



International Journal of Engineering Research and Science & Technology

www.ijerst.org

ISSN : 2319-5991

Vol. 22 No. 2(2) (2026)



ijerst.editor@gmail.com
editor@ijerst.com

Research Paper

ENHANCING PRECISION AGRICULTURE PEST CONTROL A YOLOV10-BASED DEEP LEARNING APPROACH FOR INSECT DETECTION

Mohammed Sahil Rasool Khan¹, Dr. K. M.Subramanian², Dr. Md Ateeq Ur Rehman³

¹PG Scholar, Department of Computer Science and Engineering, **Email:** msahilrk@gmail.com

²Professor, Department of Computer Science and Engineering, **Email:** kmsubbu.phd@gmail.com

³Professor, Department of Computer Science and Engineering, **Email:** mail_to_ateeq@yahoo.com
Shadan College of Engineering and Technology, Peerancheru, Hyderabad - 500086

ABSTRACT:

Precision agriculture (PA) maximizes resource usage while preserving crop output and quality through the use of state-of-the-art technologies. However, pest infestations remain a significant issue that may jeopardize these benefits. Recent deep learning frameworks such as YOLOv8 have shown promise in real-time insect detection, albeit often being limited to specific insect species or crops. To overcome this limitation and improve detection accuracy, this work explores an enhanced, generalized approach that makes use of the latest YOLOv10 object identification model. We develop and test a YOLOv10-based technology that enables more thorough and rapid field pest monitoring by identifying any kind of bug in a range of crops. A comprehensive performance investigation was conducted on a benchmark bug dataset, demonstrating notable improvements over YOLOv8, such as higher mean Average Precision (mAP) scores and faster inference rates. The findings show that YOLOv10's architectural enhancements result in more accurate, scalable, and real-time pest identification, which holds significant potential for enhancing precision agriculture pest management strategies.

1. INTRODUCTION:

By enabling farmers to track, evaluate, and react to field variability in real time, precision agriculture (PA) is transforming farming methods and ultimately increasing crop output and resource efficiency. The prompt and effective detection of pests that can seriously harm crop yields, however, is one of the most enduring issues endangering PA's success. Conventional manual monitoring techniques are frequently subjective, labour-intensive, and unable to deliver continuous real-time data. To bridge this gap, deep learning-based item detection frameworks have recently surfaced that can quickly and automatically identify pests in field photos and videos. Among these, the YOLO (You Only Look Once) family of models—particularly YOLOv8—has become well-known for balancing speed and accuracy, which makes it suitable for real-time agricultural applications. However, the YOLOv8-based solutions used in existing systems often concentrate on recognizing specific insect species that are relevant to specific crops, which limits their scalability and adaptability to new pests or agricultural contexts. Moreover, these algorithms can struggle to recognize small, partially obscured, or overlapping insects in challenging field conditions. The proposed study offers an enhanced pest detection technique in this context using the latest YOLOv10 object identification model. YOLOv10's architectural improvements include an anchor-free architecture and improved feature representation, which lead to faster inference and higher accuracy even on edge devices.

The system can identify every type of bug in a range of crops without the requirement for crop-specific retraining since it was trained to identify insects as a generalized single class. This project intends to provide a more reliable, universal, and effective pest monitoring tool by overcoming the shortcomings of earlier models and tackling practical field issues. This will ultimately assist farmers in proactive, data-driven pest management within the context of precision agriculture.

In the realm of artificial intelligence (AI), computer vision uses camera-equipped machines to process and interpret images as though they were seen by human eyes [8]. This makes it possible to identify and categorize things in an image. More methods for PA deployment have been made available by the introduction of computer vision techniques in the field of agriculture, often known as agrovision [9]. Robots and modified farming equipment used for weeding [10] and harvesting mature crops [11] are two examples of automated farming tasks. [12]. Thus, the primary emphasis of current research is Convolutional Neural Networks (CNN) and other deep-learning algorithms [13], [14]. CNN-based techniques have shown impressive results in a variety of computer vision problems since they don't require human feature extraction. Their framework, which automatically classifies photos before conducting item recognition tasks, is designed with visual perception in mind. CNN models and algorithms are the foundation of YOLO [15], an open-source real-time object detection system.

Since its first iteration, YOLO has been enhanced with additional frameworks, outperforming CNN-based algorithms such as Fast Region-CNN [16] [17]. Furthermore, YOLO has enhanced pest detection studies [18], [19], and [20]. The current YOLO pest remedies, however, have a serious flaw: they only detect one or a small number of insects, which implies that the solutions are only useful for a single type of bug or crop.

Because of this, it is challenging to extend pest detection systems as a de facto aspect of more comprehensive PA systems, which limits their usefulness. The relevance of recent research on YOLO is limited because it relies on the classification of insects into pre-established categories of a smaller fraction of insect kinds. Not every insect is present anywhere in the world. Some are exclusive to particular crops and geographical areas. Farmers who grow different crops or are located in different regions will not find this useful. This study's general approach to the issue, which allows the farmer to identify insects regardless of the particular crop or bug species, is one of its main contributions. By using our approach as an all-purpose solution, farmers may lessen the disadvantages of current systems that only address a small number of insects and may ignore other dangerous pests, leading to serious problems going unnoticed. By presenting a generalist application of YOLO for pest detection, we provide a system that considers all insect species as a single category to detect the presence of any bug in any type of crop in real time. In order to fully apply YOLO to insect detection, we thoroughly examine and assess the many models of the most recent version of YOLO, YOLOv8 [21]. The creation of a general and potent deep-learning pest detection tool to be incorporated into PA systems that use YOLOv8 is our main contribution to the scientific community and end users in general.

We use experimentation to analyse the outcomes of the training, validation, and testing phases in order to evaluate the model's performance under different conditions. The results show the benefits and potential drawbacks of YOLOv8 for insect detection. This has enabled us to produce a thorough and reliable dataset and laid the groundwork for future accurate and reliable insect identification systems. Information processing in PA has evolved as a result of recent advancements in AI. such as machine learning and deep learning, has encouraged writers to evaluate its efficacy for pest detection and classification. Artificial Neural Networks (ANN), naïve Bayes (NB), k-Nearest neighbour's (KNN), Support Vector Machine (SVM), and CNN models were examined for insect recognition and classification tasks by [22]. In particular, insects were located using contour identification and foreground extraction. The validation findings showed that CNN had the highest accuracy rates, with 91.5% classification accuracy for nine different insect classes and 90% for twenty-four different insect classes. Similarly, [12] compared machine learning and deep

learning techniques for computer vision-based tomato plant pest detection. Autonomous farm robots were the target audience for this pest detecting system.

1.2 SCOPE OF THE PROJECT

This project's scope includes using the YOLOv10 object detection framework to design, construct, and assess a generalized pest detection system for precision agriculture. By training YOLOv10 to recognize insects of any kind across several crops, the study seeks to go beyond crop- and species-specific detection algorithms, offering a scalable solution appropriate for a variety of agricultural settings. The topics covered include dataset preparation, model training, validation, testing, and performance comparison with existing YOLOv8-based systems. Because the system is designed to process images and video frames in real time, it might be included into mobile applications, drones, or field monitoring devices. By focusing on generalization, accuracy, and real-time performance, the project seeks to enhance timely insect monitoring and decision-making, ultimately promoting sustainable pest control methods within modern precision agriculture.

1.3 OBJECTIVE

The primary objective of this project is to develop and deploy a generalized, real-time pest diagnosis system for precision agriculture using the YOLOv10 object detection framework. The project seeks to address the limitations of existing YOLOv8-based systems, which are restricted to detecting specific bug species or crops, by developing a model that can identify any insect category in a range of agricultural contexts. Improving detection accuracy for small, occluded, and overlapping insects that are commonly found in real-world field conditions; optimizing the system for real-time performance suitable for deployment on edge devices like drones, cameras, or mobile applications; and training the YOLOv10 model on a large insect dataset labeled under a single class to achieve generalization are the specific objectives. The study also aims to evaluate the performance improvements enabled by YOLOv10 in terms of mean Average Precision (mAP), inference time, and resilience when compared to previous YOLOv8 implementations. By accomplishing these objectives, the project seeks to advance more effective and scalable pest monitoring techniques, enabling prompt and data-driven pest management decisions in modern precision agriculture

1.4 EXISTING SYSTEM:

One of the most sophisticated and popular deep learning-based object detection algorithms in automated pest monitoring systems for precision agriculture is YOLOv8 (You Only Look Once version 8). Convolutional neural networks are the foundation of YOLOv8, a single-stage object detector that can process an entire input image in a single forward pass while concurrently predicting bounding boxes and

class probabilities. Because of its real-time detection capability, it is ideal for agricultural applications where prompt insect identification is essential to preventing crop damage and production loss.

Typically, datasets tagged with particular bug classes relevant to certain crops in real-world deployments are used to train YOLOv8-based algorithms. This targeted approach helps achieve high accuracy in identifying the selected pest species in regulated or semi-controlled situations. Despite these benefits, using the existing technology in diverse and dynamic agricultural contexts has serious disadvantages. Its narrow scope is a major problem: models developed on datasets pertaining to a specific crop or area often fail to generalize to new insect species or crops that were not included in the original training set.

Furthermore, in real-world field situations, when insects may be small, partially obscured by leaves or stems, grouped together, or present against complicated and changeable backgrounds, detection accuracy might deteriorate dramatically. Performance can also be impacted by variances in lighting, weather, and camera quality. Furthermore, even while YOLOv8 is faster and more effective than many previous models, it can still be computationally difficult to run on low-power edge devices like field sensors or drones, which occasionally compromises real-time response. Because of these characteristics, the current YOLOv8-based system is still specialized, less scalable, and frequently unsuitable for thorough, large-scale pest monitoring in precision agriculture, even if it is a major improvement over manual insect scouting and earlier detection systems.

1.4.1 EXISTING SYSTEM DISADVANTAGES:

➤ **Restricted to particular bug species associated with particular crops:**

Insects outside of those designated classifications are not detected by models since they are often trained on limited, crop-specific datasets.

➤ **Insects that are small, hidden, or overlap reduce accuracy:**

Partially occluded, aggregated, or very small insects, which are frequently encountered in dense crop fields, have worse detection performance.

➤ **Every new crop or pest type necessitates retraining:**

Deployment is expensive and time-consuming since each new insect or crop scenario requires dataset gathering and retraining.

➤ **Real-time deployment on edge devices is limited by higher computing load:**

Even while YOLOv8 is faster than previous models, it still requires a lot of GPU or CPU power, which makes it difficult to use on drones or low-power field sensors.

➤ **Performance declines with complicated field backgrounds and shifting lighting:**

False positives or missed detections may result from variations in backdrop clutter, sunlight, and shadows.

2. RELATED WORK

The evolution of information processing in PA brought about by recent advances in AI, such as machine-learning

and deep-learning, has led authors to evaluate its performance for pest detection and classification. Artificial Neural Networks (ANN), naïve bayes (NB), k-Nearest neighbors (KNN), Support Vector Machine (SVM), and CNN models were compared for insect detection and classification tasks by [22]. In particular, insect detection was performed using foreground extraction and contour identification. The validation results showed that the highest accuracy values were obtained for CNN with 91.5% classification accuracy for 9 different insect classes and 90% for 24 different insect classes. Similarly, [12] compared machine-learning and deep-learning techniques for computer vision-based pest identification in tomato plants. This pest identification solution was intended for agricultural autonomous robots. The authors generated their own dataset with affected tomato plants. Their results from the validation tests determined that deep-learning techniques perform better in terms of accuracy, speed, and pest-type discrimination.

Analyzing the recent scientific literature, there seems to be a consensus about the superior performance of deep-learning for pest detection. Reference [13] presented a deep-learning solution for tiny pest detection employing the images obtained from cameras on sticky traps. The focus of the study was on thrips and whiteflies in greenhouses. A model called TPest-RCNN based on Faster Regional-CNN was developed, with a mean average precision (mAP) of 0.952 and a mean F1 score, which is the harmonic mean of precision and recall, of 0.944. Their results demonstrated better performance using the solution proposed by the authors than the Faster R-CNN model it was based on. A deep learning-based system for pest detection in oilseed rape crops to be employed in mobile devices was introduced by [23]. Firstly, the authors created a database by gathering images from the 12 most common pests in oilseed rape. Five different models were evaluated to determine their performance considering the model loss. These models were Faster R-CNN, R-FCN, and SSD (Single-Shot Detector), with a combination of different feature extractors such as ResNet101, Inception, or MobileNet. Data augmentation techniques as well as a dropout layer, which prevented overfitting, were added to improve the resulting mAP. The authors determined that the best solution was SSD with Inception as it provided a balance between execution time, accuracy, and memory usage. Their results showed an improvement in mAP, which achieved a value of 77.14%. The trained model was implemented in an Android platform for use in mobile phones. Furthermore, [14] developed a mobile application for pest identification based on Faster R-CNN deep-learning techniques. The application was also

connected to a database with pesticide recommendations, providing users with specific information on how to eliminate the detected pests. The pests considered were red spider, flax budworm, aphids, flea beetles, and Cicadellidae, all receiving recognition results of 99.0%. Within the options of deep-learning techniques available, YOLO has been gaining popularity due the performance demonstrated in several studies. Reference [24] reviewed insect detection techniques using deep learning.

2.1 PROPOSED SYSTEM

The proposed method aims to address the drawbacks of the existing YOLOv8-based pest identification models by using the capabilities of YOLOv10, the latest version in the YOLO object detection family. Unlike traditional methods that focus on identifying specific insect species associated with certain crops, this experiment employs a generalized detection strategy in which the YOLOv10 model is trained on a heterogeneous dataset grouped under a single, broad insect category. Because of its architecture, the model can identify and locate each kind of insect in a variety of crops and field settings without requiring recurrent retraining for every kind of pest or crop. Improved backbone networks for richer feature extraction, sophisticated neck and head modules for better multi-scale object detection, and anchor-free mechanisms that boost accuracy for small, partially hidden, or overlapping insects frequently found in agricultural settings are just a few of the architectural improvements that YOLOv10 brings over its predecessor.

YOLOv10 significantly reduces processing requirements and increases the system's appropriateness for deployment on real-time edge devices like field cameras, drones, or mobile applications by providing improved detection accuracy together with better model scaling and more efficient inference. The proposed system architecture covers the entire detection pipeline, including data preparation, real-time image and video processing, model training and validation, and bounding box visualization of detection results. By incorporating these developments, the system improves robustness against changing illumination, complex backgrounds, and other real-world difficulties in addition to increasing accuracy and speed. In the end, our generalized, real-time pest detection system seeks to facilitate more scalable and sustainable pest control within precision agriculture, lower crop losses, and assist farmers in making quicker, data-driven decisions.

2.1.1 PROPOSED SYSTEM ADVANTAGES:

- **Identification of all insects in a range of crops:**
The model can be deployed to different crops and regions without requiring retraining because it was trained to recognize insects as a single, broad class.
- **Improved accuracy for small, hidden, and clustered insects:**

The enhanced multi-scale feature extraction and architectural enhancements in YOLOv10 enable the detection of insects that are partially or completely hidden.

➤ **Quicker inference and real-time field use optimization:**

Real-time monitoring on live video feeds is made possible by reduced processing time due to improved efficiency and anchor-free architecture.

➤ **It is appropriate for edge devices due to its lower computational requirements:**

Deployment on drones, cellphones, and field cameras with constrained hardware is made possible by optimized scaling and lightweight architecture.

➤ **Increased resilience under a variety of lighting conditions and complicated backgrounds:**

Even in busy agricultural areas and unpredictable outside situations, detection becomes more stable with improved feature representation.

2.2 METHODOLOGIES

2.2.1 MODULES

1) Object Detection Engine:

Finding and recognizing insects in images and video feeds is the responsibility of this module, which forms the basis of the system. This engine uses the advanced YOLOv10 architecture to identify the recognized objects and locate their exact locations in the image. It is ideal for real-time pest detection applications because of its swift processing speed and ability to recognize many insects simultaneously, ensuring that infestations in agricultural fields are discovered promptly.

2) Model Adaptation through Pre-trained Networks:

This module employs transfer learning to enhance model performance using pre-trained YOLOv10 networks. A model that was first trained on big, wide datasets is refined using a domain-specific dataset of agricultural insects. This allows the algorithm to efficiently learn pest-specific features, which expedites model training and removes the need for large annotated datasets.

3) Performance Tuning Module:

This module makes sure the model converges precisely and effectively by concentrating on optimizing hyperparameters like learning rate, batch size, and number of epochs. By fine-tuning these settings, detection precision can be increased, allowing the system to function consistently in a variety of field circumstances and insect species.

4) Data Expansion and Augmentation:

To expand the dataset and strengthen the model's resistance to actual agricultural settings, this module makes changes including rotation, scaling, flipping, and illumination or contrast adjustments. This enhancement guarantees reliable pest detection even in a range of lighting, weather, and camera angles by enhancing the system's generalization.

5) Model Training and Validation:

This module assesses the model's performance on independent validation data and oversees the entire training process on labeled insect datasets. The model gains the ability to precisely identify and categorize insect traits through this procedure. Validation guarantees that the model is not overfitting and has good generalization, which makes it useful for actual crop monitoring.

6) Real-time Detection and Localization:

This module allows real-time pest detection from incoming video or image streams after training. Insects are instantaneously detected and their locations within the frame are marked, giving farmers or automated systems useful information to take immediate action.

7) Performance Evaluation and Monitoring:

This module uses common measures including accuracy, precision, recall, and F1-score to assess system performance. It guarantees that the system satisfies anticipated standards and offers suggestions for additional enhancements. In order to preserve effectiveness and scalability in production deployments, it also keeps an eye on system performance under large-scale, real-time data situations.

2.3 TECHNIQUE / ALGORITHM USED YOLOV8

The YOLOv8s object detection framework, which stands for "You Only Look Once version 8 small," is the foundation of the current precision agriculture pest detection system. YOLOv8s, a member of the single-stage detector family, is well-liked for real-time applications in industries like industrial inspection, traffic surveillance, and agricultural because it strikes the ideal balance between speed, precision, and computing economy. YOLOv8s simultaneously predicts item locations (bounding boxes) and their related class probabilities by processing an entire image in a single neural network forward pass. It can identify several insects simultaneously thanks to this feature, which facilitates quick decision assistance in pest management situations. Convolutional neural networks (CNNs) are used in the architecture to extract spatial information from images, which are then sent through a detection head to produce bounding boxes and confidence ratings. By adding architectural improvements such as a detached head, enhanced feature fusion, and improved backbone architecture, YOLOv8s outperforms previous iterations of YOLO, resulting in higher mean Average Precision (mAP) and quicker training convergence.

YOLOv8s models are usually trained on crop-specific datasets with annotated photos of target insect species in the context of pest identification. When identifying those recognized pests under controlled circumstances, this focused method aids in achieving high accuracy. But there are drawbacks to this specialization. First, in complex real-world field settings where lighting varies, insects are partially obscured by plants, or several insects overlap within a

single image, detection accuracy frequently declines. Second, the model may not be able to identify novel pests or insects from other crops due to the dependence on species-specific datasets, which restricts generalization. Third, real-time deployment on resource-constrained edge devices, such as drones or low-power cameras, can still pose computational issues even though YOLOv8s is lighter than larger models (such as YOLOv8m or YOLOv8l). Lastly, YOLOv8s models typically need to be retrained or fine-tuned in order to incorporate new pest species or adapt to different agricultural locations, which raises the cost of data collecting and training. Despite being one of the most sophisticated real-time object detection models, YOLOv8 still has a number of drawbacks that may impair how well it works in some situations. Its comparatively high processing and memory requirements, especially for larger model versions, are a significant disadvantage. Because of this, deployment on low-power edge computing platforms, embedded systems, and Internet of Things devices may be challenging. YOLOv8 may also struggle to recognize very small objects in complex or congested surroundings where important features may be lost during feature extraction and down-sampling processes. Another disadvantage is that it requires large, well-annotated datasets for effective training. Inadequate or uneven training data might lead to poor generalization and reduced accuracy when applied to real-world scenarios. Furthermore, even though YOLOv8 has a high detection speed, powerful GPU hardware is frequently needed to maintain real-time performance with high-resolution photos, which raises deployment costs. Additionally, the model may be susceptible to changes in lighting, motion blur, object occlusions, and unfavorable weather, all of which could lower identification reliability. Moreover, YOLOv8 still introduces extra inference overhead by using post-processing methods like Non-Maximum Suppression (NMS) to remove duplicate detections. These drawbacks emphasize the need for additional efficiency, robustness, and scalability optimization for a range of real-world applications.

2.3.2 PROPOSED TECHNIQUE / ALGORITHM YOLOv10:

By utilizing the most recent YOLOv10 object detection framework, the suggested method improves insect detection in precision agriculture. Within the YOLO family, YOLOv10 is a major evolution that introduces sophisticated architectural enhancements to boost efficiency, accuracy, and resilience. It is especially made to manage challenging detection situations, such as those found in actual agricultural fields, where insects may be present against extremely cluttered natural backgrounds, appear in different sizes, or be partially obscured. The anchor-free detection technique of YOLOv10 is one of its main innovations. The anchor-free design enables the model to detect insects with greater flexibility and accuracy than standard

anchor-based approaches, which rely on predefined box shapes and may have trouble with small or odd object sizes. Insects are frequently small, overlapping, or occur in bunches, making this especially helpful for pest detection. In order to efficiently extract multi-scale features, the model's core is additionally enhanced by adding deeper layers and improved feature aggregation. This aids in capturing the minute details required to identify tiny pests without compromising overall speed. The proposed system is trained using a broad detection strategy that identifies all bug varieties under a single "insect" class, rather than concentrating on specific species. This approach overcomes the shortcomings of earlier models by enabling the system to detect any bug across various crops and regions without retraining for each new pest kind. A pre-trained YOLOv10 model that has been trained on extensive, diverse datasets is first utilized in order to perform transfer learning. The model is then improved using a smaller, domain-specific pest dataset. Examples of data augmentation techniques that improve the training data and increase the model's capacity to generalize to new field conditions include rotations, scaling, flipping, and brightness adjustments. Performance optimization is another crucial part of the system. Hyperparameters such as learning rate, batch size, and number of epochs are carefully adjusted to provide efficient convergence and excellent detection accuracy. The technology locates and identifies insects in incoming video or picture streams in real time after being trained. Because it processes data quickly enough to be shown on edge devices like drones, field cameras, or mobile applications, farmers can receive real-time pest alerts. In order to guarantee the system's resilience and scalability in actual agricultural environments, it is lastly put through a thorough examination utilizing common metrics, such as accuracy, precision, recall, and F1-score. YOLOv10 provides a more reliable, scalable, and generalized pest detection solution by including these enhancements. It helps data-driven pest management, enables farmers to react quickly to pest threats, and advances the more general objective of precision-driven, sustainable agriculture.

YOLOv10 is a sophisticated real-time object detection method designed to outperform earlier YOLO iterations in terms of both detection accuracy and inference speed. It is appropriate for edge devices, surveillance systems, autonomous vehicles, healthcare applications, and smart city environments since it is made to effectively recognize and localize various objects inside pictures or video streams while keeping low computational needs.

YOLOv10's end-to-end object detection architecture, which does away with the necessity of conventional Non-Maximum Suppression (NMS) during inference, is its primary novelty. NMS was employed in previous object identification models to eliminate redundant bounding box predictions, but it added latency and computational overhead. By using a consistent dual assignment technique, YOLOv10 overcomes this

constraint and allows the model to directly produce precise and non-redundant predictions.

Optimized feature extraction layers, improved attention mechanisms, and effective backbone and neck designs are just a few of the architectural enhancements included in YOLOv10. These improvements improve detection performance for small, medium, and big objects by enabling the model to efficiently capture both local and global picture data. Additionally, the architecture makes use of cutting-edge training techniques that improve convergence speed and generalization capacity across various datasets.

YOLOv10's scalability is another important benefit. Various model variations, from lightweight to high-performance versions, enable deployment on both high-end computing platforms and devices with limited resources. Because of its adaptability, YOLOv10 is appropriate for real-time applications where quick decision-making is crucial.

When compared to a number of earlier YOLO models, YOLOv10 has shown better performance in real-world applications in terms of mean Average Precision (mAP), detection speed, and computing economy. It is especially useful for applications like weapon detection, traffic monitoring, industrial automation, medical image analysis, and security surveillance because of its capacity to process images in real time while maintaining high accuracy.

Overall, by integrating speed, accuracy, and efficiency into a single framework, YOLOv10 marks a substantial breakthrough in object identification technology. It is one of the best deep learning models for contemporary computer vision problems because of its end-to-end design and removal of post-processing bottlenecks.

3. SYSTEM ARCHITECTURE:

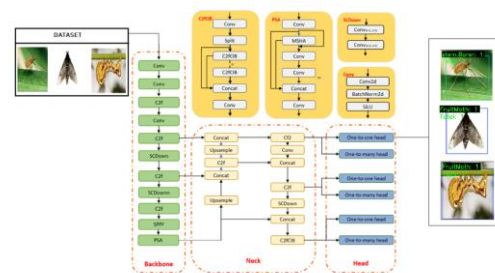


Fig 3.1: System Architecture

Bottleneck blocks are used by the C2f blocks, occasionally with a shortcut. These blocks are designed to reduce the quantity of data, which will enhance computational efficiency and reduce computational load. This is achieved by utilizing 1×1 convolutional layers to compress and then enlarge the input's volume. These are the light orange blocks in Figure 1. Concatenation layers are used to merge inputs from different network depths. However, when the input passes through the neural network, dimensions often change due to depth expansion or spatial resolution

loss. Up sample layers are used to make it easier to add inputs from different depths. In order to match a particular target resolution, these layers interpolate values. The learning of intricate and hierarchical representations is made possible by the C2f block, which uses bottleneck blocks to improve computing performance. The procedure culminates in a final Spatial Pyramid Pooling block, represented by the medium blue block in Figure 1, following the C2f block. Regardless of their size or location, the goal of spatial pyramid pooling is to collect data from various areas of an image. To ensure that contextual information is captured across many scales, the input image is divided into multiple levels and then pooled. This block divides the output into three paths after compressing the input using a 2D convolutional layer. The Max Pooling layer then addresses these outputs and concatenates them with a non-pooled output. Pooling layers are commonly used to reduce the input's dimensions, which results in information loss but speeds up calculations. Each channel of the input receives an individual application of the pooling function, often the maximum or average, which solely reduces the spatial dimensions. The combined result is then input into a 2D convolutional layer to increase the dimensionality once again. Through this procedure, the model is able to extract characteristics from several spatial dimensions and refine them to retrieve pertinent data.

4. DATASET ANALYSIS AND METRICS

In this section, we evaluate the selected dataset and detail the metrics employed to obtain the performance of the object detection model.

A. DATASET ANALYSIS

To efficiently train an object detection model to any specific task, it is necessary to properly feed images to this model. Insects are highly biodiverse [31], making their detection a challenge as it is complex to generalize physical characteristics, species similarities, and morphological differences in such a heterogeneous group. The scarce availability of public datasets referring to insects, along with the low number of images in these datasets, have intensified the effort to develop this work. Furthermore, laboratory imagery, with good light conditions and white backgrounds, does not suitably adapt to real life conditions. Some approaches to insect detection rely on traps, and the images available in these datasets are only appropriate for studying this specific situation. Even with images obtained with cameras or drones in the field, the datasets tend to focus on specific insects or pests of interest in particular crops. These circumstances only add more difficulty to an already complex issue. scale dataset of insects. This dataset contains 102 classes, which is appropriate for classification and detection tasks. Our main goal, however, is not detecting specific instances of insects but the presence of pests. Samples of the dataset images can be found in Figure 4.1.

All the classes in the original dataset have been merged into a superclass called “insect.” This approach solves the problem of class imbalance caused by a different number of samples per insect type, but introduces significant variance, that is, the resulting class becomes too heterogeneous. This is caused by the substantial difference in the number of images belonging to each class, and the model tends to predict one image as belonging to the class with the most images. To accomplish this merging process, all the images are assigned the same *class id* in their labels. The labels take the YOLO format, encompassing essential information for object classification and detection.



Fig 4.1 Example images of the IP102 dataset. Each image belongs to a different species of insect

This format includes the following key parameters: class id, a unique number assigned to each class; *x center* and *y center*, the coordinates of the bounding box center; and the width and height of the bounding box. These parameters play a crucial role in object classification and detection within an image. The detection process implies creating a bounding box, a rectangular enclosure that precisely delineates the object of interest. This bounding box serves as a spatial reference, encapsulating the detected object and facilitating subsequent analyses or actions based on its identified location.

5. RESULTS

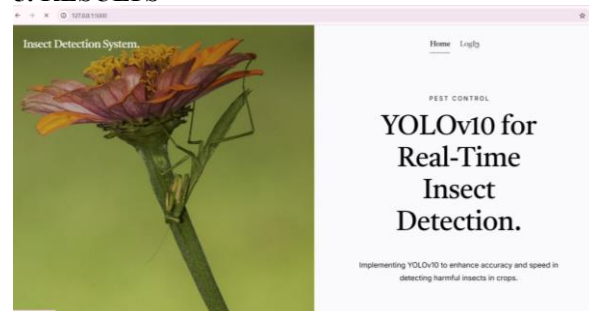


Fig 1: Title Page of Project

Please fill the details.

Your Name

Your Email

Your Password

Confirm Password

SUBMIT

Fig 2: Registration Page

Provide Login details.

admin

SUBMIT

Fig 3: Login Page

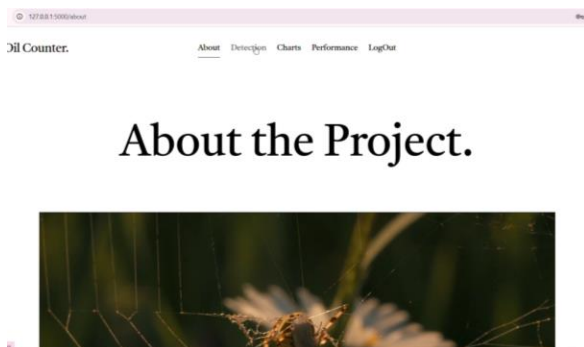


Fig 4: Brief description about Project

Detection System.

About Detection Charts Performance LogOut

Please Upload Image or Video for Detection.

Fig 5: Input page to insert image for testing

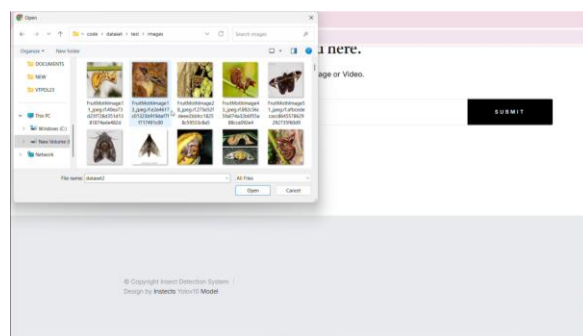


Fig 6: Select an input for Pest detection

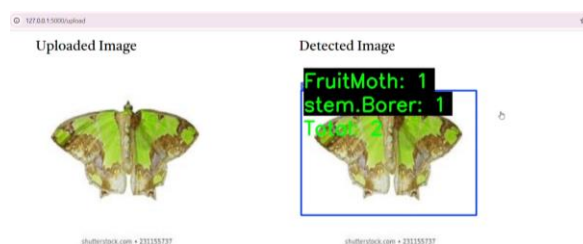


Fig 7: Pest detected output, with type mentioned

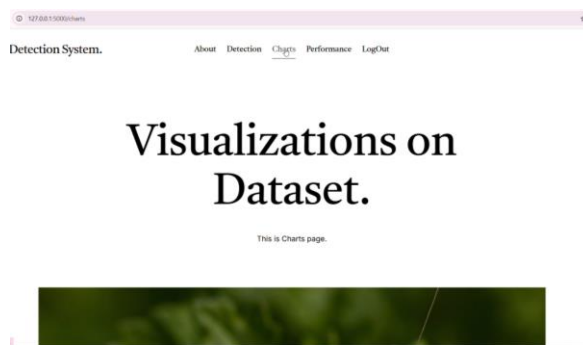


Fig 8: Data visualization for Training of Dataset



Fig 9: training dataset with 4 different classes

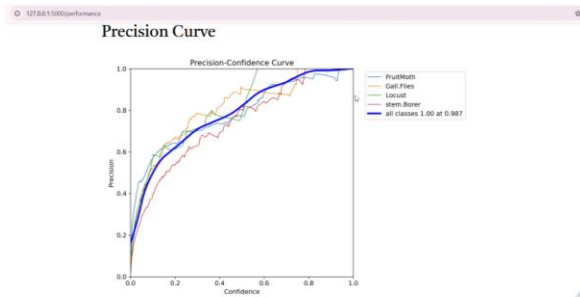


Fig 10: precision curve for the trained classes

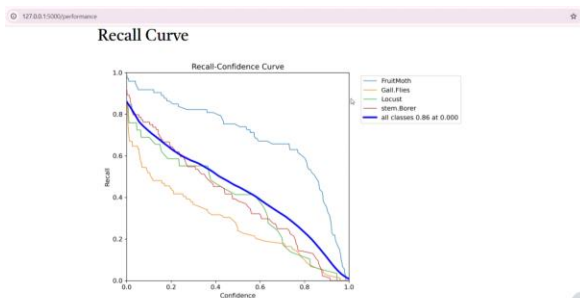


Fig 10: Recall curve for the trained classes

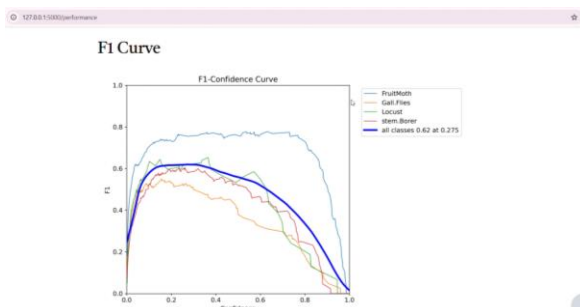


Fig 10: F1-Curve for the trained classes

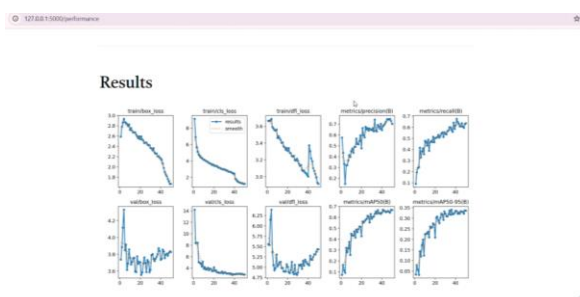


Fig 13: Analysis results for Training and Validation.

6. FUTURE ENHANCEMENTS:

Even if the suggested YOLOv10-based pest detection system is a major improvement over current techniques, there are still a number of chances for development and growth. Extending the system from generalized insect detection to fine-grained

classification is one prospective future improvement that will allow it to identify individual insect species and classify them based on their growth phases or threat levels in addition to detecting the presence of pests. This skill would help farmers make more accurate decisions and enable more focused pest management techniques. The combination of temporal analysis and video-based tracking techniques is another crucial path. The technology may track insect movement patterns over time by evaluating frame sequences rather than individual photos, assisting in the prediction of infestation spread and providing early warning notifications. This improvement may be especially useful when used in conjunction with crop growth and weather data to create predictive models that dynamically evaluate insect risk.

Deploying the detection system as a component of an edge-cloud hybrid architecture might be investigated in terms of technical scalability. Large-scale data storage, long-term trend analysis, and heavy analytics might be offloaded to cloud servers, while real-time detection would still operate on lightweight edge devices like drones or field cameras. Without compromising field-side detection speed, this hybrid solution would enable farmers to obtain sophisticated insights via dashboards or mobile apps. Integrating multi-modal data sources, such as environmental sensors that measure soil moisture, temperature, and humidity, could also be beneficial to the project. Smarter, automated pest management interventions, such as turning on pesticide sprayers only when needed, could be made possible by combining these sensor readings with visual insect detection data to create a richer context for pest risk assessment.

7. CONCLUSION

Crop pests can seriously harm crops, lowering output and raising concerns about further funding for PA systems. In this research, we examined the difficulties in detecting PA insects and evaluated the effectiveness of the YOLOv8 architecture, an open-source deep-learning (CCN) system, in this particular field. The intricacies of insect detection that result from species differences, environmental factors, and geographical viewpoints have been thoroughly investigated. Additionally, it is more difficult to take pictures of insects in natural settings without using traps or other devices because of a number of factors, including the insect's size in relation to the plant's leaves, obstructions from leaves and branches, inaccessible insect positioning on the underside of leaves and lower parts of the plants, variations in lighting, and weather conditions. These elements may interfere with an AI model's ability to function effectively by reducing the output image's visibility and clarity. By applying data augmentation techniques to change the photos into a more appropriate input for the model, this can be lessened. Furthermore, even when an AI model receives a sufficient input image from the data acquisition device or the result of data

augmentation techniques, there is an additional challenge that could make the model useless for farmers if the solution is based on classifying the insect from a smaller number of possible insect types. The output would indicate that there are no insects present when, in fact, there are if an insect is photographed and the model is unable to categorize it into one of the insect classes. Instead of classifying insects, the method described in this work uses a general strategy. Farmers would be informed when any kind of pest was present, preventing them from feeling overconfident. All the insect classes in the collection have been pooled into a single insect dataset to enable insect detection for general purpose. By extending the model's real-time detection capabilities to: i) incorporate insect species outside of the dataset and ii) take into account any arbitrary crop type, this generalization offers greater flexibility and allows the model to respond to a larger range of settings. The thorough examination of the dataset has yielded insightful information about the constraints and possible deviations that could affect the model's performance. With a mAP0.95 metric value of 0.632 for model l and a mAP0.5 value of 0.967 for model m, YOLOv8 showed acceptable performance in some instances, but difficulties emerged when handling variations in spatial distances. This emphasizes how crucial it is to include variety in the dataset in order to generalize the model in various circumstances. The results highlight the necessity of constantly improving detection models in light of the dynamic nature of datasets pertaining to insects. Thus, by highlighting the significance of the kind of information in the dataset and model architectural concerns, our study adds significant insights to the continuing research on insect detection approaches.

REFERENCES

- [1] B. Davis, E. Mane, L. Y. Gurbuzer, G. Caivano, N. Piedrahita, K. Schneider, N. Azhar, M. Benali, N. Chaudhary, R. Rivera, R. Ambikapathi, and P. Winters, Estimating Global and Country Level Employment in Agri Food Systems (FAO Statistics Working Paper Series). Rome, Italy: FAO, 2023, Doi: 10.4060/cc4337en.
- [2] Pest and Pesticide Management, Food and Agriculture Organization of the United Nations. Accessed: Jan. 10, 2024.
- [3] R. Kestur, S. N. Omar, and S. Subhash, "Unmanned aerial system technologies for pesticide spraying," in *Innovative Pest Management Approaches for the 21st Century*. Singapore: Springer, 2020, pp. 37–60, Doi: 10.1007/978-981-15-0794-6_3.
- [4] M. Abbas, M. Ramzan, N. Hussain, A. Ghaffar, K. Hussain, S. Abbas, and A. Raza, "Role of light traps in attracting, killing and biodiversity studies of insect pests in Thal," *Pakistan J. Agricult. Res.*, vol. 32, no. 4, pp. 684–690, 2019.
- [5] P. Trematerra and M. Colacci, "Recent advances in management by pheromones of thaumetopoea moths in urban parks and woodland recreational areas," *Insects*, vol. 10, no. 11, p. 395, Nov. 2019, Doi: 10.3390/insects10110395.
- [6] T. R. E. Southwood and P. A. Henderson, *Ecological Methods*. Hoboken, NJ, USA: Wiley, 2009.
- [7] K. Ennouri, S. Smaoui, Y. Gharbi, M. Cheffi, O. Ben Braiek, M. Ennouri, and M. A. Triki, "Usage of artificial intelligence and remote sensing as efficient devices to increase agricultural system yields," *J. Food Qual.*, vol. 2021, pp. 1–17, Jun. 2021, Doi: 10.1155/2021/6242288.
- [8] H. Tian, T. Wang, Y. Liu, X. Qiao, and Y. Li, "Computer vision technology in agricultural automation—A reviews. *Process. Agricult.*, vol. 7, no. 1, pp. 1–19, Mar. 2020, Doi: 10.1016/j.inpa.2019.09.006.
- [9] Y. Luand S. Young, "A survey of public datasets for computer vision tasks in precision agriculture," *Comput. Electron. Agricult.*, vol. 178, Nov. 2020, Art. no. 105760, Doi: 10.1016/j.compag.2020.105760.
- [10] W. Zhang, Z. Miao, N. Li, C. He, and T. Sun, "Review of current robotic approaches for precision weed management," *Current Robot. Rep.*, vol. 3, no. 3, pp. 139–151, Jul. 2022, Doi: 10.1007/s43154-022-00086-5.
- [11] H. C. Bazame, J. P. Molin, D. Althoff, and M. Martello, "Detection, classification, and mapping of coffee fruits during harvest with computer vision," *Comput. Electron. Agricult.*, vol. 183, Apr. 2021, Art. no. 106066, Doi: 10.1016/j.compag.2021.106066.
- [12] A. Gutierrez, A. Ansuategi, L. Susperregi, C. Tubío, I. Rankić, and L. Lenža, "A benchmarking of learning strategies for pest detection and identification on tomato plants for autonomous scouting robots using internal databases," *J. Sensors*, vol. 2019, pp. 1–15, May 2019, Doi: 10.1155/2019/5219471.
- [13] W. Li, D. Wang, M. Li, Y. Gao, J. Wu, and X. Yang, "Field detection of tiny pests from sticky trap images using deep learning in agricultural greenhouse," *Comput. Electron. Agricult.*, vol. 183, Apr. 2021, Art. no. 106048, Doi: 10.1016/j.compag.2021.106048.
- [14] M. E. Karar, F. Alsunaydi, S. Albusaymi, and S. Alotaibi, "A new mobile application of agricultural pests' recognition using deep learning in cloud computing system," *Alexandria Eng. J.*, vol. 60, no. 5, pp. 4423–44432, Oct. 2021, doi: 10.1016/j.aej.2021.03.009.
- [15] I. Ahmad, Y. Yang, Y. Yue, C. Ye, M. Hassan, X. Cheng, Y. Wu, and Y. Zhang, "Deep learning-based detector YOLOv5 for identifying insect pests," *Appl. Sci.*, vol. 12, no. 19, p. 10167, Oct. 2022, doi: 10.3390/app121910167.
- [16] J. Du, "Understanding of object detection based on CNN family and YOLO," *J. Phys., Conf. Ser.*, vol. 1004, Apr. 2018, Art. no. 012029, doi: 10.1088/1742-6596/1004/1/012029.
- [17] J. Terven, D.-M. Córdova-Esparza, and J.-A. Romero-González, "A comprehensive review of YOLO architectures in computer vision: From YOLOv1 to YOLOv8 and YOLO-NAS," *Mach. Learn.*

- Knowl. Extraction, vol. 5, no. 4, pp. 1680–1716, Nov. 2023, doi: 10.3390/make5040083.
- [18] K. Li, J. Zhu, and N. Li, “Insect detection and counting based on YOLOv3 model,” in Proc. IEEE 4th Int. Conf. Electron. Technol. (ICET), Chengdu, China, Sep. 2021, pp. 1229–1233, doi: 10.1109/ICET51757.2021.9450898.
- [19] E. Önlü, “Real time pest detection using YOLOv5,” Int. J. Agricult. Natural Sci., vol. 14, no. 3, pp. 232–246, 2021.
- [20] Z. Yang, H. Feng, Y. Ruan, and X. Weng, “Tea tree pest detection algorithm based on improved YOLOv7-tiny,” Agriculture, vol. 13, no. 5, p. 1031, May 2023, doi: 10.3390/agriculture13051031.
- [21] G. Jocher, A. Chaurasia, and J. Qi. YOLO By Ultralytics. Accessed: Nov. 6, 2023.
- [22] T. Kasinathan, D. Singaraju, and S. R. Uyyala, “Insect classification and detection in field crops using modern machine learning techniques,” Inf. Process. Agricult., vol. 8, no. 3, pp. 446–457, Sep. 2021, doi: 10.1016/j.inpa.2020.09.006.
- [23] Y. He, H. Zeng, Y. Fan, S. Ji, and J. Wu, “Application of deep learning in integrated pest management: A real-time system for detection and diagnosis of oilseed rape pests,” Mobile Inf. Syst., vol. 2019, pp. 1–14, Jul. 2019, doi: 10.1155/2019/4570808.
- [24] A. C. Teixeira, J. Ribeiro, R. Morais, J. J. Sousa, and A. Cunha, “A systematic review on automatic insect detection using deep learning,” Agriculture, vol. 13, no. 3, p. 713, Mar. 2023, doi: 10.3390/agriculture13030713.
- [25] Y. He, Z. Zhou, L. Tian, Y. Liu, and X. Luo, “Brown Rice planthopper (*Nilaparvata ugens* Stal) detection based on deep learning,” Precis. Agricult., vol. 21, no. 6, pp. 1385–1402, Dec. 2020, doi: 10.1007/s11119020-09726-2.
- [26] J.-W. Chen, W.-J. Lin, H.-J. Cheng, C.-L. Hung, C.-Y. Lin, and S.-P. Chen, “A smartphone-based application for scale pest detection using multiple object detection methods,” Electronics, vol. 10, no. 4, p. 372, Feb. 2021, doi: 10.3390/electronics10040372.
- [27] C.-J. Chen, Y.-Y. Huang, Y.-S. Li, C.-Y. Chang, and Y.-M. Huang, “An IoT based smart agricultural system for pests’ detection,” IEEE Access, vol. 8, pp. 180750–180761, 2020, doi: 10.1109/ACCESS.2020.3024891.
- [28] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Las Vegas, NV, USA, Jun. 2016, pp. 779–788, doi: 10.1109/CVPR.2016.91.
- [29] (2023). YOLOv8 Architecture Visualization, Range King.
- [30] D. Hendrycks and K. Gimpel, “Gaussian error linear units (GELUs),” 2016, arXiv:1606.08415.
- [31] N. E. Stork, “World of insects,” Nature, vol. 448, no. 7154, pp. 657–658, Aug. 2007, doi: 10.1038/448657a. 84433 VOLUME 12, 2024M. Vilar-Andreu et al.: Enhancing Precision Agriculture Pest Control
- [32] X. Wu, C. Zhan, Y. Lai, M. M. Cheng, and J. Yang, “IP102: A large-scale benchmark dataset for insect pest recognition,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Long Beach, CA, USA, Jun. 2019, pp. 8787–8796, doi: 10.1109/CVPR.2019.00899.
- [33] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, “YOLOv4: Optimal speed and accuracy of object detection,” 2020, arXiv:2004.10934.
- [34] G. H. Joblove and D. Greenberg, “Colors spaces for computer graphics,” ACM SIGGRAPH Comput. Graph., vol. 12, no. 3, pp. 20–25, Aug. 1978, doi: 10.1145/965139.807362.
- [35] H. Zhao, H. Li, and L. Cheng, “Data augmentation for medical image analysis,” in Biomedical Image Synthesis and Simulation, N. Burgos and D. Svoboda, E.Ds. Cambridge, MA, USA: Academic, 2022, pp. 279–302.
- [36] Y. Cha, W. Choi, G. Suh, S. Mahmoud Khani, and O. Büyükoztürk, “Autonomous structural visual inspection using region-based deep learning for detecting multiple damage types,” Comput. -Aided Civil Infrastructure. Eng., vol. 33, no. 9, pp. 731–747, Sep. 2018, doi: 10.1111/mice.12334.
- [37] X. Li, C. Lv, W. Wang, G. Li, L. Yang, and J. Yang, “Generalized focal loss: Towards efficient representation learning for dense object detection,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 45, no. 3, pp. 3139–3153, Mar. 2023, doi: 10.1109/TPAMI.2022.3180392.
- [38] C. M. Bishop, Pattern Recognition and Machine Learning (Information Science and Statistics), 1st ed. New York, NY, USA: Springer, 2007.
- [39] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn and A. Zisserman. (2010). The PASCAL Visual Object Classes Challenge 2010 (VOC2010) Results.
- [40] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. Lawrence Zitnick, and P. Dollar, “Microsoft COCO: Common objects in context,” 2014, arXiv:1405.0312.