A Novel Real-Time Insect Detection System on Mobile Smart Devices

Thanh-Nghi Doan [1,2]

[1] Faculty of Information Technology, An Giang University, An Giang, Vietnam
[2] Vietnam National University, Ho Chi Minh City, Vietnam
Email: dtnghi@agu.edu.vn

***Abstract*—The rapid spread of diseases and pests has significantly impacted global agricultural productivity. Farmers often struggle with pest identification, leading to the overuse of pesticides, which causes environmental harm and incurs high costs. This work presents an early real-time insect identification system using deep learning for real-time mobile insect image detection. By applying the YOLOv5-S model to a 10-species insect dataset, the system achieved a mAP@0.5 accuracy of 70.5%, and 42.9% on the IP102 dataset, optimized for low-end mobile devices. Additionally, it provides farmers with vital information on insect biology, distribution, and management to reduce production costs and promote sustainable farming.**

***Keywords*—deep learning, real-time insect identification system, YOLOv5, low-end mobile devices**

I. INTRODUCTION

Climate change has caused an increase in insect populations, posing a significant danger to world agriculture [1]. According to the Food and Agriculture Organization, pests account for up to 40% of yearly crop losses, with invasive insects alone costing more than $70 billion [2]. Farmers rely extensively on pesticides to tackle this problem, but a lack of skill in pest identification frequently leads to overuse or misuse [3]. This not only raises production costs, but it also destroys ecosystems, kills beneficial insects, and endangers human and animal health [4].

Consequently, the requirement for a pest detection system that is both effective and easily accessible has become crucial. Many farmers find traditional methods impracticable, particularly those in developing nations, because they sometimes call for costly equipment or specialized skills. A mobile-based approach provides a useful substitute, especially considering how common cellphones are. A system that supports early pest treatment should be affordable, easy to use, and able to identify insects in real time [5].

In this work, a novel real-time insect detection method for mobile smart devices is presented. Using the YOLOv5-S deep learning model, the system effectively recognizes pests from smartphone camera photos, offering comprehensive biological and management data. Accessibility for farmers in various agricultural environments is guaranteed by this strategy, which is tailored for low-end mobile technology. Real-time pest identification and educational materials are integrated into the system to improve sustainable agricultural methods and lessen need on dangerous chemicals.

II. RELATED WORKS

Previous research has been focused on developing real-time CNN architecture-based image identification systems for mobile devices. For instance, Wang *et al*. [6] created a novel technique for extracting and categorizing pictures of leaves. Using a mathematical morphological approach to segregate items in areas of adhesion, they employed a region-labeling technique to determine insect populations and disease areas inside segmented pictures. When the system was implemented on mobile smart devices, field tests revealed that it performed outstandingly in terms of efficiency and recognition.

In a different research, Nasir *et al.* [7] created a web-based platform and an Android application as part of an early warning system for insect infestations in rice farming. The Agriculture Department can identify and locate pest infestations with the use of this technology, which then notifies farmers. By entering infestation data into databases, agronomists are able to assess the danger in paddy plots by taking into account factors such as insect count, kind, location, and current circumstances. Farmers are notified via email on the condition of their paddy plots following the agronomists' evaluation.

In order to identify and count insects, Zhu *et al.* [8] presented a smartphone application and an image processing method. They used a sliding window-based binarization technique to solve the problem of non-uniform brightness in insect pictures taken using mobile phones. They used linked domain-based histogram statistics to identify and count the insects in the stored grain. The approach outperformed earlier techniques, with a 95% accuracy rate when tested on an Android application.

As shown in Ref. [9], MAESTRO is a cutting-edge framework for identifying grasshoppers that uses deep learning to identify insects in RGB photos. This two-stage deep learning training approach can be applied on both

Manuscript received December 18, 2024; revised January 21, 2025; accepted March 4, 2025; published May 9, 2025.

smartphones and desktop PCs. Similar to this, Chen *et al.* [10] used deep learning-based object recognition models including Faster R-CNN, SSD, and YOLOv4 to create an AI-based pest detection system tailored for scale pests. With 100% accuracy for mealybugs, 89% accuracy for Coccidae, and 97% accuracy for Diaspididae, the YOLOv4 model outperformed the others in this regard. Based on this concept, a smartphone application assists farmers in identifying pests and using the right pesticides.

In order to create a pest detection model appropriate for mobile information systems, Perera [11] investigated the most effective machine learning strategies. Similarly, Karar *et al.* [12] presented a deep learning-based smartphone application called Faster R-CNN for cloud-based insect pest recognition, which is used to classify pests. With five distinct pest species, this application has 99% detection accuracy and is linked to a database that provides pesticide recommendations.

In order to identify pests in vineyards, a technique for hand-held picture capturing of insect traps was given in the work of Faria *et al.* [13], which directly integrated AI into mobile devices. This method enhances picture quality and relevancy by combining many computer vision technologies. In-depth analysis of deep learning frameworks for intelligent pest monitoring was done in [14], with a particular emphasis on the identification and categorization of insect pests from field photos. The review included technical information and methods for several phases, such as data pretreatment, modeling strategies, and picture acquisition. It also covered upcoming difficulties and new developments in the sector, as well as suggesting a general framework for smart insect monitoring.

For real-time detection, AlertTrap [15] used SSD architecture in conjunction with MobileNetV1 and MobileNetV2 backbone feature extractors. The AP@0.5 rates of 0.957 and 1.0 were attained by the SSD-MobileNetV1 and SSD-MobileNetV2 models, respectively. SSD devices are better suited for real-time applications, as seen by the slower throughput of YOLOv4-tiny, despite its superior performance in AP@0.5. When it came to resistance to environmental disturbances, YOLOv4-tiny outperformed SSD devices. Furthermore, Doan [16] integrated Power mean SVM [17] with EfficientNet [18], resulting in 71.84% accuracy in state-of-the-art insect image classification on the extensive IP102 dataset. Table I provides a systematic evaluation and detailed comparison of existing insect detection approaches.

TABLE I. COMPARISON OF EXISTING INSECT DETECTION APPROACHES

Study	Methodology	Model/Algorithm Used	Dataset	Key Findings	Limitations
Wang *et al.* (2013) [6]	Traditional image processing	Region-labeling + morphological operations	Leaf images	Achieved high accuracy in segmenting insect-infested regions	Limited scalability; not real-time
Nasir *et al.* (2018) [7]	Web-based early warning system	Pest count and monitoring	Rice farming dataset	Notifies farmers about pest outbreaks	Requires internet access
Zhu *et al.* (2018) [8]	Smartphone image processing	Sliding window binarization	Stored grain dataset	95% accuracy in insect counting	Affected by lighting variations
Chudzik *et al.* (2020) [9]	Deep learning on mobile	Two-stage CNN (MAESTRO)	RGB insect images	High accuracy on smartphones	Requires GPU optimization
Chen *et al.* (2021) [10]	Object detection for scale pests	Faster R-CNN, SSD, YOLOv4	Custom dataset	100% accuracy for mealybugs, 89% for Coccidae, 97% for Diaspididae	High hardware requirements
Karar *et al.* (2021) [12]	Cloud-based deep learning	Faster R-CNN	Pest dataset	99% accuracy with cloud support	Not suitable for offline use
Le *et al.* (2021) [15]	Edge computing for insect traps	SSD-MobileNetV1 & V2	Remote trap images	SSD-MobileNetV2 achieved 1.0 AP@0.5	Limited detection range
Doan *et al.* (2022) [16]	Large-scale pest classification	EfficientNet + Power mean SVM	IP102 dataset	71.84% classification accuracy	Requires dataset balancing

Despite improvements, existing pest detection systems still have a number of drawbacks, including a narrow range of pest identifications, poor accuracy, costly equipment needs, and difficult deployment scenarios. Furthermore, most of these systems are devoid of capabilities like powerful distributed mobile information frameworks, thorough pest information, and geolocation tracking of dangerous pests. For mobile devices, there is currently no solution that offers real-time identification.

To address these issues, this paper proposes an early real-time insect recognition system that is low-cost, efficient, and designed for mobile devices with limited hardware. The study looks at lightweight network models and embedded terminal solutions, which are becoming more significant and appealing. The primary contributions of this paper are:

- A real-time insect identification system has been designed for mobile devices with limited hardware, ensuring easy installation, affordability, and user-friendliness.
- YOLOv5-S identification findings for large-scale IP102 dataset.
- A novel method collects photos and utilizes GPS to map insect dispersion in the field, leading to the creation of a complete database and distribution maps.

The paper is organized as follows: Section III describes the materials and techniques utilized to evaluate our methodology, which include an overview of our system, the YOLOv5 model, and the pest bug picture datasets. Section IV explores the experimental findings and their consequences. Section V summarizes the findings, limitations, and recommendations for further study.

III. Materials and Methods

A. Overview of Our System

The proposed real-time insect detection system is designed to assist farmers by providing accurate pest identification through mobile smart devices, as depicted in Fig. 1. Leveraging the YOLOv5-S model, the system analyzes images captured via smartphone cameras or downloaded from web resources, such as photographs of insects taken by insect traps, delivering real-time analysis with minimal hardware requirements. Key functionalities include real-time pest identification, which detects and classifies insects from images or live camera feeds while providing biological details and potential crop impact; offline and online modes, enabling cloud-based analysis for enhanced accuracy alongside offline access via an SQLite database for use in remote areas; GPS-based pest mapping, which tracks insect distribution to monitor infestations and guide pest management strategies; pesticide recommendations, offering appropriate pest control measures to minimize excessive pesticide use; and a user-friendly mobile interface, ensuring ease of use with a straightforward image capture process and instant results.

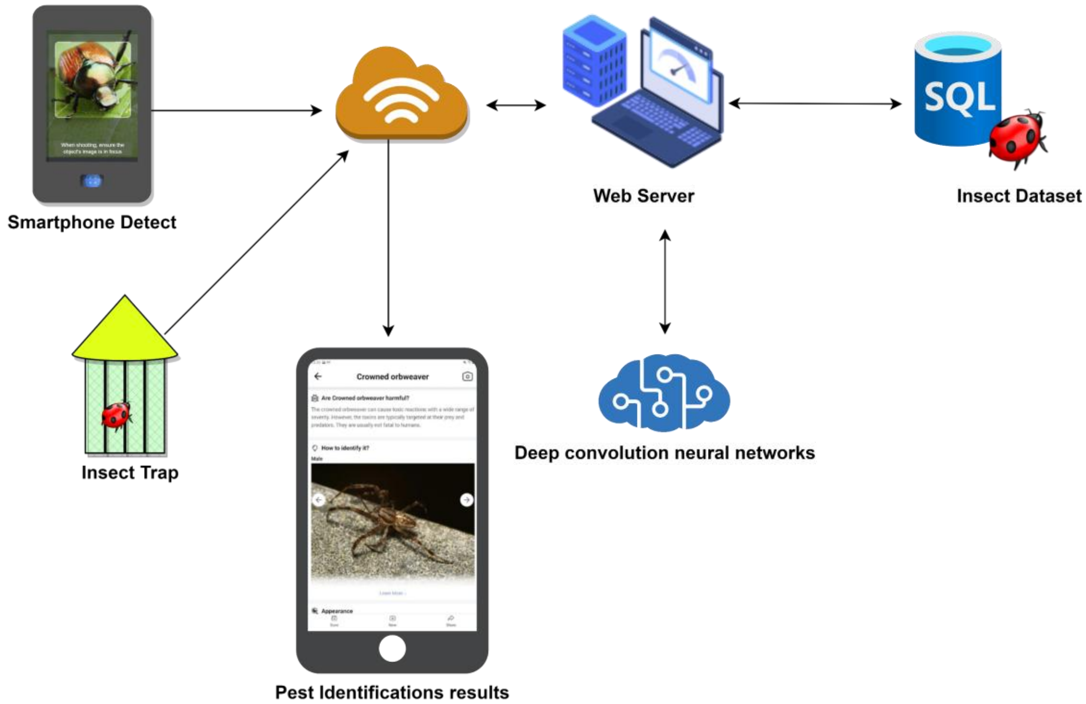


Fig. 1. An overview of our real-time insect image detection technology using mobile devices.

As illustrated in Fig. 2, the UV-equipped bug traps attract insects, which subsequently land on sticky traps designed for data collection. The deployment and operation of these traps enable the accumulation of a substantial volume of insect data, facilitating the gathering and identification of various insect species through the mobile application. To identify insects quickly, the mobile application integrates the YOLOv5-S model, which processes photos of insects or real-time camera feeds, as shown in Fig. 3. Once the insect has been correctly identified, the system gives comprehensive details about it, such as its name, biological traits, range, morphology, and methods of management.

Our insect recognition technology functions both online and offline. In the online mode, insect identification data is sent to a web server, which then analyzes it and returns full results in JSON format [19]. Users may compare this information to similar pictures in the data warehouse, browse a comprehensive list of insects with detailed information and images, and upload bug images and locations to the database. This option maintains the whole database on the server, guaranteeing that information is constantly up to date; but, the application's pace is restricted by network connectivity.

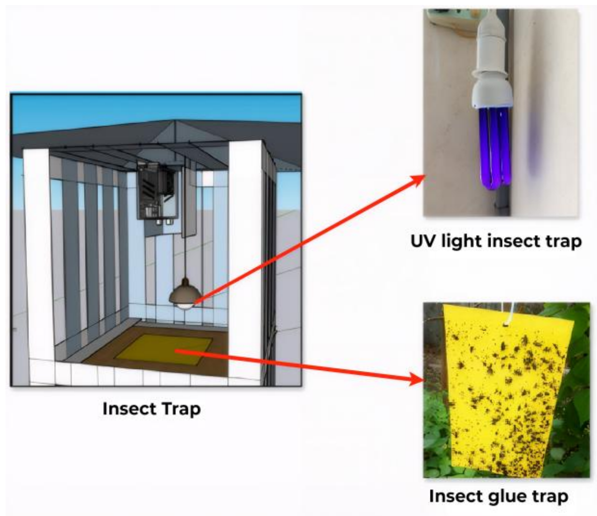


Fig. 2. Insect trap.

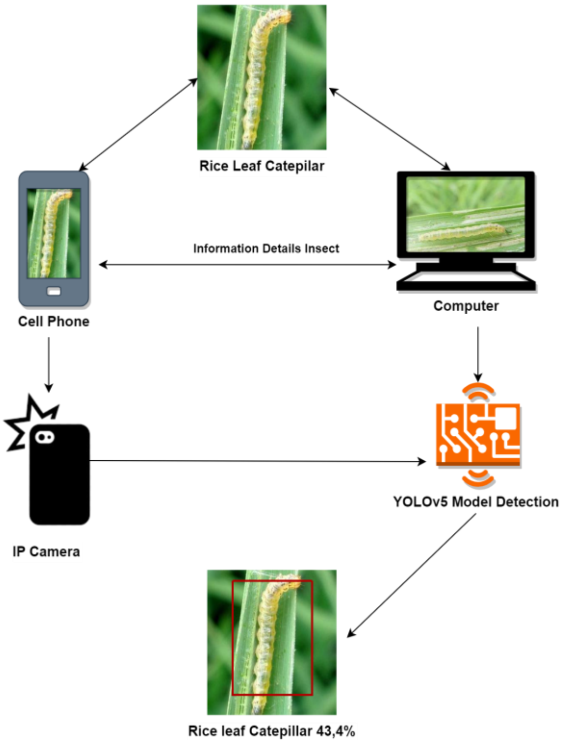


Fig. 3. Real-time mobile detection using YOLOv5s.

In offline mode, the system uses SQLite [20], a lightweight, quick, and dependable SQL database engine, to store insect data directly on mobile devices. This mode is very useful in locations without internet connection, such as isolated fields with restricted connectivity. However, some application capabilities may be limited in offline mode.

B. *YOLOv5*

YOLOv5 [21] is a single-stage object identification system that approaches object detection as a regression issue. In this method, object detection is done in a single step, estimating both the class probabilities and the coordinates of the bounding boxes that surround the objects in the input picture. The system consists of three major parts: the backbone, neck, and head. In YOLO, the head layer is in charge of ultimate object detection.

The model's core function is to identify distinguishable elements in the picture. The CSPNet [22] architecture serves as the foundation for YOLOv5. Fig. 4 shows how CSPNet divides the feature map in the base layer into two pieces. One portion integrates directly with the transition layer while the other half travels through the dense block to get there. This split contributes to a smaller model and faster inference [23].

The YOLOv5-S model is selected in this study for the development of mobile apps because of its small size, fast GFLOPs calculation speed, and excellent accuracy, which makes it appropriate for devices with constrained hardware. YOLOv5-S has a reduced disk capacity of 14.2 MB and a smaller network parameter size of 7.3 M compared to previous YOLO models like YOLOv4 [24] and YOLOX [25], which fits well with the limitations of mobile devices. As shown in Table II, the YOLOv5-S model is relatively small in size, with a network parameter of 7.2 M and a disk size of 14.2 MB, making it suitable for mobile devices with limited hardware configuration. With a GFLOPs index of 17.1, the model's processing speed is considered sufficient. Furthermore, compared to previous YOLO models, YOLOv5-S demonstrates high performance on the dataset from Table III, achieving outstanding mAPval@0.5 and speed metrics, as shown in Tables IV and V.

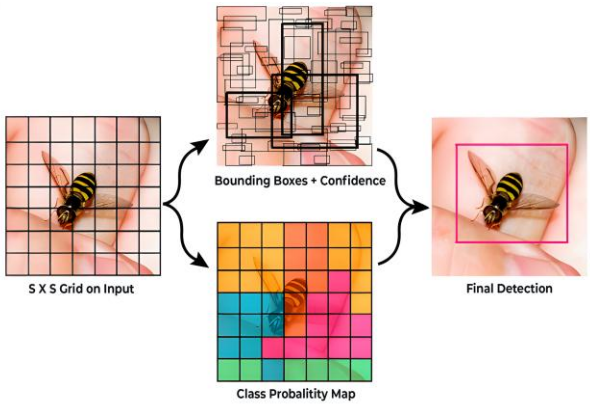


Fig. 4. Object detection in YOLO framework.

TABLE II. NETWORK PARAMETERS OF YOLO MODELS

Models	Params [M]	Size on disk [MB]	GFLOPs
YOLOv4	27.6	245.0	59.6
YOLOv4-tiny	5.88	23.1	6.8
YOLOv5-S	7.2	14.2	17.1
YOLOv5-M	21.2	40.8	51.4
YOLOv5-L	46.5	89.3	115.6
YOLOv5-X	86.7	167.1	219.0
YOLOX-S	9.0	68.5	26.8
YOLOX-M	25.3	193.0	73.8
YOLOX-L	54.2	413.0	155.6
YOLOX-X	99.1	757.0	281.9

C. *Proposed Methods*

The proposed approach involves five consecutive steps:

Image Collection: The process starts with insect pest images being gathered for both training and evaluating the models.

Data Preprocessing: The dataset is preprocessed through annotation and augmentation. Image data augmentation artificially increases the size of the training dataset by making slight modifications to existing images based on specific parameters.

Model Training: YOLO object detection models are then trained using the IP102 dataset, as depicted in Fig. 5. The dataset is split for validation purposes, allowing us to assess the detection performance of the fine-tuned models.

Performance Evaluation: The detection performance of the trained models is validated using the validation dataset, and the results are evaluated.

Model Selection: Finally, the most suitable model for practical application in a farming context is selected.

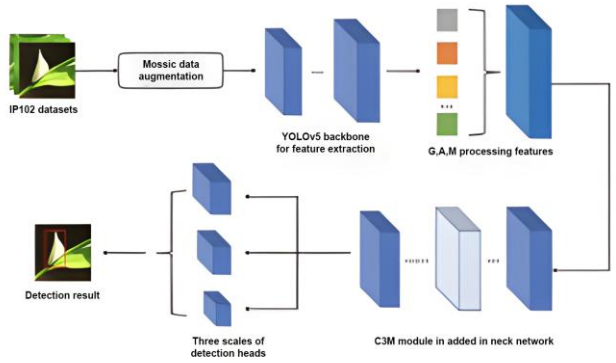


Fig. 5. Schematic flowchart of the research approach.

D. Datasets

In order to prioritize frequent and agriculturally relevant pests and retain a dataset appropriate for real-time detection on mobile devices, only 10 insect species were purposefully chosen. These species were picked because they are common in agricultural settings, have an effect on crop yields, and are pertinent to pest control plans. Restricting the dataset ensures smooth operation on low-end mobile hardware by preventing overfitting, maintaining a balanced data distribution, and improving model performance. Although adding more beneficial insects and pests might increase the system's applicability, doing so would need more computing power and larger datasets, which would not be suitable for mobile deployment. In order to enhance detection skills and offer a more thorough pest identification system, subsequent research stages will endeavor to broaden the dataset beyond the original 10 insect species. In order to give a more comprehensive context for insect identification difficulties, this study also makes reference to previous studies on large-scale pest detection, such as the IP102 dataset. Furthermore, the inclusion of beneficial insects such as Mantodea (praying mantises), who are natural pest predators, and Dermaptera (earwigs), which contribute to ecosystem balance, broadens our understanding of insect interactions in agriculture. This technique assists farmers in not only identifying and controlling destructive pests, but also recognizing and protecting beneficial species, therefore fostering sustainable pest management practices. By teaching farmers on the roles of both pests and beneficial insects, the system promotes better decision-making, decreases dependency on chemical pesticides, and promotes ecologically acceptable farming methods.

As shown in Fig. 6, 2,335 images of 10 distinct pest species were gathered from online data sources to create the insect pest database for the machine learning models. Table III displays the 1,634 pictures for training, 467 for validation, and 234 for testing that were obtained from the dataset, which was split into three categories: 70% for training, 20% for evaluation, and 10% for testing. The insect objects were manually labeled using the Label Image application [26], which produced.xml files with object location data. These files were subsequently transformed into.txt files that were compatible with YOLOv5. It was difficult to achieve high identification efficiency because of limitations in the IP102 dataset, such as the presence of numerous phases of the same insect type (e.g., larvae, caterpillars, and moths). Thus, the YOLOv5-S model was tested with the 10 insect classes collected by agriculture expert volunteers.

Fig. 6. Some images of insect samples in the Insect10 dataset.

TABLE III. THE NUMBER OF IMAGES IN THE INSECT10 DATASETS WITH 10 INSECT SPECIES

No	Insect name	Train	Validation	Test
1	Acalymma_vittatum	116	33	17
2	Achatina_fulica	258	74	37
3	Alticini	193	55	28
4	Asparagus_beetles	89	25	13
5	Aulacophora_similis	113	32	16
6	Cerotoma_trifurcata	86	25	12
7	Dermaptera	111	32	16
8	Leptinotarsa_decemlineata	234	67	33
9	Mantodea	185	53	26
10	Squash_bug	249	71	36
	Total	1634	467	234

In this study, the proposed technique was tested using large-scale insect image datasets. Because of the numerous lifecycle phases of various insect species, it is difficult to compile a comprehensive insect pest image collection. As a consequence, the publicly available IP102 dataset [27] was utilized for system evaluation. This collection comprises more than 75,000 photos covering 102 agricultural insect issue categories. The IP102 collection comprises 75,222 images and 102 insect pest categories, with the lowest category including just 71 samples. The dataset consists of 18,983 annotated images intended for object identification tasks. Following the methodology outlined in [27], the images with bounding box annotations were divided into training and testing sets, containing 15,178 and 3,798 images, respectively. It is worth noting that some images were not utilized in this process. Fig. 7 showcases a variety of sample photographs from the IP102 dataset.

Models generally perform better with bigger datasets, but gathering a significant amount of data for training can be difficult. As a result, problems with insufficient data frequently develop in data analysis. Increasing the number of training samples aids in reducing overfitting and enhancing model generalizability.

To solve the issue of limited data and probable overfitting, data augmentation techniques are used. Geometric transformation is a useful strategy for increasing the model's resilience. In this study, online augmentation is utilized, employing a variety of geometric modifications such as rotation, horizontal flipping, color correction, blurring, and saturation adjustments. As a result, each original image is transformed into 12 augmented images. Fig. 8 illustrates the application of these diverse data augmentation techniques to insect pest photographs.

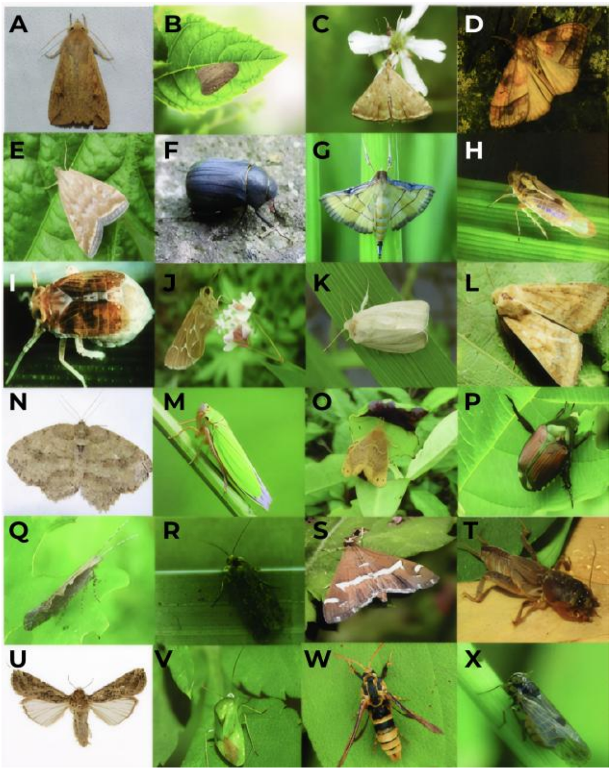


Fig. 7. Some images of insect samples in the IP102 dataset.

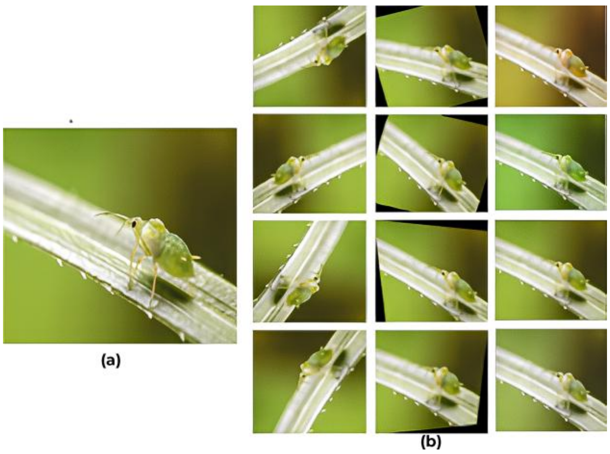


Fig. 8. The augmentation is accomplished by adding horizontal and vertical shifts, rotation, horizontal flipping, color, blur, and saturation. (a) Original picture, (b) Augmented picture.

IV. Result and Discussion

A. Experimental Setup and Training

All YOLO model training tests were carried out on Google Colab, utilizing a Tesla K80 24 GB GPU. The algorithms were written in Python and Keras. The experimental setup for training the models contained the following parameters: a learning rate of 0.01, an image size of 640 pixels, a batch size of 16, and 150 epochs for YOLOv5 and YOLOX, whereas YOLOv4 was trained for 2,000 epochs. The training epochs for YOLOv4 and YOLOv5 differed because of differences in the models' architectures and optimization efficiency levels. Being a more recent and lightweight model than YOLOv4, YOLOv5 converges faster and requires fewer epochs to achieve optimal performance. Additionally, the training process for YOLOv5 was conducted with hardware constraints, necessitating a trade-off between computer efficiency and model correctness. The choice of training epochs was made pragmatically to ensure a trade-off between detection performance and training length. The optimization approach was based on stochastic gradient descent [28]. Mobile device testing was carried out utilizing low-configuration devices, as shown in Table IV.

TABLE IV. Configuration of Smartphone Devices and the Environment for Developing Applications

Item	Description
Smartphone hardware configuration	The Samsung Galaxy A30 is equipped with a Samsung Exynos 7 Octa 7904 processor, featuring 8 cores and a clock speed of MHz. This powerful processor, combined with 3,000 MB of RAM, ensures smooth performance even with complex applications or games. The phone supports microSDXC memory cards for additional storage. It features a 15.93-megapixel rear camera and a 15.93-megapixel front camera, providing high-quality photos and videos with an excellent camera interface. The device boasts a 6.4-inch SUPER AMOLED display, offering good display quality with a balanced gradation of warm and cool colors. The operating system is Android 10.
Programinng language to build applications	Programing language: Java, Development Environment: Android Studio
The light	Normal luster intensity

B. Evaluation Metrics

Object detection models are evaluated using several key metrics that measure their accuracy, efficiency, and robustness. These metrics help assess how well the model detects and localizes objects within images.

1) Confusion matrix components

In object detection, predictions are classified based on how they match the ground truth:

True Positives (TP): The model correctly detects an object, and its predicted bounding box has sufficient overlap with the ground truth.

False Positives (FP): The model incorrectly detects an object where none exist, or the predicted bounding box does not sufficiently overlap with the ground truth.

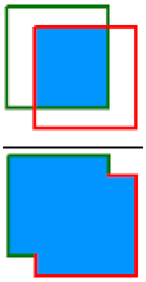
False Negatives (FN): The model fails to detect an object that is present in the ground truth.

True negatives (TN) are typically not used in object detection because the number of background pixels vastly outweighs object pixels.

2) Intersection over Union

Intersection over Union (IoU) measures the overlap between the predicted bounding box and the ground truth bounding box. It is calculated as:

$$IOU = \frac{area\ of\ overlap}{area\ of\ union} =$$



A prediction is considered correct if IoU exceeds a predefined threshold (e.g., 0.5). Fig. 9 shows automated detection of Leptinotarsa decemlineata with bounding boxes and confidence score.

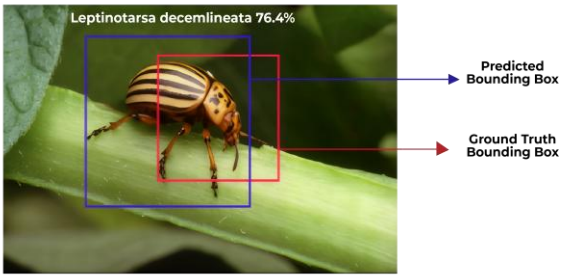


Fig. 9. Automated detection of Leptinotarsa decemlineata with confidence score.

3) Precision and Recall

These metrics evaluate how well the model detects objects:

Precision: Measures the proportion of predicted bounding boxes that are correct:

$$Precision = \frac{TP}{TP + FP}$$

Recall: Measures how many actual objects were correctly detected:

$$Recall = \frac{TP}{TP + FN}$$

4) Mean Average Precision (mAP)

Mean Average Precision (mAP) is the primary metric used to evaluate YOLO models. It calculates the Average Precision for each class and averages them.

Average Precision (AP): Computed as the area under the Precision-Recall (PR) curve:

$$AP = \sum_{k=0}^{k=n-1} [Recall(k) - Recall(k+1)] \times Precision(k)$$

Mean Average Precision (mAP): The mean of AP across all object classes:

$$mAP = \frac{1}{n} \sum_{k=1}^{k=n} AP_k$$

$$AP_k = the\ AP\ of\ class\ k, and\ n = the\ number\ of\ classes$$

$$n = number\ of\ thresholds$$

Common mAP Variants:

mAP@0.5: The AP is calculated using an IoU threshold of 0.5 (i.e., the prediction is correct if IoU > 0.5).

mAP@0.5:0.95: The AP is averaged over multiple IoU thresholds from 0.5 to 0.95 in increments of 0.05, providing a more comprehensive evaluation.

C. *Experimental Results and Discussion*

The experiment was designed to compare the performance of several model variations depending on their backbone topologies, input picture sizes, and measures such as mAP@IoU:0.5 and mAP@IoU:0.5:0.95. Table V summarizes the results for the Insect10 dataset, which are represented in Fig. 10.

TABLE V. SIMULATION RESULTS FOR THE YOLOV4, YOLOV5, AND YOLOX MODELS USING THE INSECT10 DATASET

Models	Backbone	mAPval @0.5	mAPval @0.5:0.95	Time (s)
YOLOv4	CSPDarknet53	84.9	63.2	2.8
YOLOv4-tiny	CSPDarknet53	64.4	48.3	0.3
YOLOv5-S	Darknet-53	70.5	35.9	0.6
YOLOv5-M	Modified CSP v5	76.6	42.7	1.6
YOLOv5-L	Modified CSP v5	78.9	46.8	2.3
YOLOv5-X	Modified CSP v5	73.0	40.9	3.8
YOLOX-S	Darknet-53	84.8	58.5	0.3
YOLOX-M	Modified CSP v5	82.3	61.9	1.3
YOLOX-L	Modified CSP v5	84.0	65.0	2.3
YOLOX-X	Modified CSP v5	83.0	64.0	3.2

According to the results in Fig. 10, the new mobile application has a relatively good success rate in terms of accuracy, recall, and mAP for pest object detection. For example, the Alcalymma insect has the lowest detection accuracy (mAP@IoU:0.5 of 0.45), but the Leptinotarsa insect has the highest detection capability (mAP@IoU:0.5 of 0.979).

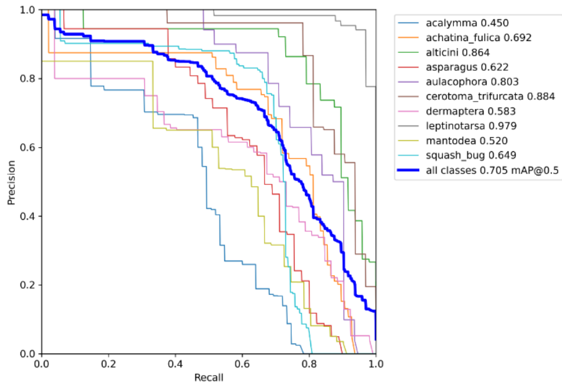


Fig. 10. Precision and recall of insect recognition findings on the Insect10 dataset using the YOLOv5-S model.

The YOLOv5-S model, utilized in our application and trained on the Insect10 dataset containing ten insect species, demonstrates superior performance compared to other object detection algorithms. It was trained with a batch size of 16 and an input image size of 640 pixels, consuming approximately 3.9 GB of GPU memory, making it an optimal choice for deployment on devices with limited computational resources. It enables quicker recognition speeds and near-real-time item identification.

Fig. 11 shows several effective examples of insect recognition on mobile devices using the Insect 10 dataset.

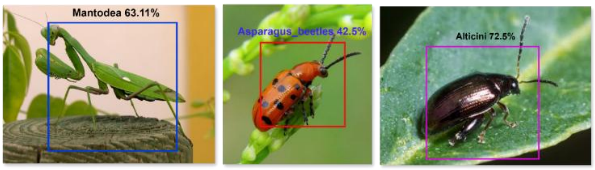


Fig. 11. Some photos were successfully recognized on mobile devices using the Insect10 dataset.

Our technique was also tested on the large-scale IP102 dataset [27] to determine its scalability. Table VI shows that the system obtained a promising mAPval@0.5 accuracy of 42.9% with the YOLOv5-S model. This outperforms numerous earlier approaches published in [27]. However, finding insect items on the IP102 dataset is still difficult due to concerns such as similar color appearances of pests and backgrounds, as well as variations in insect shape and picture blurriness, as seen in Fig. 12.

TABLE VI. SIMULATION RESULTS FOR YOLOV4, YOLOV5, AND YOLOX MODELS ON THE IP102 DATASETS

Models	Backbone	mAP @0.5	mAP @0.5:0.95	Time (s)
YOLOv4	CSPDarknet53	39.2	20.1	29.6
YOLOv4-tiny	CSPDarknet53	36.1	19.0	5.6
YOLOv5-S	Darknet-53	42.9	24.0	10.2
YOLOv5-M	Modified CSP v5	47.4	27.9	18.3
YOLOv5-L	Modified CSP v5	50.1	29.9	27.8
YOLOv5-X	Modified CSP v5	54.0	32.5	40.2
YOLOX-S	Darknet-53	52.3	34.1	9.8
YOLOX-M	Modified CSP v5	54.2	35.1	19.4
YOLOX-L	Modified CSP v5	53.9	34.7	28.5
YOLOX-X	Modified CSP v5	54.1	34.9	41.0

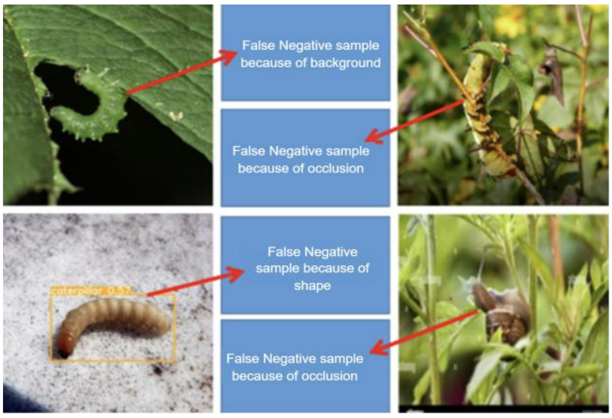


Fig. 12. Common difficulties with our object detecting system.

Our findings indicate that a number of variables contribute to the reduced accuracy, including comparable insect looks, varied life phases (e.g., larvae vs. adults), complicated backdrops, and picture quality difficulties. The IP102 dataset, in particular, poses issues due to class imbalance and overlapping characteristics amongst insect species, resulting in greater misclassification rates. Data augmentation strategies to improve model resilience, fine-tuning the model with extra training samples, and using post-processing methods like ensemble learning or attention mechanisms to refine classification accuracy are all possible enhancements. In addition, higher-resolution picture inputs are wanted to be used, and bounding box annotations are to be modified to increase detection performance in real-world agricultural contexts.

Fig. 13 illustrates effective insect recognition on a mobile device using the IP102 dataset. Despite YOLOX-S having greater accuracy (mAPval@0.5 of 52.3%) and a faster execution time (9.8 ms) than YOLOv5-S, we picked YOLOv5-S for our implementation. YOLOv5-S provides a fair trade-off between performance and computational economy, making it ideal for use on mobile devices with limited hardware. While YOLOX-S provides higher accuracy and quicker execution, YOLOv5-S offers more seamless integration and operation on low-cost mobile systems.

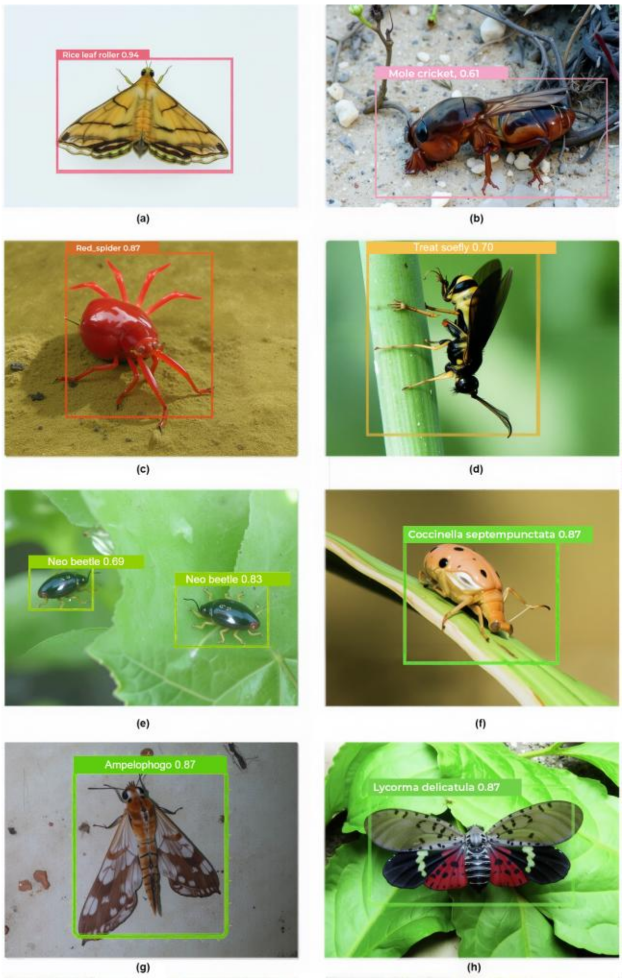


Fig. 13. Some photos were successfully recognized on mobile devices using the IP102 dataset.

Furthermore, as shown in Fig. 14, our approach combines pest classification and pesticide recommendations to give experts and farmers with practical assistance. Future implementations will incorporate additional devices like as the NVIDIA Jetson Nano Developer Kit [29], which provide more hardware combinations, cheaper prices, smaller footprints, and more durability, hence improving the system's efficacy and accessibility.

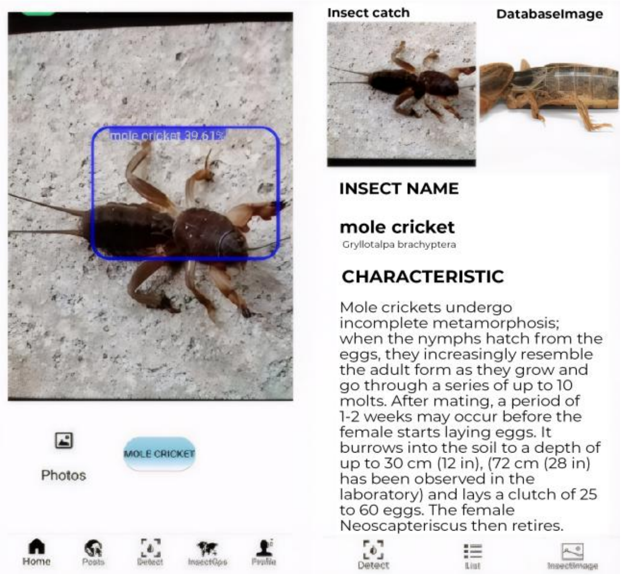


Fig. 14. The user interface panel displays successful insect detection and extensive insect information on a mobile device.

Table VII compares the proposed YOLOv5-S-based insect detection system to other detection systems, highlighting key advantages such as low hardware requirements, real-time capabilities, and offline capability. Unlike cloud-based solutions that require internet connectivity, this technology ensures accessibility in remote agricultural areas, making it more valuable to farmers. Furthermore, the table demonstrates that, while traditional image processing techniques have limited accuracy and scalability, and deep learning models like YOLOv4 and YOLOX require more computational resources, the proposed system strikes a balance between efficiency and feasibility for mobile deployment. However, despite these advantages, the table does not address the model's lower accuracy on large-scale datasets like IP102, nor does it include direct performance comparisons with other YOLO-based systems on the same dataset. Its relative performance should be more accurately assessed in future research by doing more thorough testing versus state-of-the-art models using the same datasets.

TABLE VII. DETAILED COMPARISONS WITH OTHER DETECTION SYSTEMS TO HIGHLIGHT UNIQUE ADVANTAGES

Feature	Proposed System (YOLOv5-S on Mobile)	Traditional Image Processing [8]	Cloud-based Detection [12]	Other DL-based Systems (YOLOv4, YOLOX) [15, 25]
Hardware Requirement	Low (runs on mobile devices)	Low to moderate	High (cloud-based processing)	Moderate to high (requires GPUs for inference)
Real-time Capability	Yes	Limited (due to processing constraints)	No (dependent on internet and cloud processing)	Partially (varies with model and hardware)
Accuracy (mAP@0.5)	70.5% (Insect10), 42.9% (IP102)	50–60%	99% (limited to 5 species)	YOLOv4: 84.9%, YOLOX-S: 52.3%
Processing Speed	Fast (~10.2 ms/image)	Slow (due to complex image processing steps)	Depends on cloud latency	Faster for YOLOX-S (9.8 ms/image), but higher latency in other models
Model Complexity	Lightweight YOLOv5-S (7.2M params)	Simple image processing techniques	Requires complex cloud-based inference	Larger YOLO models require more computing power
Deployment Feasibility	Highly feasible for mobile devices	Feasible for basic applications	Requires network connectivity	Suitable for high-end embedded systems
Cost of Implementation	Low	Low to medium	High (requires cloud subscription)	Medium to high (requires dedicated hardware)
Offline Functionality	Yes (SQLite database for local insect info)	Yes	No (requires cloud access)	Limited (depends on model size and hardware)
Scalability	High (adaptable to various datasets)	Low	High (if cloud resources are available)	Moderate to high
Additional Features	GPS-based insect distribution tracking, pesticide recommendations	Basic counting and segmentation	Cloud storage & processing	Some models integrate geolocation but lack real-time adaptation

The incorporation of insect GPS position and density information into the system will be beneficial to Integrated Pest Management (IPM) systems. Real-time distribution density maps, as illustrated in Fig. 15, allow users to efficiently monitor and anticipate insect infestation trends across wide areas. This skill will assist in assessing the effect of insect pests on agriculture and ecosystems, allowing for more informed decision-making and control tactics.

Although our study findings show tremendous promise for real-world applications, the system's accuracy requires additional work to improve its precision. Several variables contribute to the suggested model's low performance on the IP102 dataset:

Lifecycle Variability: Insects go through various lifecycle stages (egg, larva, pupa, adult), causing significant changes in their appearance, which makes consistent detection and classification challenging.

Diverse Backgrounds: The images in the IP102 dataset contain diverse backgrounds, often causing confusion for the model as insects blend into their natural surroundings.

Image Quality: Some photographs are of poor quality or grainy, limiting the model's capacity to recognize and categorize insects, underlining the need of clear, high-resolution images.

Morphological Similarities: Different insect species might have similar forms and sizes, which complicates the model's ability to discern between them.

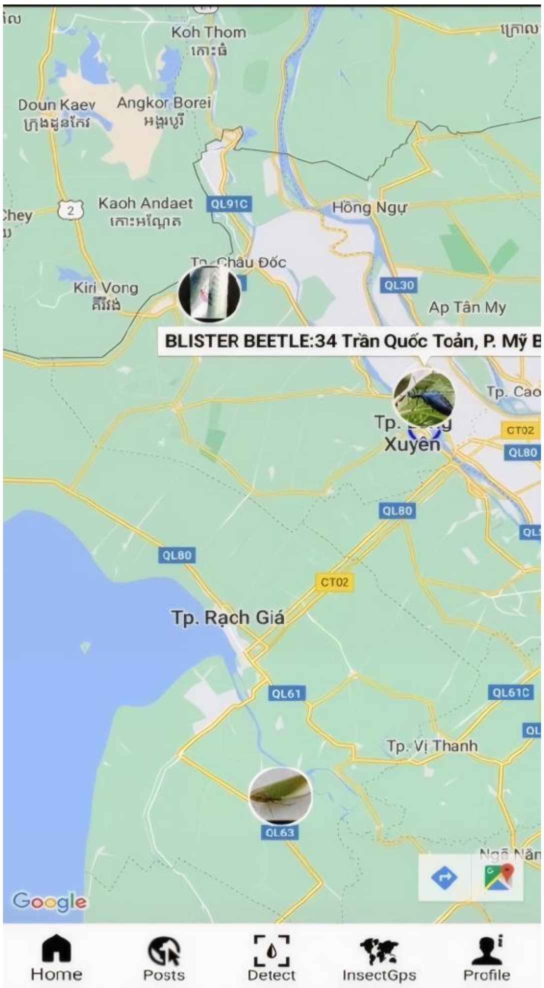


Fig. 15. The user interface panel displays successful insect detection and extensive insect information on a mobile device.

Dataset Imbalance: Despite the IP102 dataset's vast size, the number of photos per bug category may still be inadequate for training a robust model, particularly in categories with few observations, resulting in overfitting and poor generalization. As a result, our future research will focus on building more efficient identification algorithms in order to enhance accuracy and increase the number of recognized insects. Furthermore, improved mobile devices with increased CPUs, GPUs, and cameras will be investigated, allowing the installation of larger convolutional neural models.

V. Conclusion Limitation and Furture Research

This paper describes an effective real-time insect identification system that leverages mobile smart devices. It is based on the YOLOv5-S model, which has a compact design and is suited for devices with limited hardware resources. The system detects and classifies insect pests, and its benefits include real-time identification, cheap cost, ease of development, and practical implementation. Numerical experiments demonstrate that the system achieved a classification accuracy of 70.5% with mAP@0.5 on the Insect10 dataset and 42.9% on the large-scale IP102 dataset, the best reported accuracy for YOLOv5-S on that dataset. Despite these advances, mAP accuracy remains lower than the ideal threshold required for effective insect identification in agricultural settings. Future study will aim to improve the model's accuracy, particularly on large-scale datasets such as IP102. Furthermore, difficulties such as lifetime unpredictability, various backgrounds, and image quality will be addressed.

Conflict of Interest

The author declares no conflict of interest.

Funding

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

Acknowledgment

The author would like to thank the technical staff and agricultural experts from An Giang University and Vietnam National University in Ho Chi Minh City, Vietnam.

References

[1] S. Skendžić, M. Zovko, I. P. Živković, V. Lešić, and D. Lemić, "The impact of climate change on agricultural insect pests," *Insects*, vol. 12, no. 5, 2021.
[2] New standards to curb the global spread of plant pests and diseases. (2020). Food and Agriculture Organization of the United Nations (FAO). [Online]. Available: http://www.fao.org/news/story/en/item/1187738/icode/
[3] J. Pretty and Z. P. Bharucha, "Integrated pest management for sustainable intensification of agriculture in Asia and Africa," *Insects*, vol. 6, no. 1, pp. 152–182, 2015. doi: 10.3390/insects6010152
[4] D. Goulson, "An overview of the environmental risks posed by neonicotinoid insecticides," *J. Appl. Ecol.*, vol. 50, no. 4, pp. 977–987, 2013. doi: 10.1111/1365-2664.12111
[5] G. Niedbała, S. Kujawa, M. Piekutowska, and T. Wojciechowski, "Exploring digital innovations in agriculture: A pathway to sustainable food production and resource management," *Agric*., vol. 14, no. 9, pp. 2–5, 2024. doi: 10.3390/agriculture14091630
[6] K. Wang, Z. Shuifa, Z. Wang, Z. Liu, and F. Yang, "Mobile smart device-based vegetable disease and insect pest recognition method," *Intell. Autom. Soft Comput*., vol. 19, 2013. doi: 10.1080/10798587.2013.823783
[7] H. Nasir, A. N. Aris, A. Lajis, K. Kadir, and S. I. Safie, "Development of Android application for pest infestation early warning system," in *Proc. 2018 IEEE 5th International Conference on Smart Instrumentation, Measurement and Application (ICSIMA)*, 2018, pp. 1–5. doi: 10.1109/ICSIMA.2018.8688774
[8] C. Zhu, J. Wang, H. Liu, and H. Mi, "Insect identification and counting in stored grain: Image processing approach and application embedded in smartphones," *Mob. Inf. Syst.*, 2018. doi: 10.1155/2018/5491706
[9] P. Chudzik *et al*., "Mobile real-time grasshopper detection and data aggregation framework," *Sci. Rep*., vol. 10, no. 1, 1150, 2020. doi: 10.1038/s41598-020-57674-8
[10] J. W. Chen, W. J. Lin, H. J. Cheng, C. L. Hung, C. Y. Lin, and S. P. Chen, "A smartphone-based application for scale pest detection using multiple-object detection methods," *Electron*., vol. 10, no. 4, pp. 1–14, 2021. doi: 10.3390/electronics10040372
[11] S. A. L. Perera, *Pest Detecting Mobile Information System*, 2021.
[12] M. E. Karar, F. Alsunaydi, S. Albusaymi, and S. Alotaibi, "A new mobile application of agricultural pests recognition using deep learning in cloud computing system," *Alexandria Eng. J*., vol. 60, no. 5, pp. 4423–4432, 2021. doi: 10.1016/j.aej.2021.03.009
[13] P. Faria, T. Nogueira, A. Ferreira, C. Carlos, and L. Rosado, "AI-powered mobile image acquisition of vineyard insect traps with

automatic quality and adequacy assessment," *Agronomy*, vol. 11, no. 4, 2021. doi: 10.3390/agronomy11040731

[14] 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," *Ecol. Inform.*, vol. 66, 101460, 2021. doi: 10.1016/j.ecoinf.2021.101460

[15] A. D. Le, D. A. Pham, D. T. Pham, and H. B. Vo, "AlertTrap: {A} study on object detection in remote insects trap monitoring system using on-the-edge deep learning platform," arXiv preprint, arXiv:2112.13341, 2021.

[16] T. Doan, "Large-scale insect pest image classification," *J. Adv. Inf. Technol.*, vol. 14, no. 2, 2023. doi: 10.12720/jait.14.2.328-341

[17] J. Wu, "Power mean SVM for large scale visual classification," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 2344–2351. doi: 10.1109/CVPR.2012.6247946

[18] M. Tan and Q. V. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proc. the 36th Int. Conf. Mach. Learn., ICML 2019*, 2019, pp. 10691–10700.

[19] F. Pezoa, J. L. Reutter, F. Suarez, M. Ugarte, and D. Vrgoč, "Foundations of JSON schema," in *Proc. the 25th International Conference on World Wide Web*, 2016, pp. 263–273.

[20] R. D. Hipp. (2020). SQLite. [Online]. Available: https://www.sqlite.org/index.html.

[21] G. Jocher, "Ultralytics/YOLOV5: v3. 1-bug fixes and performance improvements," *Zenodo*, 2020. doi: 10.5281/zenodo.4154370

[22] C. Y. Wang, H. Y. Mark Liao, Y. H. Wu, P. Y. Chen, J. W. Hsieh, and I. H. Yeh, "CSPNet: A new backbone that can enhance learning capability of CNN," *IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Work.*, vol. 2020-June, pp. 1571–1580, 2020. doi: 10.1109/CVPRW50498.2020.00203

[23] R. Xu, H. Lin, K. Lu, L. Cao, and Y. Liu, "A forest fire detection system based on ensemble learning," *Forests*, vol. 12, no. 2, pp. 1–17, 2021. doi: 10.3390/f12020217

[24] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal speed and accuracy of object detection," arXiv preprint, arXiv:2004.10934, 2020.

[25] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, "{YOLOX:} Exceeding {YOLO} Series in 2021," arXiv preprint, arXiv:2107.0, 2021.

[26] Tzutalin. (2015). LabelImg. [Online]. Available: https://github.com/tzutalin/labelImg

[27] X. Wu, C. Zhan, Y. K. Lai, M. M. Cheng, and J. Yang, "IP102: A large-scale benchmark dataset for insect pest recognition," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2019, vol. 2019-June, pp. 8779–8788. doi: 10.1109/CVPR.2019.00899

[28] S. Ruder, "An overview of gradient descent optimization algorithms," arXiv preprint, arXiv:1609.04747, 2016.

[29] NVIDIA. (2021). Jetson Nano Developer Kit User Guide. [Online]. Available: https://developer.download.nvidia.com/embedded/L4T/r32-3-1_Release_v1.0/Jetson_Nano_Developer_Kit_User_Guide.pdf