A Novel Smart System with Jetson Nano for Remote Insect Monitoring

Thanh-Nghi Doan[1, 2], Thien-Hue Phan[3]
Faculty of Information Technology, An Giang University, An Giang, Vietnam[1]
Vietnam National University, Ho Chi Minh City, Vietnam[2]
Graduate Students, University of Information Technology, Vietnam National University, Ho Chi Minh City, Vietnam[3]

Abstract—**Insect monitoring is vital for agricultural management and environmental conservation, but traditional methods are labor-intensive and time-consuming. This paper introduces a novel smart system utilizing NVIDIA's Jetson Nano technology combined with object detection models for remote insect monitoring. The system automates the processes of detection, identification, and monitoring, thereby significantly improving the efficiency and accuracy of insect population assessments. The implementation of the YOLOv7 model on a dataset containing 10 insect species achieved a mAP@0.5 accuracy of 77.2%. This enables farmers to take timely and appropriate measures to prevent pests and diseases, reducing production costs and protecting the environment.**

Keywords—NVIDIA Jetson Nano; insect monitoring; YOLOv7

I. Introduction

Insects are crucial to ecological health and agricultural ecosystems, pollinating crops and managing pest populations. Traditional monitoring of these insects is labor-intensive and resource-heavy. However, recent technological advances offer a solution through automation and real-time data processing. Innovations in sensor technology, machine learning, and computer vision enable precise and continuous monitoring of insect populations. These automated systems enhance data collection efficiency and provide valuable insights into insect behavior, aiding researchers and farmers in making informed decisions. This leads to better crop protection and ecological balance, supporting sustainable agriculture and environmental conservation.

The Jetson Nano, with its powerful GPU capabilities and compact size, offers a promising platform for developing a remote insect monitoring system. The literature cited presents a comprehensive overview of research endeavors aimed at revolutionizing insect monitoring and detection through innovative technological solutions. The authors in study [1] delve into the realm of computer vision techniques tailored specifically for automated insect monitoring and detection, a domain ripe for the development of cutting-edge image processing algorithms on platforms like the Jetson Nano. Expanding on this foundation, article [2] meticulously scrutinizes deep learning methodologies designed for insect detection and classification. Such insights not only enrich our understanding but also pave the way for implementing on-device machine learning models seamlessly integrated with Jetson Nano's capabilities.

Moreover, the discourse in study [3] sheds light on the integration of wireless sensor networks in environmental monitoring applications, offering invaluable insights into the design and deployment of sensor nodes for remote insect monitoring, a critical aspect of effective surveillance. These insights are crucial for ensuring that the sensor nodes are not only strategically placed but also robust and reliable in various environmental conditions. Additionally, researchers in study [4] review energy-efficient communication protocols tailored for IoT applications, a knowledge pool essential for optimizing communication between Jetson Nano devices and remote servers, ensuring seamless data exchange. This optimization is pivotal for maintaining long-term operation and minimizing energy consumption, which is vital for remote monitoring systems that often rely on limited power sources. By leveraging these protocols, the efficiency and reliability of remote insect monitoring systems can be significantly enhanced, leading to more accurate and timely data collection and analysis.

Furthermore, the authors in study [5] elucidate various data fusion techniques essential for integrating information from diverse sensors in environmental monitoring systems, a pivotal step towards enhancing the accuracy and reliability of insect monitoring data. The challenges and opportunities associated with deploying IoT systems in remote environments are thoroughly explored by researchers in study [6], offering pragmatic insights crucial for implementing smart systems for remote insect monitoring. Moreover, the researchers in study [7] explore the myriad applications of the NVIDIA Jetson Nano in edge computing, providing inspiring examples and case studies that could catalyze the development of innovative smart systems for remote insect monitoring.

Deep learning's prominence is reaffirmed in study [8], where the authors explore its effectiveness in automated insect pest detection for precision agriculture using image-based data. In [9], a real-time insect detection and classification system using convolutional neural networks (CNNs) on image data is proposed, offering a pioneering and practical approach. The authors in study [10] provide a detailed overview of image-based insect identification techniques employing deep learning, enhancing our knowledge of advanced methodologies. Article [11] presents a sophisticated framework integrating image-based and sensor-based data for real-time insect pest monitoring in greenhouse crops, highlighting the synergy between different data modalities. In study [12], a fusion approach combining data from multiple sensors for improved insect pest detection in precision agriculture is proposed, emphasizing the value of diverse data sources for thorough analysis. Finally, the authors in study [13] underscore the real-time processing capabilities of Jetson Nano for deep learning-based insect detection, demonstrating its potential as a key device for future advancements in this field.

As a result, this paper proposes a novel smart system with Jetson Nano for remote insect monitoring that is low in cost, efficient, has a fast response time, and is simple to install and implement in practice using hardware devices with limited configuration. The total cost of our proposed system is detailed in Table I. The main contributions of the paper include:

- A novel system utilizing the NVIDIA Jetson Nano and object detection models for real-time detection and classification of pest insects. This system significantly enhances the efficiency and accuracy of insect population assessments.
- The implementation of the YOLOv7 model on a dataset of 10 insect species resulted in a mAP@0.5 accuracy of 77.2%. This demonstrates the system's capability to identify and distinguish between 10 common insect groups with high precision.
- The system is designed to be low-cost, efficient, and easy to install and implement using hardware devices with limited configuration, making it accessible for practical agricultural applications.
- Leverages deep learning methodologies, image processing algorithms, and wireless sensor networks to create an integrated solution for remote insect monitoring.

The rest of the article is arranged as follows. Section II describes the materials and methods used to describe overview of our system, general system design and setup, NVIDIA Jetson Nano Developer Kit, insect trap, insect detection model. The experimental results and discussion are reported in Section III. Section IV presents the conclusions, limitations, and recommendations for future research.

TABLE I. THE DETAIL COST OF OUR SYSTEM

Device	Price in USD
NVIDIA Jetson Nano	224.32
Insect traps	62.97
UV Lights Attract Insects	4.72
Sticky insect trap	1.57
YOLO test fee	20
128GB memory card	25.58
Total cost	**339.16**

II. MATERIALS AND METHODS

A. System Overview

In the initial stage, we collected and labeled image data of pest insects for training and evaluating the CNN model. Next, YOLO object detection models were trained on the insect dataset. We evaluated the model parameters based on the trained models. From the evaluation, the best model with the appropriate parameters is selected for object recognition on the Jetson Nano device. Then, the trained model is deployed on the Jetson Nano device. Finally, we implemented the real-time pest insect recognition system in the fields. Overview of our real-time insect detection system is illustrated in Fig. 1.

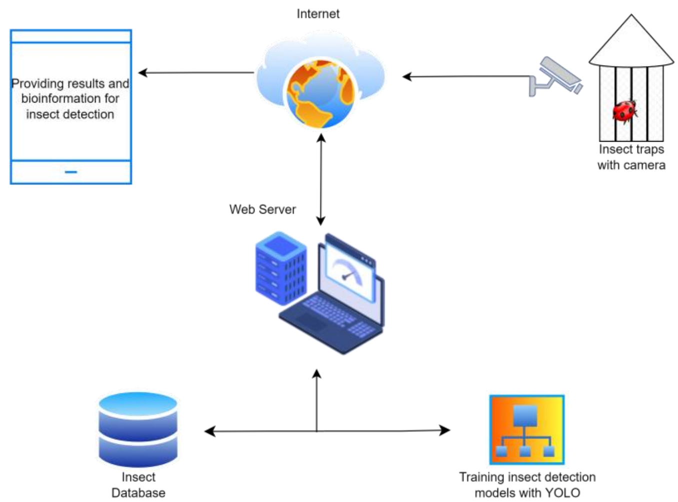

Fig. 1. Overview of our real-time insect detection system.

The Jetson Nano's MIPI CSI-2 camera serves as a monitoring system for object detection. Subsequently, the captured images of the objects are detected through OpenCV data processing and YOLO data classification on the Jetson Nano. The process is illustrated in Fig. 2.

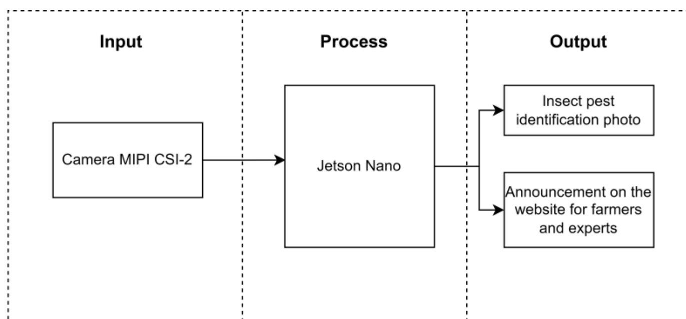

Fig. 2. General system design.

B. Equipment Setup

1) Jetson Nano developer kit: The Jetson Nano Developer Kit [15] is a compact computer developed by NVIDIA for use in artificial intelligence (AI) applications, particularly in the field of real-time image and video processing. It allows users to run multiple neural networks in parallel for image processing applications. It delivers the performance to run modern AI workloads in a small, energy-efficient (consuming as little as 5W), and cost-effective form factor. The NVIDIA Jetson Nano consists of nine basic components, as illustrated in Fig. 3.

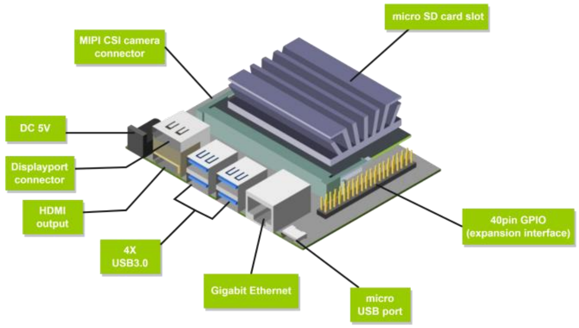

Fig. 3. NVIDIA Jetson Nano hardware overview.

The NVIDIA Jetson Nano Developer Kit is the smallest member of the Jetson product family, designed for portability and powered by a backup battery when mains power is unavailable. This makes it ideal for use outside of the office or on the go. The kit features a powerful GPU-supported system that includes a 64-bit quad-core ARM Cortex-A57 CPU, 4GB of RAM, and a video processor capable of 4K 30fps encoding and 4K 60fps decoding, as shown in Table II.

TABLE II. NVIDIA JETSON NANO DEVELOPER KIT B01 SPECIFICATIONS

Items	Technical Specifications
Model	NVIDIA Jetson Nano Developer Kit B01 (upgrade version with 2 cameras)
GPU	128-core Maxwell
CPU	Quad-core ARM A57 @1.43 GHz
Memory	4 GB 64-bit LPDDR4 25.6 GB/s
Model	NVIDIA Jetson Nano Developer Kit B01 (Upgraded version with dual cameras)
Storage	microSD
Video Encode	4K @ 30 | 4x 1080p @ 30 | 9x 720p @ 30
Video Decode	4K @ 60 | 2x 4K @ 30 | 8x 1080p @ 30 | 18x 720p @ 30
Mechanical	69.6 mm × 45 mm, 260-pin edge connector
Entire set	100mm × 80mm × 29mm
Camera	2x MIPI CSI-2 DPHY lanes
Connectivity	Gigabit Ethernet, M.2 Key E
Display	HDMI and display port
USB	4x USB 3.0, USB 2.0 Micro-B
Others	GPIO, I2C, I2S, SPI, UART

Additionally, it supports PCIe and USB 3.0 slots. The Jetson Nano delivers 472 GFLOPS for accelerated execution of modern AI algorithms. With a quad-core ARM 64-bit CPU, an integrated 128-core NVIDIA GPU, and 4GB of LPDDR4 memory, it can simultaneously run multiple neural networks and process high-resolution sensors.

Utilizing two cameras on NVIDIA Jetson Nano B01 offers several significant advantages. Firstly, it allows for image capture from two different angles, enhancing observational capabilities and covering a wider area. Secondly, with stereoscopic vision capabilities, the two cameras can create 3D images from different viewpoints, aiding in depth and distance determination, which is crucial for autonomous robots, object recognition, and navigation. Thirdly, the dual image sources enable the system to compare and eliminate errors or noise, increasing data accuracy and reliability. Fourthly, this setup optimizes performance and simplifies connections, removing the need for external adapters or USB ports. Fifthly, the Jetson Nano B01's design includes two CSI connectors, allowing for the simultaneous connection of multiple cameras, making it ideal for multi-channel applications. Lastly, NVIDIA provides robust software support for CSI cameras through Gstreamer and supporting libraries, making it easy to use commands like nvgstcapture to test and capture images from the cameras.

2) *Remote monitoring insect trap:* Weather is crucial to agricultural production, significantly impacting crops, livestock, and the environment, even with minor fluctuations. The outbreak and spread of pests and diseases are also highly dependent on weather conditions. Research has shown that temperature, humidity, rainfall, wind, and microclimate all influence the growth, reproduction, and population density of brown rice planthoppers. To address this, we propose a remote monitoring insect trap featuring an innovative model of integrated light traps that operate automatically based on sensor data, as shown in Fig. 5. These automatic light traps will continuously collect, analyze, store, and, if necessary, alert data, transmitting it to a network system. The newly trained model is eventually deployed to the Jetson Nano camera, as illustrated in Fig. 4.

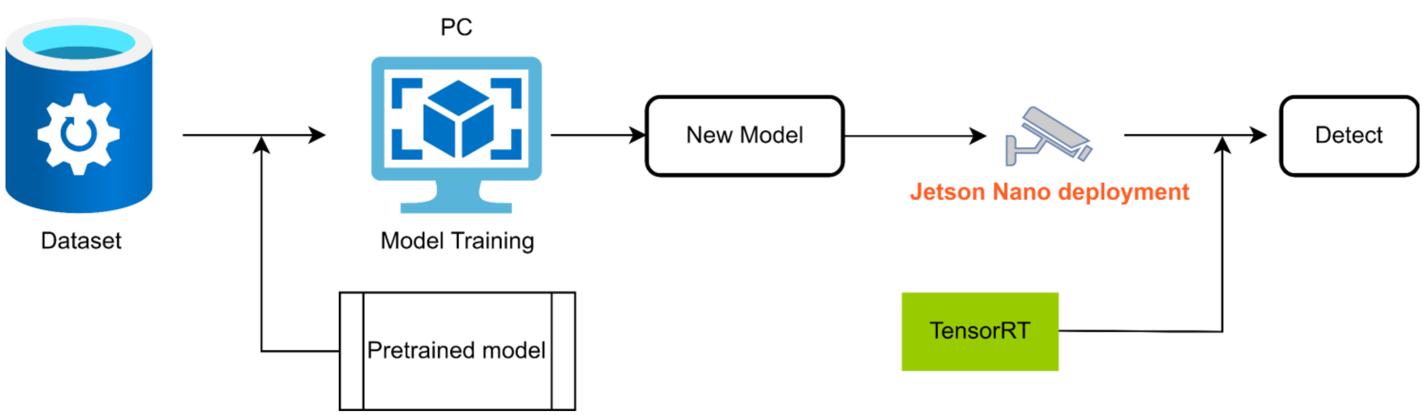

Fig. 4. Real-time insect identification system with Jetson Nano.

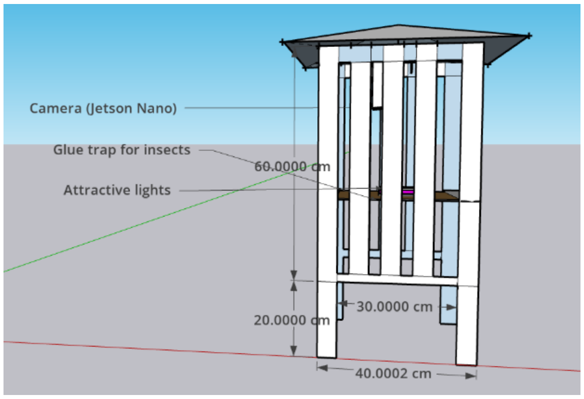

Fig. 5. Remote monitoring insect trap.

Based on the biological characteristics and behavior of certain harmful insects, insect traps can serve as an effective alternative to directly spraying pesticides onto plants, reducing the use of potentially harmful chemicals. By integrating multiple trapping methods, we can enhance overall effectiveness. Some possible methods to combine include:

- Attracting insects using sex pheromones.
- Attracting insects using bio-based traps (e.g., sweet sticky traps).
- Attracting insects using blue or yellow sticky traps.
- Attracting insects using light traps.

The study introduces a design for a light-induced insect trap with a modular design that enables straightforward assembly and disassembly of trap components. Detailed images illustrating the placement of devices within insect traps are depicted in Fig. 6. This design allows for easy relocation of the trap, operates effectively under various weather conditions, ensures high durability, and uses materials that are safe for both humans and the environment. Additionally, the modular nature of the trap makes it adaptable to different pest management needs and scalable for larger agricultural applications. This approach not only targets pest reduction but also promotes sustainable farming practices by minimizing the reliance on chemical pesticides, thereby protecting the ecosystem and promoting biodiversity.

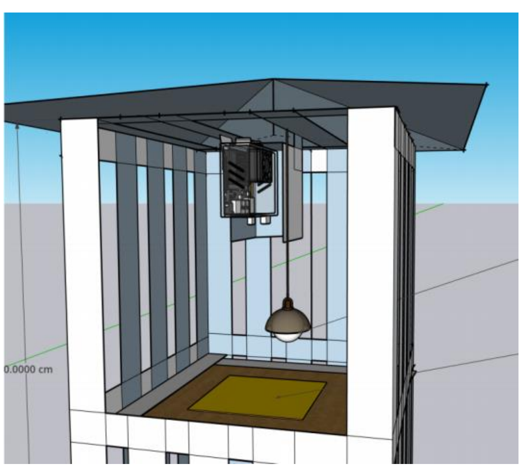

Fig. 6. Detailed images of device placement in insect traps.

3) Light attracts insects: Light is crucial for attracting insects to traps, with blue and ultraviolet (UV) light being particularly effective [21], [22]. Mosquitoes, flies, and moths are especially drawn to these wavelengths. Additionally, white light, which includes both blue and UV components, can also serve as an attractant. Using light to lure insects into traps is an effective, safe, and eco-friendly method for managing insect populations.

Examining the attractiveness of different light components enhances our understanding of their efficacy in insect attraction. Blue and UV light, with shorter wavelengths, are highly attractive to insects due to their eyes' sensitivity to these wavelengths. In contrast, red light has the lowest attraction capability, drawing only about 2% of insects in nature. Yellow light, with a slightly shorter wavelength and higher energy than red light, attracts approximately 4–5% of insects. Green light, being neutral and abundant in natural light, has average attraction capabilities, drawing around 7–8% of insects. The blue light spectrum, characterized by its short wavelength and high energy, is particularly enticing to insects, attracting roughly 20–23% of those present in nature. UV light, though not visible to the human eye, surpasses even blue light in energy and attractiveness, enticing approximately 40–50% of insects. Understanding these nuances in light spectra helps identify the most effective options for insect control.

As demonstrated by the list of light types and their respective insect attraction capabilities, UV lights exhibit exceptional ability to attract insects, drawing in approximately 40–50% of insect populations in nature. This remarkable effectiveness prompted our decision to use UV lights as the primary insect attractant in this research endeavor. The specifications of the UV insect attractant lamp are shown in Table III. The Conopery UV black light lamp, emitting purple light, operates on a convenient 220V power source, allowing for easy electrical plug-in. Its compact dimensions (5.2 cm by 17.5 cm) facilitate easy setup for insect trapping, as shown in Fig. 7.

TABLE III. THE SPECIFICATIONS OF THE UV INSECT ATTRACTANT LAMP

Product Name	**Conopery UV Black Light Lamp (Purple Light)**
Power Source	220V
Wavelength Range	300 ~ 400 nm
Peak Wavelength	365nm
Lifespan	8000 hours
Dimensions	5.2cm x 17.5cm
Type of Lamp	3U/36W Spiral

Fig. 7. UV lights attract insects

The choice of UV light in this study is also supported by its wide application in various ecological and agricultural settings. UV lights are known for their ability to attract a broad spectrum of insect species, making them highly versatile in different environments. The high attraction rates of UV lights not only enhance the efficiency of insect traps but also contribute to more accurate population assessments and monitoring in ecological studies.

Moreover, UV light traps have been shown to reduce the need for chemical insecticides, thus promoting environmental sustainability. By minimizing chemical usage, these traps help maintain ecological balance and reduce the risk of pesticide resistance among insect populations. This aligns with integrated pest management (IPM) strategies that emphasize sustainable and environmentally friendly pest control methods.

In practical applications, the Conopery UV black light lamp has been selected for its durability and ease of use in field conditions. Its design allows for seamless integration into various trapping setups, ensuring reliable operation even in remote or challenging environments. The lamp's specifications, including its wavelength emission, power requirements, and physical dimensions, have been carefully considered to maximize its effectiveness in attracting target insect species.

Overall, the deployment of UV light as an insect attractant in this study exemplifies the integration of scientific understanding and practical application. By leveraging the unique properties of UV light, this research aims to contribute valuable insights into insect behavior and improve pest management practices. The findings from this study could inform future developments in trapping technology and enhance the effectiveness of insect control strategies across diverse settings.

4) Insect sticky trap: This is a flat surface used to trap insects. One challenge is to maintain the shape of the insect when it is caught in the trap and when it dies, so that the camera can recognize it. The solution in this study is to use an insect sticky trap, as illustrated in Fig. 8.

Fig. 8. Insect sticky trap.

C. Insect Image Dataset

Creating a dataset requires thorough planning and execution to ensure its quality and relevance for the intended task. Initially, the scope and purpose of the dataset are defined, specifying criteria such as data types, sources, and required volume. Potential sources like public repositories, APIs, or manual data collection methods are then identified. Data collection protocols are implemented with ethical guidelines and privacy considerations in mind, ensuring proper consent and anonymization where necessary. Techniques such as web scraping, surveys, or crowd-sourcing are employed to gather diverse and representative samples.

The dataset is iteratively refined through processes like data cleaning, validation, and augmentation to enhance usability and reliability. Thorough documentation including metadata and usage guidelines is provided to facilitate accessibility and reproducibility for researchers and practitioners.

To evaluate the effectiveness of a new pest insect recognition system, insect images collected from the internet were utilized to train convolutional neural network (CNN) models. The primary objective was to develop a system capable of identifying and distinguishing 10 common insect groups. The dataset used, Insect10_Bbox [14], consisted of 2,335 images categorized into 10 classes.

To ensure effective learning and accurate evaluation of the models, the Insect10_BBox dataset was divided into three sets: training, validation, and testing, in a ratio of 7:2:1. This division ensures sufficient representative images for each insect class in every subset of the dataset.

Specifically, the training set comprises 1,633 images, the validation set contains 467 images, and the testing set includes 235 images. This balanced split enables the model to encounter various examples of each insect type, thereby enhancing its ability to accurately recognize and distinguish them.

D. YOLO

1) YOLO algorithm: The You Only Look Once (YOLO) algorithm is an object detection method. YOLO utilizes a unified model to simultaneously predict bounding boxes and the probabilities of classes within these boxes [16]. This method operates by applying a single convolutional neural network across the entire input image, thereby quickly providing predictions. Compared to traditional classification methods, YOLO is trained on a loss function directly related to detection performance, allowing the model to learn the best way to comprehensively detect objects, as illustrated in Fig. 9.

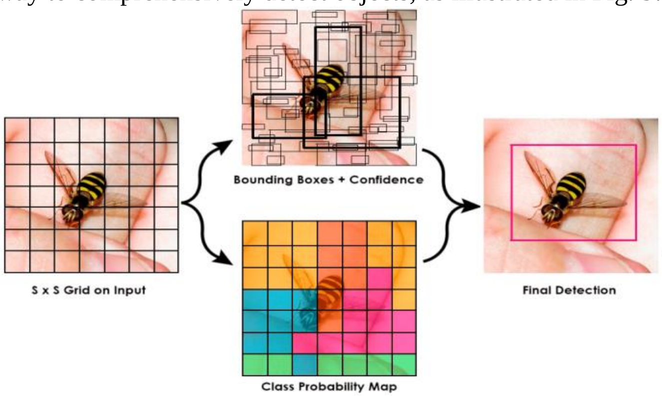

Fig. 9. YOLO algorithm for detecting insects.

In general, during classification, we determine labels from the data being tested. However, in YOLO, classification is combined with localization by providing additional information about the object's location in the form of bounding boxes. Each bounding box B consists of five predictions: x, w, y, h, and confidence score. The coordinates (x, y) represent the center of the box, determined by grid cells. Meanwhile, w (width) and h (height) predict the size of the object in the overall image [17]. The confidence score is typically used to represent the Intersection Over Union (IOU), a measure of the correlation between the predicted box and the actual object's box.

2) YOLOv7: YOLOv7 is a highly impactful algorithm in the computer vision and machine learning communities, surpassing previous object detection models and YOLO versions in both speed and accuracy [18]. It requires cheaper hardware and can be trained quickly on small datasets without pre-trained weights. Key features include an efficient backbone network, advanced optimization strategies, and novel loss functions, making it suitable for real-time applications.

YOLOv7's versatility allows it to be effectively used in domains like autonomous driving, surveillance, and medical imaging. It is easy to deploy, compatible with common machine learning frameworks, and adaptable for specific tasks, such as agricultural technology and retail. Additionally, YOLOv7 supports edge computing solutions, enabling real-time detection on devices with limited processing power. Overall, YOLOv7 sets new standards for performance and efficiency in object detection, driving innovation across various fields. The architecture of YOLOv7 is shown in Fig. 10.

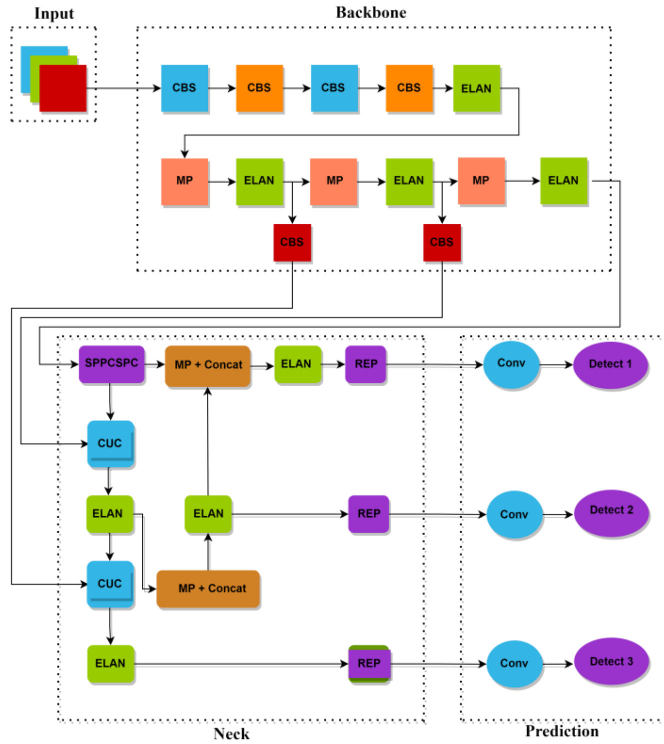

Fig. 10. YOLOv7 architecture. (Different colors represent the various functions performed within a single block).

E. Training Model

The dataset used for training the model is the Insect10_BBox of the authors in [14]. This dataset includes 10 insect classes including 'Acalymma_vittatum', 'Achatina_fulica', 'Alticini', 'Asparagus_beetles', 'Aulacophora_similis', 'Cerotoma_trifurcata', 'Dermaptera', 'Leptinotarsa_decemlineata', 'Mantodea', and 'Squash_bug'. The number of images for each insect class in the dataset used for training, validation, and testing the model is presented in Table IV.

TABLE IV. TABLE OF INSECT QUANTITIES IN THE INSECT10_BBOX DATASET

STT	Insect Name	Train	Val	Test
1	Acalymma_vittatum	115	33	17
2	Achatina_fulica	258	74	36
3	Alticini	193	55	28
4	Asparagus_beetles	89	25	13
5	Aulacophora_similis	113	32	17
6	Cerotoma_trifurcata	86	25	17
7	Dermaptera	111	32	16
8	Leptinotarsa_decemlineata	234	67	34
9	Mantodea	185	53	26
10	Squash_Bug	249	71	36
Total		**1633**	**467**	**235**

The training process for the YOLOv7 model is carried out using Google Colab, a free cloud computing service that provides a powerful Jupyter Notebook environment. Google Colab also offers access to GPUs to accelerate the model's training speed.

The first step begins by downloading the source code of YOLOv7 from GitHub and installing other necessary supporting libraries to run YOLOv7 on Google Colab in the requirement.txt file. Then, we proceed to pretrain the YOLOv7 model to evaluate its detection performance. Next, we upload the Insect10_BBox dataset to Google Drive and connect it to Google Colab. Additionally, we modify the data configuration file and YOLOv7 cfg file according to the number of object classes in the dataset.

Start training with the YOLOv7 model on the Insect10_BBox dataset. The configuration of the data.yaml file includes class names as the names of insect objects in the Insect_10BBox dataset; the content of dataset yaml file is shown in Table V.

TABLE V. DATASET YAML FILE

Item	Value
path	'/content/gdrive/My Drive/QL/Insect10_BBox'
train	'/content/gdrive/My Drive/QL/Insect10_BBox/images/train'
val	'/content/gdrive/My Drive/QL/Insect10_BBox/images/val'
test	'/content/gdrive/My Drive/QL/Insect10_BBox/images/test'
nc	10
names	['acalymma', 'alticini', 'Squash_Bug', 'asparagus', 'aulacophora', 'dermaptera', 'leptinotarsa', 'mantodea', 'Achatina_fulica', 'Cerotoma_trifurcata']

Finally, the YOLOv7 model was trained using the prepared training dataset. The training process will iterate through batches of images and update the model parameters based on the dataset used. During training, researchers can monitor evaluation metrics such as loss rate, accuracy, or mAP (mean average precision) to assess the model's performance. These evaluations can be conducted on validation or test datasets. Once the training process is complete, the model will be saved in an appropriate format. The model will undergo testing, and if the obtained results do not meet the requirements, the model will continue training until it fits the predefined parameters.

F. Evaluation Metrics

In our study evaluating the performance of an insect detection system, we employed a confusion matrix to provide a thorough understanding of its classification capabilities [19]. This matrix, a fundamental tool in classification model assessment, tabulates the counts of true positive (TP), true negative (TN), false positive (FP), and false negative (FN) predictions. Here, we elucidate these components:

- True Positives (TP): Insects correctly identified by the system.
- True Negatives (TN): Non-insects correctly identified as such.
- False Positives (FP): Non-insects erroneously identified as insects (Type I error).
- False Negatives (FN): Insects erroneously identified as non-insects (Type II error).

Leveraging these values, we computed several performance metrics: Accuracy, Precision, and Recall. Accuracy measures the ratio of correctly identified insects to the total number of insects in the test dataset. Precision, a critical metric, delineates the ratio of true positive detections to the sum of true positive and false positive detections. Similarly, recall assesses the ratio of true positive detections to the sum of true positive and false negative detections. Additionally, we employed the F1-score, serving as the harmonic mean of precision and recall, to provide a balanced evaluation of the system's performance. These performance metrics were calculated using the following equations:

$$\text{Accuracy} = \frac{TP+TN}{TP+FP+TN+FN} \times 100\% \quad (1)$$

$$\text{Precision} = \frac{TP}{TP+FP} \times 100\% \quad (2)$$

$$\text{Recall} = \frac{TP}{TP+FN} \times 100\% \quad (3)$$

To assess robustness, we tested the system across various environmental conditions, such as different lighting and backgrounds, to ensure consistent performance. The evaluation process should also include computational efficiency, assessing the system's processing speed and resource utilization. Finally, user feedback and field testing provide practical insights into the system's usability and real-world effectiveness, enabling further refinements.

G. Deployment on NVIDIA Jetson Nano

1) Preparation for Installation: To proceed with the installation of the Jetson Nano device, the following items have been prepared:

- Jetson Nano Developer Kit equipment box, including: NVIDIA Jetson module and reference carrier board.
- MicroSD card (recommended minimum 32GB UHS-1).
- MicroSD card reader to USB port.
- USB keyboard and mouse.
- Computer monitor (HDMI or DP).
- 5V-4A power supply.
- Pre-trained YOLO model.
- NVIDIA Jetpack.
- BalennaEtcher software for booting the drive onto the microSD card.
- SDCardFormatter software for formatting the microSD card.

2) Device setup: Initially, the microSD card undergoes formatting using SDCardFormatter. Subsequently, the Jetpack, obtained via download, is to be flashed onto the microSD card utilizing BalenaEtcher. Upon completion, the microSD card is to be inserted into the Jetson Nano.

The Jetson Nano operates efficiently with a 5V-4A power supply, facilitating easy connection for powering and booting up. It supports HDMI connectivity to a monitor, enabling users to visualize the interface and outcomes of AI applications. Additionally, it features a Gigabit Ethernet port for network access, facilitating internet connection and LAN device connectivity. With four USB 3.0 ports and one Micro-B USB 2.0 port, the Jetson Nano offers versatile connectivity options. The Micro-B USB 2.0 port serves dual purposes for power supply or device mode.

Integrated with two cameras, the Jetson Nano enables direct connection for recognition tasks. Furthermore, it offers various interfaces including GPIO, I2C, I2S, SPI, and UART, facilitating connection of peripheral devices such as sensors, motors, and expansion modules. Fig. 11 presents description of these device connections. The actual configuration of external devices connected to the Jetson Nano is depicted in Fig. 12.

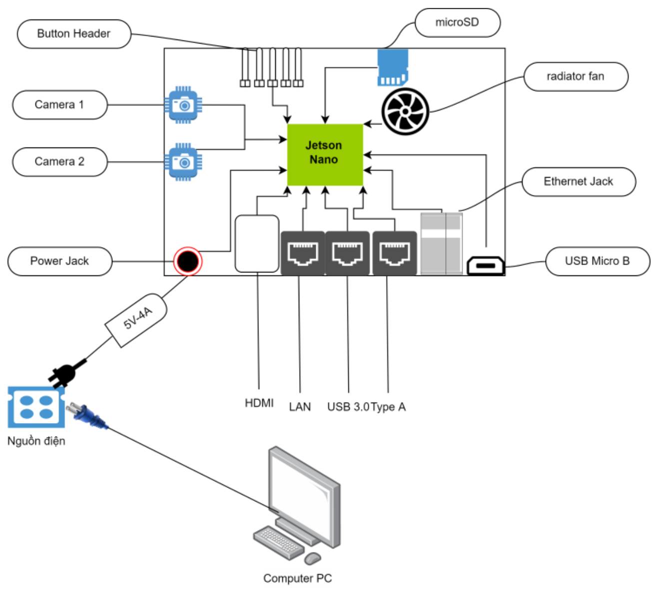

Fig. 11. Description of device connections.

Fig. 12. The actual configuration of external devices connected to the Jetson Nano.

The green LED next to the MicroUSB connector will light up. During the initial boot, the tool will guide users through the setup process, which includes:

- Reviewing and accepting the NVIDIA Jetson EULA software.
- Selecting the system language, keyboard layout, and time zone.
- Creating a username, password, and computer name.

For the APP partition size, we use the maximum recommended size. The setup process will take approximately one minute. After completion, the computer screen will boot up as shown in Fig. 13.

Fig. 13. The screen after configuration completion for Jetson Nano.

3) *Library setup:*

a) PyTorch: PyTorch offers a powerful and versatile deep learning framework built for Python. Backed by a thriving community and a rich ecosystem of tools, PyTorch excels in both research and production settings. It delivers seamless interoperability and optimized performance for your machine-learning projects.

b) *TorchVision:* TorchVision is your one-stop shop for computer vision projects using PyTorch. It streamlines development by providing pre-trained models and image transformation tools. This powerful library bridges the gap between cutting-edge research and real-world applications on your Jetson Nano.

c) CUDA: CUDA, the de facto standard for GPU acceleration, empowers you with high-performance computing tools. This comprehensive toolkit accelerates application development and unleashes the full potential of your deep learning PC or Jetson Nano.

III. RESULTS AND DISCUSSION

A. *Model Training*

This study proposes a novel model utilizing the YOLOv7 algorithm for real-time detection of harmful insects. The model is trained for 100 epochs with a batch size of 8, demonstrating high efficiency in identifying and classifying various insect species. Feature extraction and object detection training were conducted using different YOLOv7 models: YOLOv7, YOLOv7-X, and YOLOv7-W6. The comprehensive training results, including metrics such as FPS, model size, precision, and recall, are detailed in Table VI, showcasing the effectiveness of the proposed approach in diverse environmental conditions.

TABLE VI. YOLOv7 TRAINING RESULTS

Dataset	Models	FPS	Model size (MB)	Precision (%)	Recall (%)
Insect10_Bbox	YOLOv7	161	74.9	74.7	73.4
	YOLOv7-X	114	142.2	84.2	80.3
	YOLOv7-W6	84	162.7	89.6	82.5

The corresponding confusion matrix for the trained YOLOv7 model was obtained and is presented in Fig. 14. This confusion matrix reflects the performance of the classifier when evaluated on the test set. The diagonal elements indicate the number of samples correctly predicted for each insect class. As illustrated in Fig. 14, the leptinotarsa class achieved the highest accuracy at 89%, whereas the acalymma class exhibited the lowest accuracy at 63%.

To enhance the model's performance, attention should be directed towards improving the prediction results for the acalymma class. The misclassification rate for this class is 13%, as indicated by the sum of the values in the white box of column 1, representing incorrect predictions into other classes. Additionally, there is a 24% false negative rate, where the model fails to detect the presence of an insect when one is actually present. This rate is the highest among all classes. Consequently, the accuracy in predicting the acalymma class is limited to 63%.

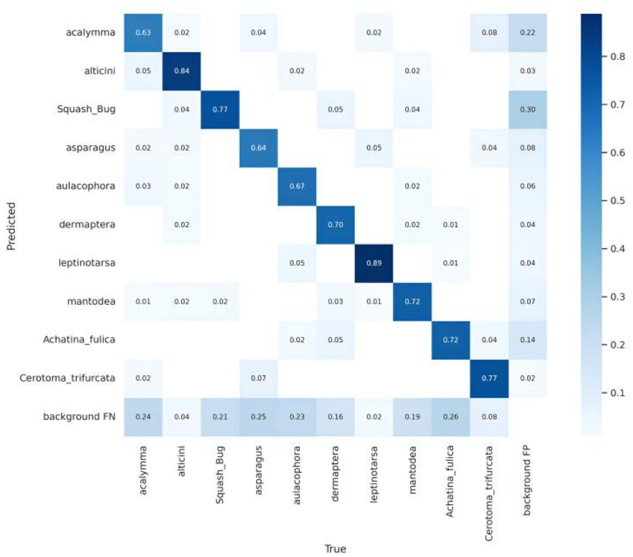

Fig. 14. Confusion matrix for training the YOLOv7 model.

The disparity in accuracy for the acalymma class can be attributed to a limited or lower quality dataset and high visual similarity with other classes, which confuses the model. To address this, several strategies are recommended: applying data augmentation techniques to increase the diversity of training samples, collecting more high-quality images of acalymma, fine-tuning the YOLOv7 model specifically for acalymma, implementing class rebalancing with weighted loss functions, and conducting feature analysis to highlight distinctive characteristics of acalymma. These approaches aim to improve the model's accuracy in predicting the acalymma class and enhance overall performance in insect classification tasks, with further experiments and validations needed for optimal results.

B. Detect on NVIDIA Jetson Nano

Upon completion of the training process, the YOLOv7 model is employed for object detection in images and videos, as referenced in study [20]. The detection outcomes for single-class insect identification are either displayed on the screen or saved to a file, as illustrated in Fig. 15. This procedure involves the model analyzing each frame or image to identify and classify insects based on the training it received. The results are then rendered visually on the screen with bounding boxes around detected insects, or alternatively, the data can be stored in a file for subsequent analysis. This dual approach allows for both immediate visual verification and detailed post-processing, enhancing the versatility and applicability of the detection system in various operational contexts.

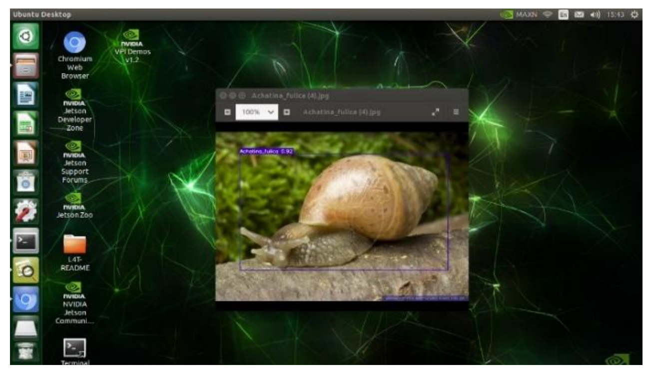

Fig. 15. Single-class insect detection.

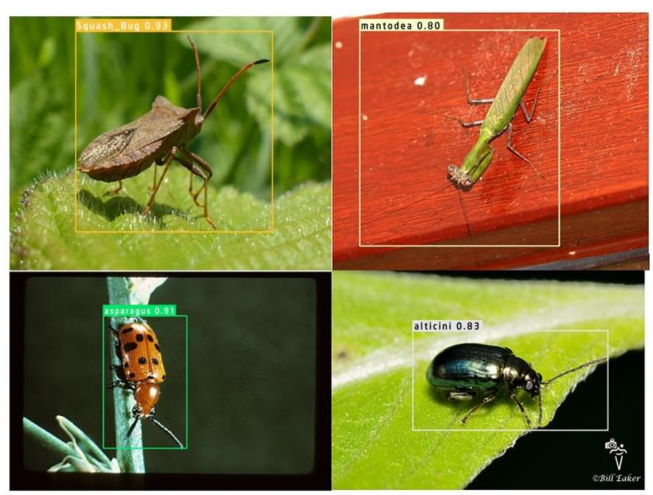

Fig. 16. Multiclass insect detection.

Fig. 15 and Fig. 16 present examples of insect object detection performed on the Jetson Nano, demonstrating both single-class and multiclass detection capabilities using images from the test dataset. These figures highlight the accuracy and efficiency of the model in identifying various insect species under different conditions. Table VII provides comprehensive details on the system's performance metrics, including frames per second (FPS) and precision, offering insights into the computational efficiency and detection accuracy of the YOLOv7 model on the Jetson Nano platform. This performance evaluation is critical for understanding the model's applicability in real-time insect monitoring and detection scenarios, ensuring reliable and efficient operation in practical applications.

TABLE VII. INSECT RECOGNITION TEST RESULTS ON NVIDIA JETSON NANO

	Insect name	Detection result	FPS	Precision (%)
Single-class object detection	Acalymma_vittatum	1 Acalymma	5,543	67%
	Achatina_fulica	1 Achatina_fulica	5,735	66%
	Alticini	1 alticini	4,999	76%
Multiclass object detection	Alticini, Squash_Bug, Mantodea, Asparagus_bee	1 alticicni, 1 Squash_Bug, 1 mantodea, 1 asparagus	4,890	Alticini 83%, Squash_Bug 93%, mantodea 80%, asparagus 91%

We conducted real-time insect detection experiments using NVIDIA Jetson Nano. The results indicate an approximate frame rate of 4 frames per second (FPS), as illustrated in Fig. 17. This frame rate demonstrates the capability of the Jetson Nano to perform real-time processing despite its limited computational resources. The experiments were designed to evaluate the practical applicability of the YOLOv7 model in field conditions, ensuring that the system can effectively detect and classify insects in real-time. The findings highlight the balance between detection accuracy and processing speed, crucial for developing efficient and responsive insect monitoring systems. Further optimization and hardware enhancements could potentially improve the FPS, making the system even more robust for large-scale deployments.

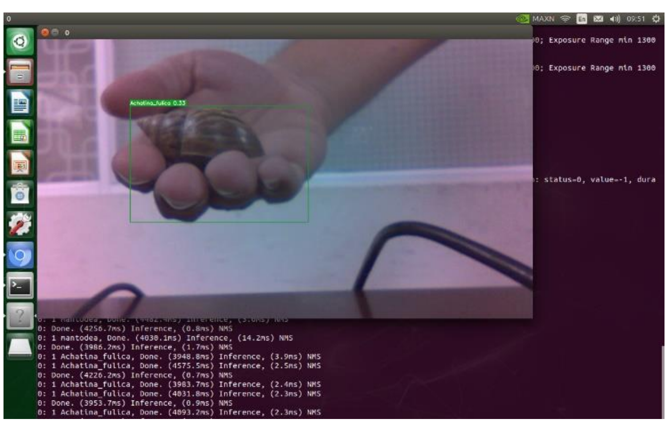

Fig. 17. Real-time insect detection.

IV. CONCLUSION AND FUTURE WORK

The novel smart system using the Jetson Nano for remote insect monitoring provides a scalable, efficient, and accurate method to assess and manage insect populations in various ecosystems. Its successful implementation can lead to more sustainable agricultural practices and enhanced environmental conservation efforts. Our system was developed based on the YOLOv7 model due to its lightweight convolutional neural network, which allows for effective insect pest detection and classification. This technology can be integrated into hardware accessible to farmers, enabling its use in diverse situations to protect crops from pests. Our method offers numerous advantages, including real-time insect identification, low cost, simple implementation, and practical applicability. Numerical results demonstrated that the system achieved a classification accuracy of 77.2% with mAP@0.5 on the Insect10 dataset. However, this mAP accuracy is still lower than what is required for effective insect detection in agricultural production. Future work will focus on refining the algorithms, expanding the range of detectable insect species, and integrating larger datasets to enhance the system's accuracy and overall effectiveness.

ACKNOWLEDGMENT

This research is funded by Vietnam National University Ho Chi Minh City (VNU-HCM) under grant number “DS2024-16-06.”

REFERENCES

[1] M. Cardim Ferreira Lima, M. E. Damascena De Almeida Leandro, C. Valero, L. C. Pereira Coronel, and C. O. Gonçalves Bazzo, “Automatic Detection and Monitoring of Insect Pests—A Review,” Agriculture, vol. 10, no. 5, p. 161, May 2020, doi: 10.3390/agriculture10050161.

[2] W. Li, T. Zheng, Z. Yang, M. Li, C. Sun, and X. Yang, “Classification and detection of insects from field images using deep learning for smart pest management: A systematic review,” Ecological Informatics, vol. 66, p. 101460, Dec. 2021, doi: 10.1016/j.ecoinf.2021.101460.

[3] C. R. Okpara, V. E. Idigo, and S. M. Oguchienti, “Wireless Sensor Networks for Environmental Monitoring: A Review,” IJETT, vol. 68, no. 1, pp. 68–71, Jan. 2020, doi: 10.14445/22315381/IJETT-V68I1P210.

[4] U. Tupe, D. S. Kadam, and D. P. N. Mahalle, “Survey Paper on Optimized Energy-Efficient Protocol for M2M Communication towards Green IoT”, Thirteenth International Conference on Recent Trends in Communication and Computer Networks- ComNet 2023.

[5] U. Ahmad, A. Nasirahmadi, O. Hensel, and S. Marino, “Technology and Data Fusion Methods to Enhance Site-Specific Crop Monitoring,” Agronomy, vol. 12, no. 3, p. 555, Feb. 2022, doi: 10.3390/agronomy12030555.

[6] S. F. Khan and M. Y. Ismail, “An Investigation into the Challenges and Opportunities Associated with the Application of Internet of Things (IoT) in the Agricultural Sector-A Review,” Journal of Computer Science, vol. 14, no. 2, pp. 132–143, Feb. 2018, doi: 10.3844/jcssp.2018.132.143.

[7] S. Valladares, M. Toscano, R. Tufiño, P. Morillo, and D. Vallejo-Huanga, “Performance Evaluation of the Nvidia Jetson Nano Through a Real-Time Machine Learning Application,” in Intelligent Human Systems Integration 2021, vol. 1322, pp. 343–349. doi: 10.1007/978-3-030-68017-6_51.

[8] A. Albanese, M. Nardello, and D. Brunelli, “Automated Pest Detection with DNN on the Edge for Precision Agriculture.” Aug. 02, 2021, IEEE Journal on Emerging and Selected Topics in Circuits and Systems PP(99):1-1. doi: 10.36227/techrxiv.15087729.v1.

[9] D. J. A. Rustia, C. E. Lin, J.-Y. Chung, and T.-T. Lin, “A Real-Time Multi-Class Insect Pest Identification Method Using Cascaded Convolutional Neural Networks”, 9th International Symposium on Machinery and Mechatronics for Agriculture and Biosystems Engineering (ISMAB), Jeju, South Korea.

[10] J. Wäldchen and P. Mäder, “Machine learning for image based species identification,” Methods Ecol Evol, vol. 9, no. 11, pp. 2216–2225, Nov. 2018, doi: 10.1111/2041-210X.13075.

[11] Y. He, H. Zeng, Y. Fan, S. Ji, and J. Wu, “Application of Deep Learning in Integrated Pest Management: A Real-Time System for Detection and Diagnosis of Oilseed Rape Pests,” Mobile Information Systems, vol. 2019, pp. 1–14, Jul. 2019, doi: 10.1155/2019/4570808.

[12] S. Dong et al., “Automatic Crop Pest Detection Oriented Multiscale Feature Fusion Approach,” Insects, vol. 13, no. 6, p. 554, Jun. 2022, doi: 10.3390/insects13060554.

[13] D. M. Nazeer, M. Qayyum, and D. A. Ahad, “Real Time Object Detection And Recognition In Machine Learning Using Jetson Nano,” vol. 11, no. 10, 2022.

[14] T.-N. Doan, “Large-Scale Insect Detection With Fine-Tuning YOLOX,” ijmst, vol. 10, no. 2, pp. 892–915, Jun. 2023, doi: 10.15379/ijmst.v10i2.1306.

[15] NVIDIA Corporation, "Jetson Nano Developer Kit," [Online]. Available: https://developer.nvidia.com/embedded/jetson-nano-developer-kit. [Accessed:12-10-2023]

[16] X. Yue, H. Li, M. Shimizu, S. Kawamura, and L. Meng, “YOLO-GD: A Deep Learning-Based Object Detection Algorithm for Empty-Dish Recycling Robots,” Machines, vol. 10, no. 5, p. 294, Apr. 2022, doi: 10.3390/machines10050294.

[17] Hadi Supriyanto, Sarosa Castrena Abadi, and Aliffa Shalsabilah, “Deteksi Helm Keselamatan Menggunakan Jetson Nano dan YOLOv7,” J. Appl. Comput. Sci. Technol., vol. 5, no. 1, pp. 1–8, Feb. 2024, doi: 10.52158/jacost.v5i1.637.

[18] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, “YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors,” in 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada: IEEE, Jun. 2023, pp. 7464–7475. doi: 10.1109/CVPR52729.2023.00721.

[19] Doe, J. (2020). “Machine Learning for Classification. Springer”. doi: 10.1007/978-3-030-12345-6

[20] H. Gomes, N. Redinha, N. Lavado, and M. Mendes, “Counting People and Bicycles in Real Time Using YOLO on Jetson Nano,” Energies, vol. 15, no. 23, p. 8816, Nov. 2022, doi: 10.3390/en15238816.

[21] Abbas, Muneer & Ramzan, Muhammad & Hussain, Niaz & Ghaffar, Abdul & Hussain, Khalid & Abbas, Sohail & Raza, Ali. (2019). Role of Light Traps in Attracting, Killing and Biodiversity Studies of Insect Pests in Thal. Pakistan Journal of Agricultural Research. 32. 10.17582/journal.pjar/2019/32.4.684.690.

[22] Fabian, S.T., Sondhi, Y., Allen, P.E. et al. Why flying insects gather at artificial light. Nat Commun 15, 689 (2024). https://doi.org/10.1038/s41467-024-44785-3.