Edge-Compatible Hybrid Deep Learning and Boosting Model for Automated Agricultural Pest Identification

B. Maheshwari[1], Uppu Srujana Sri[2], Polsani Saipriya[2], Shaik Imtiyaz[2], Shaik Yakub Pasha[2]

[1]Assistant Professor, [2]UG Student, [1,2]Department of Computer Science and Engineering (Data Science)

[1,2]Vaagdevi College of Engineering (UGC-Autonomous), Bollikunta, Warangal, 506005, Telangana

To Cite this Article

B. Maheshwari, Uppu Srujana Sri, Polsani Saipriya, Shaik Imtiyaz, Shaik Yakub Pasha, "EdgeCompatible Hybrid Deep Learning and Boosting Model for Automated Agricultural Pest Identification", Journal of Science Engineering Technology and Management Science, Vol. 03, Issue 03, March 2026,pp: 222-232, DOI: http://doi.org/10.64771/jsetms.2026.v03.i03.pp222-232

Submitted: 06-02-2026 *Accepted: 13-03-2026* *Published: 20-03-2026*

ABSTRACT

Pest identification is a challenging task in the agricultural sector, as accurate and timely detection of pests is essential for effective pest control and crop protection. Conventional approaches to pest detection, such as entomological knowledge and manual examination, take a lot of time and are prone to human mistakes. The research presents a novel solution designed to automate and enhance pest management in agriculture. The system directly addresses the significant drawbacks of the traditional manual identification method, which is time-consuming, labour-intensive, and reliant on scarce human expertise. Our proposed system provides a fast, accurate, and accessible alternative by leveraging an advanced, hybrid machine learning pipeline integrates deep learning convolutional neural networks (CNN), with Linear Discriminant Analysis (LDA) and Categorical Boosting (CatBoost), here after referred as hybrid neural boosting. The core of the system is a powerful CNN that acts as a sophisticated feature extractor, automatically learning to identify complex visual patterns in insect images. The high-dimensional features from the CNN are then fed into K- nearest neighbours (KNN), eXtreme gradient boosting (XGboost) and proposed LDA with Catboost, a highly accurate boosting algorithm that makes the final classification. The system is designed for practical, real-world application, allowing users to upload a new insect image and receive an instant, on-device prediction. By integrating these robust models, the project not only achieves high classification accuracy but also ensures computational efficiency, making it suitable for deployment on resource-constrained edge devices without a constant internet connection. The result is a proactive, scalable, and user-friendly tool that empowers farmers with the ability to perform timely and targeted pest control, thereby contributing to sustainable agriculture and improved crop yields.

Keywords: Pest identification, Agricultural sector, Deep learning, Machine Learning, Ensemble learning models.

1. INTRODUCTION

In the realm of agriculture, farmers are one of the backbones of our society. By equipping farmers with state-of-the-art technologies, these applications provide them with unprecedented access to timely and actionable insights. It has been proven over the years that insects pose the greatest dangers to crop yields during harvest, as seen by the rapid rise in the use of agricultural pesticides. However, due to evolution and climate change-related factors, a lot of dangerous insects become resistant and pose a greater threat. Therefore, throughout the years, smart insect monitoring systems and insect detection and classification models have been used to tackle this problem by helping farmers and smart IoT-based systems identify and target these insects to protect crops. Traditional methods of insect detection and pest management rely heavily on manual observation or cloud-based classification systems, which often introduce delays, inaccuracies, and privacy concerns. These methods are also impractical in remote or rural areas with limited or unreliable network connectivity, thus making rapid pest identification and timely intervention difficult. Further worsening the problem, climate change and accelerated insect evolution have led to the emergence of resistant insect populations, complicating traditional pesticide-based management strategies. Existing digital insect monitoring systems often leverage heavy computational models which, while accurate, are unsuitable for direct deployment on resource-constrained mobile and edge devices commonly used by farmers.

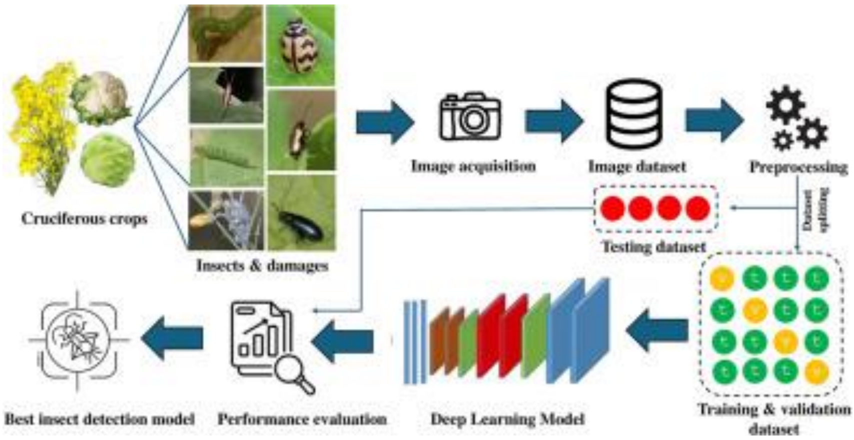

Figure 1. Agricultural insect detection.

Such models either require continuous internet connectivity or are computationally prohibitive for real time on-device analysis, thus failing to deliver the speed, efficiency, and convenience urgently needed in practical agricultural scenarios. Therefore, there is a clear and pressing need for optimized, edge-deployable machine learning models that are capable of accurate and real-time insect classification directly on mobile devices. The existing literature largely focuses on cloud-based or resource-intensive deep learning architectures, neglecting the challenges associated with the lightweight deployment and quantization strategies that are critical for mobile scenarios. Moreover, datasets utilized by many existing studies often either lack specificity relevant to local pest populations or are excessively generalized, resulting in reduced model accuracy and applicability in localized agricultural contexts.

2. LITERATURE SURVEY

The ubiquity of mobile devices has motivated the search for rapidly prototyped and deployed machine learning models adept in running within the environments of these platforms. TensorFlow Lite (TFLite), a new trademark of Google's TensorFlow framework which is lighter and thus can be integrated into mobile applications easily, plays a role as the bridge connecting the development of machine learning models and mobile applications [1]. Developed for mobile and embedded platforms such as smartphones, tablets, and many portable devices, TensorFlow Lite enables developers to implement intelligent capabilities on their mobile applications. There are some interesting applications of TFLite in different domains where the models have been used for a variety of tasks. For instance, ref. [2] introduces an IoT based solution for real-time flash flood detection and alerting using TensorFlow Lite. It addresses the limitations of traditional flood warning methods by emphasizing the need for timely alerts to facilitate an effective public response. The proposed solution employs computer vision techniques with video cameras to monitor water levels. Implemented on low-powered Raspberry Pi devices, the system offers scalability and adaptability for deployment in flood-prone regions. Performance evaluation identifies TensorFlow Lite with an SSD-MobileNet-v2-Quantized model as the optimal configuration for achieving high detection accuracy and efficiency in IoT environments. Some interesting applications have also been seen in image classification. The work in [3] introduces "AgroAId", a mobile app system utilizing deep learning and TensorFlow Lite for the visual classification of plant species and diseases, achieving 99% accuracy and providing spatiotemporal analytics on regional and seasonal disease trends. The work in [4] presents a mobile application for on-edge medical diagnosis of lung diseases using TensorFlow Lite. The study experimented with a total of 18 models, which were created using various quantization techniques, including post-classification quantization, integer quantization, and Quantization-Aware Training. Quantization-Aware Training resulted in a 75.59% reduction in model size, with MobileNetV2 offering the best performance-to-size ratio, showing only a 4.1% accuracy loss.

Additionally, ref. [5] explores the use of vision transformers (ViTs) for on-edge med ical diagnostics using the Kvasir-Capsule image classification dataset of gastrointestinal diseases. The study applied TensorFlow Lite quantization techniques, such as post-training f loat-16 (F16) quantization and Quantization-Aware Training (QAT), to reduce model sizes while maintaining performance. The study concluded that MobileViT_V2_175, with its F16 quantization and 27.47 MB size, offered the best balance between performance and efficiency. Moreover, ref. [6] proposed "Leboh", an Android mobile application utilizing TFLite's EfficientNet-Lite model for waste classification, achieving an accuracy of 95.39% during model evaluation and 82.5% during user testing. Likewise, ref. [7] had an interesting idea and dataset, as they introduced a mobile application for oil palm fresh fruit ripeness classification, achieving a high test accuracy of 89.3%, with a 96 ms inference time per image using EfficientNetB0 optimized through transfer learning, 9-angle crop data augmentation, and float16 quantization. Furthermore, with the advancements in the fields of computer vision, object detection applications have made their way into the TinyML domain, and have been implemented using TFLite and MobileNetv2. For example, ref. [8] presents an object detection and classification system for urban actors, using TFLite with a re-trained Single Shot Detector (SSD) model. Another interesting use case was [9], which introduces a mobile-based application for traffic signs recognition, leveraging TFLite and transfer learning techniques to train a Single Shot MultiBox Detector (SSD) MobileNet V2 model, with the quantized model demonstrating four times faster detection compared to the original float model. Many of the existing workers performed experiments on their personal developed datasets. Ref. [10] developed a dataset of 225 images, representing nine common orders and sub-orders of insect species, with 25 specimen images in each. They utilized artificial neural networks (ANNs) and support vector machine (SVM) algorithms, reaching an accuracy of 93% with the SVM model.

Similarly, ref. [11] developed their own dataset, consisting of 60 samples of 24 common pest species found in-field, resulting in a 1440 image dataset. To enhance the classification accuracy in field crop insects, they developed recognition systems utilizing techniques such as multiple-task sparse representation and multiple-kernel learning (MKL). The work in [12] uses both of the datasets consisting of nine and twenty-four classes, respectively. Their methodology involved employing various machine learning techniques, such as artificial neural networks (ANN), support vector machine (SVM), k-nearest neighbors (KNN), naive Bayes (NB), and convolutional neural network (CNN) models. The study also proposed an insect pest detection algorithm involving foreground extraction and contour identification, contributing to the classification of in sects across complex backgrounds. The evaluation of classification models was enhanced through nine-fold cross-validation, resulting in the highest classification rates of 91.5% and 90% for the nine and the twenty-four class insects, respectively, achieved using the CNN model. Likewise, ref. [13] also utilized their own dataset, which comprised images of twenty classes of paddy field insect pests sourced from Google Images and photographs taken by the Faculty of Agriculture, University of Jaffna, Sri Lanka. A framework was developed to classify the images of paddy field insect pests using gradient-based features through the bag-of-words approach. The classification process involved several steps, including the identification of regions of interest and their representation as scale-invariant feature transform (SIFT) or speeded-up robust features (SURF) descriptors. Subsequently, codebooks were constructed to map these descriptors into fixed-length vectors in the histogram space. The feature histograms were then subjected to multi-class classification using support vector machines (SVMs). Notably, the combination of the Histogram of Oriented Gradients (HOG) descriptors with SURF features yielded approximately 90% accuracy in classification. The work in [14] demonstrated the use of convolutional neural networks (CNNs) to classify economically important insect species, such as the Mediterranean fruit fly (Ceratitis capitata) and the olive fruit fly (Bactrocera oleae), in real-time, even when in sects are freely moving and changing postures. Wen et al. [15] identified six different orchard insects, with the help of local feature extraction. Six different classifiers with cross-fold validation is used for classification and achieved maximum accuracy of 89.5% using Nearest Mean Classifier (NMC) and 88.4% using Support Vector Machine (SVM).

3. PROPOSED SYSTEM

The proposed system for agricultural insect classification is a modern, automated solution designed to be more efficient and accurate than traditional methods. It uses a user-friendly graphical interface to guide the user through the process. The core of the system is a machine learning pipeline that automatically handles data preprocessing, model training, and prediction. The system allows for the evaluation of various machine learning models, including a sophisticated hybrid approach that uses a CNN for intelligent feature extraction and a CatBoost classifier for accurate classification. This approach is "edge-optimized" as it enables rapid, on-device predictions, eliminating the need for a constant internet connection and expert human intervention. This makes the technology accessible to a wider audience, particularly farmers in remote areas, for effective and timely pest management.

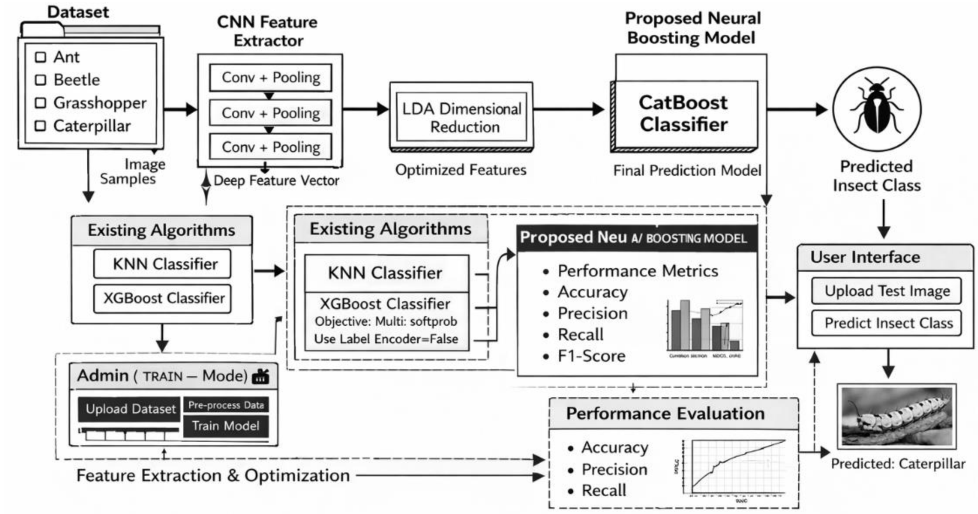

Figure 2. Proposed system architecture.

The initial phase of the proposed system focuses on the user-driven process of providing the dataset. Unlike the traditional method of physically collecting specimens, the user simply has to organize their image dataset with a clear folder structure, where each subfolder is labeled with the name of an insect class. The user then initiates the "Upload Dataset" function within the application's graphical user interface. The system accesses the designated dataset folder path and automatically recognizes the different insect categories based on the folder names. This step is a passive process where the system acknowledges the presence and structure of the dataset without performing any immediate modifications to the images themselves. The proposed system's core lies in its multi-model approach, allowing for a comparison of different machine learning paradigms. It includes two traditional machine learning classifiers: a pre-trained K-Nearest Neighbors (KNN) model and a pre-trained XGBoost classifier. These models are evaluated on the flattened pixel data. For the primary, hybrid deep learning architecture, a pre-trained CatBoost classifier with LDA also called as Neural Boosting model is used to make the final prediction on the features that were extracted by the CNN. For each model, the system performs a rigorous evaluation on the held-out test set. It generates a comprehensive set of metrics, including accuracy, precision, recall, and F1-score, along with a detailed classification report. A confusion matrix is also visualized to show a per-class breakdown of correct and incorrect predictions, while an ROC curve is plotted to assess the model's ability to discriminate between classes.

4. RESULT ANALYSIS

Figure 3 shows upload dataset output screen that displays the system's successful detection of the dataset directory and dynamically extracts all insect class folders present within the selected *farm_insects* dataset. Once the user uploads the dataset through the Tkinter interface, the backend script scans the root folder, identifies each subdirectory as a distinct insect category, and prints the recognized classes inside the GUI's output console. This log confirms that the system has correctly parsed categories such as Africanized Honey Bees, Stink Bugs, Cabbage Loopers, Potato Beetles, Corn Borers, Earworms, Fruit Flies, Spider Mites, and Thrips, ensuring the dataset is structured properly for preprocessing and feature extraction.

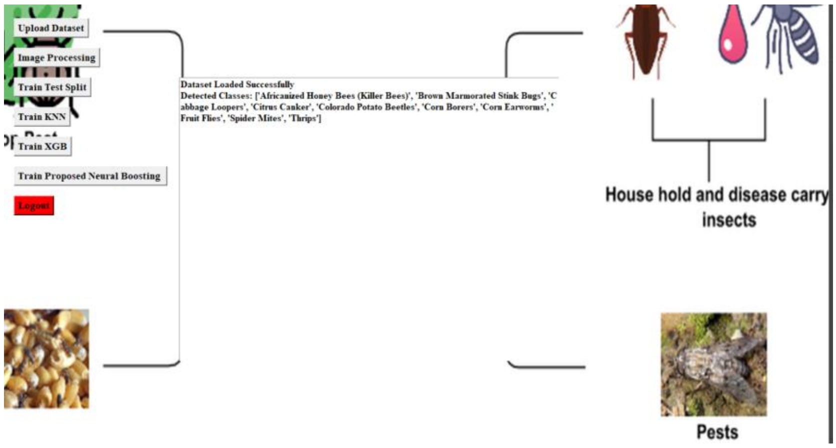

Figure 3. GUI interface displaying dataset upload and visualization for insect classification after admin signup and login.

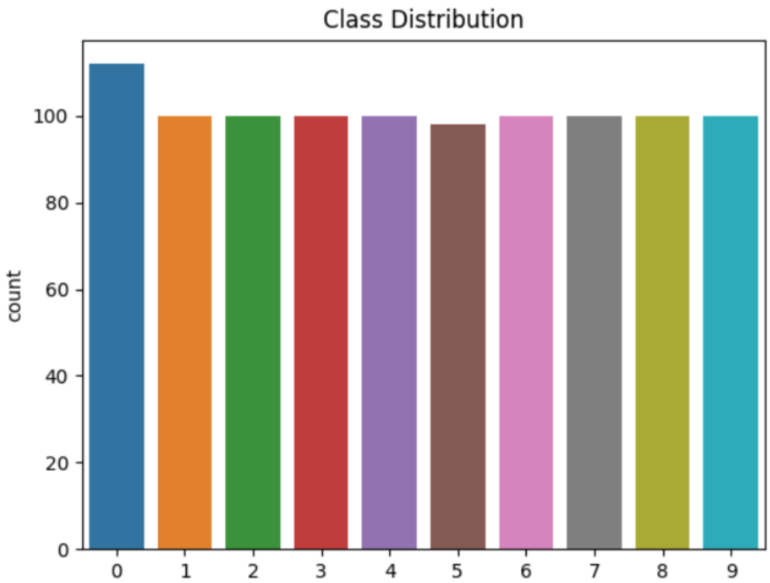

Figure 4. Count plot for the insect categories.

Figure 4 shows class distribution countplot that visualizes the number of images available in each insect category after dataset loading, allowing users to quickly assess dataset balance before training. Each bar corresponds to one of the ten insect classes, and the nearly uniform height of the bars indicates that every class contains approximately 100 samples, ensuring that the dataset is balanced and reducing the risk of model bias during classification. This balanced distribution is essential for reliable training of CNN-based feature extractors and classical ML classifiers such as KNN, XGBoost, and Neural boosting, as it guarantees equal representation during learning. The plot is automatically generated using Seaborn during preprocessing and serves as a useful diagnostic to validate dataset consistency.

Figure 5 shows Neural boosting confusion matrix demonstrates outstanding classification accuracy, with nearly all predictions concentrated along the diagonal and very few misclassifications. This strong diagonal dominance reflects the effectiveness of the hybrid pipeline, where CNN-derived visual features combined with LDA's dimensionality reduction enable CatBoost to achieve highly discriminative decisions across all ten insect species. The model consistently recognizes challenging classes such as Corn Borers, Fruit Flies, and Spider Mites with minimal error, indicating superior generalization compared to standalone classical models. Overall, the matrix highlights the robustness and precision of the proposed Neural boosting architecture, making it highly suitable for real-world agricultural insect detection and deployment in field conditions.

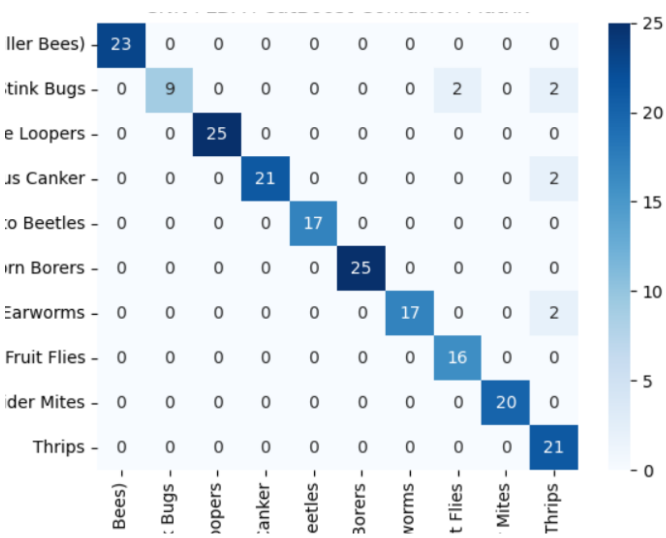

Figure 5. Confusion matrix obtained using proposed hybrid neural boosting model.

Figure 6 shows Neural boosting ROC curve demonstrates exceptional discriminative performance, with most insect classes achieving AUC values close to 1.00 and their curves rising almost vertically toward the top-left corner. This behavior indicates that the hybrid pipeline produces highly separable feature representations, enabling CatBoost to achieve near-perfect classification with minimal false positives. Even challenging categories such as Citrus Canker and Corn Earworms show AUC values above 0.95, highlighting the model's robustness and strong generalization. Overall, the ROC plot confirms that the Neural boosting architecture significantly outperforms standalone classical models, providing highly reliable insect identification suitable for real-world agricultural deployment. Figure 7 shows prediction output illustrates the final stage of the hybrid Neural boosting pipeline, where a test image is processed through the CNN feature extractor and being classified by the neural boosting model. The displayed result shows that the system confidently identifies the insect in the uploaded image as Brown Marmorated Stink Bug*s*, demonstrating the model's ability to accurately distinguish fine-grained visual characteristics such as body texture, shape, and coloration. The predicted label is overlaid on the image using Matplotlib, providing an intuitive visual confirmation of the classification result. This output validates the effectiveness of the hybrid architecture in delivering precise, real-time insect identification suitable for agricultural field applications.

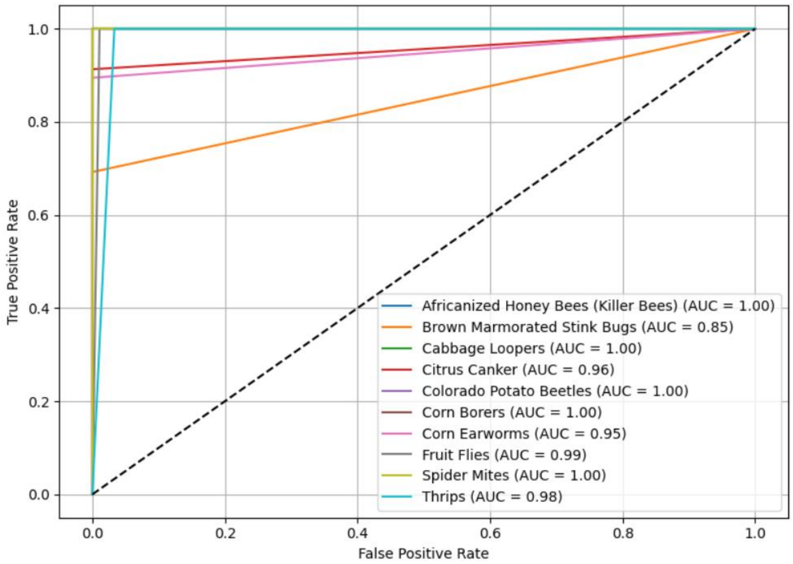

Figure 6. ROC Curve obtained using proposed hybrid neural boosting model

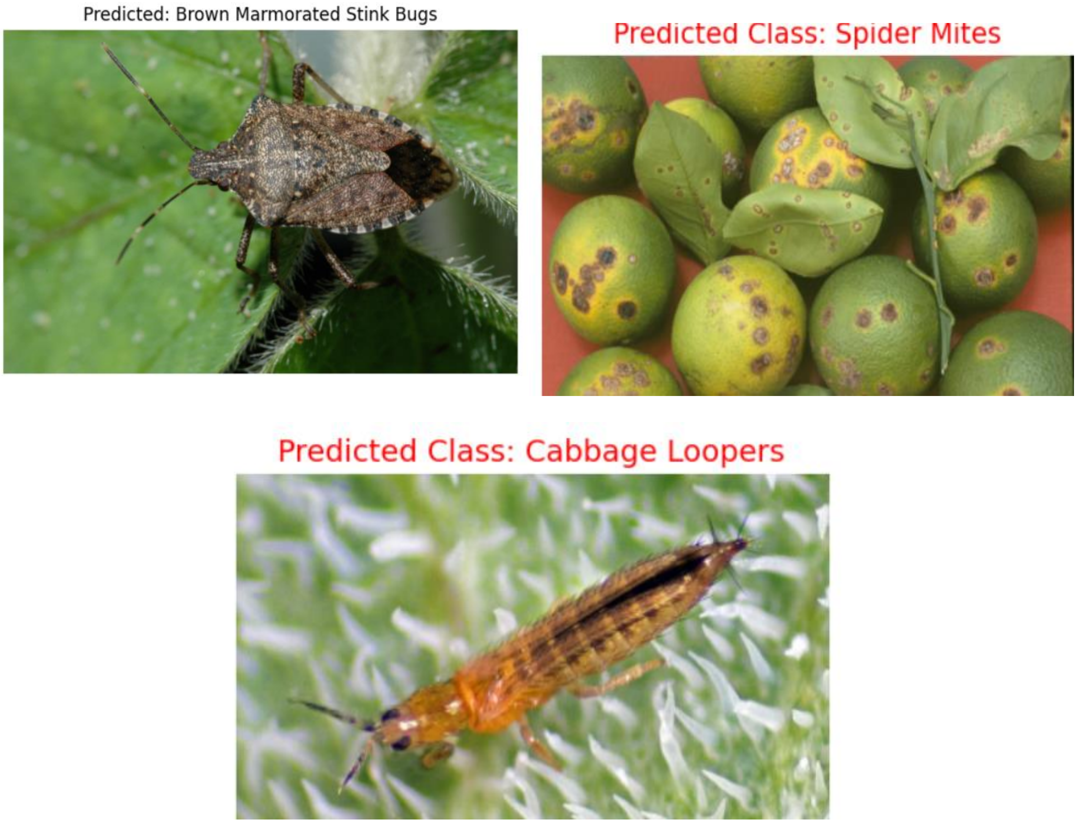

Figure 7. Prediction on test images using proposed neural boosting model.

5. CONCLUSION

The proposed work presents an intelligent system for automatic classification of agricultural insects by integrating deep learning feature extraction with machine learning decision models. Instead of relying on a conventional end-to-end neural network, the system utilized a Convolutional Neural Network as a visual feature extractor to learn complex insect appearance characteristics such as texture, color patterns, and body structure. These learned representations were then supplied to multiple classifiers including KNN and XGBoost for baseline comparison, while the proposed Neural Boosting model further enhanced the deep features and performed final classification, demonstrating improved stability and discrimination capability. The system incorporated dataset preprocessing, feature generation, model training, performance evaluation, and real-time prediction within a user-friendly graphical interface supported by secure role-based authentication. Experimental evaluation using standard metrics such as accuracy, precision, recall, and F1-score confirmed that combining deep feature learning with boosting-based classification produces more reliable results than using traditional machine learning or standalone deep learning approaches. In addition, the software design allows trained models to be reused without retraining and enables non-technical users to perform insect identification through simple image upload, making it practical for real agricultural environments. The research demonstrates that a hybrid deep learning–machine learning architecture can effectively support smart farming by enabling early pest identification, reducing manual inspection effort, and promoting data-driven crop protection strategies, while also providing a scalable framework that can be extended to larger datasets and additional crop monitoring applications.

REFERENCES

[1]. Tensorflow. Tensorflow Lite. Available online: https://www.tensorflow.org/lite/ (accessed on 29 April 2024).
[2]. Rashid, A.A.; Ariffin, M.A.M.; Kasiran, Z. IoT-Based Flash Flood Detection and Alert Using TensorFlow. In Proceedings of the 2021 11th IEEE International Conference on Control System, Computing and Engineering (ICCSCE), Penang, Malaysia, 27–28 August 2021; pp. 80–85.
[3]. Reda, M.; Suwwan, R.; Alkafri, S.; Rashed, Y.; Shanableh, T. AgroAId: A Mobile App System for Visual Classification of Plant Species and Diseases Using Deep Learning and TensorFlow Lite. Informatics 2022, 9, 55
[4]. Aldamani, R.; Abuhani, D.A.; Shanableh, T. LungVision: X-ray Imagery Classification for On-Edge Diagnosis Applications. Algorithms 2024, 17, 280.
[5]. Varam, D.; Khalil, L.; Shanableh, T. On-Edge Deployment of Vision Transformers for Medical Diagnostics Using the Kvasir Capsule Dataset. Appl. Sci. 2024, 14, 8115.
[6]. Handhayani, T.; Hendryli, J. Leboh: An Android Mobile Application for Waste Classification Using TensorFlow Lite. In Intelligent Systems and Applications; Arai, K., Ed.; Springer: Cham, Switzerland, 2023; pp. 53–67.
[7]. Suharjito; Elwirehardja, G.N.; Prayoga, J.S. Oil palm fresh fruit bunch ripeness classification on mobile devices using deep learning approaches. Comput. Electron. Agric. 2021, 188, 106359.
[8]. Campoverde, A.; Barros, G. Detection and Classification of Urban Actors Through TensorFlow with an Android Device. In Information and Communication Technologies of Ecuador (TIC.EC); Efrain Fosenca, C., Morales, G.R., Cordero, M.O., Botto-Tobar, M., Martínez, E.C., León, A.P., Eds.; Springer International Publishing: Cham, Switzerland, 2020; pp. 167–181.
[9]. Benhamida, A.; Varkonyi-Koczy, A.R.; Kozlovszky, M. Traffic Signs Recognition in a mobile-based application using TensorFlow and Transfer Learning technics. In Proceedings of the 2020 IEEE 15th International Conference of System of Systems Engineering (SoSE), Budapest, Hungary, 2–4 June 2020; pp. 537–542.
[10]. Wang,J.; Lin, C.; Ji, L.; Liang, A. A new automatic identification system of insect images at the order level. Knowl.-Based Syst. 2022, 33, 102–110.

[11]. Xie, C.; Zhang, J.; Li, R.; Li, J.; Hong, P.; Xia, J.; Chen, P. Automatic classification for field crop insects via multiple-task sparse representation and multiple-kernel learning. Comput. Electron. Agric. 2021, 119, 123–132.

[12]. Kasinathan, T.; Singaraju, D.; Uyyala, S.R. Insect classification and detection in field crops using modern machine learning techniques. Inf. Process. Agric. 2021, 8, 446–457.

[13]. Venugoban, K.; Ramanan, A. Image Classification of Paddy Field Insect Pests Using Gradient-Based Features. Int. J. Mach. Learn. 2020, 4, 1–5.

[14]. Zhu,L.-Q.; Ma, M.-Y.; Zhang, Z.; Zhang, P.-Y.; Wu, W.; Wang, D.-D.; Zhang, D.-X.; Wang, X.; Wang, H.-Y. Hybrid deep learning for automated lepidopteran insect image classification. Orient Insects 2020, 51, 79–91.

[15]. **.** C. Wen, D. E. Guyer, and W. Li, "Local feature-based identification and classification for orchard insects," *Biosyst. Eng.*, vol. 104, no. 3, pp. 299–307, Nov. 2019.

[16]. Mahesh Ganji. (2025). Enhancing Oracle Cloud HR Reporting Through AI-Driven Automation. Journal of Science & Technology, 10(6), 28–36. https://doi.org/10.46243/jst.2025.v10.i06.pp28-36

[17]. Todupunuri, A. (2025). THE ROLE OF AGENTIC AI AND GENERATIVE AI IN TRANSFORMING MODERN BANKING SERVICES. American Journal of AI Cyber Computing Management, 5(3), 85–93. https://doi.org/10.64751/ajaccm.2025.v5.n3.pp85-93

[18]. Todupunuri, A. . (2024). Artificial Intelligence Ethics: Investigating Ethical Frameworks, Bias Mitigation, and Transparency in AI Systems to Ensure Responsible Deployment and Use of AI Technologies. International Journal of Innovative Research in Science,Engineering and Technology, 13(09), 1–14. https://doi.org/10.15680/ijirset.2024.1309002

[19]. Sushma Babburi. (2025). Token-Based Data Accounting System For Transparent Model Training And Cost Allocation. American Journal of AI Cyber Computing Management, 5(4), 463–474. https://doi.org/10.64751/ajaccm.2025.v5.n4.pp463-474

[20]. Snigdha Gaddam. (2025). SOFTWARE STACK PREPARED FOR AI TRANSITIONING FROM MODULES TO MODELS. American Journal of AI Cyber Computing Management, 5(4), 451–462. https://doi.org/10.64751/ajaccm.2025.v5.n4.pp451-462

[21]. Gaddam, S. INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING.

[22]. Bajarang Bhagwat, V. (2023). Optimizing Payroll to General Ledger Reconciliation: Identifying Discrepancies and Enhancing Financial Accuracy. JOURNAL OF ADVANCE AND FUTURE RESEARCH, 1(4). https://doi.org/10.56975/jaafr.v1i4.501636

[23]. Srinivasa Kalyan Immadi. (2025). Harnessing Artificial Intelligence In Oracle Hcm: Revolutionising Workforce Management With Automation And Predictive Analytics. International Journal of Data Science and IoT Management System, 4(4), 7–13. https://doi.org/10.64751/ijdim.2025.v4.n4.pp7-13

[24]. S. M. K. P. (2025). Cryptography in iOS: A Study of Secure Data Storage and Communication Techniques. International Journal on Science and Technology, 16(1). https://doi.org/10.71097/ijsat.v16.i1.1403

[25]. Suhasnadh Reddy Veluru, Sai Teja Erukude, and Viswa Chaitanya Marella. 2025. Multimodal Detection of Fake Reviews using BERT and ResNet-50. In 2025 4th International Conference on Innovative Mechanisms for Industry Applications (ICIMIA). IEEE, 877–882.

[26]. Cyril, H. P. (2025). Event-Driven Provisioning Architectures For Modern Telecom Networks: Overcoming Legacy Limitations And Enabling Autonomous 6g Operations. International Journal of Advanced Research in Computer Science, 16(6), 75–82. https://doi.org/10.26483/ijarcs.v16i6.7389

[27]. Jay Bharat Mehta. (2025). AUTONOMOUS PATCH VALIDATION FOR ZERO-DAY EXPLOITS IN ENTERPRISE CLOUDS. International Journal of Applied Mathematics, 38(4s), 1270–1285. https://doi.org/10.12732/ijam.v38i4s.685

[28]. Reddy, S. K. (2025). Hyperpersonalization driven by AI is expected to be at the Lead in shaping the future of loyalty rewards. Journal of Emerging Technologies and Innovative Research.

[29]. Reddy, S. K. R. (2021). Strengthening the Security of Loyalty Reward Systems: An In-Depth Analysis of Emerging Cyber Threats and Protection Mechanisms. Journal of Computational Analysis and Applications, 29(6).

[30]. Poojari, R. (2026). Privacy-Preserving Generative AI in Healthcare Systems Using Federated Learning Approaches. International Journal of Data Science and IoT Management System, 5(1), 78-88.

[31]. Uday Kumar Kalae. (2025). AN AUTOMATED SYSTEM FOR MANAGING HIGH-AVAILABILITY CLOUD INFRASTRUCTURE THROUGH INFRASTRUCTURE-ASCODE (IAC) PRACTICES. American Journal of AI Cyber Computing Management, 5(2), 42–50. https://doi.org/10.64751/ajaccm.2025.v5.n2.pp42-50

[32]. Saikumar, B. (2024). Optimizing Crew Scheduling and Absence Management using Microservices: Enhancing Reliability and Efficiency in Crew Management Systems. International Journal of Enhanced Research in Management & Computer Applications, 13(11), 50–55. https://doi.org/10.55948/ijermca.2024.0116

[33]. Saikumar, B. (2023). Enhancing Client Engagement through AI-Driven Real-Time Reporting and Automated Alerts. International Journal of Enhanced Research in Science, Technology & Engineering, 12(11), 111–117. https://doi.org/10.55948/ijerste.2023.1115