UDC 004.8
DOI: 10.15587/1729-4061.2025.331285

DESIGN OF AN INTELLIGENT SYSTEM TO CONTROL THE DEVICE FOR RECOGNIZING THE BREAD STRIPED FLEA (*PHYLLOTRETA VITTULA*)

Akerke Akanova
PhD, Senior Lecture*
Galiya Anarbekova
Corresponding author
Master of Natural Sciences, Doctoral Student*
E-mail: galiya2810@mail.ru
Mira Kaldarova
PhD, Senior Lecture
School of Information Technology and Engineering
Astana International University
Kabanbay Batyr ave., 8, Astana, Republic of Kazakhstan, 010000
Nazira Ospanova
PhD, Associate Professor, Head of Department
Department of Information Technology
Toraighyrov University
Lomov str., 64, Pavlodar, Republic of Kazakhstan, 140008
Saltanat Sharipova
PhD
Astana IT University
Mangilik El ave., 55/11, EXPO Business Center, Block C1,
Astana, Republic of Kazakhstan, 010000
*Department of Computer Engineering and Software
S. Seifullin Kazakh Agrotechnical Research University
Zhenis ave., 62, Astana, Republic of Kazakhstan, 010000

The object of this study is an autonomous Raspberry Pi-based device for real-time pest detection. The task addressed relates to the lack of affordable, energy-efficient, and autonomous solutions for working in the field without an Internet connection.

The paper reports the design of an intelligent device for pest monitoring. The device is focused on automatic recognition of the striped grain flea beetle (Phyllotreta vittula) in grain crops. As a result of the study, a system was designed based on the Raspberry Pi 4.0 microcomputer using the OpenCV library and the YOLO model. The device processes the video stream, identifies pests, and saves data locally. The system provides high accuracy at low power consumption. This was made possible by a lightweight neural network architecture and optimized image processing. A distinctive feature of the solution is autonomy, mobility, and resistance to variable lighting conditions. The system also works with limited computing resources.

The results demonstrate that the device could be effectively used in precision farming systems and at scientific institutions. The device helps identify pests and make agricultural decisions at early stages of infection. The technological advancement could be adapted to other types of pests with minimal changes to the model. In the future, the system could be integrated into broader agricultural monitoring platforms with the ability to transfer data to the cloud. The practical use of the device is possible both in large farms and on private farms. This technological advancement is especially relevant for regions with limited technical infrastructure

Keywords: artificial intelligence, Raspberry Pi, YOLO, OpenCV, pest monitoring, agriculture, computer vision

Received 05.03.2025
Received in revised form 15.04.2025
Accepted 27.05.2025
Published 25.06.2025

How to Cite: *Akanova, A., Anarbekova, G., Kaldarova, M., Ospanova, N., Sharipova, S. (2025). Design of an intelligent system to control the device for recognizing the bread striped flea (Phyllotreta vittula). Eastern-European Journal of Enterprise Technologies, 3 (5 (135)), 39–48. https://doi.org/10.15587/1729-4061.2025.331285*

1. Introduction

Modern information technologies are being actively introduced into agriculture, thereby improving the efficiency of crop monitoring and protection. Artificial intelligence plays an important role in this process, allowing for automated data analysis and informed decision-making. The main idea of artificial intelligence in the agricultural sector [1, 2] is its adaptability, accuracy, and optimal use of resources.

Field-scale pest monitoring [3, 4] is crucial for assessing insect infestation in agricultural settings. The development of machine learning technologies and embedded systems allows for the automation of this process, thereby reducing dependence on manual labor and increasing diagnostic accuracy.

Various devices and systems for monitoring and protecting plants from pests are actively used in agriculture. One of the most common methods is the use of visual control systems, automated solutions based on machine vision and neural networks.

Work [5] describes automated systems with traps, LEDs, and sensors that collect and transmit insect data to servers for further analysis. The data is available online, stored in geo-referenced databases. The devices are equipped with software for image interpretation and identification of the captured insect pest. The number of captured pests is counted by a remote operator.

Recently, an efficient lightweight neural network model based on YOLO-ELWNet for object detection based on YOLOv3 was reported [6]. The architecture uses a cross-stage module with an efficient narrow block, which improves the ability to extract features and reduces computational costs. However, the device does not provide sufficient autonomy and is not adapted for field work.

An improved version of YOLOv8, the DM-YOLO model [7], was proposed, specially adapted for accurate detection of cucumber diseases. Under conditions of poor lighting and background noise, standard algorithms often face difficulties in detecting small and overlapping features. DM-YOLO solves these problems through three modules: MultiCat provides extraction of features of different sizes; ADC2f uses attention and depthwise separable convolution to focus on important areas of the image with minimal load; C2fe implements contextual analysis, highlighting significant fragments of the scene. Testing on a custom set and PlantDoc showed an increase in mAP50 by 1.2% and 3.2%, respectively, which confirms the improved accuracy and stability of the model under a variety of conditions. The main drawback of the solution is the lack of adaptation of the DM-YOLO model to work on low-power devices. Optimization for real time and ensuring efficient operation on devices with limited resources is indicated as a direction for future research.

The development of energy-efficient IoT devices for precision farming reported in [8] demonstrated the potential for using low-power embedded computing systems. Similar devices use a mini-computer and process static images, while the proposed solution is mobile and processes video streams in real time. In addition, existing systems require significant financial costs and are difficult to operate.

Monitoring crop conditions and early diagnostics of pests is a key task for sustainable agriculture worldwide. However, because of the small size and high mobility of insects, conventional monitoring methods (e.g., mechanical capture with an entomological net) are labor-intensive in large fields.

One of the common pests of grain crops in various countries of Europe and Asia is the striped grain flea beetle (*Phyllotreta vittula*). In the Republic of Kazakhstan, this species is one of the main pests of wheat crops, especially in the northern regions. There, it is regularly observed in the seedling phase, which requires the development of more effective and automated monitoring and diagnostic methods.

2. Literature review and problem statement

Our review of the literature reveals that the attention is focused on designing intelligent monitoring systems based on computer vision algorithms and neural network models.

In [9], an intelligent system for monitoring cruciferous crop pests based on images from sticky traps is proposed. To improve the accuracy, a modified Cascade R-CNN model is used. The authors achieved an accuracy of over 90% when processing images in the field. However, using the model requires high computing resources and there is no possibility of local processing.

Work [10] describes the DAMI-YOLOv8l model designed for monitoring insects using light traps. The introduction of multi-level detectors made it possible to increase the accuracy to 93% when detecting small objects. However, the solution is not adapted for energy-efficient operation on embedded devices.

The authors of [11] conducted a comparative analysis of various YOLO configurations for real time when detecting soybean pests. The best results were shown by the YOLOv5 model but, under conditions of limited energy consumption, the model's performance is significantly reduced.

Paper [12] reports the YOLOCSP-PEST model adapted for localization and classification of pests. Despite its high accuracy, the authors acknowledge that the model requires further optimization for low-power devices and autonomous systems.

Of particular interest is study [13], in which the modified YoloV8x model as part of the Smart Trap IoT system achieved accuracy of up to 94% in detecting small pests such as fruit flies and caterpillars. However, there are still limitations: high computational load, the need for a stable Internet connection, and limited autonomy of the systems.

Research based on the Transformer architecture. The authors of [14] described the Pest-PVT model, which uses the Pyramid Vision Transformer (PVT) for multi-class and dense pest detection in field settings. The model demonstrated robust accuracy on complex scenes with many overlapping objects. While it has the advantage of being able to extract features at scale, the implementation is resource-intensive and is not designed for embedded devices. This limits its applicability to mobile autonomous systems.

Work [15] focuses on improving the YOLO architecture for more accurate detection of agricultural insects. The Insect-YOLO model demonstrates high accuracy on a variety of pest image datasets. The authors implemented optimizations to enable real-time use of the model but did not provide any data on its energy efficiency or applicability to embedded systems. The model is still focused on server-side or cloud processing.

In [16], a hybrid Vision Transformer with multi-level attention was reported, which outperformed classical CNNs in the pest classification task.

In [17], an RS Transformer model was proposed, which provides detection with a limited amount of training data. However, the robustness of such solutions under changing weather conditions in agroecosystems remains poorly understood.

The authors of [18] implemented tomato disease classification using CNN on Raspberry Pi 4 using lightweight architectures. The work demonstrated the possibility of local image processing with reasonable accuracy without an internet connection. The main feature is the demonstration of the applicability of CNN models on a low-cost embedded platform; however, the implementation is limited to static images and does not provide for real-time streaming video processing.

In [19], an improved model for millet leaf disease classification is proposed using multimodal data collection and precision-aware CNN. The model is tested on Raspberry Pi, demonstrating high accuracy with limited computing resources. The work emphasizes the importance of combining multiple types of sensory data (video and images) but focuses on foliar diseases rather than insects and does not contain a real-time automatic detection component.

Study [20] deals with CNN-based classification of root crop leaves implemented on Raspberry Pi. The main contribution of this study is to confirm the feasibility of CNN operation in real time on a low-cost platform. However, the task is limited to static classification and does not cover mobile environments or complex visual noise scenarios as in field shooting.

Work [21] describes a bird scaring system using OpenCV on Raspberry Pi to process the video stream. Although the task is different (birds instead of pests), the principles of real-time processing and offline operation are similar to the problem under consideration. The system showed good responsiveness and accuracy in detecting objects under dynamic conditions.

Paper [22] describes a compact YOLO_MRC model applicable for real-time counting of *Tephritidae* pests. The TP-Transfiner model presented in [23] provides highly accurate pest segmentation in densely distributed objects. Despite these achievements, the problem of designing a mobile and energy-efficient system with the ability to locally analyze images remains unsolved.

A separate research area relates to forecasting tasks. Work [24] considers methods for forecasting the number of pests of grain crops using MLP, MT-ANN, LSTM, Transformer, and SVR models. It is shown that the Transformer model provides the highest forecast accuracy, but the integration of such models with real-time visual monitoring systems has not yet been implemented.

Despite the progress achieved, the task of designing an energy-efficient and mobile system for monitoring small pests remains unsolved. An important requirement is the ability to locally process data without connecting to external infrastructure. The solution to this problem involves designing a device based on a miniature Raspberry Pi computer. Optimized neural network models are proposed for analyzing video streams in real time.

3. The aim and objectives of the study

The aim of our study is to design an autonomous device based on Raspberry Pi for automatic recognition and counting of the striped flea beetle in the field using machine learning algorithms.

To achieve the goal, the following tasks were set:

– to design the architecture and operating principle for the device to automatically detect pests;

– to select suitable computing modules, cameras, power supplies, and other elements to enable mobility and energy efficiency;

– to test the device under real conditions to assess the accuracy of recognition, processing speed, and resistance to external factors.

4. The study materials and methods

The object of our study is an autonomous system for monitoring agricultural pests based on Raspberry Pi and machine learning algorithms, focused on recognizing and counting the striped flea beetle in the field.

The main hypothesis of the study assumes that the integration of Raspberry Pi in combination with the OpenCV library and the YOLO model would make it possible to design an autonomous and accurate system for automatic recognition and counting of pests in real time. It is assumed that such a system could operate at minimal costs and high efficiency. This would enable timely adoption of agrotechnical measures and make it possible to reduce crop losses due to early detection of pests in crop areas.

The following assumptions and simplifications have been adopted. Monitoring is carried out in stable weather and good lighting. The device's camera is placed at a fixed height for stable shooting quality. Pest recognition is carried out using a limited data set, mainly including images of the striped flea beetle. All calculations are performed locally on the Raspberry Pi without connecting to cloud services. It does not take into account factors such as dust, moving leaves, and other insects that may affect the accuracy of the system.

Materials used in the device:

– Raspberry Pi 4.0 is a miniature computer running Raspbian OS, with the OpenCV library and the YOLO model installed for processing video streams and recognizing objects;

– Raspberry Pi Camera HQ is a high-resolution camera designed to obtain clear images of pests;

– SD card is a storage medium with an operating system and software;

– USB flash drive is for saving processed data and recognition results;

– power source is a battery or power bank, enabling autonomous operation of the device in the field;

– fan is for cooling the processor during long-term operation under high load conditions;

– case is for protecting all components from environmental influences.

Research methods.

Computer vision algorithms are implemented using the OpenCV library, including video stream capture, image transformation, contour search, and object selection functions.

The YOLO (You Only Look Once) model is used to recognize objects (pests) in a video stream in real time. The model was trained on a specialized dataset of striped bread flea beetle images.

Image processing includes the following stages: initialization of the video stream from the camera, division of the video stream into frames, binarization of images, extraction of object contours, noise filtering and application of the trained neural network model.

Software implementation was carried out in the Python programming language, using the cv2 (OpenCV) and numpy libraries. The algorithm implements the stages of video stream capture, image pre-processing, and feeding the model to recognize objects.

The choice of approaches is based on well-known methods of computer vision and object recognition used in monitoring biological and agricultural systems. For automatic monitoring, the YOLO model is used as one of the effective algorithms that enables real-time object detection. The OpenCV library is used for image pre-processing, including binarization and contour extraction, which corresponds to generally accepted methods of visual analysis.

5. Results of designing and testing a device for automatic pest recognition

5. 1. Designing the architecture and operating principle of the device

As part of the study, a device for automatic recognition of the striped grain flea beetle in grain crops was designed. The device is an autonomous system based on a Raspberry Pi 4.0 single-board computer with the integration of machine learning algorithms and computer vision libraries.

All training parameters are given in Table 1. The model was trained using the YOLOv8x-seg configuration on its own dataset of pest images. Further assessment of the model's accuracy was carried out on the test set.

Table 1

YOLOv8x-seg model training parameters

Parameter	Value	Note
Training dataset size	1200 images	Annotated pest images
Test dataset size	400 images	Used for model validation
Number of epochs	50	Full training cycle
Image size	640 × 640	Input data size
Batch size	16	Number of images per training iteration
Number of threads	4	Parallel data loading
Used optimizer	Adam	With default parameters
Loss function	Composite (bbox, obj, cls)	Standard YOLOv8 function
Experiment ID	flea_beetle_model	For tracking the training

The model was trained using a custom dataset containing 1200 annotated images of the striped flea beetle (*Phyllotreta vittula*), of which 400 images were set aside for testing. The model was trained over 50 epochs using an image resolution of 640×640 pixels and a batch size of 16. The standard configuration of the YOLOv8x segmentation model was used as the loss function. The Adam algorithm with default parameters was used for optimization, and training was performed using four data processing threads (workers = 4). The experiment ID in the model configuration is flea_beetle_model.

Pests are counted automatically for each frame, and the data is saved to a USB drive for subsequent analysis.

The visualization of the device architecture and the logic of its operation is shown in the form of diagrams (Fig. 1–3) describing the processing sequence and interaction of the system components.

This technological advance involved taking into account the requirements for mobility, low energy consumption, and the possibility of using under field conditions without connecting to external infrastructure. The diagram of the device for recognizing striped grain fleas is shown in Fig. 1.

The scheme is a set of tools with the inclusion of a system for automatic recognition of objects in a video stream where the striped grain flea beetle is detected in the external environment. The main elements of the device are the following components:

1) a power source that provides power to all components of the system;

2) a control system (mechanism), a centralized control module that coordinates the operation of all components of the system (processor, SD card, software using the OpenCV library with the YOLO model;

3) a camera that captures images of the external environment and transmits images to the control system for further processing.

The mechanism of operation of the device is as follows (Fig. 1):

1) capturing an object in a video stream (the camera records a video of the external environment with pests and transmits the video stream to the control system);

2) processing the object for recognition and detection of striped grain fleas (the video stream enters the processor and is transmitted to the Open CV program with support for the YOLO model, where objects are recognized);

3) recognition of objects in the video stream, when an object (pest) is detected in the frames, it recognizes and identifies the striped bread flea, then counts it, and saves the number of objects of one video stream to the USB card. This diagram shows the process of interaction of all components of the device with automatic recognition of objects in the video stream.

The machine learning algorithm for recognizing striped grain fleas in a video stream and their processing (Fig. 2) begins with the initialization of the system, including Raspberry Pi and all connected components: camera, data storage devices. Then the camera is activated to capture the video stream. The resulting video stream is divided into frames from which objects (striped grain fleas) are selected. Contours are found on the binarized image, small objects are filtered and transform neural network algorithms are used to analyze and identify pests.

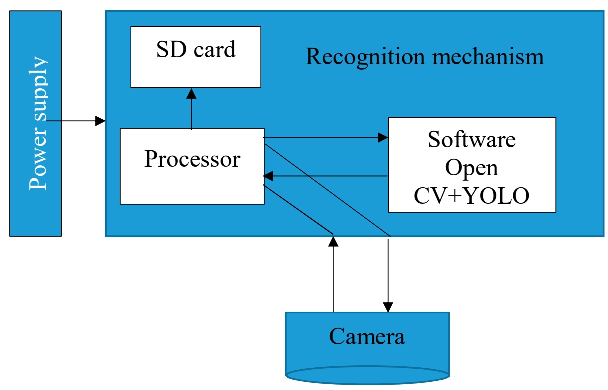

Fig. 1. Diagram of the device for recognizing grain striped flea beetles in grain crops

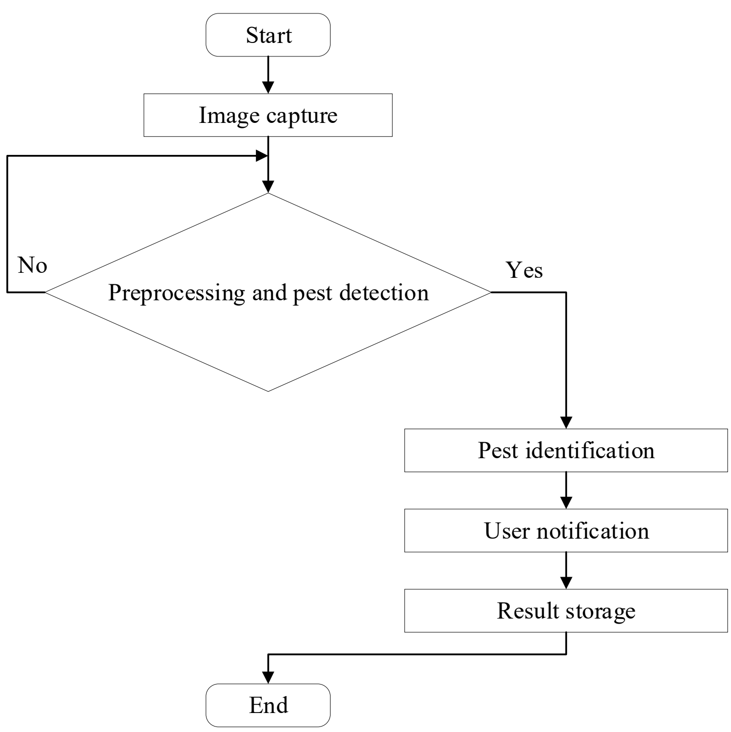

Fig. 2. Machine learning algorithm flow chart for recognition and processing

The Raspberry Pi-based device for recognizing grain striped fleas with machine learning algorithms will be widely used in agriculture. The device is also suitable for other areas related to pest control.

The device provides the ability to automatically and effectively recognize, track, and count grain striped fleas. Based on the data acquired, it becomes possible to predict the reproduction of the pest and take timely measures to reduce their numbers, which helps protect the crop.

The proposed designed device uses a miniature Raspberry Pi computer and a Raspberry Pi Camera HQ camera, which ensures the compactness and versatility of the design. In addition, the YOLO model and the OpenCV library are integrated into the system, allowing for real-time image analysis with high accuracy. Thanks to this, the device is able to operate autonomously in the field without connecting to external infrastructure, which makes it especially useful for agricultural applications. These features can significantly increase the accuracy, productivity, and cost-effectiveness of the system for recognizing pests in grain crops.

A prototype of a device for recognizing striped bread fleas has been designed, based on a miniature Raspberry Pi 4.0 computer and a Raspberry Pi Camera HQ (Fig. 3). The device is used to determine and count their number, as well as predict their increase based on machine learning algorithms. The recognition mechanism is implemented with the following code, which is written in the Python programming language using the Open CV program with YOLO support for capturing and processing the video stream from the camera and is launched on the SD card.

The code is designed to recognize striped bread fleas in real time:

1. Import libraries:

– cv2: The main OpenCV program with support for the YOLO model for computer vision.

– numpy: A library for working with arrays and numerical operations.

2. Initialize the camera:

```
cap = cv2.VideoCapture(0)
```

Opens the video stream from the camera (0 indicates the first available camera).

3. Function for recognizing striped bread fleas:

```
def detect_striped_fleas(frame):
    gray = cv2.cvtColor(frame,
    cv2.COLOR_BGR2GRAY)
    _, thresh = cv2.threshold(gray, 127, 255,
    cv2.THRESH_BINARY)
    contours, _=cv2.findContours(thresh,
    cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)
    for contour in contours:
        if cv2.contourArea(contour) < 100:
            continue
        x, y, w, h = cv2.boundingRect(contour)
        cv2.rectangle(frame, (x, y), (x + w, y + h),
        (0, 255, 0), 2)
    return frame
```

Thresholding: applies a binary thresholding transform to extract objects.

Finding edges: searches for edges in the thresholder image.

Drawing rectangles: draws rectangles around the found edges of the object.

4. Main video stream processing loop:

```
while True:
    ret, frame = cap.read()
    if not ret:
        break
    frame = detect_striped_fleas(frame)
    cv2.imshow('Striped Fleas Detection', frame)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break
```

Read Frame: captures frames from the camera in real time.

Process Frame: applies the detect_striped_fleas function to process each frame.

Display Frame: shows the processed frames in a window named 'Striped Fleas Detection'.

Check for 'q' keypress: breaks the loop and exits when the 'q' keypress is pressed.

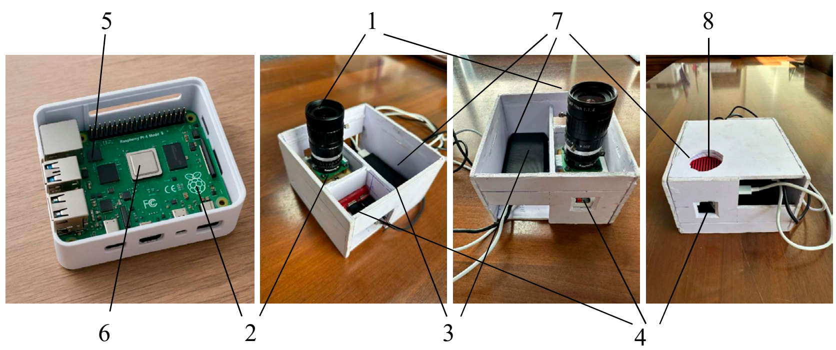

Fig. 3. Device for recognizing striped grain fleas in grain crops: 1 — camera; 2 — miniature computer Raspberry Pi 4.0; 3 — power supply (battery or power bank); 4 — SD card and USB flash drive; 5 — signal processing unit; 6 — program control unit; 7 — case; 8 — fan (for cooling the processor)

Fig. 3 shows a device for recognizing striped grain flea beetles in grain crops. The camera (1) captures a video stream, which is sent to a miniature Raspberry Pi 4.0 computer (2), which processes the data. The system is powered by a battery or an external accumulator (3). An SD card and a USB flash drive (4) are used to store data. The signal is processed in the corresponding module (5), and the program is controlled via the control module (6). All components are placed inside the case (7), supplemented by a fan (8) to cool the processor and maintain stable operation of the device in the field.

5. 2. Selection of hardware and software components

The designed device includes the following main elements:

– Raspberry Pi 4.0 mini-computer with Raspbian OS and pre-installed OpenCV software and YOLO model;

– high-quality Raspberry Pi Camera HQ camera, providing detailed images for accurate object recognition;

– autonomous power source (battery or power bank);

– SD card for storing data and program code.

The software part includes machine learning algorithms trained to recognize the striped bread flea beetle in the video stream. The system performs local processing, object selection, and counting in real time; the basic characteristics of the system are given in Table 2.

Table 2

Raspberry Pi-based system specifications

Parameter	Value
Platform	Raspberry Pi 4B (8GB RAM)
Camera	Raspberry Pi HQ Camera (12.3 MP, Sony IMX477, connecting via CSI)
Resolution	HQ (up to 4056 × 3040)
Recognition algorithm	YOLO (v4 or v5 using OpenCV and Python)
Frame rate	10–20 FPS
Recognition accuracy	80–90%
Number of images in the dataset	3000
Data transfer method	Wi-Fi (embedded Wi-Fi 802.11ac)
Camera focusing	Manual
Camera aperture	Manual
Battery life	2–4 hours (when powered by Power Bank ~20000 mAh)
Operating system	Legacy OS (Debian Bookworm x64, 64-bit)
Libraries and frameworks	OpenCV, PyTorch/ONNX (for YOLO), NumPy
Development environment	Python 3.9+, Jupyter/VSCode/Thonny

The characteristics given in Table 2 confirm that the designed system has all the necessary parameters for autonomous operation in the field. The use of an energy-efficient Raspberry Pi 4B mini-computer in combination with a high-quality camera (12.3 MP, Sony IMX477, connection via CSI) and YOLO algorithms makes it possible to achieve high accuracy of pest recognition (up to 90%) at a processing frequency of 10–20 frames per second. The compactness of the device, support for wireless data transmission, and battery life of up to 4 hours make it suitable for mobile use. The power consumption of the system was measured using a USB tester: under an idle mode (without active video stream processing), the consumption was about 0.6 A, which at 5 V corresponds to 3 W; under full load, with an active camera and a running YOLO model, the consumption increased to 0.9 A (≈ 4.5–5 W). Thus, the system is optimally balanced in terms of accuracy, energy efficiency, and functionality for intelligent monitoring tasks.

5. 3. Device testing and parameter analysis

During testing, the basic operating parameters of the device were assessed:

1. Recognition accuracy is determined by the model architecture and the quality of the training dataset. Using neural networks such as YOLO makes it possible to achieve high detection accuracy.
2. The speed of video stream processing depends on the performance of the Raspberry Pi and the optimization of the software code. Using OpenCV and hardware acceleration enables stable operation in real time.
3. The detection range is determined by the characteristics of the camera, such as resolution and viewing angle, as well as the illumination of the scene.
4. Power consumption is controlled by the choice of components and software optimization. The device can operate for a long time from an external battery.
5. Reliability and resistance to external factors are achieved due to the sealed case, high-quality materials, and a minimum number of moving parts.

To demonstrate the system's operation in real-time, a video stream processing structure was implemented, covering all stages from image capture to result display. Table 3 gives the step-by-step architecture of video stream processing. The sequence of components starts with image capture by the HQ Camera and ends with the result being displayed on the screen. The video stream is processed using the OpenCV library and transmitted via a Flask server running on a Raspberry Pi 4, which also serves as a Wi-Fi access point. Particular attention is paid to step 12 – the real-time video processing module (AI Module), where object detection is performed using the YOLO algorithm. This confirms that all analytics are performed locally, without the need to connect to cloud services, which is especially important for autonomous operation of the device in the field.

Table 3

Components and stages of video stream processing

No.	Component/stage	Description
1	HQ Camera	High quality camera (IMX477), connected via CSI
2	RPi4 Legacy System	Raspberry Pi 4 with Legacy OS (Bookworm x64)
3	PiCamera2	Library for working with CSI camera
4	cv2.VideoCapture(picamera)	OpenCV captures video, only 640×480 available
5	Flask video <div>	Video stream sent to HTML page via Flask
6	RPi4 Hotspot	RPi4 works as a Wi-Fi access point
7	Wi-Fi	Streaming to another device
8	Receiver Program	Client program on another device
9	Request library	Getting an HTML page using requests
10	HTML parsing <div>	Extracting a video stream from an HTML container
11	cv2.VideoCapture (parsed)	OpenCV captures the video from the received stream again
12	AI Module	Real-time video processing (e.g. YOLO)
13	cv2.imshow()	Displaying the result on the screen

To confirm the system's operability, an example of video frames obtained during field testing of the device is provided. The image shows the process of recognizing the striped grain flea beetle using the trained YOLO model (Fig. 4).

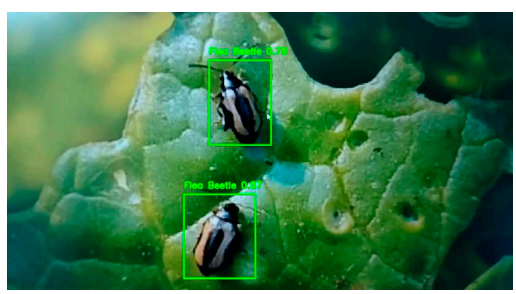

Fig. 4. The result of pest recognition by the YOLO model on a video frame (marked with green rectangles with the probability of recognition)

To evaluate the accuracy and efficiency of model training, standard metrics were used: accuracy (Precision), recall (Recall), and average accuracy (mAP). Fig. 5, 6 show the corresponding plots demonstrating the growth of the model quality as the number of training epochs increases.

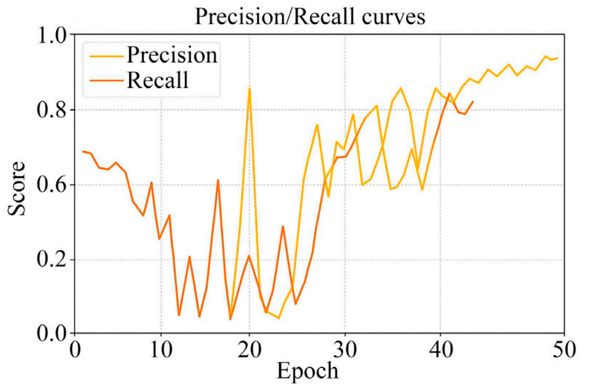

Fig. 5. Plot of Precision and Recall of the model during training

The plot shows the change in the values of the precision and recall metrics as the number of epochs increases. It is clear that after the 25th epoch, both metrics began to grow rapidly, reaching values above 0.9 closer to the 50th epoch. This indicates stable training of the model and its ability to effectively detect pests in images.

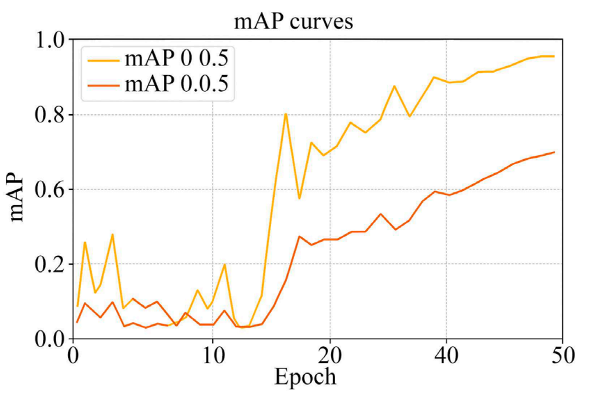

Fig. 6. Plot of changes in the mAP@0.5 and mAP@0.5:0.95 metrics during training process

The plot shows the dynamics of the growth of the mean Average Precision values at different IoU thresholds over 50 training epochs. It is evident that both metrics begin to grow rapidly after the 22nd epoch, reaching values above 0.9 for mAP@0.5 and about 0.7 for mAP@0.5:0.95 by the end of training, which indicates the high quality of the model in detecting objects of varying complexity.

To assess the quality of the trained model, quantitative metrics characterizing the accuracy of predictions and the degree of coincidence of predictions with actual values were used. Fig. 7 shows the values of the mean square error (MSE) and the determination coefficient (R^2), reflecting the accuracy of regression forecasting. Fig. 8 shows the error matrix demonstrating the distribution of correct and erroneous classifications of objects in the process of image recognition, which allows for a detailed analysis of the model's behavior for each class.

The plot shows the values of two key metrics: the mean square error (MSE) and the coefficient of determination (R^2), which characterize the accuracy of the regression model. MSE equal to 0.12 indicates a low level of errors in forecasting, and the R^2 value equal to 0.82 indicates a high degree of variance explained by the target variable by the model. The lower the MSE value and the higher R^2, the more accurate the model's predictions are, which in this case confirms its high efficiency.

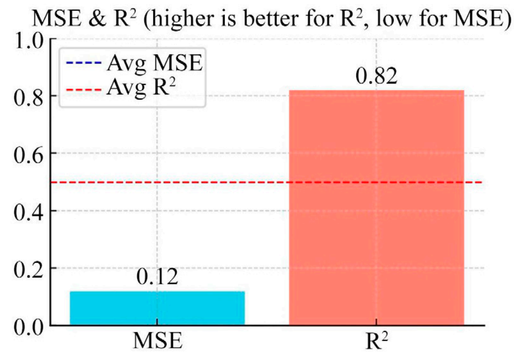

Fig. 7. Model quality assessment based on MSE and R^2 metrics

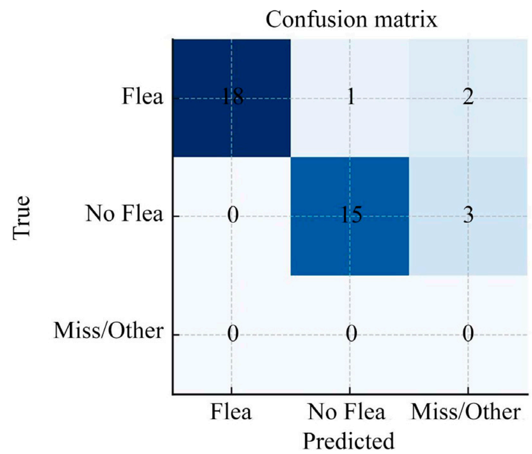

Fig. 8. Confusion matrix for object classification model

In Fig. 8, the error matrix shows how accurately the model recognized three categories: "Flea", "No Flea", and "Miss/Other". The model correctly categorized 18 images with pests and 15 without them. However, there were 3 errors: 1 case when the pest was classified as "No Flea" and 2 cases when it was classified as "Miss/Other". The model also incorrectly categorized 3 images without pests as "Miss/Other". The complete absence of errors in the "Miss/Other" category (by row) indicates that all objects not classified as "Flea" and "No Flea" were either not present in the test or did not cause false classifications. The matrix demonstrates the high accuracy of the model with a minimum number of false positives and false negatives.

To confirm the effectiveness of the proposed solution, Table 4 gives a comparison of characteristics with similar developments described in the Scopus database.

The data given in Table 4 allow us to compare the characteristics of existing solutions with the designed device. Unlike its analogs, the device based on Raspberry Pi 4.0 is focused on one specific pest – the striped grain flea beetle, which ensures high accuracy due to the adapted model architecture. The system operates offline, does not depend on a stable Internet connection, unlike solutions [25–28], and does not require expensive equipment, as is the case with UAVs [3]. At the same time, its compactness, energy efficiency, and the ability to locally process data make it suitable for use in the field. The device is a promising solution for autonomous pest monitoring in the agricultural environment.

Table 4

Comparative analysis of existing automated pest monitoring solutions based on Raspberry Pi and other technologies

Study	Technology	Advantages	Disadvantages
[25]	Raspberry Pi with YOLO algorithm to detect insects on leaves	– automatic detection of pests; – real-time notifications to the Android device; – high accuracy (> 90%)	– limited area of application (leaves only); – dependence on a stable Wi-Fi connection
[26]	Raspberry Pi with YOLOv4-tiny for monitoring insects in traps	– collection of data on the environment and the number of pests; – resistance to external interference	– slow processing speed of full YOLO on Raspberry Pi; – need for real-time optimization
[27]	Improved YOLOv10n model on Raspberry Pi 4B for pest detection on sticky traps	– reduction in the number of model parameters; – balancing the focus on local and global traits; – high detection accuracy	– limited scope of application (sticky traps); – further optimization for field conditions is required
[28]	Raspberry Pi with YOLOv10m for codling moth monitoring	– integration with the Ubidots platform for remote monitoring; – high detection accuracy (89%)	– focus on one type of pest; – dependence on a stable internet connection
[3]	Unmanned aerial vehicles (UAVs) with cameras and machine learning algorithms	– wide coverage of the territory; – high resolution of images; – the ability to monitor hard-to-reach areas	– high cost of equipment; – dependence on weather conditions; – limited flight time
Designed system	Raspberry Pi 4.0 with YOLO model for bread striped flea recognition	– autonomy and mobility; – low cost; – high recognition accuracy; – ease of use; – real-time monitoring	– limited processing power of the Raspberry Pi; – dependence on the quality of lighting; – lack of a GPS module for georeferencing data

The software based on machine learning algorithms for recognizing striped grain flea beetles includes computer vision algorithms and neural networks trained for accurate recognition in a video stream. The algorithms detect objects (pests) in real time, highlighting areas with pests, and counting their number. Advantages of the designed device (according to testing results):

1. High recognition accuracy has been confirmed during testing – the model provides up to 90% recognition accuracy of the striped grain flea beetle due to the use of optimized neural networks and YOLO model settings.

2. High data processing speed – during testing, a processing speed of up to 20 frames per second was recorded, which ensures real-time monitoring even on a platform with limited resources (Raspberry Pi).

3. A wide range of pest detection has been confirmed due to the use of a high-resolution camera (12.3 MP), which allows one to record pests at various distances and under variable lighting conditions.

4. Low power consumption – measurements have shown that the device consumes up to 5 W under an active mode, which makes it possible to use it autonomously for several hours from an external battery.

5. The reliability of the structure is ensured by a protected case and cooling system, which is confirmed by the stable operation of the device in field tests under variable weather conditions.

6. Mobility and ease of use have been tested in practice – compact dimensions and light weight make the device easy to transport and convenient for use in various places.

6. Discussion of the efficiency and limitations of the designed device

Our results confirm the hypothesis that the integration of Raspberry Pi with OpenCV and YOLO provides an autonomous and energy-efficient device for pest recognition. The plots (Fig. 5, 6) show that the model achieves high accuracy (Precision > 0.9) and mean accuracy (mAP > 0.9 at IoU = 0.5) after the 40th epoch. This is due to the use of the YOLOv8x architecture, optimized for working with limited computing resources and detecting small objects.

A significant difference of the proposed solution is local data processing on a Raspberry Pi single-board computer, without using cloud infrastructure. Unlike existing approaches [25–28] (Table 3), which require an Internet connection, or more expensive UAV systems [3], our device provides autonomy, mobility, and ease of operation. Its effectiveness is confirmed by experimental data, including regression accuracy metrics ($MSE = 0.12$, $R^2 = 0.82$, Fig. 7) and the confusion matrix (Fig. 8), demonstrating the minimum number of false negative and false positive classifications.

The study has several limitations. First, the system assumes stable lighting conditions and immobility of leaves, which may be violated under real weather conditions. Second, the model is trained on a limited data set, and its accuracy may decrease when the appearance of the pest changes or in the presence of similar insects. In addition, the device is not yet integrated with automatic agronomic response systems, which limits its use as part of a single precision farming ecosystem.

A disadvantage of our work is the lack of comparison of different neural network architectures (e.g., YOLOv8, EfficientDet) and their impact on accuracy under specific conditions. Also, energy efficiency was not assessed compared to alternative platforms such as Jetson Nano. Resolving these issues requires a wider experimental and computational resource, as well as an expansion of the set of tested models.

The study may evolve in several directions. First, it is possible to expand the functionality of the device due to multispectral shooting, which could improve the distinction between pests and the background. Secondly, it is necessary to build a scalable image database for additional training and validation of the model in different regions. Potential difficulties in this case are related to the need for manual labeling of a large number of images, setting the balance between accuracy and speed, and ensuring stable operation of the model with seasonal changes in the appearance of plants and pests.

7. Conclusions

1. The designed architecture of a mobile device for automatic pest recognition takes into account field conditions, including limitations in power supply, computing resources, and illumination. Using a Raspberry Pi mini-computer together with the YOLOv8x-seg model and the OpenCV library has made it possible to achieve autonomous processing of the video stream and confident detection of small objects (*Phyllotreta vittula*) in real time. The trained model on a custom dataset of 1200 images showed stable operation with a minimum number of false positives, maintaining performance at ≥15 frames per second on a device with limited resources. The main difference of the proposed solution is its complete energy independence, compactness, and the ability to function under an autonomous mode without communication with the cloud. This is explained by the use of an optimized model architecture and adaptation of the algorithm to the capabilities of the embedded platform.

2. A well-founded selection and integration of hardware and software components was carried out, including Raspberry Pi 4.0, Raspberry Pi Camera HQ, an autonomous power supply, and a YOLO model with OpenCV. Power limitations were taken into account when we designed it: the system operates with a power consumption of up to 5 W. The compactness of the structure and the ability to process data directly on the device were achieved through software optimization of the neural network model and minimization of dependence on external infrastructure. That has made it possible to devise a fully autonomous solution suitable for field monitoring conditions without a network connection.

3. During field tests, the device demonstrated the accuracy of recognition of the striped grain flea beetle in the range of 80–90% and the processing frequency of 10–20 frames per second. The quality of the model has been confirmed by plots for the Precision and mAP metrics, as well as the MSE and R^2 metrics. A special feature of the solution is the ability to function effectively under natural light conditions and autonomously. These parameters confirm the practical applicability of the system in agricultural production. Our results are attributed to the adaptation of the model to real field images and the optimization of the computing process, taking into account the limitations of the built-in platform.

Conflicts of interest

The authors declare that they have no conflict of interest in relation to this research, whether financial, personal, authorship or otherwise, that could affect the research, and its results reported in this paper.

Financing

This research was funded by the Ministry of Science and Higher Education of the Republic of Kazakhstan, IRN "AP 19675312 Analytical system for forecasting the dynamics of the number of pests of grain crops in Kazakhstan based on a neural network model".

Data availability

The data that support the findings of this research will be made available by the authors upon reasonable request.

Use of artificial intelligence

The authors have used artificial intelligence technologies within acceptable limits to provide their own verified data, which is described in the research methodology section.

References

1. Aijaz, N., Lan, H., Raza, T., Yaqub, M., Iqbal, R., Pathan, M. S. (2025). Artificial intelligence in agriculture: Advancing crop productivity and sustainability. Journal of Agriculture and Food Research, 20, 101762. https://doi.org/10.1016/j.jafr.2025.101762
2. Espinel, R., Herrera-Franco, G., Rivadeneira García, J. L., Escandón-Panchana, P. (2024). Artificial Intelligence in Agricultural Mapping: A Review. Agriculture, 14 (7), 1071. https://doi.org/10.3390/agriculture14071071
3. Ahmed, S., Marwat, S. N. K., Brahim, G. B., Khan, W. U., Khan, S., Al-Fuqaha, A., Koziel, S. (2024). IoT based intelligent pest management system for precision agriculture. Scientific Reports, 14 (1). https://doi.org/10.1038/s41598-024-83012-3
4. Venkateswara, S. M., Padmanabhan, J. (2025). Deep learning based agricultural pest monitoring and classification. Scientific Reports, 15 (1). https://doi.org/10.1038/s41598-025-92659-5
5. Sciarretta, A., Calabrese, P. (2019). Development of Automated Devices for the Monitoring of Insect Pests. Current Agriculture Research Journal, 7 (1), 19–25. https://doi.org/10.12944/carj.7.1.03
6. Song, B., Chen, J., Liu, W., Fang, J., Xue, Y., Liu, X. (2025). YOLO-ELWNet: A lightweight object detection network. Neurocomputing, 636, 129904. https://doi.org/10.1016/j.neucom.2025.129904
7. Ding, J.-Y., Zou, C.-M., Jeon, W.-S., Rhee, S.-Y. (2024). An Improved YOLO Detection Approach for Pinpointing Cucumber Diseases and Pests. Computers, Materials & Continua, 81 (3), 3989–4014. https://doi.org/10.32604/cmc.2024.057473
8. Brunelli, D., Minakov, I., Passerone, R., Rossi, M. (2015). Smart monitoring for sustainable and energy-efficient buildings: A case study. 2015 IEEE Workshop on Environmental, Energy, and Structural Monitoring Systems (EESMS) Proceedings, 186–191. https://doi.org/10.1109/eesms.2015.7175875
9. Gao, Y., Yin, F., Hong, C., Chen, X., Deng, H., Liu, Y. et al. (2024). Intelligent field monitoring system for cruciferous vegetable pests using yellow sticky trap images and an improved Cascade R-CNN. Journal of Integrative Agriculture. https://doi.org/10.1016/j.jia.2024.06.017
10. Chen, X., Yang, X., Hu, H., Li, T., Zhou, Z., Li, W. (2025). DAMI-YOLOv8l: A multi-scale detection framework for light-trapping insect pest monitoring. Ecological Informatics, 86, 103067. https://doi.org/10.1016/j.ecoinf.2025.103067

11. Tetila, E. C., da Silveira, F. A. G., da Costa, A. B., Amorim, W. P., Astolfi, G., Pistori, H., Barbedo, J. G. A. (2024). YOLO performance analysis for real-time detection of soybean pests. Smart Agricultural Technology, 7, 100405. https://doi.org/10.1016/j.atech.2024.100405
12. Qayyum, H., Ali, F., Saleem, K., Ahmad, I., Iqbal, M. J. (2025). YOLOCSP-PEST for Crops Pest Localization and Classification. Computers, Materials & Continua, 82 (2), 2373–2388. https://doi.org/10.32604/cmc.2025.060745
13. Hakim, A., Srivastava, A. K., Hamza, A., Owais, M., Habib-ur-Rahman, M., Qadri, S. et al. (2025). Yolo-pest: an optimized YoloV8x for detection of small insect pests using smart traps. Scientific Reports, 15 (1). https://doi.org/10.1038/s41598-025-97825-3
14. Chen, H., Wen, C., Zhang, L., Ma, Z., Liu, T., Wang, G. et al. (2025). Pest-PVT: A model for multi-class and dense pest detection and counting in field-scale environments. Computers and Electronics in Agriculture, 230, 109864. https://doi.org/10.1016/j.compag.2024.109864
15. Wang, N., Fu, S., Rao, Q., Zhang, G., Ding, M. (2025). Insect-YOLO: A new method of crop insect detection. Computers and Electronics in Agriculture, 232, 110085. https://doi.org/10.1016/j.compag.2025.110085
16. Saranya, T., Deisy, C., Sridevi, S. (2024). Efficient agricultural pest classification using vision transformer with hybrid pooled multihead attention. Computers in Biology and Medicine, 177, 108584. https://doi.org/10.1016/j.compbiomed.2024.108584
17. Wu, T., Shi, L., Zhang, L., Wen, X., Lu, J., Li, Z. (2023). RS Transformer: A Two-Stage Region Proposal Using Swin Transformer for Few-Shot Pest Detection in Automated Agricultural Monitoring Systems. Applied Sciences, 13 (22), 12206. https://doi.org/10.3390/app132212206
18. Gonzalez-Huitron, V., León-Borges, J. A., Rodriguez-Mata, A. E., Amabilis-Sosa, L. E., Ramírez-Pereda, B., Rodriguez, H. (2021). Disease detection in tomato leaves via CNN with lightweight architectures implemented in Raspberry Pi 4. Computers and Electronics in Agriculture, 181, 105951. https://doi.org/10.1016/j.compag.2020.105951
19. Tiwari, S., Gehlot, A., Singh, R., Twala, B., Priyadarshi, N. (2025). Design of an improved model for finger millet leaf disease detection with raspberry Pi using multimodal data acquisition and precision-aware CNN. Results in Engineering, 25, 103969. https://doi.org/10.1016/j.rineng.2025.103969
20. Rakesh, M. D., Jeevankumar, M., Rudraswamy, S. B. (2025). Implementation of real time root crop leaf classification using CNN on raspberry-Pi microprocessor. Smart Agricultural Technology, 10, 100714. https://doi.org/10.1016/j.atech.2024.100714
21. Upendra, M., Anuradha, T., Prakash, L. (2025). Bird Repeller Using Opencv. Procedia Computer Science, 252, 975–984. https://doi.org/10.1016/j.procs.2025.01.058
22. Wei, M., Zhan, W. (2024). YOLO_MRC: A fast and lightweight model for real-time detection and individual counting of Tephritidae pests. Ecological Informatics, 79, 102445. https://doi.org/10.1016/j.ecoinf.2023.102445
23. Wu, R., He, F., Rong, Z., Liang, Z., Xu, W., Ni, F., Dong, W. (2024). TP-Transfiner: high-quality segmentation network for tea pest. Frontiers in Plant Science, 15. https://doi.org/10.3389/fpls.2024.1411689
24. Anarbekova, G., Ruiz, L. G. B., Akanova, A., Sharipova, S., Ospanova, N. (2024). Fine-Tuning Artificial Neural Networks to Predict Pest Numbers in Grain Crops: A Case Study in Kazakhstan. Machine Learning and Knowledge Extraction, 6 (2), 1154–1169. https://doi.org/10.3390/make6020054
25. Amnerkar, P. G., Bawane, K., Raut, M., Bargat, D., Damahe, B., Ande, K. A. (2023). An IoT-Based Solution for Insect Monitoring in Agriculture Using Raspberry Pi and YOLO. International Research Journal of Modernization in Engineering Technology and Science, 5 (4). https://doi.org/10.56726/irjmets36564
26. Le, A., Pham, D. A., Thanh, D., Vo, H. B. (2024). AlertTrap: A Study on Object Detection in Remote Insect Trap Monitoring System Using on the Edge Deep Learning Platform. Journal of Computational and Cognitive Engineering. https://doi.org/10.47852/bonviewjcce42023264
27. Huang, Y., Liu, Z., Zhao, H., Tang, C., Liu, B., Li, Z. et al. (2025). YOLO-YSTs: An Improved YOLOv10n-Based Method for Real-Time Field Pest Detection. Agronomy, 15 (3), 575. https://doi.org/10.3390/agronomy15030575
28. Zarboubi, M., Bellout, A., Chabaa, S., Dliou, A., Zeroual, A. (2024). IoT-Based Pest Detection in Agriculture Using Raspberry Pi and YOLOv10m for Precision Farming. ITM Web of Conferences, 69, 04017. https://doi.org/10.1051/itmconf/20246904017