Enhanced Locust Detection in Smart Farming Using YOLOv5 and YOLOv8 with Data Augmentation: A Comparative Performance Evaluation

Pooja Vajpayee
Department of Computer Science, Banasthali Vidyapith, Rajasthan, India
pgphb20113_pooja@banasthali.in (corresponding author)

Kuldeep Kr. Yogi
Department of Computer Science, Banasthali Vidyapith, Rajasthan, India
ykuldeep@banasthali.in

Amit Kumar
School of Computer Science and Engineering, Galgotias University, Uttar Pradesh, India
amitbsbkumar@gmail.com

Received: 30 April 2025 | Revised: 22 June 2025 | Accepted: 3 July 2025

ABSTRACT

Deep learning-based object detection models have emerged as powerful tools for real-time pest monitoring in agriculture. Locust swarms pose a severe threat to crops, necessitating prompt and accurate detection. This study evaluates the performance of two state-of-the-art object detection models, YOLOv5 and YOLOv8, for locust detection in smart farming. After training and testing these models on a custom dataset, their accuracy, precision, recall, mean average precision (mAP), and F1 score are compared. The results demonstrate that YOLOv8 slightly outperforms YOLOv5 in terms of accuracy and recall, while YOLOv5 is faster and requires fewer resources. In addition to baseline testing, data augmentation techniques are applied, leading to improved model performance and accuracy. YOLOv8 achieved an accuracy of 79.80% after data enhancement, highlighting its potential for real-time locust detection. This comparative analysis provides valuable insights into the strengths and weaknesses of each model, which can contribute to smart farming initiatives, aiding in the selection of the most suitable tool for sustainable agriculture and food security, such as pest monitoring and management.

Keywords-YOLOv5; YOLOv8; data augmentation; locust detection; object detection

I. INTRODUCTION

Deep learning is a specific subfield of machine learning that implements multi-layered neural networks for identifying patterns autonomously within enormous datasets, making natural language processing and image analysis easier. Among the various applications of deep learning, object detection is a distinct function that not only recognizes but also pinpoints objects in images or videos, enclosing them with bounding boxes to indicate their exact locations [1-3]. This technology plays a crucial role in sectors such as agriculture, where the detection of locusts is critical due to their devastating effects on crops. Locust swarms can decimate large tracts of agricultural cultivation in a short time, endangering food security and causing enormous financial losses.

Early and accurate locust detection can reduce the adverse impact of locusts on the crops and also encourage swift preventive action, which can lead to significantly less harmful impact on crops [4, 5]. Timely locust identification is critical as it provides a fair chance for farmers and agricultural managers to separate the infested areas or to implement focused control measures, which can mitigate the infestation and protect the cultivation. In addition to confirming the presence of locusts, deep learning methods can recommend suitable actions depending on the extent and severity of the spread [6-8].

This study examines the detection efficiency of two YOLO model versions, YOLOv5 and YOLOv8, because both variants are widely popular and common for object detection.

A. Problem Statement

Locust manifestations cause significant damage to the total agricultural productivity, which is a serious problem for food security and the world economy. An effective and reliable detection technique is required to monitor and mitigate locust populations. Conventional methods, which require human intervention, are very time-consuming, more susceptible to error, and do not frequently provide real-time insights. This study aimed to address these limitations by curating a dataset and then developing a robust locust detection framework using a popular object detection method, YOLO. The models were also trained and tested on an augmented dataset to examine the impact of data augmentation on model performance.

B. Aims and Objectives

This work applies both YOLOv5 and YOLOv8 to a curated locust detection dataset, which is publicly accessible for reproduction and further analysis purposes. Then, data augmentation techniques are applied to further enhance and compare the performance metrics of these two models. Data augmentation techniques involve changes such as rotations, flips, and brightness adjustments in images to artificially increase the size and variety of the training dataset [9]. In this way, models can generalize better in different locust appearances and environmental conditions. This study involves:

- Preparing and augmenting a locust detection dataset.
- Training and fine-tuning YOLOv5 and YOLOv8 models for locust detection.
- Evaluating models using precision, recall, mAP, F1 score, and accuracy.
- Analyzing the impact of data augmentation on model performance.
- Comparing the accuracy and efficiency of YOLOv5 and YOLOv8.

This study aimed to go beyond simply identifying locusts, facilitating prompt and accurate locust detection to improve agricultural production and promote crop protection.

II. RELATED WORK

Several recent developments in smart farming technologies offer new research opportunities, especially in the area of object detection. Many techniques have been developed and proposed for object detection, and this section presents the latest relevant developments in the field. In [10], the performance of deep learning methods, particularly YOLOv5 and YOLOv8, was assessed in the identification of various tomato diseases using a tailored dataset consisting of approximately 3,000 images. YOLOv8 consistently demonstrated superior performance compared to YOLOv5 in both precision and recall, especially after data augmentation, which substantially improved the accuracy for both models. This study highlights the crucial role of data augmentation techniques, which improved the detection of diseases in tomatoes.

In [11], the effectiveness of YOLOv5 and YOLOv8 was examined for vehicle and license plate detection within intelligent transportation systems, using a dataset of 4,075 annotated images. YOLOv8 exceeded YOLOv5 in terms of accuracy (97.98 vs. 97.83%), recall, and F1-score, while also requiring less training time. In [12], various YOLOv8 models used for real-time traffic sign recognition were outlined, focusing on their features, performance metrics, and the training methods employed. This study underscored the significance of precise image annotation and hyperparameter adjustment to enhance model performance. The YOLOv8 nano model was the most effective for real-time use, achieving impressive accuracy in recognizing traffic signs. In [13], YOLOv8 models generally surpassed YOLOv5 models regarding precision, recall, F1-score, and mean average precision in corrosion segmentation tasks. Although there were indications of overfitting, YOLOv8 was suggested for practical applications that demand efficient and accurate corrosion detection.

In [14], an innovative approach was based on Faster R CNN to enhance grasshopper detection in smart crops, addressing the constraints of previous pest identification methods. This study achieved an accuracy of 98.62%, outperforming traditional methods, and emphasized the importance of data preprocessing and the model's potential for integration into smart agricultural monitoring systems. In [15], a novel annotated dataset was introduced, focused on weed detection in Indian cotton fields. This dataset consisted of 2300 images and 44,130 bounding boxes, spanning three categories. A total of 12 YOLO-based models (including YOLOv5 and YOLOv8) were assessed, with YOLOv5s6 delivering the best accuracy (76.50% mAP@0.5) and YOLOv5n offering the quickest inference. The findings demonstrate the effectiveness of these models for real-time weed identification, aiding in the advancement of precision farming techniques.

Rice Pest-YOLO [16] is an improved version of YOLOv8n adapted for the precise and efficient detection of rice pests. This approach incorporates ODConv, BiFPN, and Shape-IoU to improve detection accuracy, achieving 94.3% mAP@0.5 with reduced computational load. Designed for real-time use, it outperformed existing models and was well-suited for deployment on lightweight devices. In [17], a real-time waste monitoring system used lightweight YOLOv5 and YOLOv8 models for accurate trash classification. This approach enabled efficient and low-cost garbage detection and alerting, and was suitable for deployment on devices such as Raspberry Pi.

III. MATERIALS

A. Data Collection

A dataset was curated by collecting several locust images, classified into two main categories based on the background of the image, as shown in Figures 1 [18] and 2 [19].

- Artificial and Blurred: This category contains images that display a close-up of a locust, and the focus is on its detailed body, including its legs, wings, and antennae. The background in these images is out of focus, which indicates that the image was taken to emphasize the locust in the foreground and isolate the subject from its surroundings. For this purpose, high-quality locust images were selected from Roboflow.

- Natural with visible soil: This category includes locust images that mainly focus on the environment or the entire scene and do not prominently feature a locust. The background in these images is fully in focus, and the soil texture and sparse vegetation are visible, with small green plants growing unevenly across the soil. This category aims to document the entire scene and not just a specific object. Some relevant locust images from the infested field were also selected, which were captured from various angles and are sourced from https://lcas.lincoln.ac.uk/ [19].

The dataset consists of a total of 1623 images, taken from various angles and selected to ensure a balanced representation.

Fig. 1. Artificial locust images where the background is blurred.

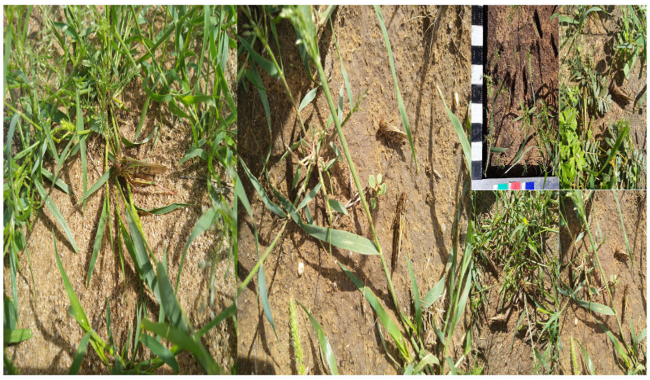
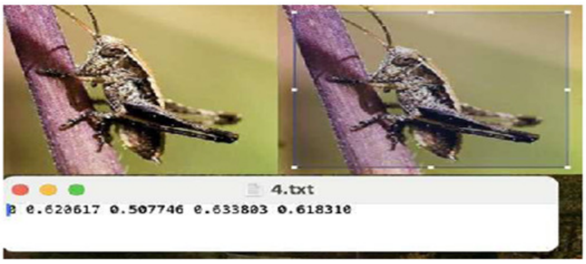


Fig. 2. Natural locust images with clear backgrounds.

B. Data Annotation

After collecting the dataset, image annotation was performed to assign labels to the images. Annotations help in training machine learning models to learn the contents of the image. The annotation process was performed using makesense.ai. Figure 3 shows an annotated image, offering insight into the bounding box drawn around every locust.



```
0 0.620617 0.507746 0.633803 0.618310
```

Fig. 3. Dataset image annotation.

C. Data Resizing

A Python program was used to resize all the images of various sizes to 300×300.

D. Data Preprocessing

To effectively train and evaluate the locust detection model, the curated dataset was cautiously partitioned into three divergent subparts.

E. Training

In the model training phase, the model learns to extract features and recognize key patterns, which are vital for identifying locusts. Thus, the training dataset is considered a pillar of the training phase.

F. Validation

The validation set is a key component to validate the model's performance during training. It ensures that the model generalizes well on unseen data through fine-tuning hyperparameters and by avoiding overfitting.

G. Testing

The conclusive performance of the model on unseen data is examined on the test set. This intended evaluation is a valid indication of how well the model is likely to function in a real-world scenario.

H. Data Augmentation

After labeling the dataset, data augmentation was applied. Data augmentation techniques are the most effective way to alleviate the need for huge training data. Data augmentation aims to expand the dataset by increasing its volume, quality, and content diversity [20]. The augmented images are derived by implementing some minor changes (such as flipping, rotating, cropping, brightness, and color shift) in the original image, as shown in Figure 4. Data augmentation is helpful to mitigate the overfitting problem when a model is being trained with a limited set of data.

TABLE I. DATASET DETAILS

Dataset	**Training images**	**Validation images**	**Testing images**	**Total images**	**Annotation type**
Non augmented	1210	341	72	1623	Bounding box
Augmented	3630	341	72	4043	Bounding box

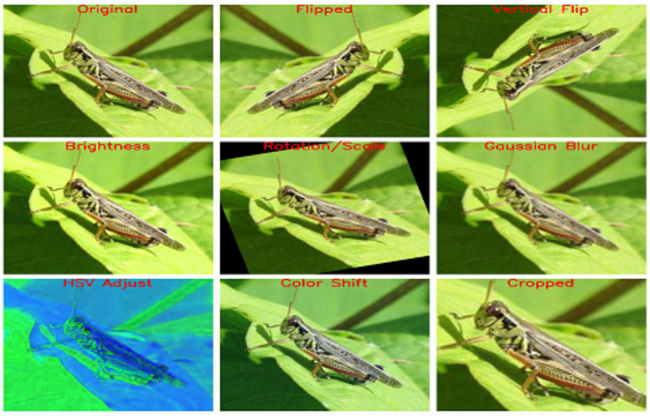


Fig. 4. Augmentation of dataset images.

IV. PROPOSED METHOD

A training environment was set up in Google Colab Pro, which offers free access to some powerful GPUs and accelerates training to offer more efficient results. A 15 GB NVIDIA Tesla T4 GPU was used for training and testing. The models were efficiently trained in roughly 0.746 hours for YOLOv8 and 0.833 hours for YOLOv5, with 50 epochs. The YOLOv5 and YOLOv8 models were separately trained to determine the suitability of the dataset. For this specific dataset, a comparative approach was implemented to examine the effectiveness and accuracy of each algorithm. Although both YOLOv5 and YOLOv8 are strong methods for object detection, they differ in their respective architecture, as illustrated in Table II, which eventually contributes to their unique features and drawbacks. In real-time applications where speed is essential, YOLOv8 could be an excellent option because of its faster inference time and lighter architecture. YOLOv5 has flexible backbone options and high accuracy with well-designed anchors, so it is mainly preferable for tasks that demand high accuracy and are not resource-constrained.

TABLE II. MAIN ARCHITECTURAL DIFFERENCES BETWEEN YOLOV5 AND YOLOV8 (ADAPTED FROM [13])

Object detector	**YOLOv5**	**YOLOv8**
Building block	C3	C2f
Detection head	Anchor Based	Anchor Free
Kernel size	6×6	3×3
Backbone output channels	128, 256, 512, 1024	128, 256, 512, 512

V. EXPERIMENTAL RESULTS AND ANALYSIS

This study evaluates two state-of-the-art YOLOv5 and YOLOv8 models for locust detection. The dataset used is publicly available in [21]. To enable reproduction of our results, the implementation code and training instructions used are publicly available in [22].

A. Experimental Environment

Table III shows the hardware and software parameters used for training and testing the models.

TABLE III. CONFIGURATION PARAMETERS OF THE EXPERIMENTAL SETUP

Specifications	**Details**
Graphics Processing Unit (GPU)	The model was run on a GPU (Tesla T4) with 15,102 MB of memory, enabling faster processing
GPU accelerator	CUDA 12.1
Compilers	Google. LLC. Collab and Jupyter

B. Training and Hyperparameter Settings for YOLO Models

Table IV shows the training settings, and Table V shows the hyperparameters that were utilized in the model training.

TABLE IV. TRAINING SETTINGS

Object detector	**YOLOv5**	**YOLOv8**
Weights	yolov5s.pt	yolo8s.pt
Class	1: Locust	1: Locust
DL framework	PyTorch 1.131	PyTorch
Language	Python 3.9	Python 3.9
Neural network type	CNN	CNN

TABLE V. HYPERPARAMETERS UTILIZED IN MODEL TRAINING

Object detector	**YOLOv5**	**YOLOv8**
Optimizer	SGD	SGD
Batch size	16	16
Learning rate	0.01	0.01
Momentum	0.937	0.937
Weight decay	0.0005	0.0005
Epochs	50	50

C. Performance Evaluation Metrics

Table V shows the confusion metrics for the models trained on the dataset, both with and without augmentation.

- Increment in True Positives (TP): After data augmentation, both YOLOv5 and YOLOv8 demonstrated higher TP for the locust class, which is a clear indication of an improvement in their detection ability.
- Reduction in False Positives (FP): YOLOv8 exhibits lower FP compared to YOLOv5 for the locust class, which suggests that precision is better in identifying locusts after augmentation.

These findings indicate that YOLOv8 displays better performance regarding both TP and FP, which signifies an improvement in overall accuracy for locust detection. In order to assess this change quantitatively, the accuracy of the model before and after data augmentation was calculated using the confusion matrices. Accuracy was chosen because it provides a holistic point of view of the model's behavior to the locust class. Since in the curated dataset every image contains locusts, there would be no scenario where the model detects an image that does not contain locusts, which is basically considered a True Negative (TN) count. Accuracy indicates the overall correctness of the model's predictions across all categories, given by:

$$\text{Accuracy} = \frac{\text{TP+TN}}{\text{TP+TN+FP+FN}} \quad (1)$$

where FN denotes False Negatives. In addition, the four most common evaluation metrics in object detection were used. Precision measures correctly identified locusts out of all identifications. The formula for its calculation is given by:

$$\mathrm{Precision} = \frac{\mathrm{TP}}{\mathrm{TP+FP}} \quad (2)$$

Recall is the ratio of correctly identified locusts out of all existing locust instances, given by:

$$\mathrm{Recall} = \mathrm{TP}/(\mathrm{TP}+\mathrm{FN}) \quad (3)$$

The mean Average Precision (mAP) is the mean value of precision for all detection categories, given by:

$$\mathrm{mAP} = (\textstyle\sum_{n=1}^{10} \mathrm{AP}_n)/\mathrm{N} \quad (4)$$

where N represents the total number of classes. F1-score is the harmonic mean of precision and recall, given by:

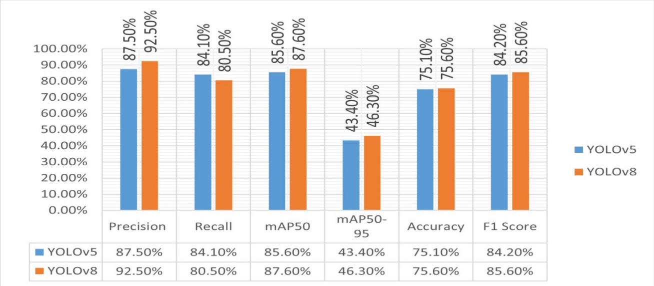
$$\mathrm{F1-score} = \frac{2\times\mathrm{Precision}\times\mathrm{Recall}}{\mathrm{Precision+Recall}} \quad (5)$$

TABLE VI. CONFUSION MATRICES

		Without Augmentation				**With data augmentation**			
		YOLOv5		**YOLOv8**		**YOLOv5**		**YOLOv8**	
Predicted		TRUE		TRUE		TRUE		TRUE	
	Locust	0.83	1	0.86	1	0.87	1	0.88	1
	Background	0.17	0	0.14	0	0.13	0	0.12	0
		Locust	Backgound	Locust	Background	Locust	Background	Locust	Background
		Actual							

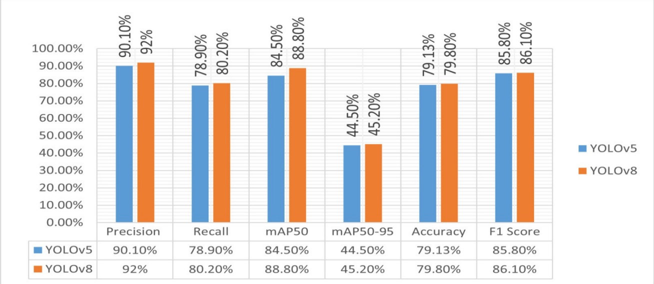
VI. .RESULTS AND DISCUSSION

At first, both the YOLOv5 and YOLOv8 models were trained for locust detection on the dataset without data augmentation, and Figure 5 illustrates their performance metrics. YOLOv8 was significantly more accurate in locust detection. Data augmentation eventually increased the size and diversity of the dataset and the overall performance as well, as shown in the results in Figure 6.



	Precision	Recall	mAP50	mAP50-95	Accuracy	F1 Score
🟦 YOLOv5	87.50%	84.10%	85.60%	43.40%	75.10%	84.20%
🟧 YOLOv8	92.50%	80.50%	87.60%	46.30%	75.60%	85.60%

Fig. 5. Performance comparison for YOLOv5 and YOLOv8 on the non-augmented dataset.



	Precision	Recall	mAP50	mAP50-95	Accuracy	F1 Score
🟦 YOLOv5	90.10%	78.90%	84.50%	44.50%	79.13%	85.80%
🟧 YOLOv8	92%	80.20%	88.80%	45.20%	79.80%	86.10%

Fig. 6. Performance comparison of YOLOv5 and YOLOv8 on the augmented dataset.

YOLOv8 achieved a higher accuracy of 79.80% on the augmented dataset. In this particular instance, this strategy can be recommended for locust detection due to its remarkable performance enhancement.

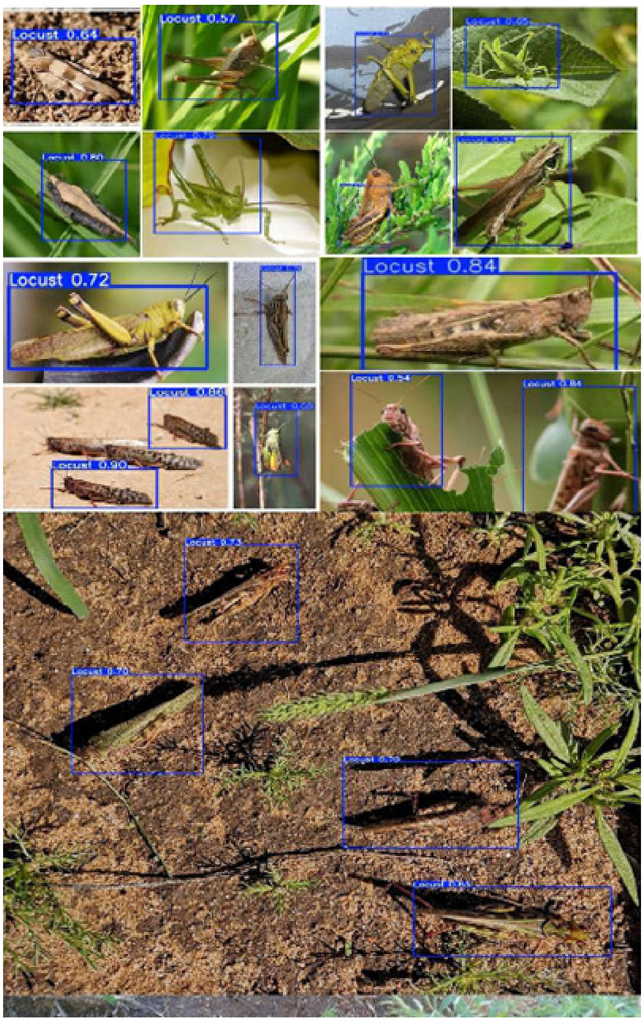


Fig. 7. Detection results of YOLOv5 after data augmentation on sample test images. The numbers represent the model's confidence in identified objects being locusts.

Figure 8 illustrates the detection results of YOLOv5 on the augmented dataset, and Figure 9 illustrates the detection results of YOLOv8 on the augmented dataset. From both figures, it is evident that the YOLOv8 confidence scores indicate the high accuracy levels of the model compared to the YOLOv5 model in recognizing locusts.

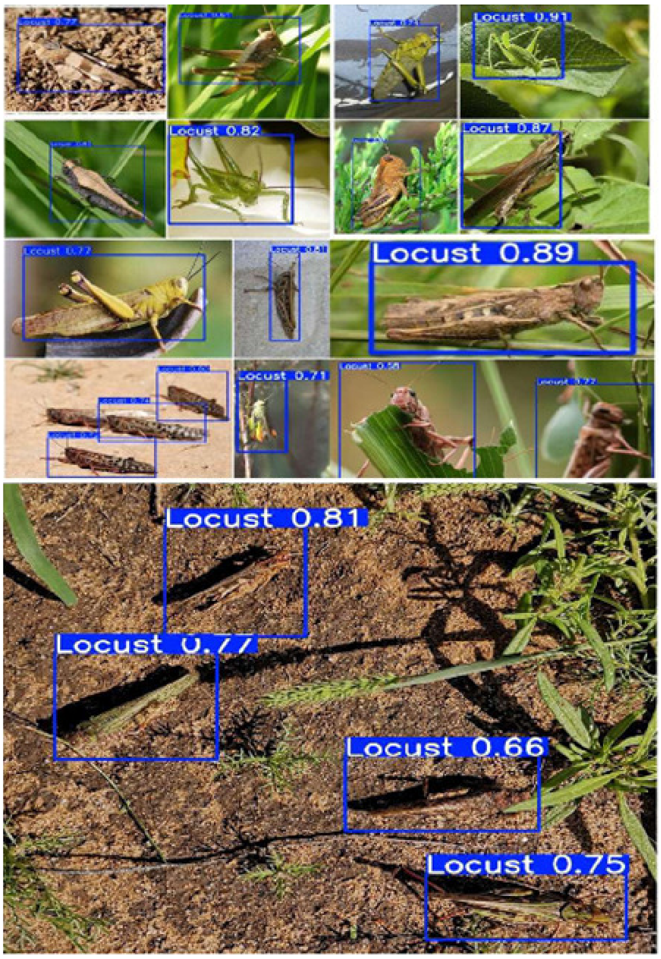


Fig. 8. Detection results of YOLOv8 after data augmentation on sample test images. The numbers represent the model's confidence in identified objects being locusts.

Locust detection in real-time using YOLOv5 and YOLOv8 highlights the efficacy of deep learning in processing visual data in dynamic environments. Other comparable object detection methods are commonly employed in AR/VR systems for real-time scene understanding, interactive simulations, and motion tracking. This demonstrates the broader relevance of this approach, as real-time object detection is a key technology in both agricultural monitoring and interactive visual computing environments.

VII. CONCLUSION

This study focused on enhancing locust detection ability in smart farming by comparing two popular object detection algorithms, YOLOv5 and YOLOv8, for locust detection. These two models were trained and validated on a publicly available dataset [21]. The results show that YOLOv8 achieved better results than YOLOv5. Data augmentation makes models more complex, but benefits both of them. YOLOv8 achieved higher detection accuracy due to its enhanced architecture's capacity, which makes it better at generalizing to unseen data. In addition, YOLOv8 has a shorter training time compared to YOLOv5 for the scenario in question. Therefore, YOLOv8 is the most suitable option for real-world agricultural scenarios, where accuracy and robustness need to be high, such as prompt locust detection in diverse environments. The findings of this study confirm the benefits of using advanced augmentation techniques and demonstrate that YOLOv8 has more potential for real-time locust monitoring. The primary focus of this study was on smart farming, contributing to precision agriculture by establishing proactive and swift pest management methods with the use of deep learning techniques. Nevertheless, the detection techniques reviewed here are of wide relevance for other real-time detection tasks in robotics, autonomous systems, and AR/VR applications.

VIII. FUTURE SCOPE

Despite numerous recent advances, locust detection remains a challenging process in an uncontrolled environment. Future studies could focus on:

- Fine-tuning YOLO models to accurately detect locusts in poor lighting conditions in challenging environments.
- Enhance models to tackle small-scale objects and objects in cluttered backgrounds to improve detection performance.
- Optimize the model to deploy in real-time scenarios to monitor locust infestation in farming using drone cameras and 3D environmental reconstruction, further bridging the gap between agricultural AI and computer vision applications in interactive and immersive computing. Hardware acceleration can help in real-time applications.
- Use satellite-based large-scale surveillance systems for prompt detection in response to locust swarms, because locust outbreaks can be widespread in no time.
- Integrate models with other sensor modalities, such as thermal cameras, LiDAR, or multispectral imaging, to improve detection accuracy in challenging environments, such as detecting locusts during night or in dense vegetation.
- Few-shot learning and transfer learning techniques can minimize the reliance on huge annotated datasets. These techniques would make the models more applicable in a scenario where there are fewer training examples with limited labeled data.

The future of locust detection holds promising potential for more effective, thorough, and sustainable solutions to pest management in agriculture and beyond, provided that the scientific community addresses these challenges and explores innovative concepts.

DECLARATIONS

Conflict of Interest: The authors declare no conflict of interest.

Data availability: Curated datasets related to this article can be accessed publicly at [21].

REFERENCES

[1] Y. Amit, P. Felzenszwalb, and R. Girshick, "Object Detection," in *Computer Vision*, Springer, 2021, pp. 875–883.

[2] X. Wu, D. Sahoo, and S. C. H. Hoi, "Recent advances in deep learning for object detection," *Neurocomputing*, vol. 396, pp. 39–64, Jul. 2020, https://doi.org/10.1016/j.neucom.2020.01.085.

[3] Y. Xiao *et al.*, "A review of object detection based on deep learning," *Multimedia Tools and Applications*, vol. 79, no. 33, pp. 23729–23791, Sep. 2020, https://doi.org/10.1007/s11042-020-08976-6.

[4] Y. Zhang, X. Li, L. Zhang, and H. Qi, "A Locust Swarm Recognition Algorithm Based on Faster-RCNN and GCN," in *2023 IEEE 6th International Conference on Computer and Communication Engineering Technology (CCET)*, Beijing, China, Aug. 2023, pp. 97–101, https://doi.org/10.1109/CCET59170.2023.10335122.

[5] A. Dharshini, A. Monisha, M. B. Malini, and S. K. Vinoth, "Method to prevent and track Locust's Intrusion using Object Detection Algorithms," in *2022 International Conference on Communication, Computing and Internet of Things (IC3IoT)*, Chennai, India, Mar. 2022, pp. 1–6, https://doi.org/10.1109/IC3IOT53935.2022.9768027.

[6] D. Zaveri, D. Shah, S. Joshi, T. Ahuja, N. M. Patil, and S. C. Iyer, "A Device for Detection and Deterrence of Locust Threats," in *2023 International Conference on Emerging Trends in Networks and Computer Communications (ETNCC)*, Windhoek, Namibia, Aug. 2023, pp. 1–6, https://doi.org/10.1109/ETNCC59188.2023.10284949.

[7] Y. Zhang, X. Li, and A. Wu, "A Locust Recognition Method Based on Graph Convolution Neural Network," in *2021 IEEE 4th International Conference on Computer and Communication Engineering Technology (CCET)*, Beijing, China, Aug. 2021, pp. 58–62, https://doi.org/10.1109/CCET52649.2021.9544266.

[8] P. Vajpayee, K. K. Yogi, P. Vajpayee, and K. K. Yogi, "Recognition and Early Stage Detection of Phytophthora in a Crop Farm Using IoT," in *Agro-Economic Risks of Phytophthora and an Effective Biocontrol Approach*, IntechOpen, 2021.

[9] B. Zoph, E. D. Cubuk, G. Ghiasi, T. Y. Lin, J. Shlens, and Q. V. Le, "Learning Data Augmentation Strategies for Object Detection," in *Computer Vision – ECCV 2020*, 2020, pp. 566–583, https://doi.org/10.1007/978-3-030-58583-9_34.

[10] R. Ahmed and E. H. Abd-Elkawy, "Improved Tomato Disease Detection with YOLOv5 and YOLOv8," *Engineering, Technology & Applied Science Research*, vol. 14, no. 3, pp. 13922–13928, Jun. 2024, https://doi.org/10.48084/etasr.7262.

[11] M. H. F. Afonso, E. H. Teixeira, M. R. Cruz., G. P. Aquino, and E. C. Vilas Boas, "Vehicle and Plate Detection for Intelligent Transport Systems: Performance Evaluation of Models YOLOv5 and YOLOv8," in *2023 IEEE International Conference on Computing (ICOCO)*, Langkawi, Malaysia, Oct. 2023, pp. 328–333, https://doi.org/10.1109/ICOCO59262.2023.10397996.

[12] E. Soylu and T. Soylu, "A performance comparison of YOLOv8 models for traffic sign detection in the Robotaxi-full scale autonomous vehicle competition," *Multimedia Tools and Applications*, vol. 83, no. 8, pp. 25005–25035, Mar. 2024, https://doi.org/10.1007/s11042-023-16451-1.

[13] E. Casas, L. Ramos, C. Romero, and F. Rivas-Echeverría, "A comparative study of YOLOv5 and YOLOv8 for corrosion segmentation tasks in metal surfaces," *Array*, vol. 22, Jul. 2024, Art. no. 100351, https://doi.org/10.1016/j.array.2024.100351.

[14] P. Vajpayee and K. Kr. Yogi, "Enhanced Grasshopper Detection in Smart Crops Using Faster R-CNN Deep Learning Model," in *2024 3rd Edition of IEEE Delhi Section Flagship Conference (DELCON)*, New Delhi, India, Nov. 2024, pp. 1–7, https://doi.org/10.1109/DELCON64804.2024.10866725.

[15] A. K. Kanade, M. P. Potdar, A. Kumar, G. Balol, and K. Shivashankar, "Weed detection in cotton farming by YOLOv5 and YOLOv8 object detectors," *European Journal of Agronomy*, vol. 168, Jul. 2025, Art. no. 127617, https://doi.org/10.1016/j.eja.2025.127617.

[16] G. Liu, J. Di, Q. Wang, Y. Zhao, and Y. Yang, "An Enhanced and Lightweight YOLOv8-Based Model for Accurate Rice Pest Detection," *IEEE Access*, vol. 13, pp. 91046–91064, 2025, https://doi.org/10.1109/ACCESS.2025.3569819.

[17] M. M. Abo-Zahhad and M. Abo-Zahhad, "Real time intelligent garbage monitoring and efficient collection using Yolov8 and Yolov5 deep learning models for environmental sustainability," *Scientific Reports*, vol. 15, no. 1, May 2025, Art. no. 16024, https://doi.org/10.1038/s41598-025-99885-x.

[18] "Locust Dataset > Overview," *Roboflow*. https://universe.roboflow.com/new-workspace-ocpp0/locust-kvcoz.

[19] P. Chudzik *et al.*, "Mobile Real-Time Grasshopper Detection and Data Aggregation Framework," *Scientific Reports*, vol. 10, no. 1, Jan. 2020, Art. no. 1150, https://doi.org/10.1038/s41598-020-57674-8.

[20] A. Mumuni and F. Mumuni, "Data augmentation: A comprehensive survey of modern approaches," *Array*, vol. 16, Dec. 2022, Art. no. 100258, https://doi.org/10.1016/j.array.2022.100258.

[21] P. Vajpayee and K. Yogi, "Locust Images Dataset." Zenodo, Mar. 09, 2025, https://doi.org/10.5281/ZENODO.14964987.

[22] P. Vajpayee, "poojavjpy/Locust-Detection." May 01, 2025, [Online]. Available: https://github.com/poojavjpy/Locust-Detection.