Volume 4 Issue IV, April 2025, No. 68, pp. 866-877
Submitted 22/4/2025; Final peer review 28/5/2025
Online Publication 26/5/2025
Available Online at http://www.ijortacs.com

DESIGN OF A SMART AGRICULTURE SYSTEM FOR PEST MANAGEMENT BASED ON TRANSFER LEARNING AND IOT TECHNIQUES

[1]Nnaemeka Frances O., [2]Ogochukwu Okeke
Department of Computer Science; Chukwuemeka Odumegwu Ojukwu University,
Uli, Anambara State, Nigeria
[1]**obynnaemekaf@gmail.com**; [2]**ogoookeke@yahoo.com**
Corresponding Author's Email : **obynnaemekaf@gmail.com**

Abstract

Pests are organism that causes damage to crops in a farm land. For many years, this organism has continued to evolve and some have even become resistant to the traditional pest control measures which are more of a reactive approach. Several existing literatures which have also provides recommendation to better manage pest, however there is gap in the need of models which considers region specific pest to ensure a reliable system for pest management. Hence, the aim of this study is design and implementation of pest management system for precision agriculture using integrated transfer learning and internet of things technique. The methodology to be used for this work is the dynamic system development model. The tested for primary data collection is Aninri and the pests considered are weevil, caterpillar, and whitefly. The secondary data source is Kaggle. Then total sample size of data collected is 18138. A notification algorithm was developed with simple mail transfer protocol, while rule-based approach was applied to develop pest control model, using data collected from pest related domain experts. The models were integrated as a system for pest management in smart agriculture. Experimental validation was carried out considering insects collected from different farms, the results recorded successful pest classification, and notification of control recommendations. In conclusion, this work has successfully presented a reliable solution of the real time management of pest.

Keywords: Pests Management; Precision Agriculture; IoT; Transfer Learning; Smart Agriculture

1. INTRODUCTION

Pests pose a significant threat to agricultural productivity, potentially causing substantial damage to crops and leading to significant economic losses for farmers (Thomas et al., 2023). These pests encompass a wide range of organisms, including insects, rodents, fungi, bacteria, viruses, and other unwanted species as in Figure 1. Their impact can manifest through direct consumption of crops, transmission of diseases, or disruption of the plant's natural growth processes. Addressing the challenges posed by pests is crucial for ensuring food security and sustainable agricultural practices (Nayagam et al. 2023).

The timely and accurate detection of pests is essential for effective pest management in agriculture. Traditional methods of pest detection, such as visual inspection, can be labour-intensive and may not provide early identification. With advancements in technology, pest detection has seen a transformation, with the integration of AI, IoT devices, and sensor networks (Azfar et al., 2023). Automated image recognition using machine learning and deep learning algorithms allows for the swift identification of pest presence through the analysis of images captured in the field (Prasath and Akila, 2023). Sensor networks and IoT devices provide real-time data on environmental conditions, enabling early detection of anomalies associated with pest activity. These technological innovations offer farmers the ability to monitor their crops more efficiently and implement targeted pest control measures.

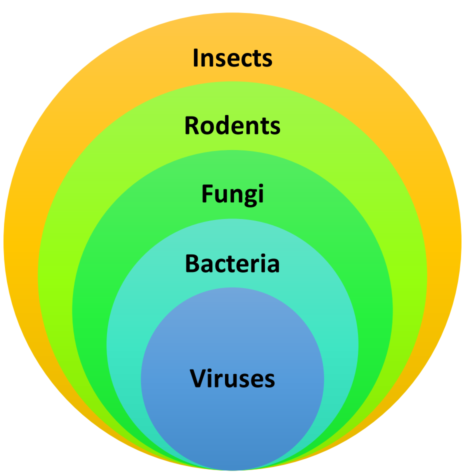

Figure 1: Examples of pest (Thomas et al., 2023)

Controlling pests in agriculture involves a range of strategies aimed at minimizing their impact on crops. Traditional methods include the use of chemical pesticides, biological control using natural predators, and cultural practices like crop rotation. However, concerns about the environmental impact and sustainability of chemical pesticides have led to a shift towards integrated pest management (IPM) approaches (Azfar et al., 2023).Drones for precision agriculture, precision technologies, automated image recognition, sensor network and internet of things, biological control approach. These recent methods of pest detection and control highlight the ongoing evolution of agricultural practices toward precision, efficiency, and sustainability (Sun et al., 2023). Integrating these technologies into pest management strategies requires addressing challenges related to accessibility, scalability, and the need for ongoing research to optimize their effectiveness in diverse agricultural settings.

In the field of agriculture, IoT plays a crucial role in transforming traditional farming into smart farming by integrating technology to enhance productivity, efficiency, and sustainability (Karar et al., 2022). And deep learning, a subset of machine learning, has shown exceptional prowess in pest detection, particularly in tasks involving image analysis. Convolutional Neural Networks (CNNs) have revolutionized image-based pest identification by automatically learning hierarchical features from images, enabling precise recognition of pest species and early signs of damage (Sun et al., 2023). Transfer learning, another deep learning technique, allows models pre-trained on extensive datasets to be fine-tuned for specific crops or regions, addressing the challenge of limited labeled data (Anwar and Masood, 2023). Deep learning's ability to automatically extract intricate features from complex datasets contribute to the unprecedented accuracy and efficiency observed in modern AI-driven pest detection systems.

Several existing literatures like (Azfar et al., 2023;Vemuri, 2023; Nayagam et al., 2023; Prasath and Akila, 2023; Thomas et al., 2023; Debauche et al., 2020) which have also provides recommendation to better manage pest, however there is gap in the need of models which considers region specific pest to ensure a reliable system for pest management. Among the most recent studies reviewed, Kumar and Kalita (2017) applied light weight YOLOV-5 for the classification of pest using field adaption method. The work considered 15000 samples of pest in classes of coleoptera, araneae, hemiptera, hymenoptera, lepidoptera, odonata, and diptera and then train YOLOV-5 to generate model for the classification of pest in a farm. While the study recorded significant contribution to pest disease management, there is need for a model which perform real-time pest monitoring considering region specific pest and also the integration of recommendation measures to help manage this problem.

2. THE PROPOSED SYSTEM

The proposed system will used image data of pest collected from the Enugu State Ministry of Agriculture and then fine tune existing pest data which will be collected in Robowflow repository to develop a comprehensive data model. The data will be used to train a deep learning algorithm, specifically YOLOv10 which is a more recent and advance version of YOLOv10, then it will be trained to generate model for the real time classification of pest in the farm. To make the model reliable, a pest detection decision algorithm will be developed, using the classification model as a foundation to decide if the farm is infected with disease or not. In the next phase of the proposed system, IoT algorithm will be developed and integrated with the classification model to facilitate real-time notification to the farmer on the event of pest in the farm. Finally, a decision-based model will be developed which inform the farmer on the right pesticide to be applied on the farm to help control the pest. The proposed system diagram was presented in the Figure 2.

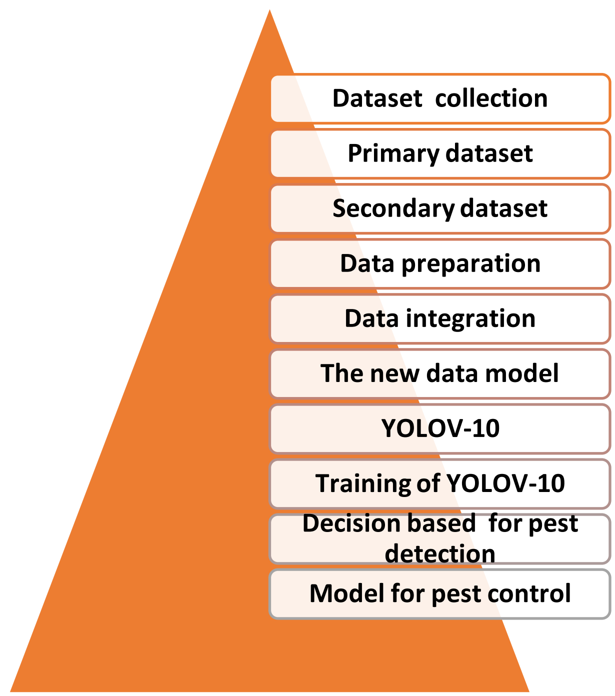

Figure 2: Diagram of the proposed system

The Figure 2 presents the block diagram of the proposed system. This system began with data collection of primary and secondary dataset. Both datasets will be processed through annotation and labelling, then collectively integrated to create a new data model. YOLOv10 will be trained with the data to generate model for real time pest classification. To address issues of false alarm, a decision-based algorithm will be developed and integrated with the deep learning-based classifier. This will facilitate accurate detection of pest in the farm and then address issues of reliability as characterized in the existing system. To ensure real time notification of the farmer on the issue of pest, an IoT algorithm will be developed which used email to notify the farmer of the issues of pest. Finally, the problem will be controlled through attached information which informs the farmer on the right type of pesticide to be applied to help management the problem.

2.1 Data Collection

The data used for this work was collected from three farms at Ndeabo in Aninri local Government Area, Enugu State. The geographical coordinate of the site is at Latitude 6.01.30N and Longitude 7.34.30E. The instruments for the data collection are HD USB camera, raspberry pi, and Laptop to collect the files. The secondary dataset, titled "Insect Pest Detection for YOLO", provided a comprehensive collection of 17,641 labelled images featuring a wide variety of pests and insects commonly found in agricultural environments. This dataset significantly enriched the diversity and quantity of training data, helping to improve the model's ability to detect and classify pests across different farm settings.

The test data used for evaluating the pest detection and notification system was carefully selected to represent a broad range of real-world scenarios in agricultural environments. The dataset consisted of a diverse collection of images containing various types of pests, such as aphids, bollworms, caterpillars, and leaf miners, in agricultural settings like crop fields and greenhouses. A total of 1,500 images were gathered from both public datasets and field recordings to ensure variability in lighting, background, and pest visibility. The images were annotated with bounding boxes around each pest and labelled with their corresponding species. Additionally, to simulate different environmental conditions, the dataset included images with varying degrees of blur, occlusion, and noise, as well as images taken under different lighting conditions (e.g., bright sunlight, cloudy skies, and low light). This diverse set of test data ensured that the model was evaluated on its ability to handle various challenges that may arise in real-world pest detection scenarios. Figure 3 presents the test dataset samples used to evaluate the model performance.

Figure 3: Sample test data of pest evaluate the model

2.2 Database Design

The database design for the smart farm pest detection and management system was structured to ensure efficient storage, retrieval, and processing of images, annotations, user access, and control recommendations. It comprises four main tables:

i. Image Table
ii. Annotation and Labelling Table
iii. Pest Control Recommendation Table

These tables work in coordination to support data-driven detection, classification, and pest control advisory processes in the smart farm system.

Table 1: Images Table

Field Name	Data Type	Key	Description
Image_ID	INT	Primary Key	Unique identifier for each image
File_Name	VARCHAR(200)		Name of the image file
Timestamp	DATETIME		Date and time the image was captured
Device_ID	INT	Foreign Key	References the device that captured the image
Image_Format	VARCHAR(20)		Format type (JPEG, PNG, etc.)
Resolution	VARCHAR(20)		Image resolution (640x640)

The image table acts as the foundational component, storing key metadata about the images captured from farm environments. This includes the file name, date of capture, image format, resolution, and source (whether collected locally or externally, such as from Kaggle). Each image is assigned a unique ID, which links it to annotations and other related data, ensuring traceability and structured data flow. The annotation and labelling table is presented as;

Table 2: Annotations and labelling

Field Name	Data Type	Key	Description
Annotation_ID	INT	Primary Key	Unique ID for the annotation
Image_ID	INT	Foreign Key	References the image being annotated
Class_Label	VARCHAR(50)		Object class (insect and pest)
X_min	FLOAT		Left coordinate of bounding box
Y_min	FLOAT		Top coordinate of bounding box
X_max	FLOAT		Right coordinate of bounding box
Y_max	FLOAT		Bottom coordinate of bounding box
Confidence_Score	FLOAT		Model's confidence in the label (0–1)

The Annotation and Labelling Table is directly linked to the images and houses all information regarding the objects (pests) identified within those images. For each annotation, the system records the pest type, precise bounding box coordinates ($X\_min, Y\_min, X\_max, Y\_max$), and a confidence score from the detection model. This table is essential for both training and validating the machine learning model, as it enables supervised learning through clearly labeled visual data. Moreover, the pest type recorded here serves as a key link to the pest management system. The pest control and recommendation table is presented as;

Table 3: Pest Control Recommendation

Field Name	Data Type	Description

Recommendation_ID	INT	Unique ID for each pest control recommendation
Pest_Type	VARCHAR(100)	Pest category
Control_Method	TEXT	Recommended method of control
Suggested_Product	VARCHAR(100)	Name of product or technique
Application_Dosage	VARCHAR(50)	Dosage or usage amount

To provide meaningful action after detection, the Pest Control Recommendation Table is introduced. It matches pest types with suggested control methods, products, and appropriate dosage levels. This enables automated advisory support to farmers or agricultural officers once a pest is detected. Together, these tables form a cohesive system that supports smart pest detection, annotation, recommendation, and controlled user access.

3. Pest Management System

The pest management system is made of the transfer learning model, the IoT model and the control model. The transfer learning model is the YOLOv10. The IoT model used SMTP while the control model is actions necessary to mitigate the impact of the pest on the farms.

3.1 Model of the proposed transfer learning technique

The transfer learning technique adopted for this work is the YOLOv10. The model is made of three main sections which are the backbone, the neck and the head. The backbone accepts images from the images dimensioned and then through the application of Cross Stage Partial with 2 bottlenecks (C2F) layer, convolutional layer, C2F-Convolutional Inverted Bottleneck (CIB) C2FCIB layer, Spatial Channel Decoupled Downsampling (SCDown), Spatial Pyramid Pooling Function (SPPF) and Parallel Self Attention (PSA), extracts the information, concatenate in the neck and then classify in the output. The reason why YOLOV-10 was adopted was due to its ability for small object detection, making it very suitable for pest classification. The C2F layer contains two bottleneck, convolutional layer and concatenation layer. The bottleneck allows balance between learning nee features and reusing existing one. Each bottle neck is made of convolutional layers connected in series. For the C2F (True) layer, the bottleneck is connected in series and also has additional channel from the input to the output. For the C2F (False), the bottle neck has only two convolutional layer connected in series. The convolutional layers allow for feature convolution with filters to extract local image information. The C2FCIB is an improved C2F layer using CIB. This section expands feature dimension and facilitate more abstract representation of features, while maintaining computational. The SPPF layer contains convolutional layer, concatenation layer and maximum pooling layers interconnected as shown in Figure 4.

The role is to maximize feature extraction in multiscale and varying sizes. The PSA is an attention mechanism made of Multi Head Self Attention (MSHA) and Feed forward Neural Network (FFN), which collectively allows the model to have focus on more important features through localization of spatial regions in the image strides. This helps in the capturing of long-range dependencies and optimizes model ability in small object detection like pest in 11 dimensions. The SCDown first increase the channel dimension with point wise 11 convolution, then spatial downsampling with 33 depth wise convolution to reduce the spatial dimension of each feature. Overall, the SCDown expands number of channels through downsampling. Figure 4 presents the internal architecture of these discussed YOLOV-10 modules.

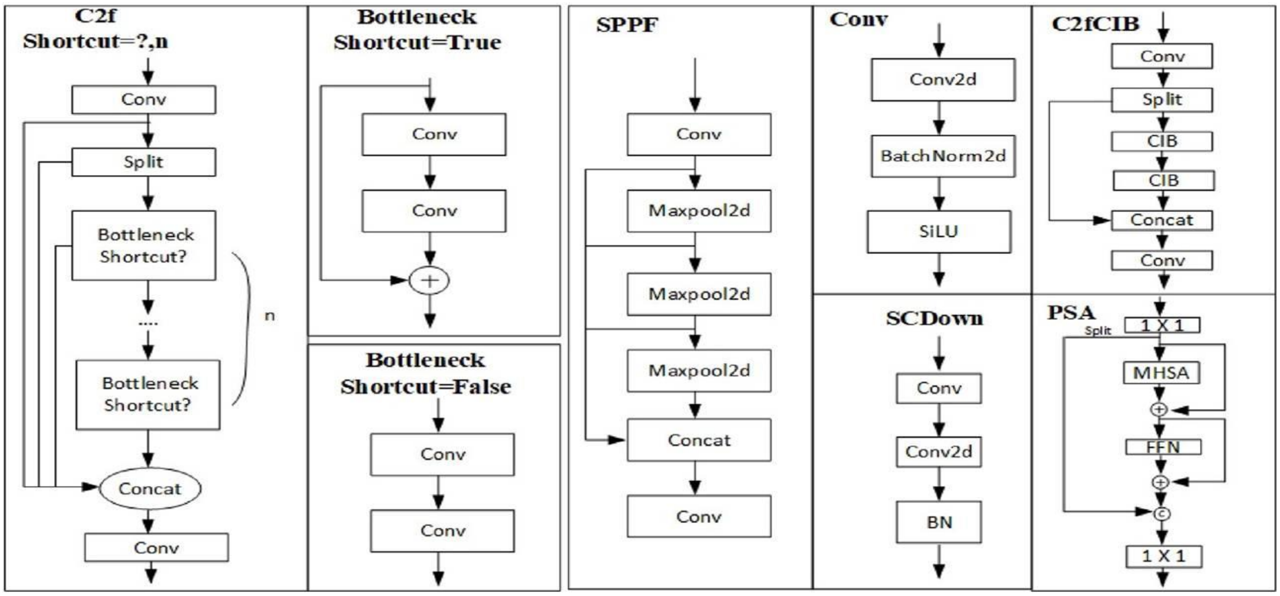

Figure 4: Internal architecture of the YOLOV-10 Components

3.2 Training of the YOLOV-10 Model

The training process of the YOLOv10 model for pest detection was conducted using a hybrid dataset comprising 497 custom-captured pest images and an additional 17,641 annotated pest samples sourced from the public Kaggle repository. The combined dataset was pre-processed and formatted to comply with the YOLO training pipeline using Roboflow for annotation standardization, bounding box formatting (in YOLO txt format), and class label encoding. The training was carried out using the YOLOv10 architecture, initialized with pre-trained weights from the COCO dataset to applied transfer learning and expedite convergence. The dataset was split into 70% training, 20% validation, and 10% testing to ensure proper generalization and to avoid overfitting. During training, a multi-scale training regime was employed, wherein the input resolution was randomly varied between 640×640 and 1280×1280. This approach enabled the model to generalize better across different pest sizes, particularly aiding in the detection of tiny or partially occluded pests. Training was conducted over 300 epochs on an NVIDIA GPU platform, with check-pointing and early stopping mechanisms based on validation mAP (mean Average Precision) to ensure optimal performance and prevent overfitting.

Table 4: Training Setup and Environment

Parameter	Description / Value
Model Architecture	YOLOv10 (Baseline with enhanced SPPF and PSA)

Pretrained Weights	COCO Pretrained Weights (YOLOv10)
Framework	Ultralytics YOLOv10
Annotation Tool	Roboflow
Dataset Size	497 (custom) + 17,641 (Kaggle) = **18,138 images**
Classes	20 pest categories
Image Resolution	Dynamic: 640 × 640 to 1280 × 1280 (Multi-scale Training)
Data Split	70% Training, 20% Validation, 10% Testing
Loss Functions	CIoU Loss (bbox), Focal Loss (cls), Binary Cross-Entropy
Optimizer	Adam with Weight Decay
Learning Rate Scheduler	Cosine Annealing
Batch Size	16
Epochs	300
Early Stopping	Enabled (based on validation mAP)
Hardware	NVIDIA RTX 3090 (24GB VRAM)
YOLOv10 Version	Ultralytics YOLOv10 (Release 2024.2)
Python Version	3.10
CUDA Version	11.8
PyTorch Version	2.1.0
Operating System	Windows 10

4. RESULTS AND DISCUSSIONS

The performance metrics employed for this evaluation include accuracy, training loss, precision, and mean Average Precision (mAP). The baseline performance of the traditional YOLOv10 model is documented in Table 5, capturing values for each of the aforementioned metrics across training epochs. The training curve for YOLOv10 reveals a steady learning progression, though with room for improvement in multi-scale feature extraction, particularly in detecting small or partially occluded pests.

Table 5: Result of the YOLOV-10+SPFF training (See appendix A)

Table 6 reported the final training performance for the traditional YOLOV-10.

The comparative results of the training process were recorded in Table 7.

Table 6: Result of the YOLOV-10 + SPFF training

Train Box Loss	Val Box Loss	Train Obj Loss	Val Obj Loss	Train Cls Loss	Val Cls Loss	Precision	Recall	mAP@0.5	mAP@0.5:0.95
0.03516	0.03685	0.02479	0.02866	0.01447	0.02075	0.88519	0.84463	0.82746	0.58272

Testing of the model utilized the image set to test the model ability to correctly detect, classify and label different pests on the farm. The Figure 5 presented the result when the model was tested on different pest images collected from the farm. Figure 6 also recorded the performance of the model when tested on different pest images collected from the testing dataset. Figure 5 and Figure 6 respectively showed the performance of the model when deployed to evaluate its ability for pest detection in different farm environments. From the outcome of the results, it was observed that the model was able to correctly classify the pest from the farm image, detect the pest, and label it successfully with confidence score. Figure 7 showed the result of the model performance when evaluated with the test data.

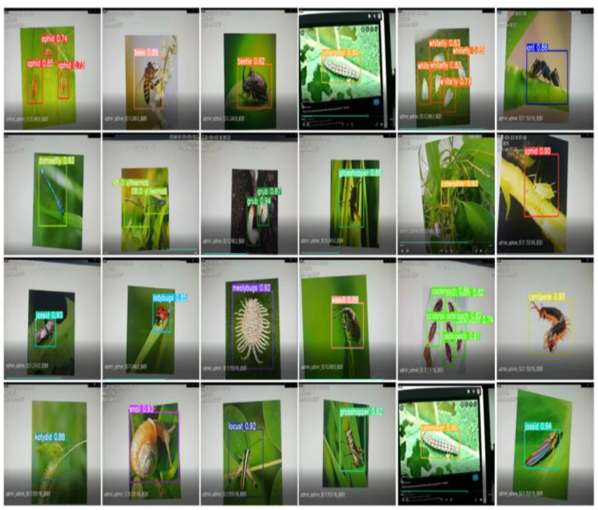

Figure 5: Testing result from pest images on the dataset

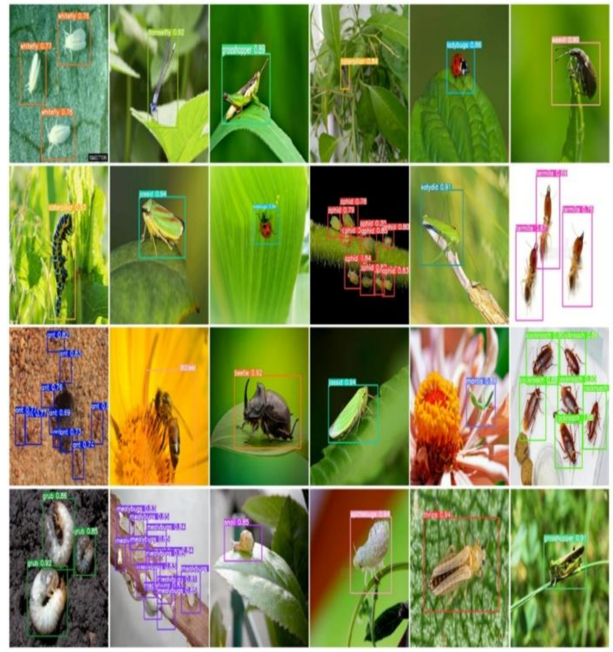

Figure 6: Testing result of the model on different pest images on the farm

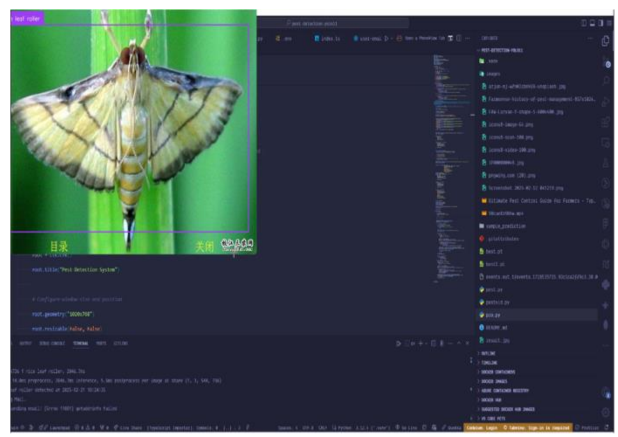

Figure 7: Result of test data A

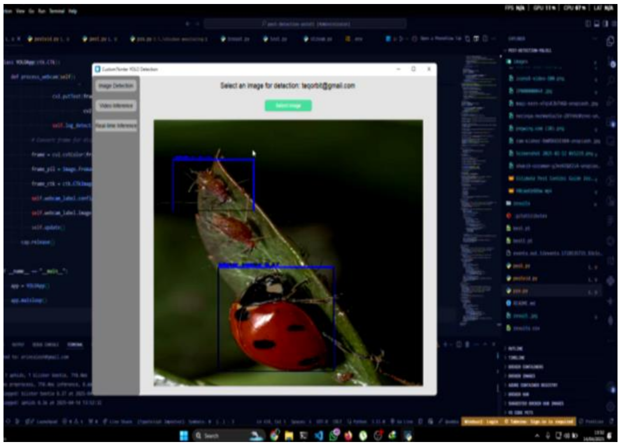

Figure 8: Result of test data of pest in Cassava farm

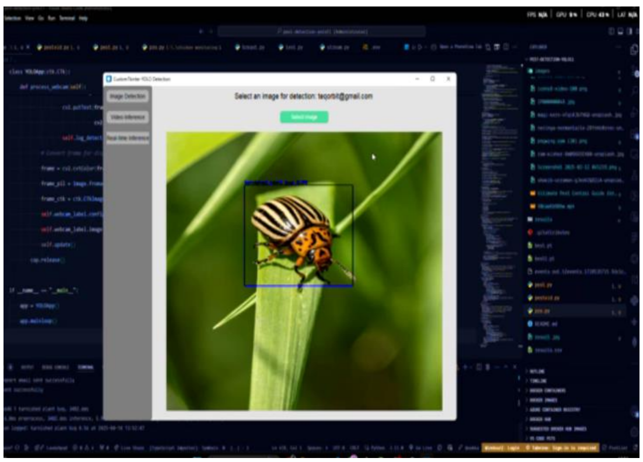

Figure 9: Result of test data B at maize farm

Figure 7-9 presents the performance of the model when evaluated considering different pest features. From the results, it was observed that the model was able to correctly detect pest and then classify it with high confidence score. The Figure 10 and Figure 11 present the result of the notification and control measures.

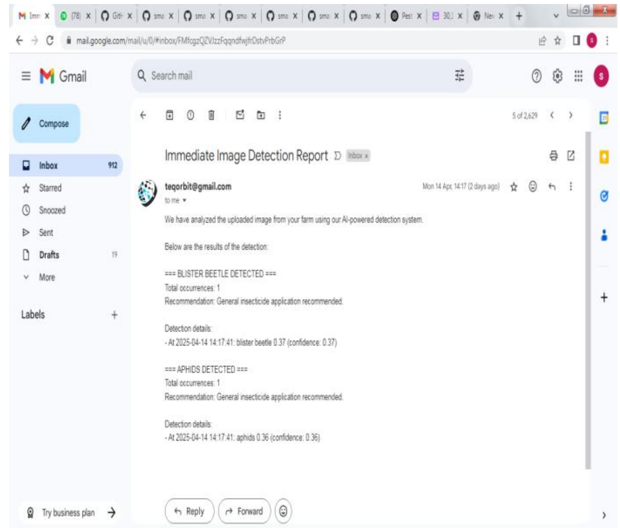

Figure 10: Result of notification and recommendation for threat control

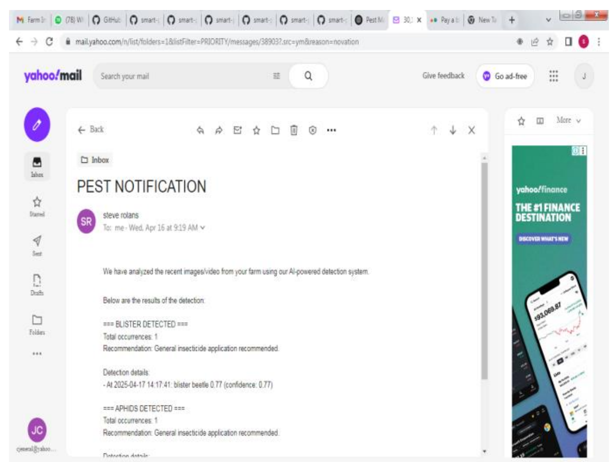

Figure 11: Result of notification and recommendation for threat control

The Figure 10 presents the result of nonfiction of pest on the farm and also recommended control measure to help arrest the problem. The Figure 11 reported the results when tested with another user email address. It was observed that upon classification of pest, the model was able to correctly notify the user of the problem through email and also make necessary recommendations to help address it.

5. CONCLUSION

The pest detection and notification system developed in this study represents a significant advancement in the use of artificial intelligence for agricultural pest management. By applying the YOLOv10 model, the system accurately identifies and classifies pests in agricultural fields, providing farmers with real-time notifications that help mitigate the damage caused by these pests. This system's ability to process images quickly and efficiently makes it a practical solution for large-scale farming operations, where the timely detection of pest infestations can have a profound impact on crop yield and quality.

Throughout the development and testing phases, the system demonstrated strong performance in detecting and classifying different pest images. The model's high precision and recall rates indicate its robustness and effectiveness in identifying a wide variety of pests, even under different farm lands. In terms of system integration, the solution is designed to be scalable across various agricultural contexts. The modular design of the system allows for easy updates and the inclusion of new pest species as more data becomes available. Moreover, the user interface ensures that both administrators and end-users can easily interact with the system, maximizing its utility and accessibility. The training and documentation provided further ensure that the transition to using the system is smooth, empowering users to applied its full potential with minimal effort. The findings from this dissertation highlight the role of deep learning and computer vision in modern agriculture, especially in pest management.

6. REFERENCES

Anwar, Z., & Masood, S. (2023). Exploring deep ensemble model for insect and pest detection from images. *Procedia Computer Science, 218*, 2328–2337.

Azfar, S., Nadeem, A., Ahsan, K., Mehmood, A., Almoamari, H., & Alqahtany, S. (2023). IoT-based cotton plant pest detection and smart-response system. *Applied Sciences, 13*(3), 1851. https://doi.org/10.3390/app13031851

Debauche, O., Mahmoudi, S., Elmoulat, M., Mahmoudi, S., Manneback, P., & Lebeau, F. (2020). Edge AI-IoT pivot irrigation, plant diseases, and pests identification. In *Proceedings of the 11th International Conference on Emerging Ubiquitous Systems and Pervasive Networks (EUSPN 2020)*.

Karar, M., Abdel-Aty, A., Algarni, F., Hassan, M., Abdou, M., & Reyad, O. (2022). Smart IoT-based system for detecting RPW larvae in date palms using mixed depthwise convolutional networks. *Alexandria Engineering Journal, 61*, 5309–5319. https://doi.org/10.1016/j.aej.2021.10.050

Kumar, D., & Kalita, P. (2017). Reducing postharvest losses during storage of grain crops to strengthen food security in developing countries. *Foods, 6*(1), 8.

Nayagam, M., Vijayalakshmi, N., Somasundaram, K., Mulunthan, M., Yogaraja, C., & Partheeban, P. (2023). Control of pests and diseases in plants using IoT technology. *Measurement: Sensors, 26*, 100713. https://doi.org/10.1016/j.measen.2023.100713

Prasath, B., & Akila, M. (2023). IoT-based pest detection and classification using deep features with enhanced deep learning strategies. *Engineering Applications of Artificial Intelligence, 121*, 105985. https://doi.org/10.1016/j.engappai.2023.105985

Sun, L., Cai, Z., Liang, K., Wang, Y., Zeng, W., & Yan, X. (2023). An intelligent system for high-density small target pest identification and infestation level determination based on an improved YOLOv5 model. *Expert Systems with Applications, 239*, 122190. https://doi.org/10.1016/j.eswa.2023.122190

Thomas, J., Manikandarajan, S., & Subha, T. (2023). AI-based pest detection and alert system for farmers using IoT. *E3S Web of Conferences, 387*, 05003. https://doi.org/10.1051/e3sconf/202338705003

Vemuri, H. (2023). Pest detection system. *International Journal of Science, Engineering and Technology*, Online ISSN: 2348-4098; Print ISSN: 2395-2404.

Appendix A (Result of YOLOV Training)

Epoch	Train Box Loss	Val Box Loss	Train Obj Loss	Val Obj Loss	Train Cls Loss	Val Cls Loss	Precision	Recall	mAP@0.5	mAP@ 0.5:0.95
1	0.12224	0.11272	0.08962	0.08345	0.05548	0.05867	0.62008	0.55812	0.44429	0.24167
2	0.11831	0.11233	0.08799	0.08215	0.05618	0.05949	0.62504	0.57015	0.45343	0.25076
3	0.11729	0.11017	0.08706	0.08299	0.05614	0.05176	0.6367	0.57527	0.46634	0.2619
4	0.11653	0.10701	0.08508	0.08098	0.05501	0.0546	0.6455	0.58224	0.47657	0.2744
5	0.11058	0.10612	0.07835	0.07804	0.05283	0.05113	0.65122	0.59007	0.48763	0.28192
6	0.10821	0.10514	0.07743	0.07669	0.05383	0.05044	0.66153	0.60228	0.49857	0.29104
7	0.10954	0.10605	0.07858	0.07743	0.04934	0.04924	0.66338	0.60616	0.51	0.29736
8	0.10569	0.10064	0.07688	0.07216	0.05011	0.04774	0.66939	0.61965	0.51866	0.31044
9	0.10106	0.09888	0.07523	0.07295	0.04959	0.04863	0.67506	0.61919	0.53189	0.31884
10	0.10098	0.09634	0.0803	0.07001	0.04937	0.04597	0.68815	0.63374	0.53707	0.33022
11	0.09693	0.09083	0.07219	0.06858	0.04877	0.0473	0.69282	0.63756	0.54768	0.33338
12	0.09495	0.09285	0.07181	0.06985	0.04415	0.04581	0.69756	0.64353	0.5561	0.3423
13	0.09445	0.09132	0.06998	0.06798	0.0424	0.0446	0.70421	0.64902	0.56853	0.34999
14	0.08828	0.09446	0.06795	0.06668	0.04712	0.04247	0.70721	0.65711	0.57443	0.36005
15	0.08685	0.08754	0.06464	0.06642	0.04436	0.04235	0.71998	0.66226	0.58426	0.36389
16	0.08742	0.08696	0.06545	0.06264	0.04135	0.04426	0.72081	0.67038	0.58714	0.37161
17	0.08482	0.08477	0.06109	0.0628	0.04512	0.04302	0.72608	0.67407	0.59833	0.37903
18	0.08582	0.08103	0.0609	0.05968	0.04146	0.03935	0.73245	0.67965	0.60702	0.38476
19	0.08178	0.08422	0.05918	0.05984	0.04281	0.04082	0.73696	0.68396	0.61264	0.38986
20	0.07921	0.08206	0.05913	0.05786	0.03983	0.04146	0.74129	0.68975	0.62163	0.39733
21	0.08346	0.08079	0.06244	0.05728	0.04309	0.03596	0.74397	0.69551	0.62528	0.40509
22	0.07862	0.07608	0.05296	0.05727	0.04177	0.03976	0.75106	0.69636	0.63574	0.40671

23	0.07778	0.07944	0.05698	0.05346	0.03707	0.03674	0.75831	0.70182	0.64394	0.41468