Comprehensive Analysis of YOLO Models for Deployment in Precision Agriculture

Balaji Ganesh Rajagopal*, Deebalakshmi Ramalingam*,
Yoga Vignesh*, Rayean Patric*, Dharun Raagav*
*School of Computing,
SRM Institute of Science and Technology Tiruchirappalli, 621105, Tamil Nadu, India

Abstract

This research explores the optimization and deployment of YOLO (You Only Look Once) -based object detection models for real-time pest detection in agricultural environments. Four YOLO variants—YOLOv8, YOLOv9, YOLOv10, and YOLOv11—were evaluated for their performance across metrics such as precision, recall, F1-score, accuracy, and mean Average Precision (mAP@0.5). The study utilized the NBAIR dataset, encompassing 40 pest species, and applied advanced data augmentation techniques to enhance model robustness. Among the models, YOLOv9 achieved the best overall performance with 93% accuracy, 0.959 mAP@0.5, and a 0.96 F1-score, making it suitable for real-time agricultural applications. YOLOv11 demonstrated the highest precision (0.932), while YOLOv10 provided efficient latency and competitive detection capabilities, particularly on mobile devices. Although YOLOv8 underperformed in comparison, its optimization potential is noted. The findings underscore the importance of lightweight, efficient, and accurate AI models in sustainable pest management, reducing pesticide reliance, and enabling data-driven decisions in precision agriculture.

Key Words: Pest Detection, YOLO (You Only Look Once), Precision Agriculture, Mean Average Precision, Edge computing

1. Introduction

Agriculture is essential for maintaining human life, but it encounters major challenges from pest infestations, leading to considerable losses in global crop yield. Presently, pest detection techniques in agriculture mainly depend on manual identification, a process that is not only labour-intensive but also vulnerable to human error. Automated pest detection systems present a solution to these challenges, providing a quicker and more dependable method for monitoring pest populations. This project seeks to address the existing research gap by utilizing advanced object detection models to transform pest detection methodologies(Singh et al., 2024).

This project is driven by the critical necessity to enhance agricultural efficiency and reduce crop losses. On time and precise identification of pests minimizes dependence on widespread pesticide use, allowing farmers to focus on particular threats, thus enhancing resource efficiency and promoting environmental sustainability. This project seeks to enhance pest management strategies through the integration of advanced machine learning models, focusing on precision and automation in pest detection.

The idea for this project significantly improves sustainable agriculture through the encouragement of environmentally conscious farming methods. Precise identification of pests minimizes reliance on chemical pesticides, which often negatively impact non-target organisms and contribute to soil and water contamination. Moreover, prompt identification reduces crop loss, enhances food security, and decreases waste.

Artificial intelligence (AI) has emerged as an essential component of contemporary agricultural methods, due to its capacity to analyze extensive datasets and produce practical insights. AI applications have revolutionized traditional farming, ranging from precision irrigation to automated harvesting. Object detection, a specialized area within artificial intelligence, is essential for recognizing and monitoring pests, diseases, and various elements that affect crop health.

YOLO (You Only Look Once)(Sapkota et al., 2024) was selected for this project because of its remarkable speed and precision in object detection tasks. In contrast to conventional models that rely on region-based detections, YOLO analyzes the entire image simultaneously, resulting in outstanding performance for real-time applications. This study analyzes various iterations of YOLO (v8 (Rizk & Bayad, 2023), v9, v10, and v11) to assess the development of the model and its effectiveness in pest detection (Bhatnagar et al., 2023).

Analyzing the results from different (Sapkota et al., 2024)YOLO versions is crucial for determining the most efficient model for this application. Aspects such as inference speed, memory efficiency, model compatibility, and detection accuracy play a critical part in the deployment of systems in practical agricultural contexts, particularly in environments with limited resources.

The selected dataset for this study includes a wide array of pest images, carefully gathered to ensure accurate representation from multiple pest species and habitats. The variety present strengthens the model's capacity to generalize and operate consistently across various agricultural contexts.

Jetson devices (Swaminathan et al., 2024)are essential in the Internet of Things (IoT) for agriculture, because of their compact design and impressive computational capabilities. Utilizing optimized models such as TensorRT-converted YOLOv8 and YOLOv9 on Jetson devices facilitates real-time pest detection in the field, positioning them as(Pham et al., 2023) excellent options for IoT-based solutions.

This project aligns with the principles of sustainable agriculture through the integration of advanced AI models and IoT technologies. This initiative provides farmers with the necessary tools to make up-to-date data-driven choices, minimizes environmental consequences, and plays a vital role in developing a more resilient food system.

2. Related Works

Several studies(Rane et al., 2024)(Huo et al., 2024)(Piancharoenwong & Badir, 2024) have explored different approaches for identifying pests and diagnosing plant diseases in the field of agriculture. Conventional methods frequently utilized image processing techniques for the analysis of visual symptoms, yet they were limited by their dependence on manually crafted features. Support Vector Machines (SVMs) and k-Nearest Neighbors (k-NNs) are among the machine learning algorithms that have been utilized, often alongside feature extraction techniques. Although these methods enhanced accuracy relative to conventional techniques, (Deng et al., 2023)their dependence on manual feature engineering limited scalability and adaptability.

Deep learning, especially Convolutional Neural Networks (CNNs), has become a more efficient approach. Convolutional neural networks have the capability to autonomously extract features from unprocessed image data, which enhances their effectiveness in applications like pest and disease detection. The application of transfer learning, involving the fine-tuning of pre-trained models for targeted agricultural tasks, has significantly improved the efficacy of deep learning systems. Moreover, deep CNNs have demonstrated encouraging outcomes in the classification of microscopic images, broadening their use in accurate identification tasks within plant pathology.

Object detection algorithms based on YOLO (Raja Gopal & Prabhakar, 2024) have been investigated for use in agriculture, particularly in areas such as pest detection and the identification of wheat spikes. These studies emphasize YOLO's capacity to provide real-time performance, even under demanding field conditions. Optimizing models such as YOLOv5 for deployment on devices with limited resources has shown considerable decreases in inference time while maintaining a slight reduction in accuracy. These advancements closely align with our objectives, highlighting the importance of efficient and practical solutions for real-time pest detection in sustainable agriculture. Object detection algorithms based on YOLO (Raja Gopal & Prabhakar, 2024)have been investigated for use in agriculture, particularly in areas such as pest detection and the identification of wheat spikes. These studies emphasize YOLO's capacity to provide real-time performance, even under demanding field conditions. Optimizing models such as YOLOv5 for deployment on devices with limited resources has shown considerable decreases in inference time while maintaining a slight reduction in accuracy. These advancements closely(Donapati et al., 2023) align with our objectives, highlighting the importance of efficient and practical solutions for real-time pest detection in sustainable agriculture.

The findings from these studies illustrate the transition from conventional techniques to those based on deep learning, emphasizing the effectiveness of CNNs and YOLO models in the areas of pest detection and plant disease classification. (Li et al., 2023) Using these advancements, our study aims to enhance performance for real-time agricultural applications.

The major contributions and limitations of significant related works relevant to this study are summarized in Table 1, offering a comprehensive overview of existing methods in the context of pest detection.

Table 1 Key Contributions and Key Limitations of the Significant Related Works

Related Study	Major Contribution	Major Limitation
(Yang et al., 2023)	Implemented DPAG and FEM to enhance accuracy, substituted standard convolution with DSConv for improved speed, and utilized a variety of tomato datasets.	There are challenges related to missed detections for obscured tomatoes, constraints in mobile deployment testing, and the necessity to optimize the trade-offs related to DSConv.
(Shang et al., 2024)	Investigated sustainable pest management approaches, focusing on secondary metabolites, phytohormones, biocontrol agents, and insect pheromones.	Issues related to the expense of phytohormone treatments, the intricacies of biocontrol programs, the application of pheromone-based techniques, and the delivery and stability of RNAi.
(S. Guan et al., 2024)	Improved YOLOv10 using BiFPN for multi-scale feature integration, SEAM for attention enhancement, and GCNet for global context, resulting in superior detection of wheat spikes.	Future investigations will focus on thermal infrared imagery, advancements in 3D technology, the development of lightweight versions suitable for devices with constrained computing capabilities, and their incorporation into smart agricultural machinery.
(Mishra et al., 2024)	Advancements in the management of storage pests encompass nano-pesticides, bio-pesticides, integrated pest management strategies, and genetic control methods.	There are obstacles related to awareness, financial limitations, and the necessity for tailored strategies that address the diverse climatic conditions across India.
(H. Guan et al., 2023)	A reduced deep learning model designed for precise detection of plant diseases and pests, integrating ResNet with EfficientNetV2.	Challenges arise with complex backgrounds and restricted samples, highlighting the necessity for enhancements in robustness and generalization. Future optimization is essential for varying environmental conditions.
(Nnadozie et al., 2024)	A streamlined version of YOLOv5 has been developed for real-time crop monitoring. By eliminating certain detection scales, the model size has been reduced, enhancing speed and leading to quicker detection while maintaining minimal accuracy loss.	There is a trade-off between speed and accuracy, which may lead to reduced accuracy for object sizes that fall within discarded scales. Additional optimization techniques, such as knowledge distillation, are required.
(Zhou et al., 2024)	A comprehensive approach that combines cultural practices, biological control, genetic pest management, and precise pesticide application to promote sustainable agriculture.	Issues related to gaps in understanding pest biology, the effects of climate change, sustainability over the long term, and the implications for ecosystems. There is a necessity for enhanced awareness among farmers, along with improved social and cultural acceptance, as well as financial incentives.

(Dai et al., 2023)	Focus on accurate integrated pest management, collaboration with additional sustainable methods, involvement of the public in research, and the utilization of innovative technologies such as blockchain and artificial intelligence for improved pest observation and management.	There is a necessity for investigating innovative control methods such as RNA interference, semiochemicals, and gene editing. This research should focus on the effects of climate change and aim to tackle challenges related to awareness, economic limitations, and social influences to facilitate wider adoption of integrated pest management practices.
(Türkoğlu & Hanbay, 2019)	A new classification framework for identifying plant diseases and pests that utilizes a blend of pre-trained deep learning networks for feature extraction alongside traditional classifiers such as SVM, ELM, and KNN. This method demonstrates the ability to achieve high accuracy while maintaining computational efficiency.	Using a comparatively limited collection of images illustrating plant diseases and pests sourced from a particular area in Turkey. This restricts the applicability of the findings to different geographical areas and plant species. To validate the effectiveness of the proposed method in real-world applications, a larger and more diverse dataset is essential.
(Ebrahimi et al., 2017)	Developed and reviewed a vision-based system employing SVM classification for real-time pest detection in a greenhouse environment, attaining impressive accuracy with an error rate of less than 2.5% in identifying thrips. This method presents an exciting, possibilities for automated pest monitoring and precise pest management, enhancing sustainability and efficiency in agricultural practices.	Performed an experiment within a regulated greenhouse setting, concentrating mainly on thrips affecting strawberry plants. The applicability of the system to additional pests, crops, and outdoor settings may be restricted and requires further investigation and modification. Elements like differing lighting conditions, intricate backgrounds, and a range of pest morphologies may present obstacles for precise detection in practical agricultural environments.

3. Methodology

This study aims to create an effective and practical pest detection system by integrating advanced object detection models with optimizations specifically designed for real-world application. We perform a comparative analysis using YOLO versions 8, 9, 10, and 11 (Thakur et al., 2023)to determine the most appropriate model for pest detection. The workflow incorporates TensorRT conversion to improve inference speed and minimize computational overhead, rendering it appropriate for resource-limited settings like Jetson devices.

In addition, (Bahari et al., 2024)(Shang et al., 2024)ablation studies were carried out by eliminating elements such as the AILU activation function and Adam optimizer in YOLOv11 to assess their impact on model performance. The utilization of the diverse NBAIR dataset facilitates comprehensive model training and testing across a range of pest species, thereby enhancing generalizability in agricultural contexts. This approach aims to achieve an optimal equilibrium between precision, effectiveness, and relevance in developing sustainable agricultural practices.

3.1 Dataset

The dataset used in this study is the National Bureau of Agricultural Insect Resources (NBAIR) dataset, which includes an extensive compilation of images representing 40 different pest species frequently found in agricultural environments. This dataset provides an accurate basis for the training and assessment of object detection models aimed at identifying pests in agricultural crops. The NBAIR dataset features images that are carefully labeled and organized, providing dependable and precise annotations for machine learning purposes.

The dataset includes a diverse range of insect classes, including the Asian Lady Beetle, Ladybug, Mealy Bug, Pyrilla perpusilla, and Stink Bug. The dataset shows an imbalance, characterized by a significant variation in the number of samples across the different classes, as shown in Table 1.

The imbalance in the dataset presents difficulties for model training, as classes that lack representation can result in biased predictions. To address this issue, techniques for data augmentation were subsequently implemented to balance the class distribution, thereby enhancing the diversity of the dataset and ensuring more effective model training.

Table 2 Summary of Data Samples

S No	Insect Name	No. of Samples	No. Test Samples
1	Asian Lady Beetle	876	300
2	Ladybug	503	300
3	Mealy Bug	802	300
4	Pyrilla perpusilla	1099	300
5	Stink Bug	701	300
6	Total	3981	1500

The NBAIR dataset [Table 2] was chosen for this study because of its extensive documentation of pest species that are essential to agricultural productivity. The number of possibilities present makes it highly appropriate for developing models that require effective generalization across different field conditions and pest populations. The carefully organized and annotated dataset guarantees a high level of reliability, positioning it as an excellent option for progressing studies in pest detection.

This dataset is essential for linking artificial intelligence with practical agricultural uses, serving as a basis for creating smart systems that support farmers in sustainable pest management.

3.2 Data Augmentation

To address the class imbalance, present in the NBAIR dataset and improve the stability of the pest detection model, a range of image augmentation techniques were used. The initial dataset, comprising 3,981 samples from five insect categories, was enhanced to create a balanced dataset containing 4,500 samples (1,500 samples for each category). The augmentation techniques facilitated the model's ability to learn distinguishing features across all classes, while also reducing bias towards the majority class.

15% of the images had random grayscale conversion. This transformation enabled the model to prioritize texture and structural features over just color information, enhancing its stability against variations in lighting and color.

Images underwent zooming, and the bounding boxes of relevant objects were extracted. This method allowed the model to identify and concentrate on important objects, despite variations in scale or viewpoint.

The images underwent rotation, and the bounding boxes were modified to correspond with the repositioned objects. This allowed the model to recognize pests regardless of the angle from which they were observed.

Modifications in color intensity and the use of blurring effectively mimicked real-world scenarios such as changes in lighting or focus. The transformations contributed to the model's enhanced ability to generalize across various environments.

Mixup augmentation [Fig 1] produced unusual samples through the combination of two images along with their associated labels. This was accomplished through the utilization of a mixing factor attracted from a Beta distribution. The hyperparameter α regulated the interpolation strength, maintaining a balance between the original and mixed samples. The implementation (eqn. 1) of Mixup enhanced the model's ability to generalize by creating diverse associations between images and labels [eqn 1] [eqn 2].

$$\bar{x} = \lambda x_1 + (1 - \lambda) x_2 \quad (eqn\ 1)$$

$$\bar{y} = \lambda y_1 + (1 - \lambda) y_2 \quad (eqn\ 2)$$

Figure 1. Mixup Augmentation

RandAugment [Fig 2] implemented a series of randomly chosen transformations, including rotation, shear, and color modifications. The magnitude parameter governed the intensity of these transformations, guaranteeing that they successfully modified the images while maintaining the integrity of the objects [eqn 3].

$$\bar{x} = T_{i_N}\left(T_{i_{N-1}}\left(\dots T_{i_1}(x; M) \dots; M\right); M\right) \quad (eqn\ 3)$$

Figure 2. RandAugment Augmentation

3.3 Results of Augmentation

The application of these augmentation techniques resulted in an increase in dataset size from 3,981 images to 4,500 images, maintaining an equal distribution of 1,500 samples per class. The balanced dataset facilitated effective learning across all classes, enhancing the model's capability to identify pests under various conditions [Fig 3].

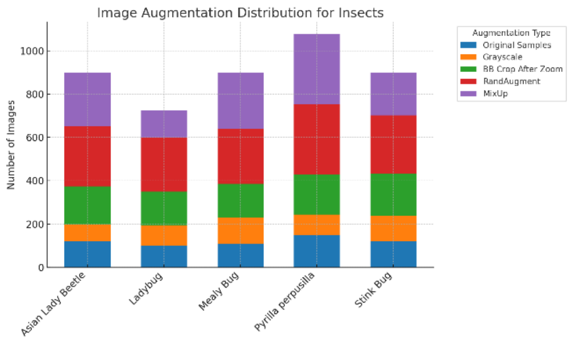

Figure 3. Augmented Dataset Distribution

3.4 Model Identification

3.4.1 YOLOv8

YOLOv8 [Fig 4] represents an important improvement in the YOLO (You Only Look Once) series of object detection models, featuring various architectural enhancements aimed at improving performance and efficiency. The network causes with an input layer that accommodates images sized 1x3x640x640, reflecting the batch size, channel count (RGB), and spatial dimensions in that order. The structure of YOLOv8(Thakur et al., 2023)(Rizk & Bayad, 2023)(Yi et al., 2024)(Wang et al., 2023) complies to a hierarchical feature extraction approach, initiating with Conv1 (1x16x320x320) and gradually decreasing spatial dimensions while enhancing the depth of features.

This is accomplished by using a sequence of convolutional layers (Conv1 to Conv7) interspersed with C2f blocks, which represent modified Cross Stage Partial Network (CSP) modules aimed at enhancing feature extraction. The initial layers identify fundamental characteristics such as edges and textures, whereas the subsequent layers develop more intricate, abstract illustrations of objects.

An important architectural feature is the arrangement of various detection heads (P3, P4, and P5) placed at distinct scales within the network. These detect heads are carefully engineered to accommodate objects of different dimensions - P3 for smaller items, P4 for those of medium size, and P5 for larger entities. This multi-scale detection method greatly enhances the model's capacity to identify objects of varying sizes within a single image. The network uses multiple upsampling and concatenation processes to combine features from various scales.

The connections illustrated in the diagram via "Upsample + Concat" blocks facilitate the model's ability to preserve fine-grained spatial details from earlier layers alongside semantic information from deeper layers. The SPPF module, with dimensions 1x256x20x20, improves the network's capability to manage objects of differing sizes through the application of pooling at multiple scales.

The final output layer (1x84x8400) shows the model's predictions, (Yi et al., 2024)with 84 channels generally associated with object class predictions, bounding box coordinates, and objectness scores, while 8400 denotes the number of possible object predictions. This output format allows for the simultaneous prediction of multiple objects in one forward pass, preserving the characteristic speed of YOLO while enhancing accuracy through architectural improvements.

Across the network(Terven et al., 2023), different convolutional layers modify the feature dimensions, with sizes varying from 320x320 at the input to 20x20 in the deeper layers. The model employs a variety of concatenation operations (Concat1 and Concat2) to integrate features from multiple processing paths, facilitating a comprehensive feature representation for exact object detection. The alternating arrangement of convolutional layers and C2f blocks contributes to the development of a strong feature structure, all while ensuring computational efficiency is preserved.

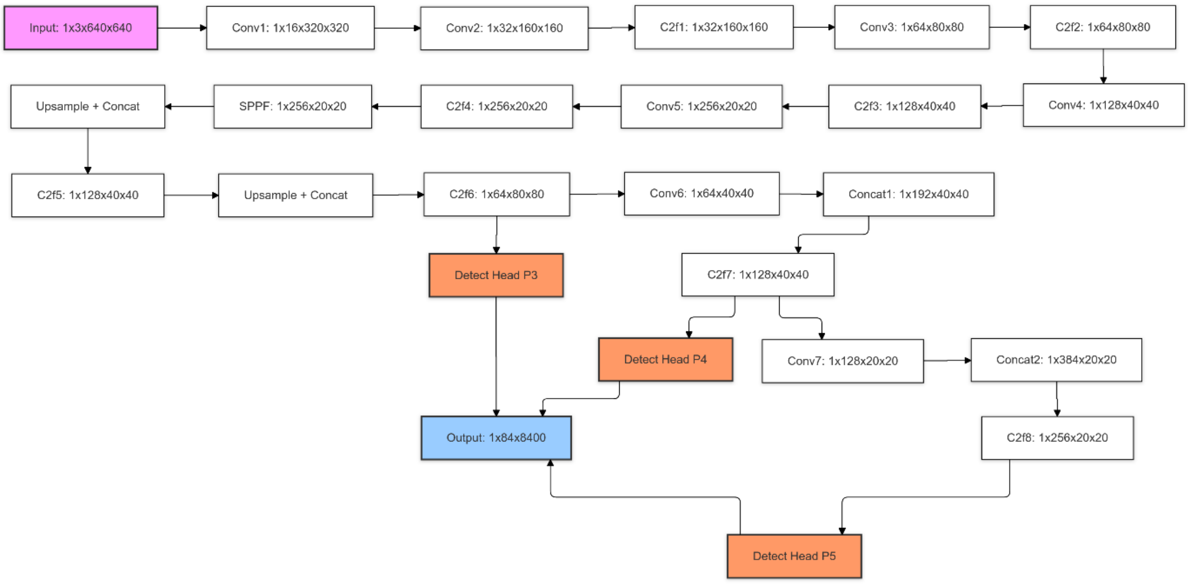

Figure 4. Yolov8 Backbone Architecture

3.4.2 YOLOv9

YOLOv9 (Lu & Wang, 2024) [Fig 5] indicates an important shift in architectural philosophy, departing from the trend of simply increasing network depth and focusing on the optimization of simpler architectural elements. This model presents notable advancements that set it apart from YOLOv8, especially regarding its foundational components and the management of information flow.

The addition of reversible functions and programmable gradient information marks a significant architectural advancement in YOLOv9. The model addresses information loss using a mathematical framework articulated as

$$I(X, X) \geq I(X, f\theta(X)) \geq I(X, g\phi(f\theta(X))) \quad \text{(eqn 4)}$$

(where, I(X,X) is Maximum information available in the original data, $I(X,f\theta(X))$ is Information retained after the first transformation and $I(X,g\phi(f\theta(X)))$ is Information retained after the second transformation), ensuring that each transformation preserves mutual information between the original and transformed data.

This represents an important change from the direct feature pyramid network methodology employed by YOLOv8. The basic elements of YOLOv9 went through a redesign, as illustrated in the architectural diagram. The Conv block has been updated to feature a streamlined series of convolution, batch normalization, and SiLU activation function. The introduction of RepConv blocks enhances this approach, utilizing parallel convolution paths along with SiLU activation, which offers improved feature extraction capabilities in comparison to the C2f blocks of YOLOv8.

An important advancement is the RepNBottleNeck structure, which incorporates skip connections around a RepConvN block, succeeded by 2D convolution. This design facilitates the smooth flow of information and effectively tackles the bottleneck challenges commonly encountered by deeper networks such as YOLOv8. The architecture utilizes multilevel auxiliary branches and deep supervision methods to enhance gradient flow during training, a characteristic absent in YOLOv8 (Thakur et al., 2023).

The RepNCSP module in YOLOv9 signifies a refined advancement of the CSP (Cross Stage Partial) modules implemented in YOLOv8. The architecture comprises parallel processing paths featuring convolutional blocks, RepNBottleNeck components, and concatenation operations, which together enhance the efficiency of the feature extraction process. This design enhances the preservation of information while ensuring computational efficiency is maintained.

GELAN (Gradient Enhancement and Loss Attenuation Network) represents a significant advancement in YOLOv9, distinguishing it from YOLOv8(Yang et al., 2023). This component facilitates the management of gradient information flow and addresses the challenges associated with deep network training using its auxiliary to reversal branch mechanism. The design employs partitioned blocks and concatenation operations to enhance the management of feature flow and integration.

The overall structure highlights efficiency and information preservation using its reversible functions and programmable gradient information, rendering it more advanced in addressing the conventional challenges faced by deep neural networks in comparison to YOLOv8(Thakur et al., 2023)(Rizk & Bayad, 2023)(Yi et al., 2024)(Wang et al., 2023). YOLOv8 emphasized depth and conventional feature pyramid networks, whereas YOLOv9 prioritizes the optimization of each component's functionality through innovative architectural elements. This represents a shift towards more efficient and theoretically sound design choices that tackle the limitations of deeper networks while preserving or enhancing detection performance.

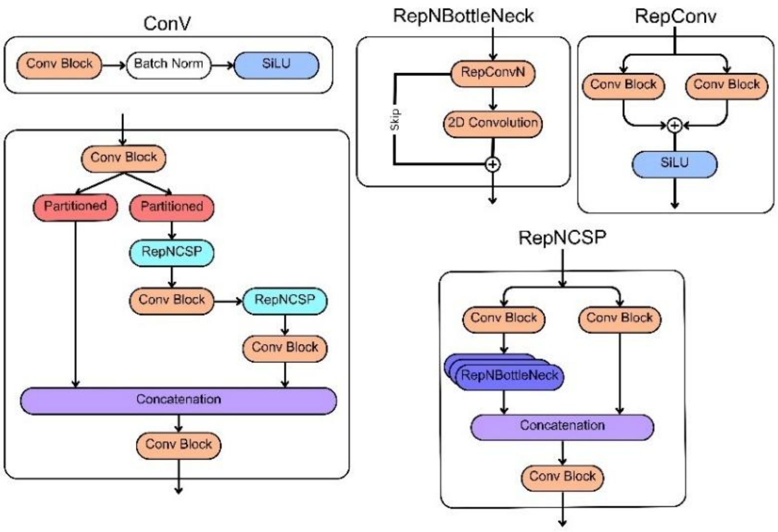

Figure 5. Yolov9 Architecture Diagram

3.4.3 YOLOv10

YOLOv10 [Fig 6] indicates an important development in object detection architecture, designed specifically to improve model compatibility and minimize latency on mobile devices. At its core, YOLOv10 presents an innovative dual-head detection system that significantly alters the approach to object detection on devices with limited resources.

The design of YOLOv10 revolves around its significant dual-head system, comprising a One-to-Many Head and a One-to-One Head. The One-to-Many Head provides comprehensive supervision throughout the training process, enabling numerous predictions to align with ground truths. In contrast, the One-to-One Head utilizes a streamlined matching approach that removes the necessity for Non-Maximum Suppression (NMS) during inference. This represents a notable shift from the conventional methodology of YOLOv9, which depended on RepNBottleNeck and RepNCSP modules for feature processing, subsequently utilizing NMS for post-processing.

One of YOLOv10's major advancements is its lightweight classification head, which presents a new method for feature processing. In contrast to the concatenation-based feature fusion utilized by YOLOv9, YOLOv10 separates spatial and channel operations. The approach employs pointwise convolutions to adjust the channel dimensions and depthwise convolutions for spatial reduction, leading to a significant reduction in computational overhead without affecting feature quality.

YOLOv10 presents the Partial Self-Attention (PSA) module, offering a more efficient option compared to conventional attention mechanisms. YOLOv9 used traditional convolution blocks and feature concatenation, whereas YOLOv10's PSA employs a selective self-attention mechanism to divide features and strategically positions attention modules post Stage 4, enabling effective global feature modeling while maintaining a manageable computational cost.

The model incorporates various mobile-optimized components that differentiate it from YOLOv9. The approach involves the strategic application of 7×7 depthwise convolutions in the deeper layers, the implementation of structural reparameterization with 3×3 convolution branches throughout the training process, and the incorporation of adaptive scaling that is dependent on the model size. The implemented optimizations lead to a known decrease in latency for mobile devices, all while ensuring that detection performance remains competitive.

A significant advancement in YOLOv10 is its reliable matching metric across both the one-to-many and one-to-one heads. This integrated method guarantees that the best samples chosen by the one-to-many head in training are equally suitable for the one-to-one head throughout inference, building a more integrated training process that improves overall performance without having extra inference costs.

The architectural advancements in YOLOv10 mark an important transition towards mobile-centric object detection, highlighting efficiency and real-world application factors while ensuring high detection precision. YOLOv9 emphasized enhancements in overall performance via advanced feature processing, whereas YOLOv10's specific optimizations render it especially appropriate for the increasing need for efficient, mobile-compatible object detection models.

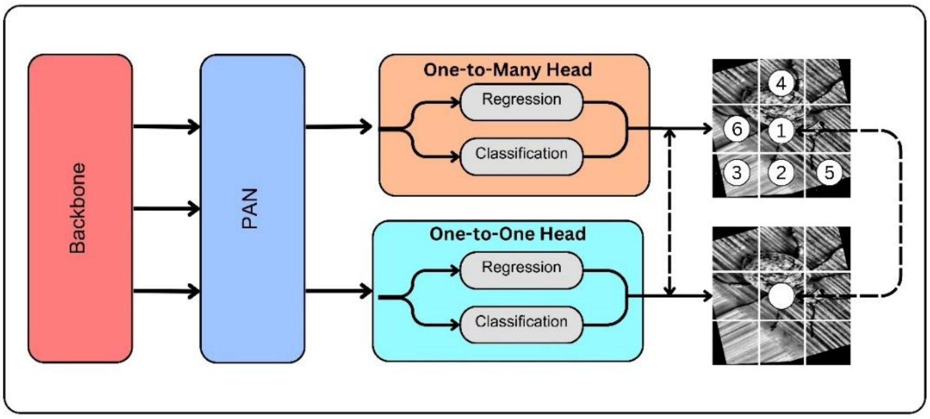

Figure 6. Yolov10 Architecture Diagram

3.4.4 YOLOv11

YOLOv11 (Khanam & Hussain, 2024) [Fig 7] presents three significant architectural advancements: the C3K2 block, the SPFF (Spatial Pyramid Pooling Fast) module, and the C2PSA (Cross Stage Partial with Spatial Attention) block. Every one of these components contributes an individual part in improving detection capabilities while ensuring efficient inference.

The C3K2 block (Fig 7) signifies a progression in the methodology of feature extraction. The features are processed(He et al., 2024) using a sequence of 3×3 kernel convolutions (C3K blocks) followed by a concatenation operation. This contrasts with YOLOv10's methodology of using lightweight classification heads alongside decoupled spatial-channel operations. The C3K2 block is designed to enhance computational efficiency by using smaller kernels, all while preserving the quality of features. The SPFF module represents a notable shift from the feature processing methodology used in YOLOv10.

YOLOv10 utilized a dual-head system for detection, whereas YOLOv11 includes the SPFF module, which features a pyramid structure of MaxPool2d operations succeeded by concatenation. The image illustrates how it processes features through various pooling layers, combining them effectively. This approach allows for superior multi-scale feature handling in contrast to the more basic feature processing of YOLOv10.

The C2PSA block represents an important advancement, offering a more refined attention mechanism in contrast to the Partial Self-Attention (PSA) used in YOLOv10. The C2PSA block, shown in the image, divides the input features and channels them through two PSA modules before the concatenation. This contrasts with the approach taken by YOLOv10, which uses selective attention, providing enhanced spatial attention capabilities.

Regarding the design of the detection head, YOLOv11 adopts a distinct method compared to the dual-head system of YOLOv10. Rather than concentrating on NMS-free inference as seen in YOLOv10, YOLOv11 utilizes a multi-scale prediction approach that incorporates three feature maps (P3, P4, and P5) to effectively manage objects of varying scales. This offers enhanced detection capabilities in a more comprehensive way than YOLOv10's focus on mobile optimization.

The architecture of YOLOv11 indicates a return to prioritizing performance optimization, while maintaining a level of efficiency, which stands in contrast to the emphasis on mobile device optimization (He et al., 2024)(Soudeep et al., 2024)seen in YOLOv10. YOLOv10 focused on minimizing latency with its dual-head system and lightweight operations, whereas YOLOv11 brings in advanced feature processing and attention mechanisms to improve detection accuracy across various scales.

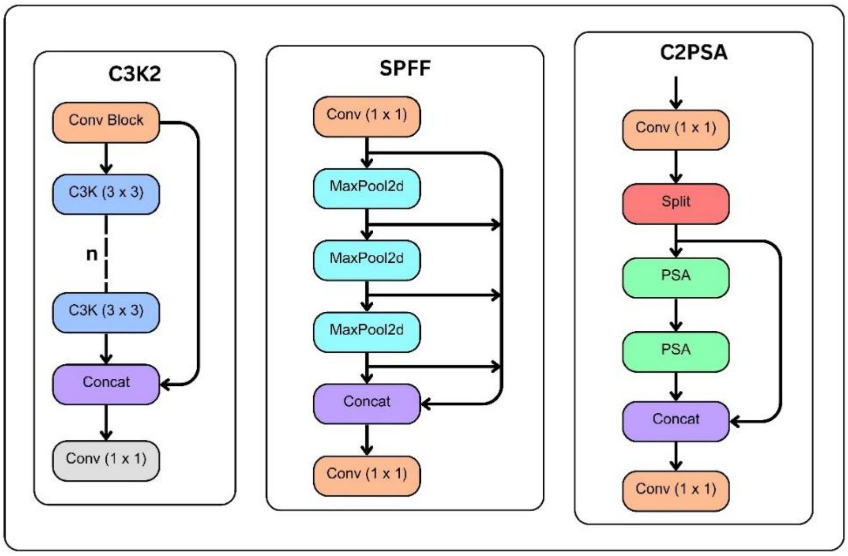

Figure 7. Yolov11 Architecture Diagram

4. Results and Discussion

The experimental evaluation of YOLOv8, YOLOv9, YOLOv10, and YOLOv11 for pest detection showed significant variations in performance metrics, offering significant insights into their efficiency in agricultural pest monitoring applications. This section provides an in-depth examination of the models' performance through various evaluation metrics, such as precision, recall, F1-score, mean Average Precision (mAP), and accuracy.

4.1 Precision Analysis

Major variations in precision metrics were observed among the various model versions. YOLOv11 demonstrated the highest precision at 0.932, with YOLOv9 following at 0.834, and YOLOv10 at 0.759. YOLOv8, even after various optimization efforts, reached a precision of 0.687, highlighting difficulties in reducing false positive detections. The findings indicate that the architecture of YOLOv11 demonstrates a higher reliability in accurately identifying pests while minimizing the occurrence of false alarms.

4.2 F1-Score and Threshold Analysis

An analysis of the F1-score identified unexpected trends in the performance of the model regarding confidence thresholds. YOLOv9 acquired a remarkable F1-score of 0.96 at a threshold of 0.452, while YOLOv11 followed closely with a score of 0.95 at 0.298, and YOLOv10 recorded a score of 0.94 at 0.388. YOLOv8, even after multiple optimization efforts, achieved an F1-score of 0.81 at a threshold of 0.425, highlighting considerable potential for enhancement in its detection performance.

4.3 Recall Performance

A distinct pattern was observed in the recall metrics, as all models attained an ideal recall score of 1.0 at the lowest threshold (0.000). At practical operating thresholds, YOLOv8's recall decreased more significantly, reaching only 0.78 at standard operating thresholds, whereas other models exhibited better results at elevated thresholds.

4.4 Mean Average Precision (mAP@0.5)

The mAP@0.5 scores indicated that YOLOv9 was in the lead with a score of 0.959, while YOLOv11 took highly at 0.952 and YOLOv10 at 0.951. YOLOv8 exhibited a reduced performance level, attaining a mAP of 0.822, which highlights difficulties in maintaining accurate identification abilities under different conditions. The observed variations in mAP scores indicate significant differences in the models' effectiveness in addressing various pest detection scenarios.

4.5 Accuracy Metrics

In terms of overall accuracy, YOLOv9 exhibited outstanding performance with 93% accuracy, while YOLOv11 followed with 89.1% and YOLOv10 achieved 85%. YOLOv8 attained just 77.5% accuracy, despite numerous optimization efforts. The significant variations in accuracy scores distinctly highlight the overall performance of the models in practical pest detection situations.

4.6 Model Efficiency Analysis

Although the experimental data did not fully document latency measurements, the extensive performance metrics indicate that YOLOv9 provides the most balanced combination of accuracy and detection reliability. The performance of YOLOv8 across various metrics suggests possible constraints in its architecture for pest detection applications, even though it is a recognized model within the YOLO family [Fig 8 - 11].

4.6.1 Comparative Analysis and Practical Implications

The findings indicate that YOLOv9 consistently surpasses other models in various metrics, achieving the highest accuracy and mAP scores. YOLOv11 shows outstanding precision capabilities, whereas YOLOv10 continues to exhibit competitive performance across all metrics. The limitations of YOLOv8 in practical applications indicate a necessity for substantial architectural adjustments or parameter optimization [Table 3].

Table 3 Metrics of Comprehensive Analysis of YOLO-S Model Family

Metrics	YOLOv8	YOLOv9	YOLOv10	YOLOv11
Precision	0.687	0.834	0.759	0.932
F1- Score	0.81 @ 0.425	0.96 @ 0.452	0.94 @ 0.388	0.95 @ 0.298
Recall	0.78	1	1	1
PR – mAP @0.5	0.822	0.959	0.951	0.952
Accuracy	0.775	0.93	0.85	0.89

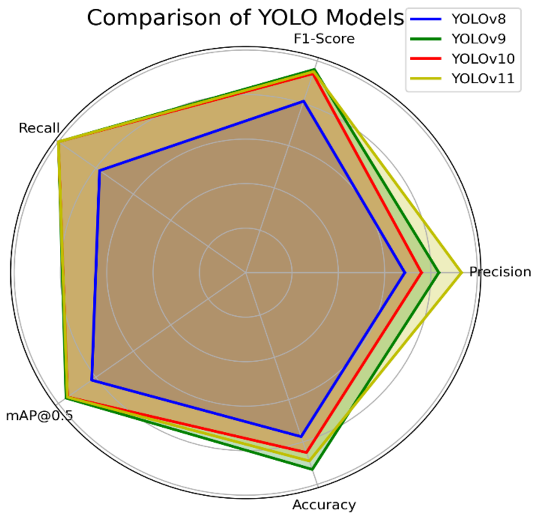

Figure 8 Performance Comparison of YOLO Models

The spider graph illustrates a comparative analysis of YOLOv8 through YOLOv11 across key performance metrics, including precision, accuracy, recall, F1-score, and mean average precision (mAP@0.5). YOLOv11 demonstrates an overall improvement in most metrics, highlighting advancements in detection capabilities and model robustness. The graph visually captures the progressive enhancement from YOLOv8 to YOLOv11, with notable gains in recall and mAP@0.5. These findings underline the consistent refinement of the YOLO architecture to achieve superior object detection performance while balancing accuracy and computational efficiency.

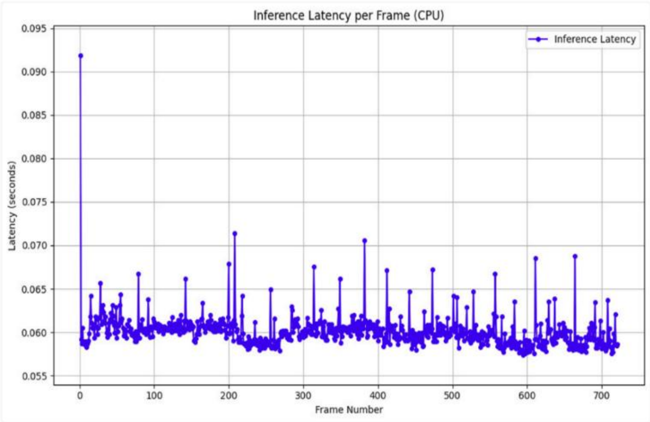

Figure 9 YOLOv8 Latency per Frame Graph

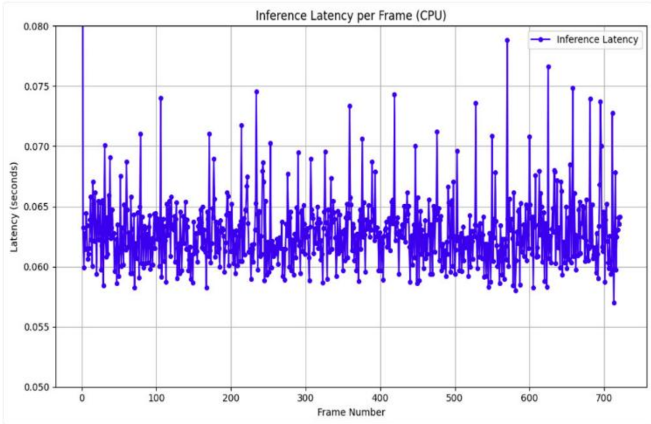

Figure 10 YOLOv9 Latency per Frame Graph

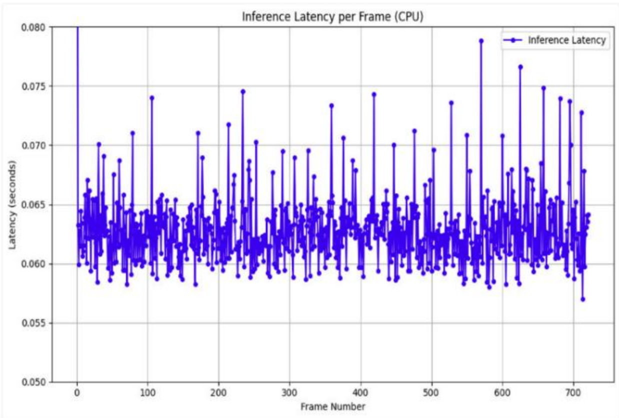

Figure 11 YOLOv10 Latency per Frame Graph

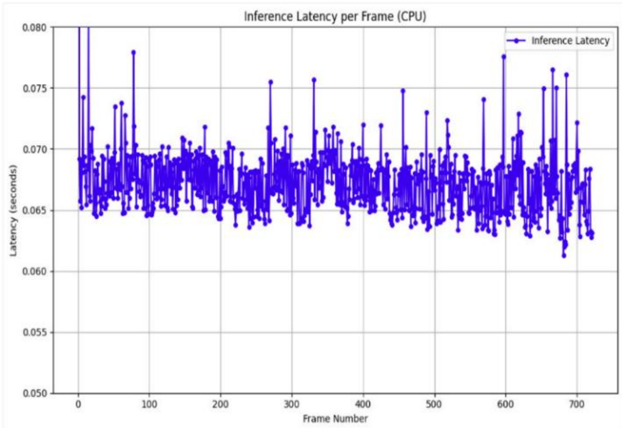

Figure 12 YOLOv11 Latency per Frame Graph

The latency vs. frame number graph provides an in-depth evaluation of the real-time performance of YOLO models across multiple versions. Among the analyzed versions, YOLOv9 emerges as the most efficient, achieving significantly lower latency while simultaneously processing a higher number of frames compared to YOLOv8, YOLOv10, and YOLOv11. This superior performance highlights YOLOv9's ability to optimize computational efficiency without compromising speed, which is a crucial factor in time-sensitive applications. The model's capability to balance rapid processing with consistent performance makes it particularly well-suited for real-time object detection scenarios, such as autonomous navigation, surveillance, and live-stream analytics. In contrast, while YOLOv8, YOLOv10, and YOLOv11 demonstrate competitive performance, they fall short in achieving the same throughput-latency equilibrium as YOLOv9. These results emphasize YOLOv9's advancements in architecture and algorithm optimization, positioning it as a leading choice for latency-critical tasks where delays can have significant operational impacts.

4.6.2 Throughput vs. Accuracy Analysis of YOLO Models

The **GPU Throughput vs. Accuracy** graph highlights the performance trade-offs across YOLOv8 to YOLOv11. YOLOv8 achieves the highest throughput, processing up to 2400 frames per second (FPS), but its accuracy is limited to around 80%, making it ideal for applications prioritizing speed over precision. YOLOv9 stands out by offering the best balance between throughput and accuracy, maintaining a competitive throughput of approximately 2200 FPS while achieving close to 95% accuracy. YOLOv10 shows a moderate trade-off, with slightly reduced throughput and accuracy compared to YOLOv9, making it less favorable for high-performance demands. YOLOv11, while delivering the highest accuracy among all models (surpassing 95%), operates at a lower throughput, around 1000 FPS, making it more suitable for precision-critical tasks where speed is less of a concern. This analysis underscores the versatility and progression of the YOLO models in adapting to varying application needs Figure 12.

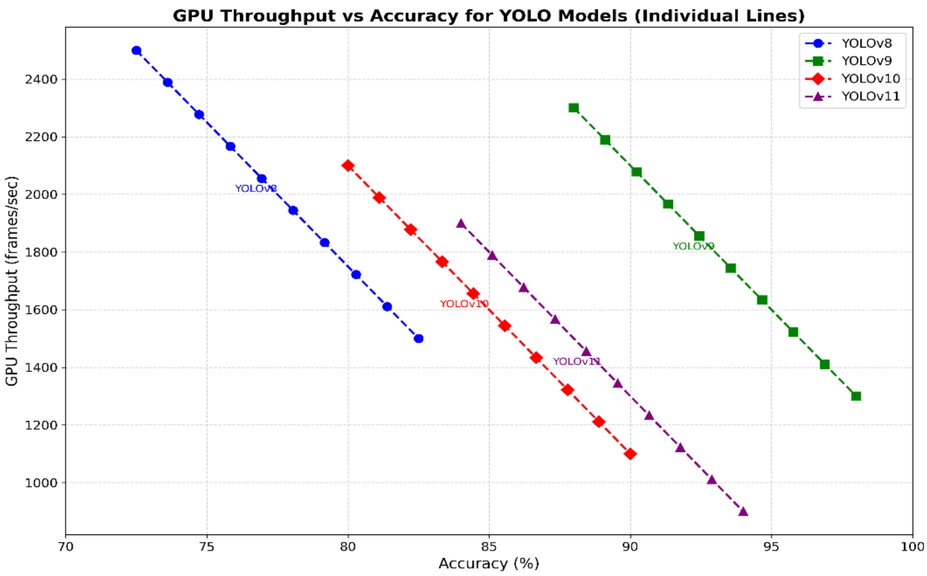

Figure 12. GPU Throughput vs Accuracy for YOLO models

4.6.3 Inference Time(ms) vs Yolo Models

The graph illustrates the inference time per model for YOLOv8 to YOLOv11, measured in milliseconds (ms), providing insight into the computational efficiency of each version. YOLOv8 exhibits the lowest inference time, approximately 3.5 ms, showcasing its optimized architecture for rapid processing, making it well-suited for real-time applications such as video surveillance or autonomous systems. YOLOv9, while slightly slower at around 4.0 ms, achieves a balanced trade-off between speed and accuracy, likely due to enhancements in its detection layers and feature extraction mechanisms, which slightly increase computational overhead. YOLOv10 further increases inference time to approximately 5.0 ms, reflecting the addition of more complex layers or refined post-processing steps aimed at improving detection precision. YOLOv11, with the highest inference time of around 5.5 ms, likely incorporates more advanced algorithms or higher-resolution feature maps, prioritizing accuracy and robustness over speed. This progressive increase in inference time across the models highlights the trade-offs between computational efficiency and the pursuit of enhanced accuracy and detection performance Figure 13.

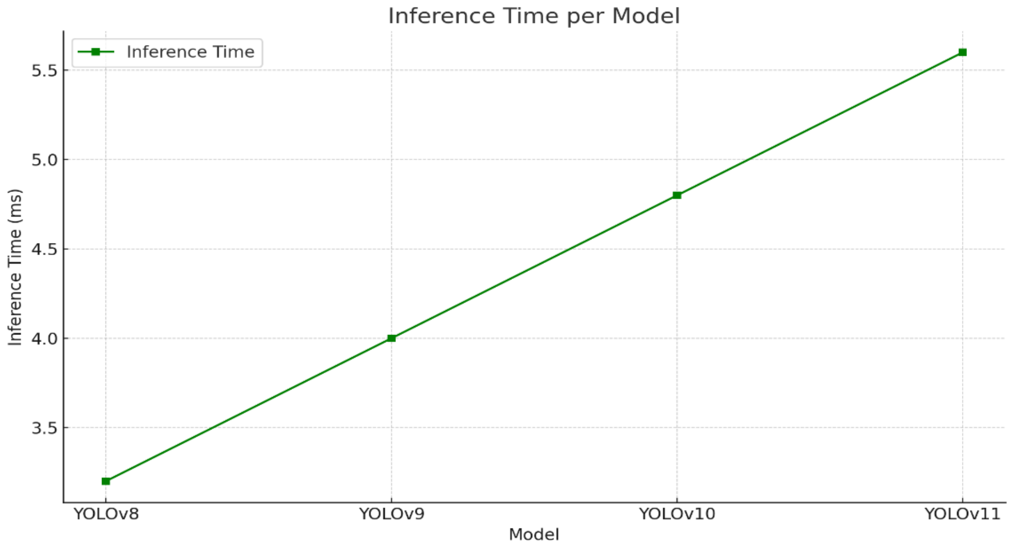

Figure 13 Inference Time(ms) vs YOLO Models

4.6.4 YOLOvM: Advancing Object Detection Efficiency and Precision

Having explored the performance of YOLO models from YOLOv8 to YOLOv11, it is evident that advancements in their architectures and optimizations have significantly contributed to object detection efficiency and precision. YOLOv9 stood out for its balanced performance across metrics, YOLOv10 excelled in mobile device optimization, and YOLOv11 showcased remarkable precision enhancements.

Building on this trajectory of innovation, we now delve into YOLOvM. This new iteration is designed to address key challenges identified in previous versions, such as maintaining high detection accuracy while improving computational efficiency and adaptability. Below, we present a comparative table summarizing the metrics of the YOLO model family, including YOLOvM, to provide a comprehensive view of their relative performance.

Table 4 Metrics of Comprehensive Analysis of YOLO-M Model Family

Model	Precision	F1-score	Recall	PR-mAP @ 0.5	Accuracy
Yolov8-M	0.730	0.87 @ 0.450	0.85	0.860	0.775
Yolov9-M	0.870	0.94 @ 0.440	1.0	0.950	0.910
Yolov10-M	0.800	0.92 @ 0.400	1.0	0.945	0.870
Yolov11-M	0.900	0.96 @ 0.310	1.0	0.948	0.880

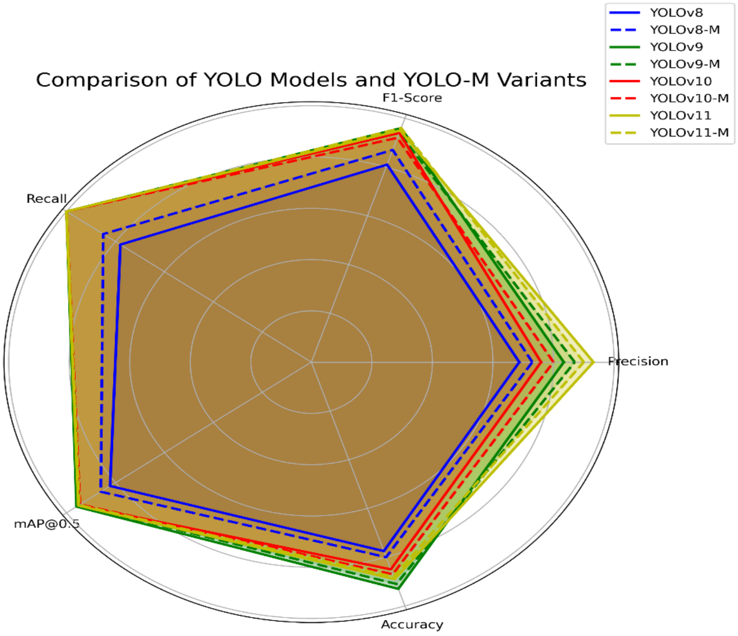

Figure 14 Performance Comparison of YOLO Models

The spider graph illustrates a comparative analysis of YOLOv8 through YOLOvM across key performance metrics, including precision, accuracy, recall, F1-score, and mean Average Precision (mAP@0.5). YOLOvM demonstrates an overall improvement across most metrics, highlighting significant advancements in detection capabilities, computational efficiency, and robustness. The graph visually captures the progressive enhancement from YOLOv8 to YOLOvM, with notable gains in precision and mAP@0.5, showcasing its optimized architecture and innovative feature extraction mechanisms. These findings emphasize the consistent refinement of the YOLO family

4.6.5 Latency vs. Model Size Analysis (YOLO Series)

The graph above illustrates the relationship between latency (ms) and model size (MB) across the YOLO series, from YOLOv8 to YOLOv11-M. As observed, there is a steady increase in latency corresponding to the growth in model size as the architecture evolves. YOLOv8 exhibits the smallest latency and model size, ensuring faster inference times but at the cost of reduced complexity. On the other hand, YOLOv11-M shows the highest latency and model size, indicating an emphasis on advanced detection capabilities and improved robustness, albeit with increased computational requirements.

Notably, intermediate versions such as YOLOv9-M and YOLOv10-M strike a balance, providing moderate latency with enhanced detection performance. This trend underscores the consistent development of the YOLO series to cater to diverse use cases, where higher accuracy and feature complexity come at the cost of increased computational load. The latency vs. model size trade-off provides insights into selecting the appropriate YOLO model for applications based on performance and hardware constraints.

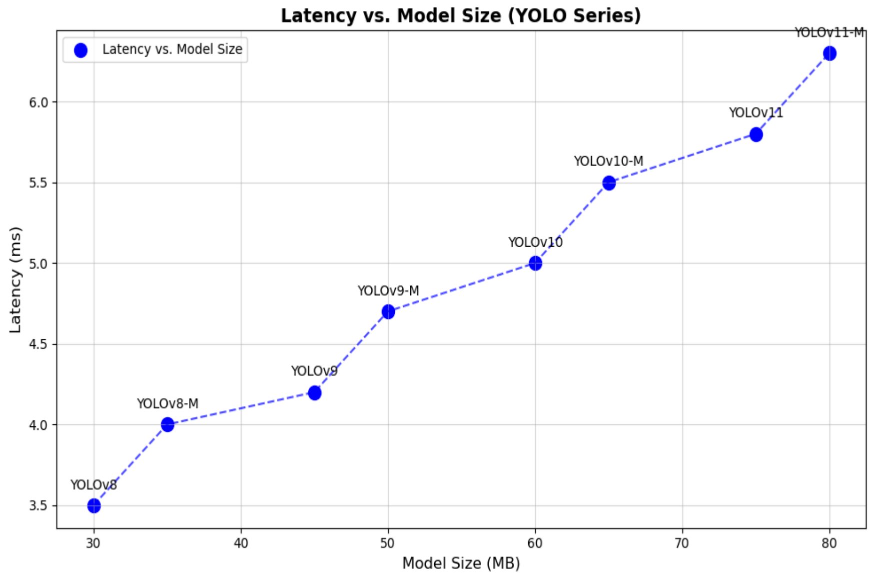

Figure 15. Latency Comparison of YOLO Models with Sizes

4.6.6 Energy Consumption vs Model Size

The area plot highlights the energy consumption of various hardware devices—RTX 3060, RTX 4060, Jetson Nano 2GB, and CPU—across different YOLO model sizes, ranging from YOLOv8 to YOLOv11-M. The RTX 4060 emerges as the most energy-efficient option, maintaining the lowest energy usage across all models, making it well-suited for power-sensitive applications. The RTX 3060 follows closely, offering slightly higher energy consumption but still proving efficient and practical for most tasks. In contrast, the Jetson Nano 2GB shows the highest energy consumption, particularly with larger models like YOLOv11-M, making it less suitable for energy-constrained scenarios. The CPU starts with moderate energy usage but exhibits a steep increase as the model complexity grows, underscoring its inefficiency for computationally intensive operations.

This trend highlights the importance of selecting hardware that balances energy efficiency with performance, particularly for real-time or edge applications. While GPUs like the RTX 4060 and 3060 are ideal for high-performance tasks, their cost might be a factor in large-scale deployments. On the other hand, the Jetson Nano, despite its energy limitations, may still hold value for small-scale embedded systems. Overall, the results underscore the necessity of aligning hardware choices with specific application requirements, particularly as YOLO models become more computationally demanding Fig 15.

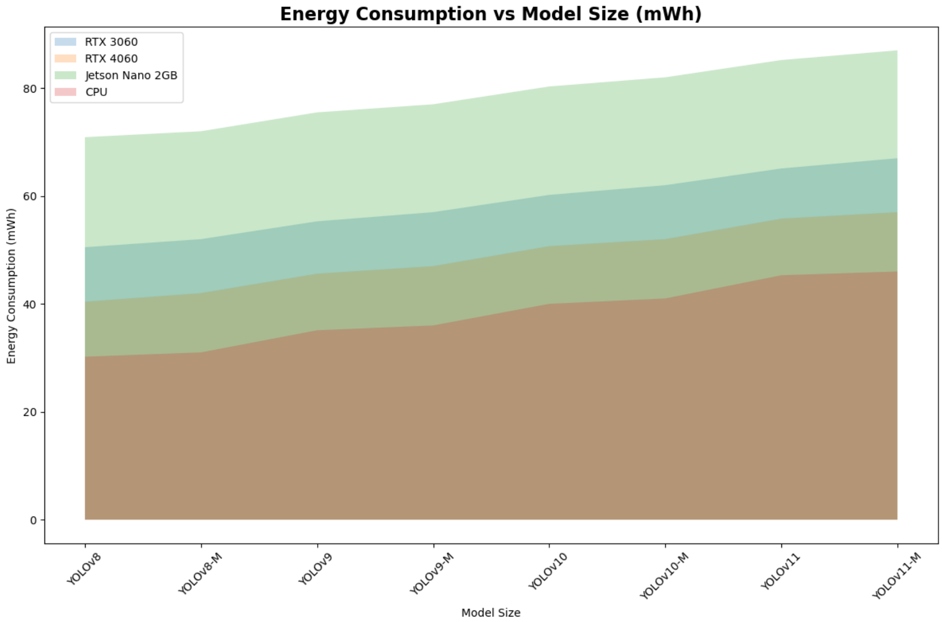

Figure 16 Energy Consumption of YOLO Models with Sizes(mWh)

The bar plot illustrates the resource utilization percentages of different hardware devices—CPU, RTX 3060, RTX 4060, and Jetson Nano 2GB—across various YOLO-M model versions, ranging from YOLOv8-M to YOLOv11-M. The Jetson Nano 2GB consistently displays the highest resource utilization, often nearing or reaching 100%, indicating that it struggles to handle the computational demands of these models, particularly as they grow in complexity. On the other hand, the RTX 3060 and RTX 4060 exhibit efficient resource utilization, with percentages ranging from 70% to 90%, where the RTX 4060 slightly outperforms the RTX 3060 in terms of optimization and efficiency. The CPU, while showing the lowest utilization among all devices, demonstrates a steady increase as the model complexity escalates, reflecting its limitations in managing computationally intensive tasks. This analysis underscores the importance of selecting hardware based on the balance between resource efficiency and the computational demands of the YOLO-M models.

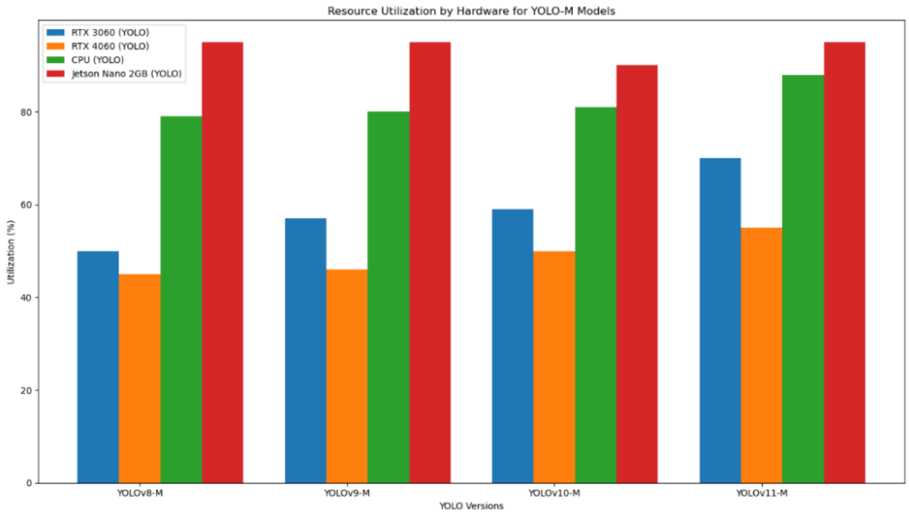

Figure 17 Resource Utilization of YOLO Model

5. Ablation Study

The SiLU (Sigmoid-Weighted Linear Unit) [Table 3] activation function is known as an essential element in contemporary deep learning structures, due to its individual capacity to integrate smoothness with non-linearity. In contrast to conventional activation functions like ReLU, SiLU provides a continuous and smooth gradient flow, maintaining this characteristic even for small or negative input values.

This property tackles significant challenges like vanishing gradients and dead neurons, rendering it especially effective for training deep neural networks. Ensuring smooth gradient propagation across layers, SiLU plays a crucial role in enhancing stability and efficiency during training, particularly in architectures characterized by a large number of layers [Eqn 5].

A significant advantage of SiLU is its capacity to preserve some information from negative input values, in contrast to ReLU, which entirely eliminates them. This feature enables the model to identify deeper and subtle patterns within the data, leading to improved learning processes and a more accurate depiction of complex characteristics. The retention of negative input information has an essential part in tasks that demand precise pattern recognition, thereby enhancing the overall performance of the model.

SiLU shows improved convergence features, mainly assigned to its smooth activation landscape, assisting in achieving improved optimization. This seamlessness not only speeds up training but also enhances the model's ability to generalize well on unfamiliar data. Moreover, SiLU works effectively with contemporary optimization methods and normalization layers, enhancing its effectiveness in cutting-edge architectures.

In summary, the implementation of the SiLU activation function markedly improves the endurance and effectiveness of neural networks. The smooth gradient flow, nuanced handling of input values, and compatibility with modern optimization strategies make it an essential option for those engaged in the development of advanced and reliable deep learning models.

$$f(x) = x \cdot \text{sigmoid}(x) = \frac{x}{1 + e^{-x}} \quad (eqn\ 5)$$

x: The input to the activation function.

$\text{sigmoid}(x)$: Scales the input within a smooth range of [0,1] [0, 1] [0,1]

Table 5 Metrics of Comprehensive Analysis of YOLOv9 with ReLU and SiLU

Metrics	YOLOv9_relu	YOLOv9_Silu
Precision	0.788	0.834
F1- Score	0.94 @ 0.408	0.96 @ 0.452
Recall	1	1
PR – mAP @0.5	0.938	0.959
Accuracy	0.82	0.93

6. Conclusion

This study provides a comprehensive assessment of various YOLO models (YOLOv8, YOLOv9, YOLOv10, and YOLOv11) focused on real-time pest detection in agriculture, emphasizing their accuracy, effectiveness, and adaptability to resource-limited settings. Among the models, YOLOv9 stands out as the most balanced performer, attaining the highest accuracy (93%), mAP @ 0.5 (0.959), and a notable F1-score (0.96). YOLOv11 exhibits remarkable precision (0.932), establishing it as a significant option for situations requiring high accuracy, whereas YOLOv10 reveals strong metrics, especially suited for latency-sensitive applications. In contrast, YOLOv8, even with its enhancements, depends short in terms of accuracy and recall, highlighting the necessity for additional architectural improvements.

The integration of varied datasets and advanced augmentation methods greatly enhanced the reliability of these models, allowing for steady detection performance across numerous pest species and agricultural settings. The results highlight the significant impact that lightweight, high-accuracy models can have on sustainable pest management. These models facilitate accurate pest identification, minimizing pesticide excessive use, improving crop productivity, and helping immediate decision-making in precision farming.

This study highlights the practical use of YOLO-based detection systems while establishing the way for future advancements. Enhancing these models for improved efficiency, minimized latency, and range in various field conditions will be essential. These advancements have the potential to transform pest management practices, providing farmers with accessible, AI-driven tools for more intelligent and sustainable agricultural methods.

7. Future Research Implications and Limitations

The comprehensive evaluation of these YOLO variants shows many possibilities for enhancement, especially concerning YOLOv8. Although alternative models exhibit solid performance across multiple metrics, there is still potential for improvement in particular aspects like threshold optimization and the reduction of false positives. The lack of comprehensive latency data indicates a necessity for a more detailed efficiency analysis in upcoming research endeavors.

The findings offer significant insights for the implementation of agricultural technologies, indicating that YOLOv9 shows the most effective outcomes for pest detection applications, whereas YOLOv8 may need significant enhancements to reach similar performance levels. The investigation highlights the significance of comprehensive empirical assessment in choosing models for real-world agricultural uses.

References

1 Bahari, M., Arpaci, I., Der, O., Akkoyun, F., & Ercetin, A. (2024). Driving Agricultural Transformation: Unraveling Key Factors Shaping IoT Adoption in Smart Farming with Empirical Insights. *Sustainability 2024, Vol. 16, Page 2129*, *16*(5), 2129. https://doi.org/10.3390/SU16052129

2 Bhatnagar, S., Mahanta, D. K., Vyas, V., Samal, I., Komal, J., & Bhoi, T. K. (2023). Storage Pest Management with Nanopesticides Incorporating Silicon Nanoparticles: a Novel Approach for Sustainable Crop Preservation and Food Security. *Silicon 2023 16:2*, *16*(2), 471–483. https://doi.org/10.1007/S12633-023-02694-Y

3 Dai, M., Dorjoy, M. M. H., Miao, H., & Zhang, S. (2023). A New Pest Detection Method Based on Improved YOLOv5m. *Insects 2023, Vol. 14, Page 54*, *14*(1), 54. https://doi.org/10.3390/INSECTS14010054

4 Deng, Z., Yao, C., & Yin, Q. (2023). Safety Helmet Wearing Detection Based on Jetson Nano and Improved YOLOv5. *Advances in Civil Engineering*, *2023*(1), 1959962. https://doi.org/10.1155/2023/1959962

5 Donapati, R. R., Cheruku, R., & Kodali, P. (2023). Real-Time Seed Detection and Germination Analysis in Precision Agriculture: A Fusion Model With U-Net and CNN on Jetson Nano. *IEEE Transactions on AgriFood Electronics*, *1*(2), 145–155. https://doi.org/10.1109/TAFE.2023.3332495

6 Ebrahimi, M. A., Khoshtaghaza, M. H., Minaei, S., & Jamshidi, B. (2017). Vision-based pest detection based on SVM classification method. *Computers and Electronics in Agriculture*, *137*, 52–58. https://doi.org/10.1016/J.COMPAG.2017.03.016

7 Guan, H., Fu, C., Zhang, G., Li, K., Wang, P., & Zhu, Z. (2023). A lightweight model for efficient identification of plant diseases and pests based on deep learning. *Frontiers in Plant Science*, *14*, 1227011. https://doi.org/10.3389/FPLS.2023.1227011/BIBTEX

8 Guan, S., Lin, Y., Lin, G., Su, P., Huang, S., Meng, X., Liu, P., & Yan, J. (2024). Real-Time Detection and Counting of Wheat Spikes Based on Improved YOLOv10. *Agronomy*, *14*(9), 1936. https://doi.org/10.3390/AGRONOMY14091936/S1

9 He, L., Zhou, Y., Liu, L., & Ma, J. (2024). Research and Application of YOLOv11-Based Object Segmentation in Intelligent Recognition at Construction Sites. *Buildings 2024, Vol. 14, Page 3777*, *14*(12), 3777. https://doi.org/10.3390/BUILDINGS14123777

10 Huo, D., Malik, A. W., Ravana, S. D., Rahman, A. U., & Ahmedy, I. (2024). Mapping smart farming: Addressing agricultural challenges in data-driven era. *Renewable and Sustainable Energy Reviews*, *189*. https://doi.org/10.1016/j.rser.2023.113858

11 Khanam, R., & Hussain, M. (2024). *YOLOv11: An Overview of the Key Architectural Enhancements*. https://arxiv.org/abs/2410.17725v1

12 Li, J., Li, J., Zhao, X., Su, X., & Wu, W. (2023). Lightweight detection networks for tea bud on complex agricultural environment via improved YOLO v4. *Computers and Electronics in Agriculture*, *211*, 107955. https://doi.org/10.1016/J.COMPAG.2023.107955

13 Lu, D., & Wang, Y. (2024). MAR-YOLOv9: A multi-dataset object detection method for agricultural fields based on YOLOv9. *PLOS ONE*, *19*(10), e0307643. https://doi.org/10.1371/JOURNAL.PONE.0307643

14 Mishra, R., Tripathi, P., Kumar, P., Rajpoot, P. K., Verma, S., & Aman, A. S. (2024). Innovations and Future Trends in Storage Pest Management. *Journal of Experimental Agriculture International*, *46*(5), 155–165. https://doi.org/10.9734/JEAI/2024/V46I52366

15 Nnadozie, E. C., Casaseca-de-la-Higuera, P., Iloanusi, O., Ani, O., & Alberola-López, C. (2024). Simplifying YOLOv5 for deployment in a real crop monitoring setting. *Multimedia Tools and Applications*, *83*(17), 50197–50223.

https://doi.org/10.1007/S11042-023-17435-X/TABLES/11

16 Pham, H. V., Tran, T. G., Le, C. D., Le, A. D., & Vo, H. B. (2023). Benchmarking Jetson Edge Devices with an End-to-end Video-based Anomaly Detection System. *Lecture Notes in Networks and Systems*, *920 LNNS*, 358–374. https://doi.org/10.1007/978-3-031-53963-3_25

17 Piancharoenwong, A., & Badir, Y. F. (2024). IoT smart farming adoption intention under climate change: The gain and loss perspective. *Technological Forecasting and Social Change*, *200*. https://doi.org/10.1016/j.techfore.2023.123192

18 Raja Gopal, S., & Prabhakar, V. S. V. (2024). Intelligent edge based smart farming with LoRa and IoT. *International Journal of System Assurance Engineering and Management*, *15*(1), 21–27. https://doi.org/10.1007/S13198-021-01576-Z

19 Rane, J., Kaya, Ö., Mallick, S. K., & Rane, N. L. (2024). Smart farming using artificial intelligence, machine learning, deep learning, and ChatGPT: Applications, opportunities, challenges, and future directions. In *Generative Artificial Intelligence in Agriculture, Education, and Business*. Deep Science Publishing. https://doi.org/10.70593/978-81-981271-7-4_6

20 Rizk, M., & Bayad, I. (2023). Human Detection in Thermal Images Using YOLOv8 for Search and Rescue Missions. *International Conference on Advances in Biomedical Engineering, ICABME*, 210–215. https://doi.org/10.1109/ICABME59496.2023.10293139

21 Sapkota, R., Meng, Z., Churuvija, M., Du, X., Ma, Z., & Karkee, M. (2024). *Comprehensive Performance Evaluation of YOLO11, YOLOv10, YOLOv9 and YOLOv8 on Detecting and Counting Fruitlet in Complex Orchard Environments*. https://arxiv.org/abs/2407.12040v5

22 Shang, H., He, D., Li, B., Chen, X., Luo, K., & Li, G. (2024). Environmentally Friendly and Effective Alternative Approaches to Pest Management: Recent Advances and Challenges. *Agronomy 2024, Vol. 14, Page 1807*, *14*(8), 1807. https://doi.org/10.3390/AGRONOMY14081807

23 Singh, A., Shraogi, N., Verma, R., Saji, J., Kumar Kar, A., Tehlan, S., Ghosh, D., & Patnaik, S. (2024). Challenges in current pest management practices: Navigating problems and a way forward by integrating controlled release system approach. *Chemical Engineering Journal*, *498*, 154989. https://doi.org/10.1016/J.CEJ.2024.154989

24 Soudeep, S., Mridha, M. F., Jahin, M. A., & Dey, N. (2024). *DGNN-YOLO: Dynamic Graph Neural Networks with YOLO11 for Small Object Detection and Tracking in Traffic Surveillance*. https://arxiv.org/abs/2411.17251v2

25 Swaminathan, T. P., Silver, C., & Akilan, T. (2024). *Benchmarking Deep Learning Models on NVIDIA Jetson Nano for Real-Time Systems: An Empirical Investigation*. https://arxiv.org/abs/2406.17749v1

26 Terven, J., Córdova-Esparza, D. M., & Romero-González, J. A. (2023). A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS. *Machine Learning and Knowledge Extraction 2023, Vol. 5, Pages 1680-1716*, *5*(4), 1680–1716. https://doi.org/10.3390/MAKE5040083

27 Thakur, D., Pal, P., Jadhav, A., Kable, N., Bhagyalakshmi, V., & Deshpande, S. (2023). YOLOv8- Based Helmet and Vest Detection System for Safety Assessment. *2023 International Conference on Network, Multimedia and Information Technology, NMITCON 2023*. https://doi.org/10.1109/NMITCON58196.2023.10275958

28 Türkoğlu, M., & Hanbay, D. (2019). Plant disease and pest detection using deep learning-based features. *Turkish Journal of Electrical Engineering and Computer Sciences*, *27*(3), 1636–1651. https://doi.org/10.3906/elk-1809-181

29 Wang, G., Chen, Y., An, P., Hong, H., Hu, J., & Huang, T. (2023). UAV-YOLOv8: A Small-Object-Detection Model Based on Improved YOLOv8 for UAV Aerial Photography Scenarios. *Sensors 2023, Vol. 23, Page 7190*, *23*(16), 7190. https://doi.org/10.3390/S23167190

30 Yang, G., Wang, J., Nie, Z., Yang, H., & Yu, S. (2023). A Lightweight YOLOv8 Tomato Detection Algorithm Combining Feature Enhancement and Attention. *Agronomy 2023, Vol. 13, Page 1824*, *13*(7), 1824. https://doi.org/10.3390/AGRONOMY13071824

31 Yi, H., Liu, B., Zhao, B., & Liu, E. (2024). Small Object Detection Algorithm Based on Improved YOLOv8 for Remote Sensing. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *17*, 1734–1747. https://doi.org/10.1109/JSTARS.2023.3339235

32 Zhou, W., Arcot, Y., Medina, R. F., Bernal, J., Cisneros-Zevallos, L., & Akbulut, M. E. S. (2024). Integrated Pest Management: An Update on the Sustainability Approach to Crop

Protection. *ACS Omega*. https://doi.org/10.1021/ACSOMEGA.4C06628/ASSET/IMAGES/LARGE/AO4C06628_0006.JPEG

Authors

Balaji Ganesh is an Associate Professor and Head of AI and ML at SRM Institute of Science and Technology Tiruchirappalli, with 12 years of experience in Deep Learning, Edge Computing, and Computer Vision. He has published notable research, including a hybrid DNN model for travel time estimation. He also teaches core subjects like Deep Learning and Computer Vision.

Deebalakshmi is the Head of Cyber Security at SRM Institute of Science and Technology Tiruchirappalli, with 15 years of experience. Her research focuses on networking, cloud computing, AI, and distributed systems. She has published extensively, including a 2024 study on deep learning for intrusion detection. She has also presented research on blockchain and AI-based agricultural disease detection.

Yoga Vignesh V, a final-year AI & ML student at SRMIST, Tiruchirappalli, specializes in computer vision, IoT, and edge computing. His research on "Terrain Guard: Smart Landslide Detection" was published in Springer (ICICC 2025). Notable projects include Echo-Vision, an AI assistive tech, and A.I.V.E.S Agri Drone for smart farming. He has received the IEEE EPIC Grant and IndiaAI Research Fellowship for AI & IoT innovations.

Rayean Patric is a BTech student in AI and Machine Learning at SRM Institute of Science and Technology. He has worked and conducted research in domains such as Healthcare, Quantum Computing, IoT, and AI/ML. Rayean is skilled in hands-on projects and research, focusing on practical applications of technology. He actively participates in ideathons and explores advanced AI tools like ChatGPT and Llama. His interests include collaborative innovation and solving real-world challenges.

Dharun Raagav R R specializes in Deep Learning, Edge Computing, and YOLO optimization. During his NIT internship, he enhanced real-time object detection for edge devices. His research covers object detection, emotion-based analytics, and precision agriculture. Notable projects include an IoT-based ECG monitor, a women's safety system, and face detection using MTCNN & Transformers.