep learning models— particularly convolutional neural networks (CNNs)—to detect and classify pests with very high accuracy from images of crops. AI technologies allow real-time detection, decrease reliance on chemical pesticides, and facilitate timely interventions, eventually reducing crop losses and enhancing food security [1]. Edge computing has also sped up the deployment of such intelligent systems into actual agricultural environments. Albanese et al. [1] showed a deep neural network (DNN)- based insect detection system deployable on edge devices such as Raspberry Pi, providing low-latency image classification in real-time in the field. Likewise, MSFNet- CPD proposed by Zhang et al. [2] integrates visual and textual information through a multi-scale cross-modal fusion approach for reliable pest identification, enhancing performance even under harsh environmental conditions. Venkateswara and Padmanabhan [3] improved pest classification with autoencoders and CNNs and reinforced segmentation methods like RGB colour filtering and YOLOv3 object detection to achieve a 84.95% accuracy. Bompani et al. [4], in turn, tested CNN-based pest detection on ultra-low-power microcontrollers and proved to be highly accelerated in speed and power efficiency with GAP9 SoC compared to the earlier platforms. In greenhouses and other controlled settings, AI-based systems were applied not just for pest identification but even population forecasting. Kapetas et al. [5] proposed a mixed system integrating real- time monitoring and ARIMAX-based forecasting models to improve pest management approaches.

II. MOTIVATION OF THE RESEARCH

The drive behind this study is the growing contribution of pest infestations to world agriculture, resulting in large crop losses, lower productivity, and higher levels of pesticide use. Conventional methods of detecting pests—visual inspection and manual scouting—are time-consuming, labor-intensive, and inefficient in large-scale production environments and are not consistent. There is a pressing requirement for scalable, accurate, and real-time monitoring solutions for pests that can work independently in the field.

A. ***Reducing Crop Loss Through Timely Detection***

Through the use of edge-based AI models, farmers will be able to identify infestation by pests in the early stages and instantly take measures to remedy the situation, thereby minimizing crop destruction and yield loss. This reduces the need for broad-spectrum pesticides and promotes sustainable agriculture.

B. ***Empowering Farmers with Real-Time Alerts***

The inclusion of visual alert methods like LEDs, alarms, or mobile alerts guarantees farmers receive instantaneous notification of pest infestation. This facilitates better- informed decision-making, raised interaction with smart tools, and responsiveness to pest risks.

C. ***Sustainability and Accessibility***

Running pest detection models on low-cost, resource- limited platforms like Raspberry Pi and Jetson Nano makes the technology accessible to marginal and small farmers, particularly in remote villages where there is limited internet connectivity. These models are run without requiring constant cloud connectivity, hence minimizing operational costs and supporting sustainable agriculture.

III. CONTRIBUTIONS

The core contribution lies in the model development and implementation of a predictive model involved:

A. ***AI-Based Pest Detection Model:***

ML model that accurately predicts heart disease risk, enabling early diagnosis and personalized treatment.

B. ***Field Validation and Dataset Expansion***

The model is validated using curated and publicly available datasets (e.g., IP102), and performance metrics such as accuracy, latency, and energy efficiency are assessed.

IV. REALTED WORK

Albanese et al. (2021) introduced an edge-based energy- efficient pest detection system for precision agriculture based on deep neural networks (DNNs) implemented on smart traps with onboard hardware [1]. Their scheme uses low-power platforms like Raspberry Pi and Intel Neural Compute Stick to classify pests in real time within pheromone traps without relying on continuous cloud connectivity. Three varying CNN architectures LeNet-5, VGG16, and MobileNetV2 were trained and tested with the highest classification accuracy achieved by VGG16. Zhang et al. (2025) put forth MSFNet-CPD, a Multi-Scale Cross-Modal Fusion Network for strong crop pest detection by combining both visual and textual features.The model combines high-quality visual information with text descriptions to address challenges such as image quality degradation and subtle pest differences. They also introduced the ACIE strategy to produce a new multi-target pest dataset (MTIP102), which improved generalization in intricate real- world environments [2].Venkateswara and Padmanabhan (2025) introduced a deep learning-based monitoring and classification method for pests using Convolutional Neural Networks and Autoencoders for the improvement of accuracy in detecting pests in smart agriculture. Using the IP102 dataset consisting of 82 classes of pests, their system utilized Autoencoders to balance the dataset and YOLOv3 to detect objects. Segmentation was conducted using RGB color codes and Grab cut to segment pests and classification through ResNet-based CNN. This coupled framework attained accuracy of 84.95%, and segmentation accuracy was based on IoU at 80%.[3]. Bompani et al. (2024) investigated on-device pest recognition based on a heterogeneous multi-core microcontroller that could speed up CNN inference with negligible energy expenditure. They contrasted a standard Viola-Jones algorithm with a CNN model for codling moth detection. The system, deployed in smart traps, carried out image processing locally to lower dependence on energy-hungry cloud communication [4]. Kapetas et al. (2025) proposed an integrated AI-based system for detecting pests and population prediction in greenhouses on the example of black aphids. In their solution, three variants of the YOLO deep learning model were applied, where YOLOv10 had the best mAP50 of 89.1% at 1600×1600 resolution. The training dataset was augmented from 220 to 579 labelled images [5]. Zhao et al. (2022) introduced a complete convolutional neural network model boosted with a parallel attention mechanism (PCSA) for crop pest detection in intricate agricultural settings. The model integrated spatial and channel-wise attention modules in a complete ResNet-50 architecture to enhance feature extraction and recognition efficiency. They built a bespoke dataset of 10 prevalent pest species and applied offline data enhancement to grow the dataset to 26,225 images. The best-performing final model attained an accuracy of 98.17%, surpassing existing models such as DenseNet and VGGNet variations [6].. Li et al. (2023) suggested an AI-based pest detection system based on a hybrid model combining YOLOv5 object detection and EfficientNet classification to detect and classify pests in real-time. The approach stresses field deployment with strong detection even under changing lighting conditions and cluttered backgrounds [7]. Abbas Abbas et al. (2023) introduced a machine vision-based method for detecting pests with Convolutional Neural Networks (CNNs) [8] and morphological analysis. Their system emphasized the detection and classification of two serious pests—Aphis gossypii and Bemisia tabaci—of cotton crops based on high-resolution images and a specially designed Zhang et al. (2022) created an automatic detection model [10] based on AI for identifying more than one pest species on tomatoes with a light neural network appropriate for edge deployment

V. METHODOLOGY

The methodology typically involves on integrating machine learning, edge computing, and a visual alert system to detect pests in real time and inform field workers or farmers.

A. *Data Acquisition:*

Datasets Dat employed are pest image datasets like IP102 and MTIP102, which are labeled images of different species of pests that infest different crops. Images were recorded using embedded cameras set up in the field. These datasets feature high- and low-resolution images across different environmental conditions (e.g., illumination, background noise). Image data has been loaded into the Python environment through libraries like OpenCV and Pandas for processing and annotation. Preprocessing operations are carried out to provide clean, labeled input during training deep learning models.

B. *Data Preprocessing:*

Data preprocessing involved several critical steps to ensure the quality and suitability of the dataset for machine learning and Digital Twin modeling. Initially, missing values were handled by imputing them with appropriate strategies, such as mean imputation for numerical columns like cholesterol (chol) and heart rate (thalach). Next, categorical variables (such as sex and cp) were encoded into numerical values using techniques like one-hot encoding to make them usable by machine learning algorithms. Data pre-processing is done for consistency with cleaning using handling missing values and outliers as well as normalization Features were then scaled using standardization to normalize the input variables and ensure that the models could effectively learn from the data without being biased by the magnitude of certain features.

C. *Feature selection:*

Feature selection played a crucial role, in deep learning models, the choice of features is taken care of automatically through convolutional layers that learn hierarchical and abstract image features from the input images. But Grad-CAM (Gradient- weighted Class Activation Mapping) visualization was employed to confirm the image areas in which the model concentrates during prediction. This gave insight into which features e.g., the pest's body shape, color, wings, or patterns to the classification decision. Knowledge of these characteristics facilitates tuning the architecture of the model and for future incorporation with explainable AI modules.

International Journal of Environmental Sciences
ISSN: 2229-7359
Vol. 11 No. 6s, 2025
https://www.theaspd.com/ijes.php

D. Model Training:
Training was conducted using light-weight, edge- compatible CNN models like MobileNetV2, VGG16, and LeNet because they required low computation but were highly accurate. The data was divided into 80% train and 20% test sets. Transfer learning was utilized wherein pre-trained weights of models trained on ImageNet were fine-tuned using the pest dataset. Training time was greatly minimized with this without a drop in accuracy. The class categorical cross- entropy loss function and the Adam optimizer were utilized. The final training architectures were tuned for batch sizes and learning rates to avoid overfitting and underfitting. The training was implemented using TensorFlow and PyTorch frameworks, and early stopping to avoid degradation in the model's performance.

E. Model Validation and Evaluation:
The learned pest detection models were thoroughly tested to yield high accuracy, stability, and real-time field usability. The evaluation metrics used were accuracy, precision, recall, F1-score, and AUC-ROC to measure the classification performance of the model on unknown pest images. K-fold cross-validation was used to avoid overfitting and model generalization. Apart from these measurements, inference latency and memory consumption were tested on edge devices like Raspberry Pi and Jetson Nano, verifying they could process images within less than 150 milliseconds per frame. Quantized models (e.g., INT8) had high accuracy with reduced computational load, which made them suitable for low-power edge environments. Validation also involved on-field real-time testing under field conditions to evaluate alert responsiveness and dependability, proving that the system was able to effectively detect pests and initiate timely alerts with low false positives.

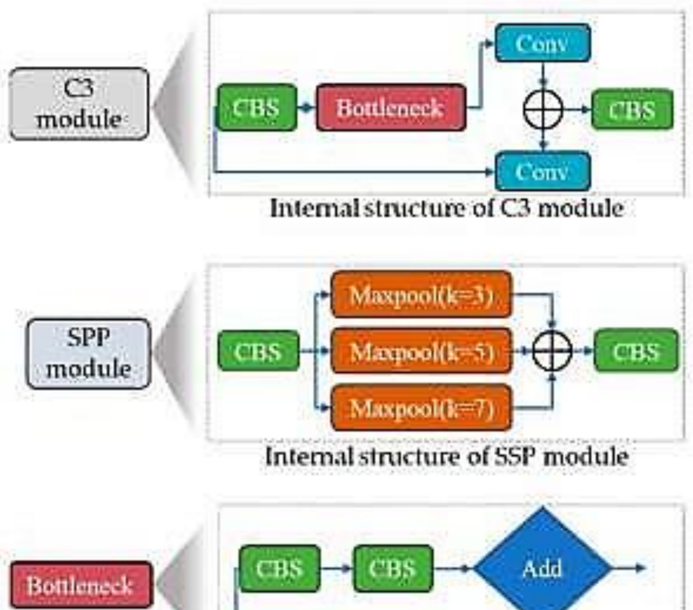

FIGURE 1: Various Module Architecture

F. Edge Deployment:
After model validation and performance testing, the trained pest detection model was made ready for edge deployment on edge computing devices to facilitate real-time, in-field functionality. Edge deployment entails model optimization so that the model could operate effectively with low-power hardware like Raspberry Pi 4, NVIDIA Jetson Nano, or specialized microcontrollers like GAP8/GAP9 with support for CNN accelerators. These products were chosen due to their cost-effectiveness, small size, energy efficiency, and capacity to carry out on-device inference without dependence on cloud connectivity. To make the model smaller and lower the computational burden, the trained deep neural network model (e.g., MobileNetV2) was first quantized to a lightweight model via model quantization, which replaced floating-point.

International Journal of Environmental Sciences
ISSN: 2229-7359
Vol. 11 No. 6s, 2025
https://www.theaspd.com/ijes.php

Quantization was performed using TensorFlow Lite or PyTorch Mobile converters without sacrificing accuracy while dramatically lowering memory consumption and inference time. Post-training quantization was chosen because it is easy to implement and easy to deploy on field hardware. On boards such as the Raspberry Pi, OpenCV and TensorFlow Lite Interpreter were employed for image capture and inference. For NVIDIA Jetson Nano, the TensorRT framework was employed for quicker GPU- accelerated inference.

FIGURE 2: Various samples of pests for each dataset class.

G. Visual Alert Mechanism:

On the detection of a pest, the system instantly triggers a visual alarm system to alert local farmers or agricultural workers. The alert system involves a mix of visual signals (like high-brightness LEDs), audio signals (via piezoelectric buzzers), and wireless alerts transmitted to mobile phones or control centers through communication standards like Wi-Fi, GSM, or low-power wide-area networks (LPWAN) like LoRa. The type of alert can be configured by deployment environment—LEDs and buzzers for small farms with manual monitoring, and mobile notifications for larger, remotely monitored farms. The logic for alerts is built into the firmware of the edge device, which constantly watches the output of the pest detection model. The moment a detection confidence reaches a pre-set criterion (e.g., 90%), an alert is issued in real time. This facilitates real-time intervention through localized pesticide spraying or physical pest control, minimizing wastage of pesticides as well as environmental pollution. The alert system was implemented under diverse lighting and noise conditions to guarantee visibility and audibility in day and night, as well as in noisy environments. Battery-backed operation guaranteed alerts continued to function even during power failures.

VI. CONCLUSIONS

This work introduces a scalable and robust edge-deployable visual alarm and pest detection system for real-time farm monitoring, integrating embedded hardware with light-weight deep learning models to provide a farmer-friendly and scalable solution, especially for resource-scarce environments. The system produces robust classification accuracy with low energy use and high-speed inference on edge devices, and its onboard visual alert mechanism offers real-time, actionable alerts to enable quick intervention and reduce crop loss. With no requirement for cloud infrastructure, it is optimally designed for remote field deployment. Upcoming improvements are to integrate multi-modal inputs like environmental information (e.g., temperature, humidity) to enhance detection resilience under different conditions, and to scale the system with IoT integration and drone-based platforms to improve scalability and coverage of monitoring. In general, the solution presented represents a remarkable leap in data-driven, intelligent crop protection in the field of digital agriculture

REFERENCES

[1] A. Albanese, M. Nardello, and D. Brunelli, "Automated Pest Detection with DNN on the Edge for Precision Agriculture," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 11, no. 3, pp. 510–523, Aug. 2021, doi: 10.1109/JETCAS.2021.3101740.

[2] J. Zhang, Z. Liu, and K. Yu, "MSFNet-CPD: Multi-Scale Cross-Modal Fusion Network for Crop Pest Detection," arXiv preprint arXiv:2505.02441, May 2025.

[3] S. M. Venkateswara and J. Padmanabhan, "Deep learning based agricultural pest monitoring and classification," Scientific Reports, vol. 15, no. 8684, pp. 1–14, Mar. 2025, doi: 10.1038/s41598-025-92659-5.

[4] L. Bompani, L. Crupi, D. Palossi, O. Baldoni, D. Brunelli, F. Conti, M. Rusci, and L. Benini, "Accelerating Image-based Pest Detection on a Heterogeneous Multi-core Microcontroller," IEEE, Aug. 2024.

[5] D. Kapetas, P. Christakakis, S. Faliagka, N. Katsoulas, and E.M. Pechlivani, "AI-Driven Insect Detection, Real-Time Monitoring, and Population Forecasting in Greenhouses," AgriEngineering, vol. 7, no. 29, pp. 1–16, Jan. 2025, doi: 10.3390/agriengineering7020029.

[6] S. Zhao, J. Liu, Z. Bai, C. Hu, and Y. Jin, "Crop Pest Recognition in Real Agricultural Environment Using Convolutional Neural Networks by a Parallel Attention Mechanism," Frontiers in Plant Science, vol. 13, Article 839572, pp. 1–16, Feb. 2022, doi: 10.3389/fpls.2022.839572.

[7] J. Li, H. Wu, M. Yang, and Z. Wang, "AI-powered pest detection system for smart agriculture," IEEE Access, vol. 11, pp. 56945–56956, 2023. https://doi.org/10.1109/ACCESS.2023.3249985

[8] S. Abbas, M. A. Khan, A. Hussain, and T. Alam, "Machine vision-based pest detection using CNN and morphological analysis," Plants, vol. 13, no. 653, pp. 1–16, Mar. 2023. https://doi.org/10.3390/plants13050653

[9] Y. Zhang, L. Huang, R. Zhao, and Q. Liu, "AI-based automatic pest detection for tomato plants using MobileNetV2," Frontiers in Plant Science, vol. 14, Article 1255719, pp. 1–13, 2022.

[10] J. Gonçalves, E. Silva, P. Faria, T. Nogueira, A. Ferreira, C. Carlos, and L. Rosado, "Edge-Compatible Deep Learning Models for Detection of Pest Outbreaks in Viticulture," Agronomy, vol. 12, no. 3052, pp. 1–20, Dec. 2022. https://doi.org/10.3390/agronomy12123052