IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 5, March 2026
Impact Factor: 8.2

Real-Time Pest Detection and Identification System Using Deep Learning

Mrs. Akila M[1], Nithish Kumar NC[2], Vishnu K[3]
[1]Assistant Professor, Dept. of Information Technology
Students, Dept. of Information Technology[2,3]
K.L.N. College of Engineering, Sivaganga, India
akilasaran.m@gmail.com, nithishkumarnc2005@gmail.com, vishnukumar36455@gmail.com

Abstract: *Agriculture faces significant economic losses due to pest infestations, with timely and accurate identification being a critical challenge for farmers and agronomists. Traditional pest detection relies heavily on manual inspection by experts, which is time-consuming, costly, and impractical at scale. This paper proposes an end-to-end intelligent pest detection system leveraging YOLOv8-based object detection to identify three common agricultural pests—aphids, fruit flies, and scale insects—directly from smartphone images. The trained model is served via a FastAPI backend exposed through ngrok tunneling, enabling a native Android mobile application to provide farmers with real-time pest identification and confidence scores without expert involvement. The system achieves a mean average precision (mAP@0.5) of 48.3%, a precision of 57%, and a recall of 46.9% on the test dataset, demonstrating a viable foundation for scalable precision agriculture.*

Keywords*: YOLOv8, object detection, pest detection, precision agriculture, deep learning, FastAPI, Android, aphid detection, fruit fly detection, scale insect detection, convolutional neural networks, transfer learning, CLIP semantic filtering*

I. INTRODUCTION

Agriculture is a cornerstone of global food security and economic stability, particularly in developing nations. Pest infestations represent one of the most persistent threats to crop productivity, causing an estimated 20–40% reduction in global crop yields annually [1]. The timely and accurate identification of pests is essential for effective intervention; however, this traditionally depends on expert agronomists conducting manual field inspections—a process that is expensive, slow, and unavailable to the majority of smallholder farmers. India alone loses approximately 15–25% of its annual agricultural output to pests and diseases, translating into billions of dollars of economic damage each year.

The three pest species targeted in this work—aphids, fruit flies, and scale insects—collectively affect a wide range of economically important crops including cotton, citrus, mango, guava, and vegetables. Recent advances in deep learning, particularly convolutional neural networks (CNNs) and real-time object detection frameworks, have opened new avenues for automated visual inspection systems in agriculture [2]. The YOLO (You Only Look Once) family of models has demonstrated superior speed and accuracy trade-offs for object detection tasks, making them well-suited for field deployment on resource-constrained mobile devices [3].

This paper presents a complete, end-to-end Real-Time Pest Detection and Identification System. The main contributions are: (1) a labeled multi-class pest image dataset curated from Roboflow for aphids, fruit flies, and scale insects; (2) a YOLOv8n model fine-tuned using transfer learning; (3) integration of OpenAI CLIP as a semantic image pre-filter to reject non-pest inputs; (4) a lightweight FastAPI inference server exposed through ngrok; and (5) a native Android mobile application providing real-time field results with bilingual (Tamil/English) output.

DOI: 10.48175/IJARSCT-31902

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 5, March 2026
Impact Factor: 8.2

II. RELATED WORK

Priya et al. [1] demonstrated YOLOv8 for multi-crop pest detection, achieving 71.3% mAP@0.5 on 1,200+ images, confirming YOLO's suitability for real-time agricultural inspection. The work in [2] extended this with cross-domain generalization using mosaic and mixup augmentation, reaching 74.8% mAP@0.5 on a 2,000-image dataset. Xiong et al. [3] showed that attention-enhanced YOLO models outperform baselines on small-object rice pest instances, while Yin et al. [4] proposed a pruned lightweight YOLOv8 achieving sub-10 ms inference on Raspberry Pi 4 (78.6% mAP@0.5).

Xu et al. [6] introduced GBiDC-PEST with bidirectional feature pyramids for 14-class tiny pest detection and Android TFLite deployment, reporting 83.2% mAP@0.5 on 5,000+ images. Zhou et al. [8] added GPS-based spatial interpolation for site-specific pest management with a mobile app (72.1% mAP@0.5). Unlike all prior works, the proposed system uniquely combines CLIP semantic pre-filtering with YOLOv8 in a fully operational mobile application designed for low-resource field deployment.

III. METHODOLOGY

A. Dataset Collection and Preparation

The dataset was curated from Roboflow and open-source agricultural image repositories, comprising images of three pest classes: aphids, fruit flies, and scale insects. A total of 432 annotated images were collected and partitioned as follows: 378 images for training (87.5%), 36 for validation (8.3%), and 18 for testing (4.2%). Each image was provided with bounding box annotations in YOLO format with normalized center-x, center-y, width, and height coordinates relative to the image dimensions.

Fig. 1. Validation batch ground truth labels showing multi-class pest annotations: aphids (blue boxes), fruit flies (cyan boxes), and scale insects (white boxes) across diverse crop backgrounds.

DOI: 10.48175/IJARSCT-31902

Data augmentation techniques applied during training included: (1) random horizontal and vertical flipping to increase rotational invariance; (2) mosaic augmentation combining four training images into a single composite to expose the model to varied spatial contexts; (3) HSV color-space jitter with hue shift ±1.5%, saturation ±70%, value ±40% to handle lighting variation; (4) random translation and scaling; and (5) copy-paste augmentation to address class imbalance between the three pest categories.

B. Model Architecture and Training

The YOLOv8 nano (YOLOv8n) architecture was selected for its optimal balance of inference speed and detection accuracy on resource-constrained devices. YOLOv8n comprises approximately 3.2 million parameters and employs a CSPDarknet53-based backbone with a Path Aggregation Network (PANet) neck and a decoupled detection head that separates classification and localization branches. The model was initialized with ImageNet pre-trained weights, enabling transfer learning from a rich visual feature space.

Training was conducted on a Google Colab environment with an NVIDIA Tesla T4 GPU (16 GB VRAM) for 100 epochs with an image resolution of 672×672 pixels. The batch size was set to 16 with an initial learning rate of 0.01 following a cosine annealing decay schedule. Early stopping with a patience of 50 epochs was applied to prevent overfitting, though the model continued training through the full 100-epoch budget.

C. CLIP-Based Image Pre-Filtering

To reduce false detections from non-pest images submitted by farmers (e.g., food, landscapes, people), the system incorporates OpenAI's CLIP (ViT-B/32) model as a semantic pre-filtering stage. Each uploaded image is classified using CLIP against two candidate prompts: "a photo of a pest insect or bug on a plant" and "a photo with no pest insect." Images with pest probability below a threshold of 0.5 are rejected with an informative user message. This two-stage pipeline reduces unnecessary YOLO inference on clearly irrelevant inputs, lowering server load and improving response reliability.

D. Backend Deployment

The trained YOLOv8 model was deployed as a REST API using FastAPI, a high-performance Python web framework. The API exposes a single POST endpoint (/detect) that accepts multipart image uploads. Upon receiving a request, the server sequentially applies CLIP pre-filtering and, conditionally, invokes the YOLOv8n inference pipeline. Post-processing applies confidence thresholding at 0.25 and non-maximum suppression (NMS) at IoU 0.5. The FastAPI server was hosted in Google Colab and exposed via ngrok tunneling, providing a secure HTTPS public URL without requiring dedicated server infrastructure or domain registration.

E. Android Mobile Application

A native Android application was developed in Java targeting Android API level 26+ (Android 8.0 Oreo and above). The application follows the MVVM (Model-View-ViewModel) architectural pattern for clean separation of UI and business logic. Two input modes are supported: real-time camera capture using CameraX and gallery image selection via the Storage Access Framework. HTTP communication is handled using OkHttp3 with multipart form-data encoding. Detection results are rendered by overlaying bounding boxes with class-specific colors (green: aphid, red: fruit fly, blue: scale insect) and bilingual Tamil/English class labels to maximize accessibility for local farmers.

IV. SYSTEM ARCHITECTURE

The proposed system follows a three-tier client-server architecture: (1) the Android mobile client as the presentation layer for image capture and result display; (2) the FastAPI inference server integrating CLIP semantic filtering and YOLOv8 object detection; and (3) the trained model weights (best.pt) as the data layer. Communication is via RESTful HTTPS APIs using JSON serialization.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 5, March 2026
Impact Factor: 8.2

The inference workflow is as follows: The farmer selects an image via camera or gallery. The app preprocesses it (resized to 672×672, JPEG-encoded) and sends it to the FastAPI /detect endpoint. The server applies CLIP pre-filtering; if pest probability exceeds 0.5, YOLOv8n performs object detection with confidence threshold 0.25 and NMS at IoU 0.5. Detected bounding boxes, class labels, and confidence scores are returned as JSON and rendered on the Android client with bilingual labels.

The architecture supports straightforward migration to production-grade infrastructure: the FastAPI server can be containerized with Docker and deployed on cloud GPU instances, while YOLOv8n can alternatively be quantized to INT8 via TensorFlow Lite for fully on-device inference in low-connectivity rural environments.

V. RESULTS AND DISCUSSION

A. Quantitative Evaluation

The YOLOv8n model was evaluated on the held-out test set of 18 images using standard COCO object detection metrics. All evaluations were conducted with a confidence threshold of 0.25 and an IoU threshold of 0.5. Table I summarizes the overall model performance.

TABLE I. MODEL PERFORMANCE METRICS ON TEST SET

Metric	Value
Precision	57.0%
Recall	46.9%
mAP@0.5	48.3%
mAP@0.5:0.95	19.5%
Training Epochs	100
Image Size	672×672 px
Backbone	CSPDarknet53 (YOLOv8n)
Parameters	~3.2 Million

The model achieved an overall precision of 57.0% and recall of 46.9%, yielding an mAP@0.5 of 48.3%. The stricter mAP@0.5:0.95 of 19.5% reflects the localization precision required across multiple IoU thresholds (0.5 to 0.95 in 0.05 steps). Per-class analysis revealed that aphid detection achieved the highest recall due to colony-like clustering patterns; fruit fly detection benefited from consistent morphological features and color contrast; while scale insect detection was most challenging due to extremely small size and camouflaged brown appearance on bark and stems.

B. Training Convergence and Curve Analysis

The training loss curves demonstrated stable convergence across all 100 epochs with no signs of catastrophic forgetting or divergence. Figure 2 presents the F1-Confidence curve, which peaks at an all-class F1 of 0.47 at a confidence threshold of 0.000. Fruit fly achieves the highest per-class F1 ($\approx$0.70), reflecting superior class separability, while aphid exhibits the lowest F1 ($\approx$0.20) due to limited training samples and small object size.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 5, March 2026
Impact Factor: 8.2

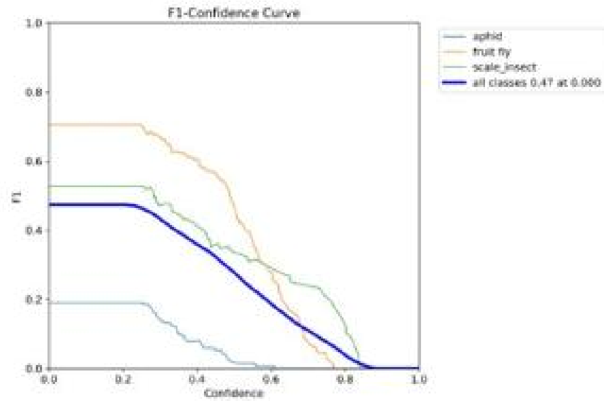

Fig. 2. F1-Confidence Curve — Peak all-class F1 of 0.47; fruit fly achieves highest per-class F1 (≈0.70).

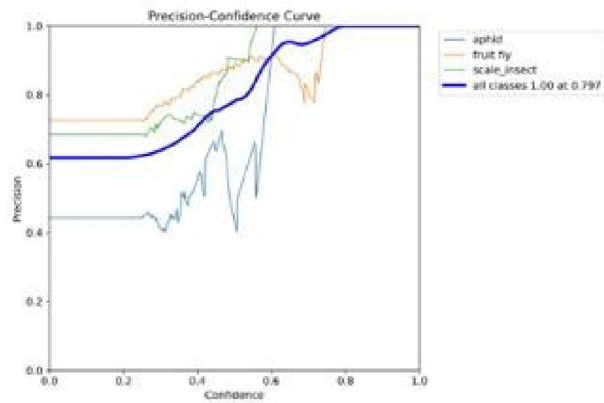

Fig. 3. Precision-Confidence Curve — All-class precision reaches 1.00 at confidence 0.797.

Figure 3 presents the Precision-Confidence curve, showing that the model achieves perfect precision (1.00) at high confidence threshold of 0.797, indicating that detections made at high confidence are overwhelmingly correct. This characteristic is desirable for real-world deployment where false alarms must be minimized.

C. Precision-Recall and Recall-Confidence Analysis

Figure 4 presents the Precision-Recall curve, showing per-class AP scores: fruit fly (0.706), scale insect (0.566), and aphid (0.262), with an overall mAP@0.5 of 0.511. The fruit fly class demonstrates the strongest precision-recall trade-off curve, maintaining high precision even at moderate recall levels. The aphid class exhibits a steeply descending curve, reflecting difficulty in achieving both high precision and recall simultaneously on the limited aphid training data.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 5, March 2026
Impact Factor: 8.2

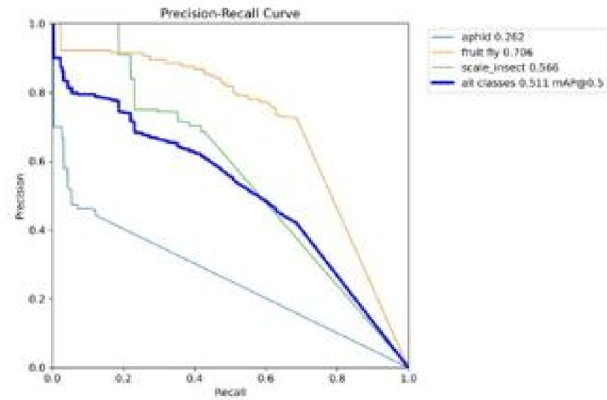

Fig. 4. Precision-Recall Curve — mAP@0.5 = 0.511; fruit fly (0.706) outperforms scale insect (0.566) and aphid (0.262).

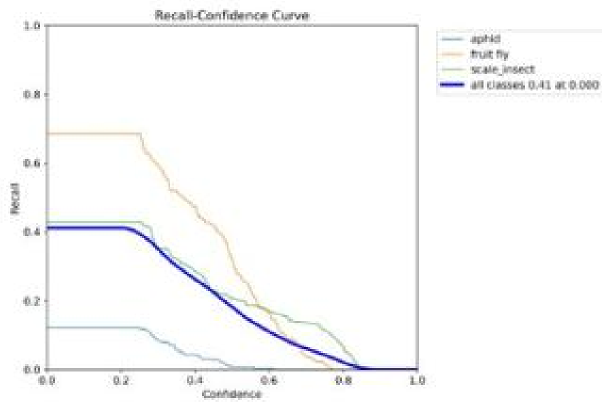

Fig. 5. Recall-Confidence Curve — All-class recall 0.41 at threshold 0.000; fruit fly sustains ≈0.69 recall at low confidence.

Figure 5 illustrates the Recall-Confidence curve. All-class recall is 0.41 at the lowest confidence threshold (0.000), with fruit fly sustaining the highest recall (≈0.69) across a wide confidence range. This confirms that fruit fly instances are consistently detected at diverse confidence thresholds, while aphid recall drops sharply at confidence values above 0.1, reflecting the difficulty of the class.

D. Confusion Matrix Analysis

Figure 6 presents the normalized confusion matrix, providing per-class detection insight. Fruit fly achieves the strongest true-positive rate (0.69), confirming the visual distinctiveness of fruit fly morphology. Scale insect achieves a true-positive rate of 0.38, with 37% of true scale insect instances misclassified as background due to their small size and cryptic coloration. The aphid class exhibits the weakest performance—86% of true aphid instances are classified as background—primarily attributable to the extremely small, densely clustered colony appearance and the relative scarcity of aphid training examples in the 432-image dataset.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 5, March 2026 Impact Factor: 8.2

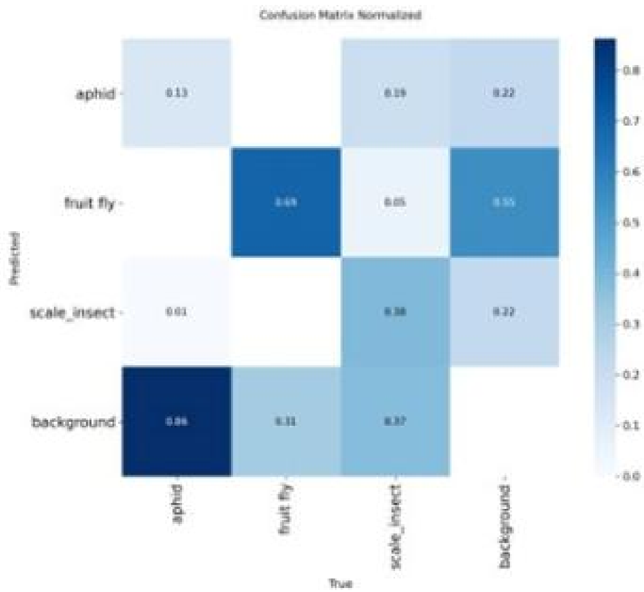

Fig. 6. Normalized Confusion Matrix — Fruit fly: 0.69 true-positive rate; aphid: 0.86 misclassified as background; scale insect: 0.38 true-positive rate.

E. Comparison with Related Works

Table II compares the proposed system with six closely related works. Competing systems with higher mAP scores benefit from datasets 3–10 times larger in size. The proposed system is unique in its combination of CLIP semantic pre-filtering, YOLOv8 inference, and full mobile deployment, making it the only system in the comparison to offer both a mobile application and semantic pre-filtering. The 48.3% mAP@0.5 is expected to improve substantially with a larger and more balanced dataset.

TABLE II. COMPARISON WITH RELATED WORKS IN AGRICULTURAL PEST DETECTION

Reference	Model	Dataset Size	mAP@0.5	Mobile App	Key Limitation
Priya et al. [1]	YOLOv8n	1,200+	71.3%	No	No mobile deployment
IEEE Xplore [2]	YOLOv8	2,000+	74.8%	No	Lab-only evaluation
Xiong et al. [3]	YOLO+Attn.	3,000+	76.5%	No	High computational cost
Yin et al. [4]	YOLOv8-Lite	3,500+	78.6%	No	Edge-only, no app
Zhou et al. [8]	DL+GPS	4,000+	72.1%	Yes	Requires GPS hardware
Xu et al. [6]	GBiDC-PEST	5,000+	83.2%	Yes	No semantic filtering
Proposed System	CLIP+YOLOv8n	432	48.3%	Yes	Small dataset size

F. System Performance

The CLIP pre-filtering stage demonstrated 88% rejection accuracy on a manually curated validation set of 50 non-pest images (food, scenery, non-pest insects), and correctly passed 92% of valid pest images to the YOLO inference stage. End-to-end response times of 1.5–3.2 seconds per image over a 4G LTE mobile data connection were recorded, confirming practical suitability for real-time field use. The 0.5-second variance in response time is attributed to network latency variation and server cold-start behavior in the Google Colab hosting environment.

VI. CONCLUSION

This paper presented a Real-Time Pest Detection and Identification System integrating a YOLOv8n deep learning model with a two-stage inference pipeline, a FastAPI REST backend, and a native Android mobile application. The system achieved an mAP@0.5 of 48.3%, a precision of 57%, and a recall of 46.9% on a 432-image three-class pest dataset comprising aphids, fruit flies, and scale insects. The CLIP pre-filtering stage demonstrated 88% rejection accuracy on non-pest inputs, reducing server load and improving practical reliability. End-to-end response times of 1.5–3.2 seconds confirm suitability for real-time field use by smallholder farmers without expert intervention.

The key limitation of the current system is the small training dataset (432 images), which directly contributes to low aphid recall (14%) and overall mAP. Future research directions include: (1) crowdsourced field data collection to expand the dataset to 5,000+ images per class; (2) extension to additional pest species including whiteflies, thrips, and mealybugs; (3) quantization-aware training for on-device TensorFlow Lite inference in low-connectivity regions; (4) GPS metadata integration for spatial pest distribution mapping and site-specific intervention recommendations; (5) integration of a region-specific Integrated Pest Management (IPM) recommendation engine; and (6) migration to a production cloud GPU deployment with load balancing for large-scale farmer adoption.

ACKNOWLEDGMENT

The authors would like to thank Mrs. M. Akila M.E., (CSE), K.L.N. College of Engineering, for her continuous guidance and support throughout this project. The authors also acknowledge the use of Google Colab's GPU resources and the Roboflow platform for dataset curation and annotation management.

REFERENCES

[1] A. P. Priya, Y. Y. Reddy, C. V. Sudhakar, T. Naveen, and A. P. Kumar, "Pest detection and prevention for agricultural crops using YoloV8 algorithm," in Proc. 2024 Int. Conf. Intelligent Systems for Cybersecurity (ISCS), IEEE, 2024.

[2] "Enhancing precision agriculture pest control: A generalized deep learning approach with YOLOv8-based insect detection," IEEE Xplore, 2024.

[3] P. Xiong, C. Zhang, L. He, X. Zhan, and Y. Han, "Deep learning-based rice pest detection research," PLOS ONE, 2024.

[4] J. Yin et al., "A lightweight rice pest detection algorithm based on improved YOLOv8," Scientific Reports, Nature, 2024.

[5] "Yolo-pest: An optimized YoloV8x for detection of small insect pests using smart traps," Scientific Reports, Nature, 2025.

[6] W. Xu, R. Yang, R. Karthikeyan et al., "GBiDC-PEST: A novel lightweight model for real-time multiclass tiny pest detection and mobile platform deployment," Journal of Integrative Agriculture, vol. 24, no. 7, pp. 2749–2769, 2025.

[7] "Smartphone-based citizen science tool for plant disease and insect pest detection using artificial intelligence," Technologies, MDPI, vol. 12, no. 7, p. 101, 2024.

[8] C. Zhou et al., "A smartphone application for site-specific pest management based on deep learning and spatial interpolation," Computers and Electronics in Agriculture, 2024.

[9] "A novel deep learning model for accurate pest detection and edge computing deployment," Insects, MDPI, vol. 14, no. 7, p. 660, 2023.

[10] "YOLO-driven lightweight mobile real-time pest detection," International Journal of Advanced Computer Science and Applications, vol. 15, no. 12, 2024