

# Smart Pest Monitoring and Management System with Integrated Deep Learning and Unmanned Aerial Vehicle (UAV) Technologies

Ogidi Patient C., Asogwa T.C.

Department of computer science, Enugu State University of Science and Technology, Nigeria

DOI: <https://doi.org/10.51584/IJRIAS.2025.10100000182>

Received: 02 November 2025; Accepted: 10 November 2025; Published: 21 November 2025

## ABSTRACT

Infestation of pest is one of the leading agricultural problems which have led to a lot of losses in the yield and risks food security. The use of conventional forms of pest control is usually inefficient, consumes a lot of chemicals and requires response time. The paper describes a smart pest monitoring and management system, combining the deep learning, Internet of Things (IoT) and Unmanned Aerial Vehicle (UAV) technologies to detect pests in real-time and give decision support. The annotated mixed data consisting of primary pest images gathered in the Federal College of Agriculture, Ishiagu, and secondary data in the Kaggle repository was used to generate a model based on YOLOv10. The model had a precision of 0.84, a recall of 0.82 and a mean Average Precision (mAP@50) of 0.83 indicating that the model was very effective in detecting and classifying a variety of pest species at an acceptable level of accuracy. A recommendation algorithm based on rules was installed to offer specific pesticide recommendations depending on the identified type of pest and the IoT-based email notification module provided the real-time notifications to the farmers to take immediate action. To realise remote sensing and aerial pest surveillance, the UAV was simulated and designed in the Simulink environment to ensure the efficient coverage and the reliable data capture. The integrated system offers a smart and long-term solution to the pest management process by eliminating false alarms, reducing pesticide waste, and enhancing the reaction time. The limitation of the study lies on the condition that the system was only being implemented as a simulation and lacks real-world validation, hence, it is recommended that future studies should adopt the real-world implementation approach for the identification of pests.

**Keywords:** Smart Agriculture; Pest Detection; YOLOv10; Internet of Things (IoT); Unmanned Aerial Vehicle (UAV); Deep Learning

## INTRODUCTION

Pests in agriculture are organisms that harm crops, livestock, or agricultural infrastructure, posing a threat to food production and economic stability. These pests include a wide range of species such as insects, rodents, fungi, weeds, bacteria, and viruses, all of which can cause significant damage to crops by reducing yield, quality, and overall productivity (Karar *et al.*, 2022). Insect pests, for example, feed on plant tissues, while fungal infections can cause disease that rots crops. As a result, pest management has become a crucial aspect of modern farming practices aimed at safeguarding agricultural output (Masood, 2023). The impact of pests on agriculture extends beyond crop damage; they also affect food security and environmental sustainability. For instance, pest infestations can lead to severe crop losses, which ultimately drive-up food prices and affect both local and global food supply chains (Alanazi *et al.*, 2023). Additionally, pests can reduce the quality of harvested products, making them unsuitable for consumption or sale, leading to economic losses for farmers.

Managing pests in agriculture requires a comprehensive approach that encompasses traditional methods such as chemical pesticides and biological controls as well as modern technologies. Integrated pest management (IPM) aims to use a combination of strategies to control pest populations while minimizing environmental harm (Anwar and Masood, 2023). However, growing resistance to chemical pesticides and the adverse effects of excessive chemical use on the environment have pushed the agricultural industry toward more sustainable, tech-driven solutions, such as precision agriculture, which uses sensors, drones, and machine learning models to detect and manage pests more effectively. This shift highlights the need for continued innovation in pest control to maintain crop health and ensure long-term agricultural sustainability (Karar *et al.*, 2022).

Pest detection is a critical component of integrated pest management (IPM) in agriculture, as it helps identify infestations early and ensures timely intervention to protect crops. Various methods are employed for detecting pests, ranging from traditional manual techniques to advanced technological approaches (Anwar and Masood, 2023). Each method has its advantages and limitations, and modern agriculture increasingly incorporates automation and technology to improve accuracy and efficiency. Machine learning and computer vision are modern methods used to detect pests through image recognition and data analysis. Pre-trained neural networks are used to analyze images of crops, identifying pests or pest damage with high accuracy (Bhoi *et al.*, 2021; Sochima *et al.*, 2025; Deepika and Arthi, 2022).

Deep learning is increasingly being integrated into autonomous robotic systems for pest management (Kekong *et al.*, 2019). Robots equipped with cameras and deep learning algorithms can navigate fields, scanning crops for signs of pest infestations (Sun *et al.*, 2023). These systems are not only capable of detecting pests but also performing targeted pest control measures, such as spraying pesticides or releasing beneficial organisms. This approach enhances the efficiency of pest management and reduces labor costs, while minimizing the environmental impact by focusing pest control efforts only where necessary.

Pest has remained a major issue for farmers over the years and has continued to witness increased research attention with diverse proposals to help address the problem. Among the most recent studies reviewed, Kumar *et al.* (2023) applied light weight YOLOV-5 for the classification of pest using field adaption method. While this work reported good classification success for pest detection in a farm, there is gap due to false alarm, region specification of the model that is, the model is limited to best within the region of data collection), misclassification, and may not be reliable for real time remote sensing applications.

## RESEARCH METHOD

The system will be made of three major components which are the classification model, real time notification system and UAV. The proposed classification and notification model will use image data of pest collected from the Federal College of Agriculture, Ishiagu, Ebonyi State, Nigeria and then fine tune existing pest data which will be collected in roboflow repository to develop a comprehensive data model. The data will be used to train a deep learning algorithm, specifically YOLO-V10 which is a more recent and advance version of YOLO-V, 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 block diagram was presented in Figure 1.

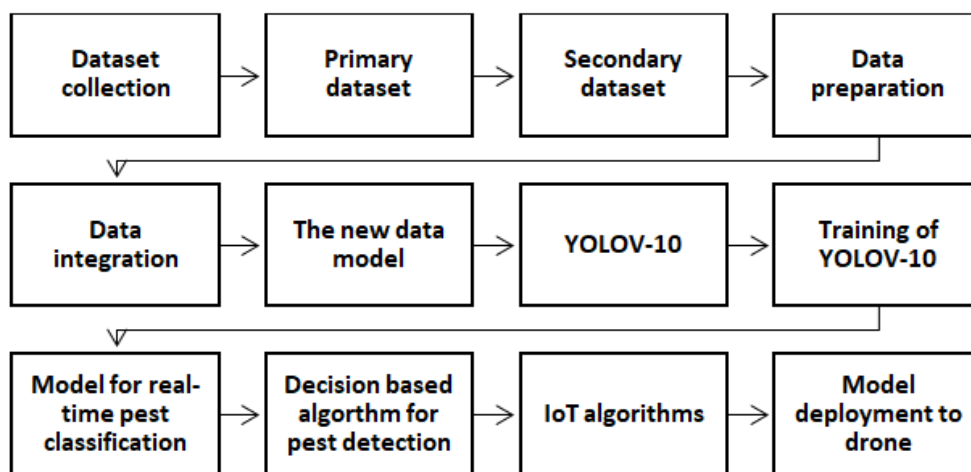


Figure 1: Block diagram of the proposed system

Figure 1 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 YOLOv-10 which 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 model will be deployed into the drone for remote sensing purposes.

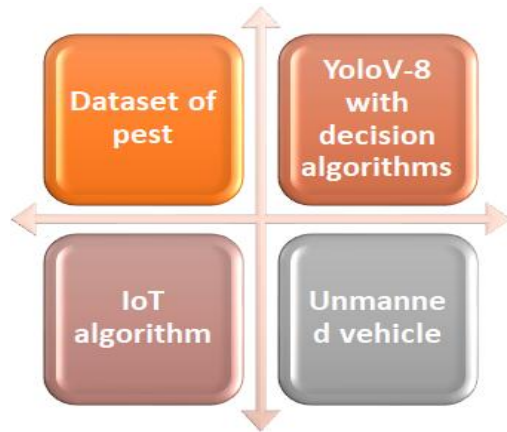


Figure 2: Proposed system for the management of pest in agriculture

Figure 2 presents the proposed system for the management of pest in agriculture using dataset of pests, deep transfer learning, an IoT algorithm and control measures in this order. The comprehensive data model which composed of localized pest in Nigeria was used to train a YOLOV-10 algorithm and generate a model for the real time classification of pest in the farm. The model was used as foundation to develop a decision-based model for the classification of pests in the farm, while the classified output was communicated to the farmer using IoT algorithm, which will be incorporated with the UV for real time monitoring purposes.

## The Research Area

The research was conducted at the Federal College of Agriculture, Isiagu, located in Ivo Local Government Area of Ebonyi State, Nigeria. This institution is known for its agricultural research and training, making it an ideal site for collecting pest-related image data in real field conditions. The geographical coordinates of the location are approximately latitude  $6.0055^{\circ}$  N and longitude  $7.5609^{\circ}$  E. The region falls within the tropical rainforest zone of southeastern Nigeria, characterized by high humidity and substantial rainfall, which creates a suitable environment for both crop cultivation and pest proliferation. These conditions provided a realistic setting for observing and capturing diverse pest infestations affecting various crops. Figure 3 presents the research area Map.

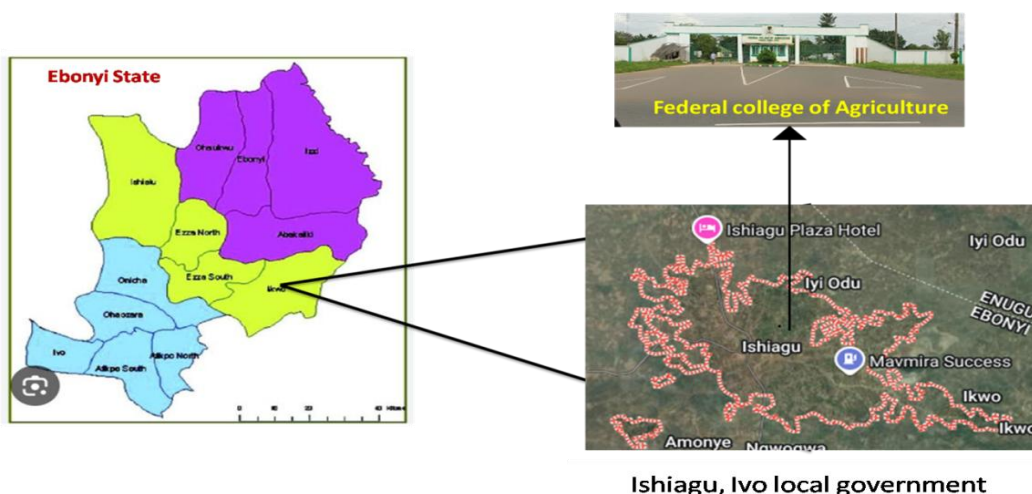


Figure 3: Research Area Map

## Data collection

The data used for this work are secondary and primary data. The primary data was collected at the Federal College of Agriculture, Ishiagu, Ebonyi State, Nigeria. The sample size of data collected was 209 images of pest (aphids, armyworms, bees, beetle, bugs, loopers, caterpillar, citrus canker, beetles, corn earth-worms and corn borers). The experimental setup for this work constitutes several materials such as a universal serial bus, high definition camera, Raspberry Pi, the farm, power supply bank, and laptop. The secondary data is the insect pest detection dataset from Kaggle repository (source: <https://www.kaggle.com/datasets/cubeai/insect-pest-detection-for-yolo>). The sample size of the data is 17641. The overall data size is 17850. The data sample is presented as Figure 4;



Figure 4: Pest data from secondary source (Source: Kaggle)

## Data Preparation

The data collected was prepared through a systematic process of annotation and labeling using the Roboflow Toolbox. This tool enabled precise identification and classification of various pests in the images, such as aphids, armyworms, bees, beetles, bugs, loopers, caterpillars, citrus canker, corn earworms, and corn borers. Each image was annotated with bounding boxes to indicate the exact location of pests, and labeled accordingly to reflect the pest type. After annotation, the dataset was split into training, validation, and testing sets to ensure efficient machine learning model development. Image augmentation techniques such as rotation, flipping, and brightness adjustment were also applied within Roboflow to increase dataset diversity and improve model generalization during training.

## The Computer Vision Systemfor Real-Time Pest Detection and Classification using Pre-Trained Technique

This section presents the computer vision algorithm model of this work using pre-trained model. The type of pre-trained model adopted is YOLOV-10 (Ebere *et al.*, 2025). The architecture is presented in Figure 5.

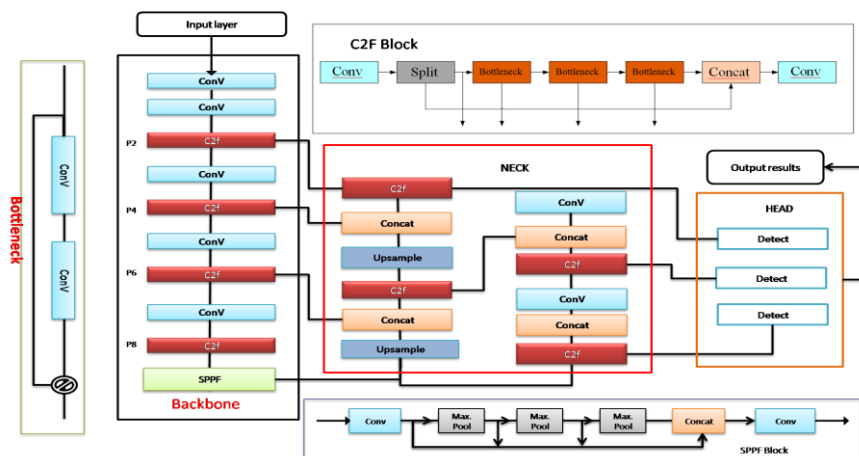


Figure 5: Architecture of the YOLOV-10 (Ebere *et al.*, 2025)

Figure 5 illustrates the architecture of the YOLOv10 model, which is organized into four key components: the input, backbone, neck, and head. Each of these plays a distinct role in the object detection pipeline, contributing to the model's overall performance and accuracy. The input stage defines the dimensions of the image provided to the model. Typically, images are resized to a fixed resolution ( $640 \times 640$  pixels) to ensure consistency during training and inference. This input is then passed into the backbone for further processing.

The backbone is responsible for feature extraction. It processes the input image using a series of convolutional layers to capture low-level and mid-level features such as edges, textures, and object shapes. This stage plays a crucial role in building a detailed representation of the image content. Following the backbone is the neck, which performs multi-scale feature fusion. It combines features from various layers of the backbone to ensure that the model can detect both small and large objects effectively. This aggregation improves the model's robustness when working with objects of varying sizes and spatial locations. The head produces the final detection output. It interprets the processed features to generate bounding boxes, class probabilities, and confidence scores. These outputs are then used to identify the presence and position of objects in the image.

Several specialized modules support the internal functioning of the model. The Conv layers (convolutional operations) are used for scanning the input and extracting essential features. The C2f module (Cross-Stage Partial with Fusion) houses the bottleneck layers, which often incorporate pre-trained models to improve learning efficiency and reduce overfitting. The Concat operation merges features from different layers, enriching the feature map with more diverse information. Finally, the SPPF (Spatial Pyramid Pooling Fast) module is a critical component at the end of the backbone, designed to extract features across multiple receptive fields by applying pooling at various scales. This enables the model to better understand spatial hierarchies within the image.

## Pest Management Model

The pest management model was developed through expert consultation at the Research area (FCA, Ishiagu). Qualitative data was collected on pest and their recommended control solution. The results of the data collected were reported in the Table 1.

Table 1: Pest control Data (Source: FCA, ISHIAGU)

| S/N | Pest             | Recommendation pesticides   |
|-----|------------------|-----------------------------|
| 1   | Aphids           | Insecticidal soap           |
| 2   | Armyworms        | Ampligo                     |
| 3   | bees             | No control method for now   |
| 4   | Canker           | CypeForce                   |
| 5   | Bugs             | Imidacloprid                |
| 6   | Lopers           | CaterpillarForce            |
| 7   | Caterpillar      | CaterpillarForce            |
| 8   | Citrus canker    | CypeForce                   |
| 9   | Beetles          | Cypermethrin DudUALL 450 EC |
| 10  | Corn earth worms | No chemical control         |
| 11  | Corn borers      | CypeForce                   |

The rule-based recommendation algorithm is presented as;

| Rule-Based Recommendation Algorithm                                                                                                                                                                                                                                                                                        |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <ol style="list-style-type: none"> <li>1. Initialize the system.</li> <li>2. Connect to pest classification models.</li> <li>3. Load Table 1.</li> <li>4. Column 1: Serial Number (S/N)</li> <li>5. Column 2: Pest</li> <li>6. Column 3: Recommended Control Pesticide(s)</li> <li>7. For each pest in Table 1:</li> </ol> |

8. Identify the pest.
9. Apply the corresponding pesticide(s) listed in Column 3.
10. End For
11. Stop the process.

## UAV for Remote Sensing and Pest Monitoring

The modelling of the UAV was adopted from Osiogun *et al.* (2019) and developed using structural method via architectural universal modelling diagram to present the various section of the system as shown in the Figure 6;

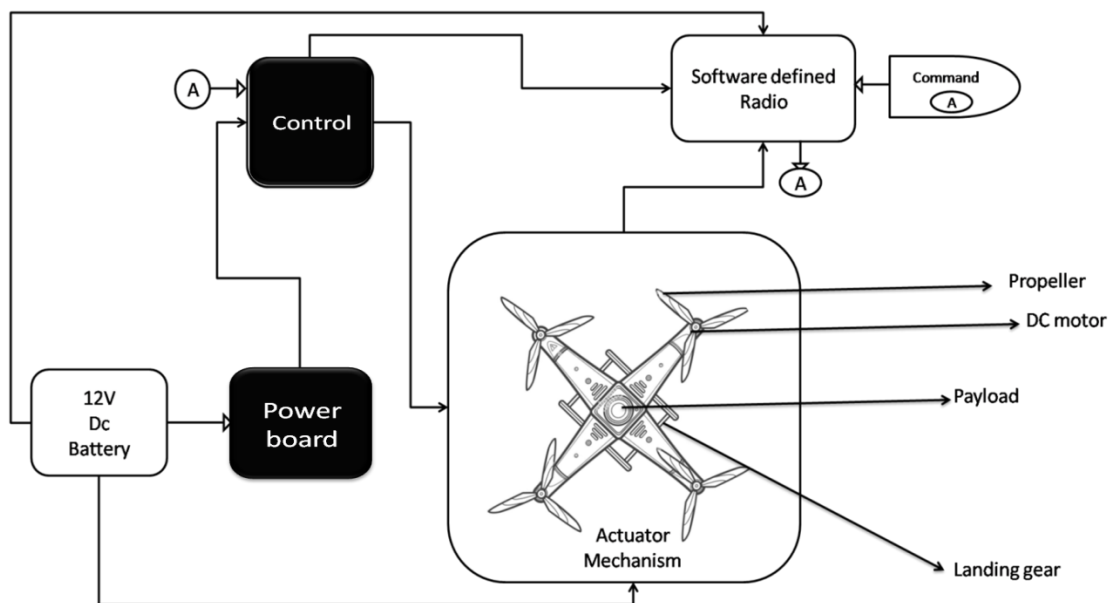


Figure 6: Architectural Model of the Unmanned Vehicle (Osiogun *et al.*, 2019)

The UV model in Figure 6 is made of the various sections which are the power supply, control system, actuator system, sensors and software defined radio. The power supply is a mini drone battery which supplied regulated direct current to the system. The control section of it was made of pitch and yaw adjustment controller and also a fuzzy logic control system (Habor *et al.*, 2021). The pitch and yaw adjustment controller was used for the aerodynamic stability control of the whole system, while the fuzzy logic controller was used for the training and control of the sensor. The actuator mechanism is the kinematic section of the system which is composed of the propeller, DC motor, payload (Software Defined Radio (SDR)) and the landing gears. The SDR is a communication system equipped with 4G wireless devices which received command from the operators to control the drone and also sends the results of the data collected from the sensor for observation and analysis to the monitoring centre (Ulagwu- Echegu *et al.*, 2021).

## Model of the UAV Behaviour in the Farm

UAV is generally described with kinematics and dynamic model. The kinematic model established the geometric relationship between positions, velocity, orientation and angular rate without considering impact of external forces. It is dependent on the transformation of frames matrices to map out body velocities into inertia-frame position and relate Euler angles and angular rates changes with time. The dynamic model on the other hand is based on the Newton Euler laws model, the UAV translational and rotational acceleration to the total external forces and moments which acts on it. Collectively these formulations provide the foundation for the UAV modelling.

## The kinematic model of the UAV

The kinematics model is developed considering the linear and angular kinematic motion model. The linear motion is represented considering velocity components  $(v_x, v_y, v_z)$ , along each axis within the coordinated

frame of the body. The translational position  $(x, y, z)$  for UAV is determined in an inertial reference frame. To relate these velocity vectors in the rate of change of position vector and body frame, rotational transformation and differentiation are applied as shown in Equation 1 (Ahmed et al., 2025).

$$\frac{d}{dt} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = R_b^i \begin{pmatrix} v_x \\ v_y \\ v_z \end{pmatrix} \quad (1)$$

Where  $x$  is the position along  $x$  axis,  $y$  is the position along  $y$  axis and  $z$  the position in the  $z$  axis.  $v_x, v_y, v_z$  are the linear velocities of objects along the three axis  $(x, y, z)$ .  $R_b^i$  is the rotational matrix transformation of the UAV body frame and represented in Equation 2 (Rahul et al., 2018; Gu et al., 2023).

$$R_b^i = \begin{pmatrix} \cos\theta \cos\varphi & \sin\theta \sin\varphi \cos\theta - \cos\theta \sin\varphi & \cos\theta \sin\varphi \cos\theta - \sin\theta \sin\varphi \\ \cos\theta \sin\varphi & \sin\theta \sin\varphi \sin\theta - \cos\theta \cos\varphi & \cos\theta \sin\varphi \sin\theta - \sin\theta \cos\varphi \\ -\sin\theta & \sin\theta \cos\theta & \cos\theta \cos\theta \end{pmatrix} \quad (2)$$

### 2.7.2. Angular motion of the UAV

This describes the relationship between the Euler angles  $(\phi, \theta, \varphi)$  from the three distinct coordinate's frames and angular rates  $(p, q, r)$  in the body frame and represented in Equation 3.

$$\begin{pmatrix} \phi \\ \theta \\ \varphi \end{pmatrix} = R_b^r \begin{pmatrix} p \\ q \\ r \end{pmatrix} \quad (3)$$

Where  $p$  is the roll rate and signifies the angular velocity within the  $x$ -axis,  $q$  is the pitch rate which is the angular velocity within  $y$  axis and corresponds to pitch axis.  $R$  is the yaw rate around the  $z$  axis angular velocity.  $\phi$  is the roll angle,  $\theta$  is the pitch angle and  $\varphi$  is the yaw angle. The transformation matrix of these frames to the body frame is represented as Equation 4;

$$R_b^r = \begin{pmatrix} 1 & \sin\phi \tan\theta & \cos\phi \tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \frac{\sin\phi}{\cos\theta} & \frac{\cos\phi}{\cos\theta} \end{pmatrix} \quad (4)$$

### Dynamic model

This section presents the dynamic model of the UAV, explaining how the system behaves under external force, based on the Newton-Euler Law of motion. The model was adopted from Ahmed et al. (2025) and developed with assumption that the ground level is flat. The movement of the UAV therefore is converted from body frame to earth frame through rotational matrices. Based on these relations, the dynamic movement of the UAV is defined Equation 5

$$\frac{d}{dt_i} P = \frac{d}{dt_b} P + w_{b/i} * P \quad (5)$$

Where  $P$  is the frame derivatives as it changes, while  $w$  is the angular velocity in the inertia frame. To model the UV motion during translation, the second Newton's law was applied as shown in Equation 6;

$$\sum F = \frac{d}{dt_i} (\text{linear momentum}) = mV_i \quad (6)$$

Where  $F$  is the external force which arising from aerodynamic forces, gravity and propulsion control of the UAV.  $m$  is mass,  $V_i$  is inertia velocity derived from Equation 5 and expressed in angular velocity defined in Equation 7, while the body frame is defined in Equation 8.

$$\frac{dV_B}{dt_i} = \frac{dV_b}{dt_b} + w_{b/i} * V_b \quad (7)$$

$$\sum F = mV_b + w_{b/i} * V_b \quad (8)$$

Where  $F_b(f_z, f_x, f_y)$ ,  $V_b(v_x, v_y, v_z)$  and  $w_{b/i}(p, q, r)$ ,  $V_b$  the velocity vector in the body frame. Consequently, the three equations that govern translational dynamics can be outlined as follows;

$$\begin{pmatrix} \dot{v}_x \\ \dot{v}_y \\ \dot{v}_z \end{pmatrix} = \begin{pmatrix} rv_y & -qv_z \\ pv_z & -rv_x \\ qv_x & -pv_y \end{pmatrix} + \frac{1}{m} \begin{pmatrix} f_x \\ f_y \\ f_z \end{pmatrix} \quad (9)$$

The rotational motion model was also developed with Newton second law which postulated Equation 10.

$$EM = \frac{d}{dt_i}(\text{angular momentum}) = \frac{dH}{dt_i} \quad (10)$$

Where M is the external moment. This equation is expanded applying Equation 10 to the rate of change of angular momentum with respect to time in the inertia frame as shown in Equation 11;

$$EM = \frac{dH}{dt_i} + w_{b/i} * H_b \quad (11)$$

Where  $H_b$  is the angular momentum as a result of the inertia matrix (J) by the angular vector (Lopez-Briones et al., 2020).

$$H_b = jw_{b/i} \quad (12)$$

From Equation (11 and 12), the Equation 13 and 14 are obtained as;

$$\sum M_b = \frac{djw_{b/i}}{dt} + w_{b/i} * jw_{b/i} \quad (13)$$

$$w_{b/i} = \begin{pmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{pmatrix} = J^{-1}[-w_{b/i} * (jw_{b/i}) + M] \quad (14)$$

$$\text{Where } J = \begin{pmatrix} j_{xx} & j_{xy} & j_{xz} \\ j_{yx} & j_{yy} & j_{yz} \\ j_{zx} & j_{zy} & j_{zz} \end{pmatrix} \quad (15)$$

The J diagonal matrix represents moments of inertia, which indicates the UAV resilience to acceleration within specified axes of motion (Chi et al., 2015). The off-diagonal matrix is the inertia product with certain inertia neglected with assumed zero minimal impact.

$$\text{Where } J = \begin{pmatrix} j_{xx} & 0 & 0 \\ 0 & j_{yy} & 0 \\ 0 & 0 & j_{zz} \end{pmatrix} \quad (16)$$

## Develop a real time notification algorithm

In smart agricultural systems and other intelligent monitoring platforms, timely communication of critical events is essential to enable users to take prompt and effective action. This section presents a real-time notification algorithm designed to automatically alert users through email when predefined events are detected by the system. The algorithm ensures that relevant stakeholders are informed with actionable recommendations, helping them respond efficiently to emerging situations. The notification mechanism operates by continuously monitoring system conditions, generating personalized alert messages, and

dispatching them through secure email channels. Below is the stepwise process of the notification algorithm. The stepwise is presented as;

#### Real-Time Email Notification Algorithm (Stepwise)

1. Trigger Event Detection%% Monitor for specific trigger conditions
2. IF system\_event == TRUE AND confidence\_level  $\geq$  THRESHOLD
3.  $\rightarrow$  proceed to notification
4. Set user information
5. Retrieve User Information %% Fetch user data (e.g., email address, name)
6. Generate Message Content %% Generate the email message content
7. Send Email Notification %% Connect to an SMTP server and send the email
8. Log Notification Record%% Store a log entry for traceability and confirmation.
9. End

#### System Integration

This section presents the integrated system developed for real-time pest monitoring and decision support. The integration process began with the successful training and validation of the pest detection model using the annotated pest dataset. Once the model achieved satisfactory performance, it was exported and deployed on a lightweight edge device compatible with the UAV system. The email notification module was then integrated by linking it to the model's inference output. Whenever pests were detected, the system automatically generated an alert and sent an email containing actionable information to the farmer's registered address. This was achieved using a Python-based SMTP script embedded within the inference pipeline. Figure 7 presents the architecture of the UAV.

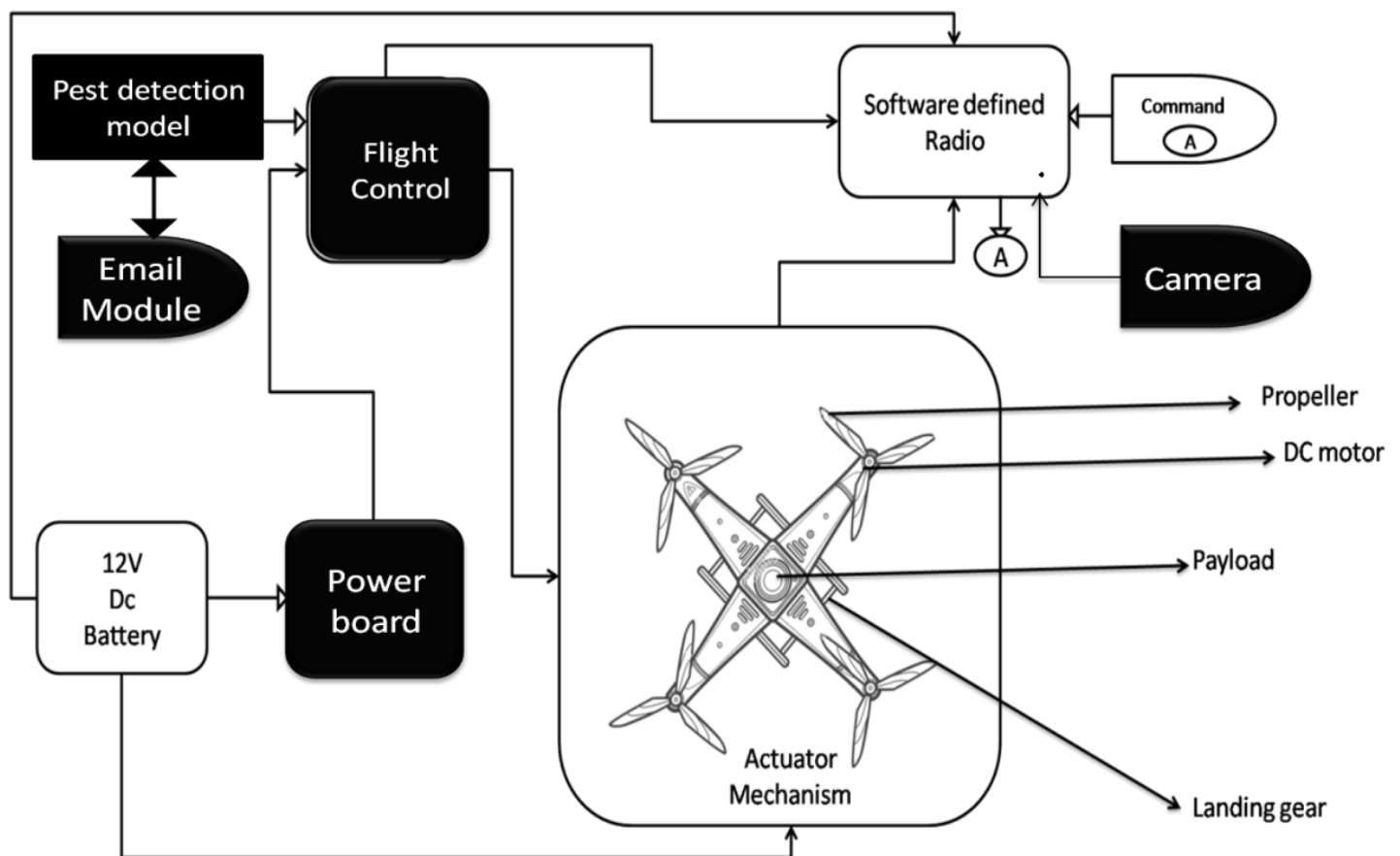


Figure 7: Architecture of the UAV with pest detection and notification module

#### Training of the model with the pest dataset

The YOLOv10 model was trained using a custom pest dataset consisting of annotated images representing various pest classes such as aphids, armyworms, beetles, bugs, and bees. The dataset was pre-processed to

ensure uniformity in image dimensions, quality, and labelling format. All images were resized to  $640 \times 640$  pixels to align with the model's input requirements, while data augmentation techniques such as horizontal flipping, scaling, and rotation were applied to enhance the model's generalization capability. The dataset was divided into training, validation, and test sets in a ratio of 70:20:10. This ensured that the model was trained on a substantial portion of the data while being evaluated on unseen samples for performance assessment. Labelling was performed using the Roboflow annotation tool, and the dataset was converted to the YOLO format, which includes text files containing class indices and normalized bounding box coordinates for each image.

Training was conducted using the PyTorch framework on a GPU-enabled environment to accelerate computation. A batch size of 16 and an initial learning rate of 0.001 were used. The Adam optimizer was selected for efficient convergence, and the learning rate was dynamically adjusted using a cosine annealing scheduler. The model was trained over 100 epochs, with performance metrics such as training loss, validation loss, mean Average Precision (mAP), and F1-score monitored at each epoch to guide early stopping and model checkpointing.

### Result Of The Yolov-10 Training

This section discusses the performance results of the YOLOv10 benchmark model after training it on the annotated pest dataset. The training process was carefully monitored, and several evaluation metrics were recorded to assess the model's accuracy in detecting and classifying different pest species. The results demonstrate how well the model learned from the data and its capacity to generalize to new, unseen images.

During the training phase, the model achieved a bounding box loss of 0.65. This metric indicates the accuracy with which the model could predict the location of pests in an image. A low bounding box loss value signifies that the model was effective at learning object localization. Similarly, the training classification loss was recorded at 0.64, showing the model's ability to correctly assign labels to the pests it identified. The training focus loss, which evaluates how well the model concentrated on the most relevant parts of the image, stood at 0.54, reflecting good spatial awareness during training. Figure 8 presents the training results.

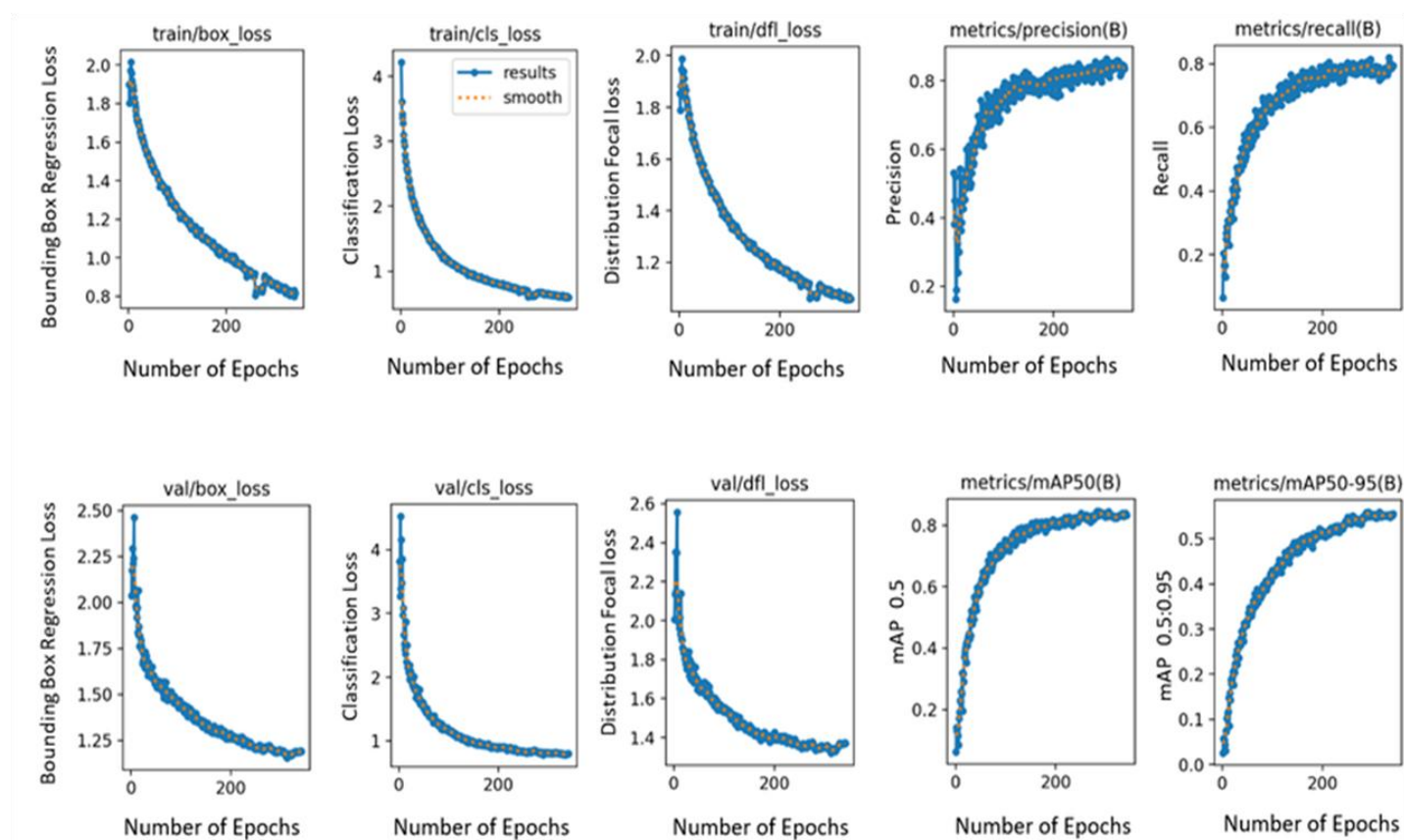


Figure 8: Training result of the benchmark YOLOV-10N

On the validation dataset, the model recorded a bounding box loss of 1.002. While slightly higher than the training value, this is still within an acceptable range and suggests that the model has not significantly overfitted the training data. The validation classification loss was 0.68, and the focus loss increased to 1.2. These validation losses indicate that the model maintained reasonable accuracy in pest identification on new data, although the increased focus loss suggests some difficulty in attending to exact object regions during validation.

In terms of detection accuracy, the model achieved a precision score of 0.84. This means that 84% of the objects detected by the model were actual pests, with few false positives. The recall was 0.82, indicating that the model was able to successfully detect 82% of all true pest instances in the dataset, minimizing false negatives. Furthermore, the model scored 0.83 on mean Average Precision at an intersection over union threshold of 0.5 (mAP@50). This shows that the model was highly effective in identifying pests with at least 50% overlap with the ground truth boxes. When evaluated across a more rigorous range of IoU thresholds (mAP@50–90), the model recorded a score of 0.60, which still reflects solid performance across various levels of detection difficulty.

The YOLOv10 benchmark model performed strongly across all major training and validation metrics. The losses remained within acceptable ranges, and the evaluation scores confirm that the model is capable of accurately identifying and classifying pests in real-time. These results support the model's integration into a practical decision support system for pest management using UAVs or smart agricultural surveillance platforms.

### The UAV simulation results

This section presents the result of the UVA applied for remote sensing of the farm environment for pest detection. The results were generated with Simulink as the tool and the simulation parameters used was adopted from Oti-Owom et al. (2024) and was applied to simulate the flying of the UAV, while capturing data from the camera and then classifying pest. Figure 9 present the Simulink block diagram of the UAV, while Table 2 presents the simulation parameters.

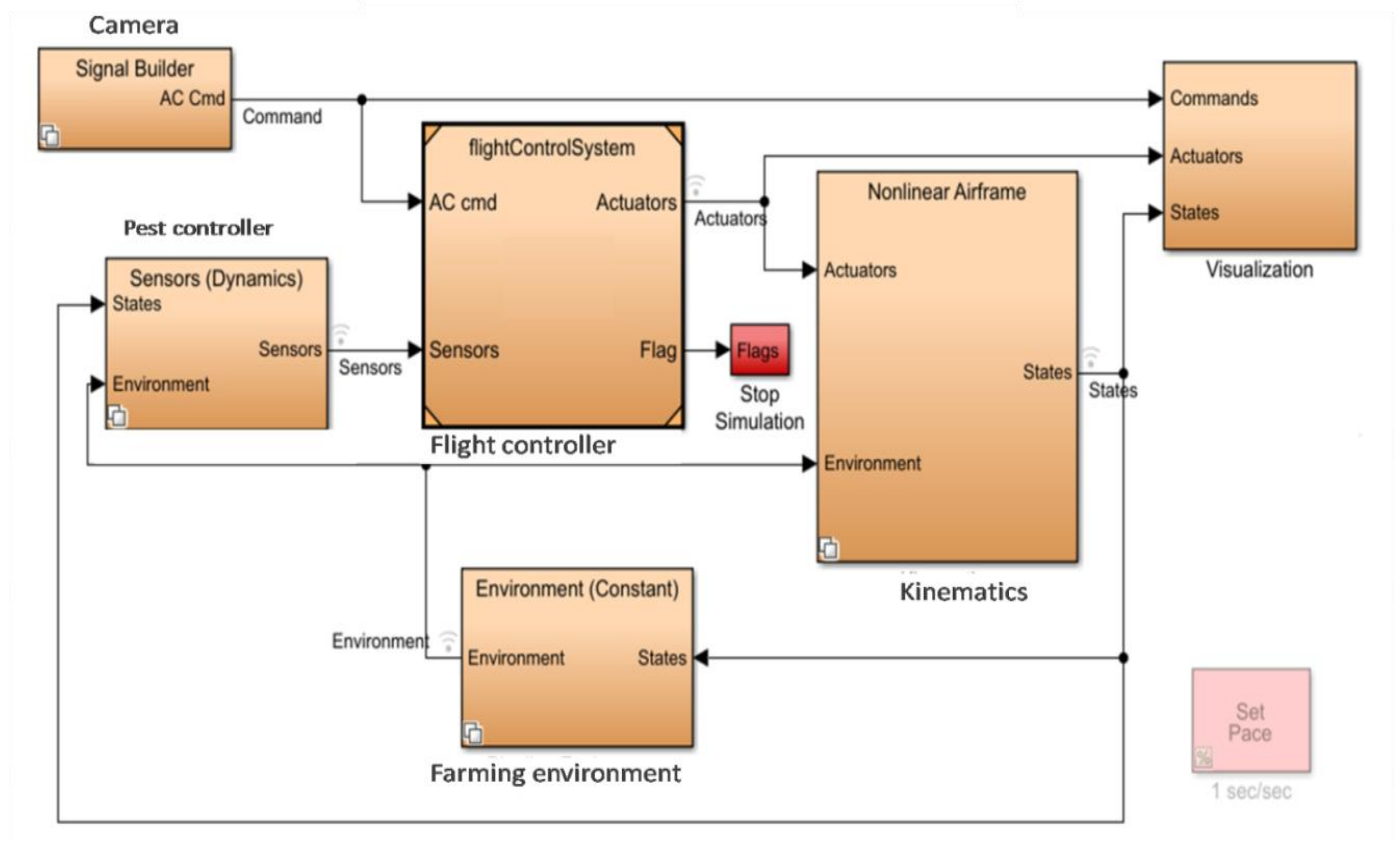


Figure 9: Simulink block diagram

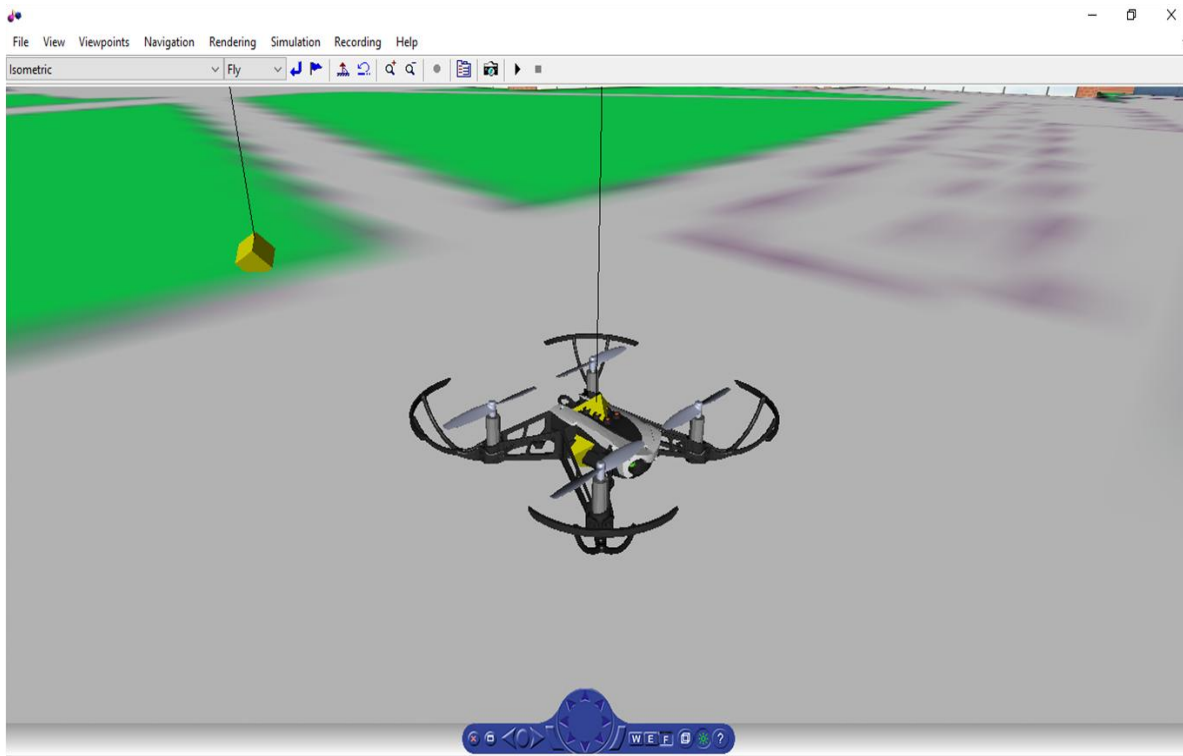


Figure 10: Result of the UV in Simulink

The Figure 10 presents the results of UAV in Simulink. This result was achieved from the simulation using simulation parameters of UV adopted from Osisiogwu et al. (2019) to run the program in Simulink 3D. The parameters used to run the simulation program are all reported in Table 2.

Table 2: Simulation parameters (Osisiogwu et al., 2019)

| Parameter                 | Value / Description         | Unit / Type             |
|---------------------------|-----------------------------|-------------------------|
| UAV Model                 | DJI Phantom 4 Pro           |                         |
| Flight Altitude           | 20 – 30                     | meters                  |
| Camera Resolution         | 3840 × 2160                 | pixels (4K UHD)         |
| Camera Type               | RGB with fixed focal length |                         |
| Frame Rate                | 30                          | frames per second (fps) |
| Image Capture Interval    | 2                           | seconds                 |
| Flight Speed              | 3 – 5                       | m/s                     |
| Flight Time per Session   | 20 – 25                     | minutes                 |
| Battery Capacity          | 5870                        | mAh                     |
| GPS Accuracy              | ±1                          | meter                   |
| Area Coverage per Flight  | 2 – 3                       | hectares                |
| Number of Images Captured | 209                         | images                  |
| Detection Algorithm       | YOLOv10n + Improved C2F     | -                       |

The UAV-based pest monitoring system proposed using YOLOv10 was effective in the real-time detection with the precision and recall values of 0.84 and 0.82, respectively. The combination of IoT-based alerts and rule-based pesticide suggestions offers the entire decision-support system to the farmers. The UAV simulation established the stable flight behaviour and dependable image data collection to monitor the pests in remote areas which confirmed that the system is ready to be utilised in realistic setting of precision agriculture.

## CONCLUSION

This paper came up with a smart and technology based real-time pest detection and management in agriculture through deep learning, IoT and integration of UAVs. The study focused on the weaknesses of conventional pest

management practises and models regionally by presenting an extensive framework of integrating image-based pest detection and automated decision-making and notification systems. Primary sources were used to form a hybrid dataset at the Federal College of Agriculture, Ishiagu, and secondary data at the Kaggle repository. The data were labelled and trained on the YOLOv10 model, the type of which was selected due to its high speed and its ability to perform the task of object detection.

The trained YOLOv10 model had good performance scores of 0.84, 0.82, and 0.83 of precision, recall, and mAP@50 respectively, which proved to identify and classify different pest species accurately. A recommendation algorithm based on rules was also designed to recommend appropriate actions to be taken to deal with each of the identified pests. To guarantee the farmers responded in time, an email notification algorithm with the help of IoT was incorporated with a detection system, allowing real-time notification of the farmers in case pests were detected. The simulated UAV model based on Simulink gave a dependable aerial pest surveying model with steady flight controls and onboard image processing.

To summarise, the suggested system is able to combine innovative computer vision, IoT, and UAV to improve efficiency in pest monitoring and management in agriculture. The capabilities of the model to identify pests with high precision and provide useful insights that can be acted on by farmers in real-time can be used to show its feasibility in the application of the model to address precision farming. In addition to minimising the need of chemical pesticides, it reduces losses and enhances informed decision-making to support sustainable agriculture by manipulating aspects of crop health and productivity.

## REFERENCES

1. Ahmed, E., Ehab, S., Mohamed, R., & Ahmed, M. (2025). Model-based simulation and validation of small fixed wing UAV. *Engineering Science and Military Technologies*, 9(1). <https://doi.org/10.21608/EJMTC.2024.243244.1267>
2. Alanazi, A., Shakeabubakor, A., Abdel-Khalek, S., & Alkhalaf, S. (2023). IoT enhanced metaheuristics with deep transfer learning based robust crop pest recognition and classification. *Alexandria Engineering Journal*, 84, 100–111. <https://doi.org/10.1016/j.aej.2023.11.008>
3. Anwar, Z., & Masood, S. (2023). Exploring deep ensemble model for insect and pest detection from images. *Procedia Computer Science*, 218, 2328–2337.
4. Bhoi, S., Jena, K., Panda, S., Long, H., Kumar, R., Subbulakshmi, P., & Jebreen, H. (2021). An Internet of Things assisted unmanned aerial vehicle-based artificial intelligence model for rice pest detection. *Microprocessors and Microsystems*, 80, 103607. <https://doi.org/10.1016/j.micpro.2020.103607>
5. Chi, S., Lee, S., Ryu, H., Shim, H., & Ha, C. (2015). Dynamics and simulation of the effects of wind on UAVs and airborne wind measurement. *The Japan Society for Aeronautical and Space Sciences*, 58(4), 187–192.
6. Deepika, P., & Arthi, B. (2022). Prediction of plant pest detection using improved mask FRCNN in cloud environment. *Measurement: Sensors*, 24, 100549. <https://doi.org/10.1016/j.measen.2022.100549>
7. Ebere Uzoka, C., Anoliefo, E., Udanor, C., Chijindu, T., & Nwobodo, L. (2025). A blind navigation guide model for obstacle avoidance using distance vision estimation based YOLO-V8n. *Journal of the Nigerian Society of Physical Sciences*, 2292–229. <https://doi.org/10.46481/jnsps.2025.2292>
8. Gu, Y., Zhang, G., Bi, Y., Meng, W., Ma, X., & Ni, W. (2023). Pitch mathematical modeling and dynamic analysis of a HALE UAV with moving mass control technology. *Aerospace*, 10(11), 918. <https://doi.org/10.3390/aerospace10110918>
9. Harbor M.C, Eneh I.I. Ebere U.C. (2021). Precision control of autonomous vehicle under slip using ANN. *International Journal of Research and Innovation in Applied Science (IJRIAS)*. Vol 6; Issue 9. <https://rsisinternational.org/journals/ijrias/DigitalLibrary/volume-6-issue-9/62-68.pdf>
10. 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 depth wise convolutional networks. *Alexandria Engineering Journal*, 61, 5309–5319. <https://doi.org/10.1016/j.aej.2021.10.050>
11. Kekong P.E, Ajah I.A., Ebere U.C. (2019). Real-time drowsy driver monitoring and detection system using deep learning based behavioural approach. *International Journal of Computer Sciences and Engineering* 9 (1), 11-21; [http://www.ijcseonline.isroset.org/pub\\_paper/2-IJCSE-08441-18.pdf](http://www.ijcseonline.isroset.org/pub_paper/2-IJCSE-08441-18.pdf)

12. López-Briones, F., Sánchez-Rivera, M., & Arias-Montano, M. (2020). Aerodynamic analysis for the mathematical model of a dual-system UAV. *17th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE)*, 1–6.  
<https://doi.org/10.1109/CCE50788.2020.9299157>
13. Oti-Owom, J. O., Eke, J., & Umeozulu, A. (2024). Modelling of intelligent robot for gas pipeline leakage detection and control using deep learning technique. *Journal of Engineering and Technology*, 1(7).
14. Rahul, A., Divya, P., & Shishirkumar, K. (2018). Mathematical modeling and simulation of quadcopter-UAV using PID controller. *4th International Conference on Engineering Confluence & Inauguration of Lotfi Zadeh Center of Excellence in Health Science and Technology (LZCODE) – EQUINOX 2018*.
15. Sochima V.E. Asogwa T.C., Lois O.N. Onuigbo C.M., Frank E.O., Ozor G.O., Ebere U.C. (2025)”; Comparing multi-control algorithms for complex nonlinear system: An embedded programmable logic control applications;  
DOI: <http://doi.org/10.11591/ijpeds.v16.i1.pp212-224>
16. 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>
17. Ulagwu-Echefu A., Eneh .I.I. Ebere U.C. (2021). Enhancing realtime supervision and control of industrial processes over wireless network architecture using model predictive controller. *International Journal of Research and Innovation in Applied Science (IJRIAS)*; vol 6; Issue 9.  
<https://rsisinternational.org/journals/ijrias/DigitalLibrary/volume-6-issue-9/56-61.pdf>