Large-Scale Insect Detection with Fine-Tuning YOLOX

Thanh-Nghi Doan[1]

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

Abstracts: With the aim of detecting insect pests at an early stage, there has been an increasing demand for insect pest detection and classification, particularly in large-scale setups. Therefore, the aim of this research is to introduce a new real-time pest detection technique using a deep convolutional neural network, which not only offers improved accuracy but also faster speed and less computational effort. The networks were constructed using various modern object detector models such as YOLOv4, YOLOv5, and YOLOX. Our proposed networks were evaluated on a standard large-scale insect pest dataset, IP102, as well as on our collected dataset, Insect10. The experimental results demonstrate that our system surpasses previous methods and achieves satisfactory performance with 84.84% mAP on the Insect10 dataset and 54.19% mAP on the IP102 dataset. Our system can deliver precise and real-time pest detection and identification for agricultural crops, enabling highly accurate end-to-end pest detection that can be applied in realistic farming scenarios.

Keywords: Deep convolutional neural networks, Large-scale insect detection, Large-scale insect pest datasets

1. INTRODUCTION

Insects, pests, diseases, and weed populations impact an estimated 40% of agricultural products [1]. Pests damage between 20% and 40% of global production each year, according to the Food and Agriculture Organization (FAO) [2], [3]. Similarly, diseases of plants harm the economy by $220 billion, while insects harm the economy by $70 billion each year [4]. Climate change has increased the number of diseases and pests, causing them to appear in previously unknown places [5]. Caterpillars are major pests in agriculture, horticulture, and fruit production [6], [7], [8]. Currently, spraying pesticides is the most popular method of insect pest control due to its speed of action and its ability to bring high efficiency [9], [10], [11]. Every year, more than 2.36 billion kg of pesticides are used worldwide and more than 85% of them are in agricultural production [12]. However, if the pest control method is not suitable, it can reduce the yield by 70% and increase the production cost. Furthermore, prolonged use of these pesticides can cause environmental pollution as well as potentially dangerous diseases such as cancer, severe respiratory and hereditary infections, and fetal death [13]. Due to increasing environmental and health awareness, the use of pesticides is less and less. One of the most important ways to reduce pesticide use is to spray precisely where it is needed. The authors in [14] reported that spraying pesticides in the right places can reduce the cost of pesticides by up to 76% with different forward speeds respectively compared to the traditional method. Meanwhile, studies by [15], [16], [17] indicate that spot spraying pesticides can reduce the cost of spraying pesticides by 90%, which can reduce environmental contamination and restrain beneficial insects like honeybees. The location of the insect pest must first be identified before performing spot spraying. Manual methods are typically used to identify pests, which are labor-intensive and thus prone to error [18]. Fortunately, thanks to recent advances in computer vision in precision agriculture, detecting insect and diseases has become an essential part of collecting information about crop growth and health [3]. Furthermore, detecting objects at various phases of agricultural development is critical for predicting future yields, activating intelligent spraying systems, and controlling autonomous pesticide spraying robots for large farms and orchards. However, due to the resemblance of form, complexed backdrop, overlying of target objects due to dense dispersal, variability of light in the large topography of orchards, and numerous other variables, identifying target objects with acceptable precision is difficult. However, as technological advances, image processing techniques can be used to identify insect. Therefore, people became increasingly interested in precision agriculture to handle these challenges. To conduct out pest detection and spot spraying, visual information acquisition and processing via computer vision are unavoidable. Due to the ability to automatically extract image features and complex relationships, methods of pest detection by image processing, convolutional neural networks (CNN) and deep learning have been widely used to develop insect identification systems in practice [19]. CNNs have currently demonstrated their accuracy in object classification and are used in object recognition and detection algorithms. These object recognition algorithms are usually divided into two approaches: (i) two-stage object recognizers based on classifiers (two-stage detectors) and (ii) single-stage object recognizers based on regression operations. In particular, two-stage detectors have higher object detection accuracy than single-stage detectors, but they are slower in inference and recognition speed [20]. Therefore, many studies have used YOLO's single-stage detector approach [21] to develop pest identification systems [22] - [23]. However, there are fewer studies using YOLO to develop insect identification systems in a large-scale setup and evaluated on large-scale datasets likes IP102. Although deep learning algorithms have better generalizability and higher certainty than traditional algorithms [24], most insect objects are small in size, thus recognition systems will encounter problems for the following main reasons. First, image features are extracted with less useful information. Since small objects occupy fewer pixels in the image and carry less information, it makes it difficult to extract distinguishing features that are not affected by ambient factors. Second, the requirements for positioning accuracy are high. Whether during training or during prediction, the bias of the bounding box is quite large for a small target detection error. Third, the problem of object synthesis. As it happens, after being displayed on the deep feature map through multiple downsampling, the targets will be grouped into a single point, resulting in indistinguishability of the different objects. Besides, it will make the bounding boxes difficult to regress and the model difficult to converge. Therefore, this study aims to develop an efficient large-sacle insect object detection system for insect datasets with different species using red-green-blue (RGB) digital images/videos, this system can overcome the mentioned disadvantages of previous studies. Overview of our real-time insect recognition system using YOLO models is shown in **Error! Reference source not found.**. The main contributions of the paper include:

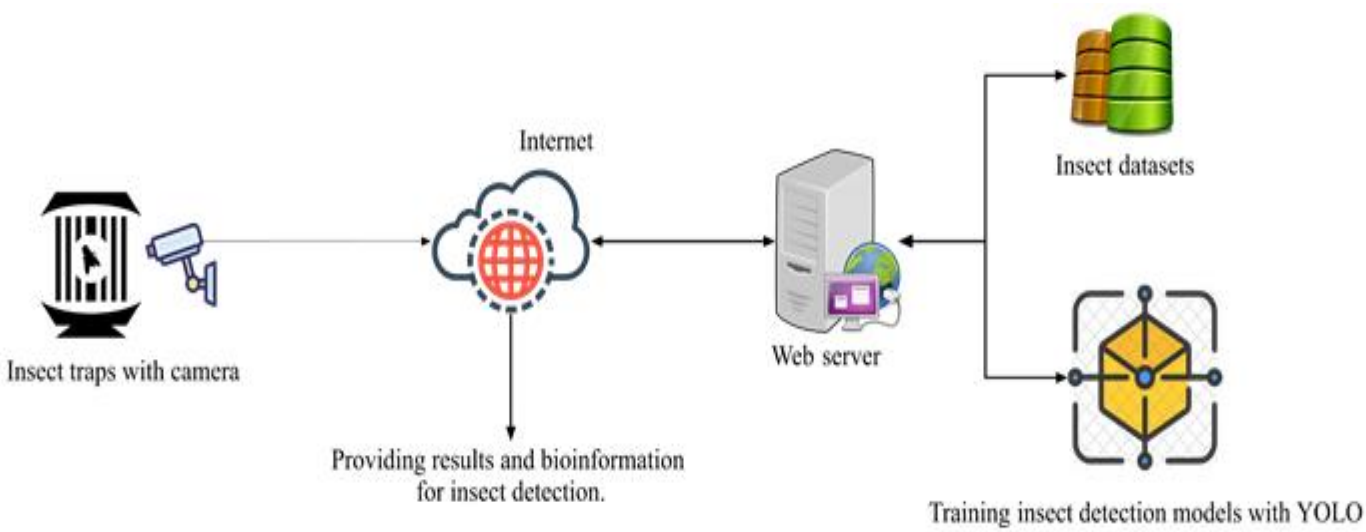

Figure 1. Overview of Real-Time Insect Recognition System using YOLO Models.

- Develop a large-scale object recognition system for detecting and classifying crop-damaging insect. The proposed method is based on fine tuning YOLO object detection architectures including YOLOX (Nano, Tiny, S, M, L), YOLOv5 (N, S, M, L, and X), YOLOv3, YOLOv4, SSD300, and RefineDet.

- Collect 2,335 images of 10 different insect species in the wild under different illumination and background conditions to train the underlying object detection approaches.

- Propose an insect pest detection system that works efficiently and correctly detects and identifies insect, and that can be used in a farming setting.

The rest of the paper is organized as follows: Section II articulates the related works, Section III presents the materials and methods, Section IV illustrates the results and discussion, and Section V concludes the study.

2. RELATED WORKS

In recent years, many researchers have used image processing and machine learning techniques to detect crop diseases [25]. Revathi et al. [26] presented Technological Strategies for categorizing illnesses using mobile-captured symptoms of Cotton Leaf Spot pictures. The classifier is being trained to accomplish intelligent farming, such as early disease detection in the trees, selective fungicide application, and so on. The suggested work is built on Image Edge detection Segmentation methods, and the recorded pictures are first handled for enrichment. Then, to obtain target areas (disease spots), R, G, B color feature picture segmentation is performed. Later, picture characteristics such as boundary, form, color, and texture are taken from disease patches in order to identify diseases and make pest management recommendations. Vinushree et al. [27] introduced a clustering method in which the famous clustering algorithm, kernel-based fuzzy c-means clustering algorithm (KFCM), is used to detect pest abundance in plants. A supervised neural network was used to classify leaf feature extraction. The methods investigated are for increasing productivity and decreasing subjective error caused by human specialists in spotting bugs in plants. Martin et al. [28] suggested an integrated pest management system that uses an image processing algorithm and expanded region-based growing to recognize the pest and quantify the pest to estimate the quantity of insecticide to be used. This extended area grow algorithm offers the finest pest detection and counting. Preetha Rajan et al. [22] suggested an image-processing-based automated pest detection system. The color feature is used for training the SVM to distinguish between pest and leaf images. Morphological procedures are used to eliminate undesireable components from the classified picture. Yogesh Kumar et al. [29] used a rapid feature recognition algorithm to develop an innovative and fast technique for detecting and enumerating pests in an image. The amount of pesticides used in farmland pollutes the environment; however, with advanced machine vision systems that apply this method, they can build machines that use pesticides efficiently by selectively targeting bugs using image processing. Human labor is currently used for physical pest identification, which is not very precise. Automation is needed in this area because it would be more effective in detecting insects in agriculture. To identify the pest, Apurva Sriwastwa et al. [30] used a color-based image segmentation technique. Extensive simulation findings on different pest images demonstrate that the suggested method outperforms Otsu's technique and edge detection segmentation. Vivek Agnihotri [31] recognized all of the pests that are present in the agricultural area and implemented specific steps to prevent them from destroying crops. Their method classifies pests in a field by using a microprocessor, infrared camera, and regular camera connected to a quadcopter that will hover over the field and identify the pest. The proposed system by Rajesh et al. [32] uses a decision tree to recognize and classify leaf disease and improves detection accuracy while taking less time than the current system. However, because conventional machine vision techniques are less robust in complex scenes, meeting the requirements of complex scenes is still very challenging for several object detection systems.

On the other hand, Convolutional neural networks (CNNs) have been effectively used in farming research to overcome the shortcomings of conventional techniques [33]. In the automated identification and classification of pest infestations, CNN models beat conventional methods [34]. Tu-Liang Lin et al. [35] used Faster R-CNN to build a knowledge base system that can detect plant pests as well as diseases autonomously. Li et al. [36] developed a real-time plant disease and pest identification system on video using faster R-CNN as an object detection framework. Gambhir et al. [37] designed a CNN-based dynamic android and web UI for agricultural pest and disease detection. The findings suggested that the proposed method could identify previously unseen rice diseases on video. In computer vision, real-time object detection is a very important task. YOLO is a popular family of real-time object detection algorithms. The initial YOLO object detector was published in 2016 [21]. This architecture is much quicker than other object detectors and has become the cutting-edge technology for real-time computer vision applications. YOLO is now extensively used in plant pest detection. Zhang et al. combined spatial pyramid pooling with YOLOv3 to accomplish inverse by merging upsampling and convolution processes. Convolution can successfully identify small-sized plant pest samples in images, with an average identification rate of 88.07% [22]. Zhong et al. [38] suggested a visual flying insect identification system on a Raspberry Pi using the YOLO design as a detector to determine the number of flying insects and classify them using an SVM model. They got 90.18% classification accuracy and 92.50% total accuracy. The authors of [39] suggested the YOLOv4_MF model, which uses MobileNetv2 as the feature extraction block and substitutes conventional convolution with depth-wise separated convolution to minimize model parameters. In addition, the coordinate attention method was incorporated in MobileNetv2 to improve feature information. A symmetric structure comprised of a three-layer spatial pyramid pool is given, and an improved feature fusion structure was developed to fuse the target information. To improve the network's learning of tiny objects, focal loss was used instead of cross-entropy loss in the loss function. The experimental findings revealed that the YOLOv4_MF model has 4.24% higher mAP, 4.37% higher precision, and 6.68% higher recall than the YOLOv4 model. Using the YOLOv5 object recognition framework, the authors of [15] developed a real-time video detection system to identify the thistle caterpillar (Vanessa cardui), which is found in Turkey and can cause harm to sunflower farming. Their findings show that the device is operational and capable of detecting the thistle larva at 65 frames per second. The authors of [40] proposed AgriPest-YOLO, a lightweight pest detection model for reaching a good equilibrium between efficiency, accuracy, and model size for pest detection. Their technique was tested on a large size multi pest picture collection with 24 pest groups and 25k images. Experiment results show that AgriPest-YOLO achieves end-to-end real-time pest detection with high accuracy, achieving 71.3% mAP on the test dataset, outperforming classical detection models while exhibiting better balanced performance in terms of model size, detection speed, and accuracy. The Pest-YOLO is a pest detector for multi-category thick and tiny bugs developed by the authors of [41]. To enhance the attention of hard samples, the concept of focal loss is first incorporated into the loss function via weight distribution. The confluence method is then used, which is a non-intersection over Union bounding box selection and reduction technique. To the greatest degree possible, the confluence approach can eradicate pest detection errors and failures caused by occlusion, adhesion, and unlabeling among minuscule dense pest individuals. The suggested Pest-YOLO model is validated using a large-scale pest image dataset, Pest24, which contains over 20k images of pests labeled by farming specialists and classified into 24 categories. Numerical results show that the Pest-YOLO can obtain 69.59% for mAP and 77.71% for mRecall on the 24-class pest dataset, which is 5.32% and 28.12% higher than the benchmark model YOLOv4. The authors of [42] built an object identification system for detecting and classifying crop-damaging insect. To decrease farms' dependence on pesticides, the current study suggests an automatic system in the shape of a smartphone IP-camera to identify insect from digital images and videos. The suggested method is built on YOLO object detection designs such as YOLOv5, YOLOv3, YOLO-Lite, and YOLOR. On their IP-23 dataset, the designed model YOLOv5-X outperforms the state-of-the-art model with an average accuracy value of (mAP@0.5) 98.3%, (mAP@0.5:0.95) value of 79.8%, precision of 94.5% and recall of 97.8%, and F1-score of 96%. The authors of [23] suggested Maize-YOLO, a novel high-precision and real-time technique for maize pest detection. The network is built on YOLOv7 and includes the CSPResNeXt-50 and VoVGSCSP modules. It can increase network detection precision and speed while decreasing the model's computational effort. Maize-YOLO's efficacy was tested on a standard large-scale pest dataset IP102 with only 13 classes. The experimental findings indicate that their approach outperforms the present state-of-the-art YOLO family of object detection algorithms, with 76.3% mAP and 77.3% recall. The authors of [43] suggested a YOLOX-based forest pest recognition method. To begin, they ultilize Mosaic, Mixup, and random erasure data enhancement to preprocess the pictures because there are few image data of actual deep forest insects in the field. Second, shallow information is incorporated into the current network design to extract fine-grained features, and a two-way cross-scale feature fusion method is used. Finally, on the public forest pest dataset IP102, the improved YOLOX algorithm suggested in this article produced the greatest results. Although previous studies have successfully applied the YOLO models in insect pest detection, the number of identified insect species is still small. There are currently a few research and development projects on the YOLO application for large-scale setup and real-time evaluation on large-scale datasets like IP102. Therefore, this paper will focus on studying real-time insect recognition systems using several prominent single-stage detectors object recognition architectures including YOLOv4, YOLOv5 and YOLOX. Our method will be evaluated on two data sets, the large-scale dataset IP102 and the insect dataset collected by us, Insect10. The different architecture versions of YOLOv5 and YOLOX will be evaluated in detail in the experiments, they include YOLOv5-N, YOLOv5-S, YOLOv5-M, YOLOv5-L, YOLOv5-X, YOLOX-Nano, YOLOX-Tiny, YOLOX-S, YOLOX-M, YOLOX-L. Shows how the CNN model works in YOLO models to detect insect pest object.

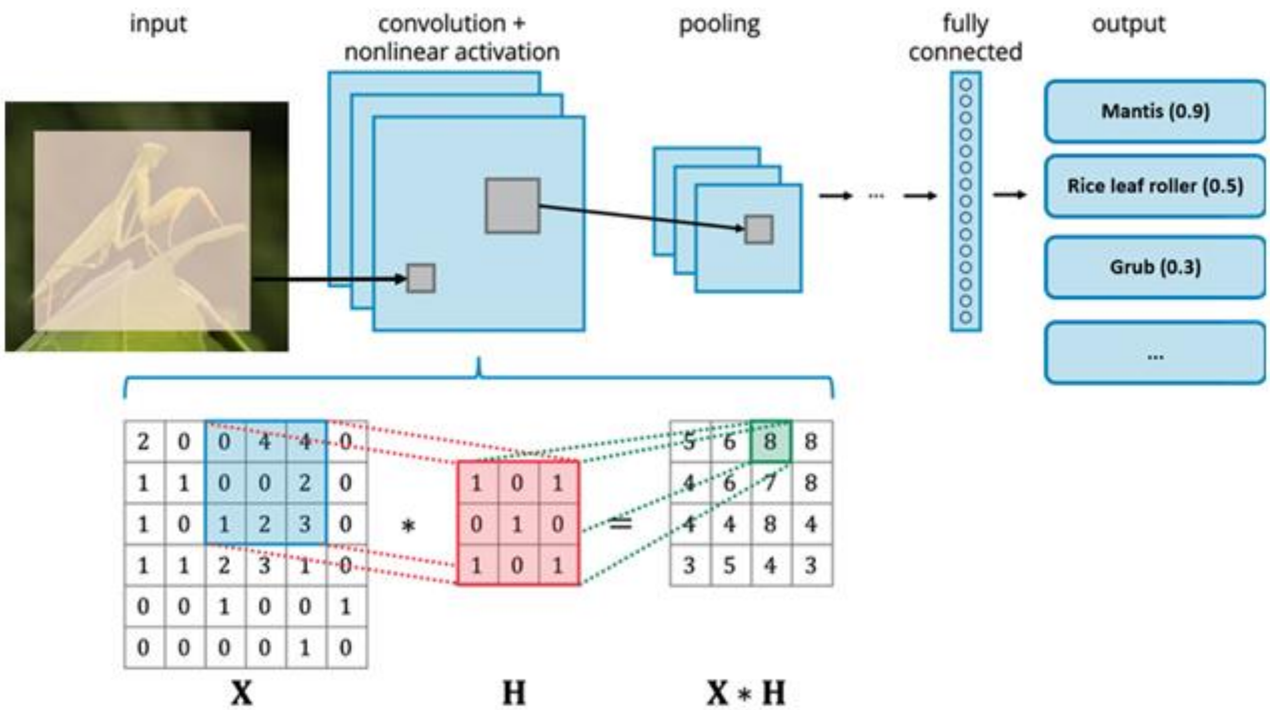

Figure 2. The Description of how the CNN Model Works in YOLO Models to Detect Insect Pest Object.

3. MATERIALS AND METHODS

3.1. Insect Image Datasets

3.1.1. Insect10

To evaluate the proposed insect pest identification system, the insect pest images are collected from the internet as a source to train CNN models. The primary goal of the study on insect was to recognize and discover the 10 insect classes. First, for the data gathering, we looked up images using various databases and search engines, including Kaggle, Google, Baidu, Iostock, Dream, Flickr, and Bing. As a result, we have accumulated more than 5,000 candidate images for the Insect10 dataset. Then, unsatisfactory images will be deleted as part of the data cleaning process. Finally, we obtained 2,335 insect images and divided them into 10 different classes, with the lowest class having 123 samples, as shown in **Error! Reference source not found.**. There are enough images in the training, validation, and testing sets for each class. For training the pest detection models, we split all datasets into a training set, a validation set and a testing set in a ratio of 7:2:1. The training, validation, and testing sets are split at the class level. Therefore, the Insect10 dataset was divided into 1,634 training images, 467 validation images, and 234 testing images for the detection task. Some representative samples from Insect10 dataset are presented in Figure 1.

Table 1. Details of the Collected Insect10 Images Dataset.

ID	Insect Species	# Training	# Testing	# Validation
1	Acalymma_vittatum	116	17	33
2	Achatina_fulica	258	37	74
3	Alticini	193	28	55
4	Asparagus_beetles	89	13	25
5	Aulacophora_similis	113	16	32
6	Cerotoma_trifurcata	86	12	25
7	Dermaptera	111	16	32
8	Leptinotarsa_decemlineata	234	33	67
9	Mantodea	185	26	53
10	Squash_bug	249	36	71
	Total	1634	234	467

Figure 1. Some Representative Samples from Insect10 Dataset.

3.1.2. IP102

IP102 [44] is a large-scale insect dataset with 102 species of plant pests. This dataset contains 75,222 insect images gathered from the Internet using popular search engines, with the English name and corresponding synonyms of each class used as keywords when searching for candidate images. For each keyword, only the top 2,000 results are retained. Then the authors of [44] conducted a search on various professional agriculture and insect science websites. Based on the crop affected by the insect pest, each insect pest is assigned an upper-level class (denoted as super-class in the following). In other words, each insect pest is a subordinate class (referred to as a sub-class in the following) of a larger class. As described in [44], this dataset contains several factors that influence and challenge the performance of classification and image recognition models when compared to other existing insect datasets such as [44], [45], [46], [47], [48], [49], [50], [51], and [52]. Firstly, the insect are difficult to identify due to their similar coloration and backgrounds. Second, each layer contains an image of the entire life cycle of an insect pest and is therefore difficult to classify, especially the larval stage. Third, many pests and diseases belong to different classes but have similar images. Fourth, this dataset is greatly imbalanced, the least prevalent class (Erythroneura apicalis) has only 71 images, while the most frequent class (Cicadellidae) has nearly 5,740 images. These are factors that cause many difficulties and challenges when designing image classification and recognition algorithms on this dataset. On the IP102 dataset, the current best classification accuracy is reported as 67.13% in [53], while the recognition accuracy is reported as 25.67% mAP [44]. There should be enough examples of each group on the testing set for more reliable IP102 results. As a result, they have an approximately 6:1:3 divide. The training, validation, and testing sets are divided by subclass. For the classification task, the IP102 is divided into 45,095 training images, 7,508 validation images, and 22,619 testing images. Furthermore, there are a total of 18,983 annotated images for the task of object detection. They divided the images with bounding box annotation into testing and training sets of 15,178 and 3,798 images, respectively. Some representative samples from the IP102 dataset are presented in Figure 4.

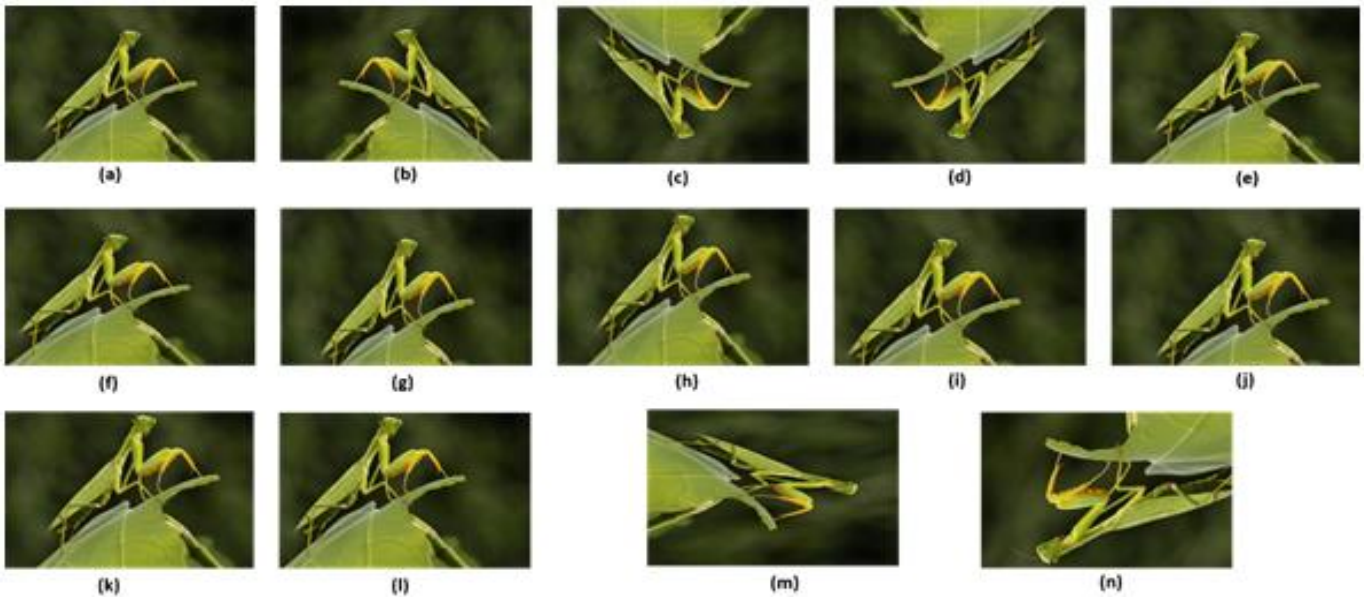

Figure 4. The augmentation is carried out by applying shifts in horizontal and vertical directions, rotation, horizontal flipping, hue, blur, and saturation.

Data augmentation: In general, more data can improve the performance of CNN algorithms. Collecting large amounts of data for training purposes, on the other hand, is a difficult job. As a result, the issue of inadequate data frequently arises in data analysis. Increasing the number of training examples can help the CNN model generalize data and avoid overfitting in training process. Fortunately, data augmentation methods can help address overfitting issues and improve the performance of CNN models. Geometric transformation is one of the most recent techniques of data augmentation [54], [55]. In this study, several geometric transformation techniques are utilized, including rotation, horizontal flipping, shifting, hue, blurring, and saturation. **Error! Reference source not found.** demonstrates the use of the data augmentation techniques on the insect pest images, with (a) the original image, and transformed images by (b) flipping horizontally, (c) flippingvertically, (d) rotating 180 °, (e) Shift 3 pixels to the right , (f) Shift 3 pixels to the left, (g) Shift 3 pixels down, (h) Shift 3 pixels up, (i) Shift 3 pixels to the left and down, (j) Shift 3 pixels to the right and down, (k) Shift 3 pixels right and up, (l) Shift 3 pixels left and up, (m) rotate 90^0, (n) rotate 270^0.

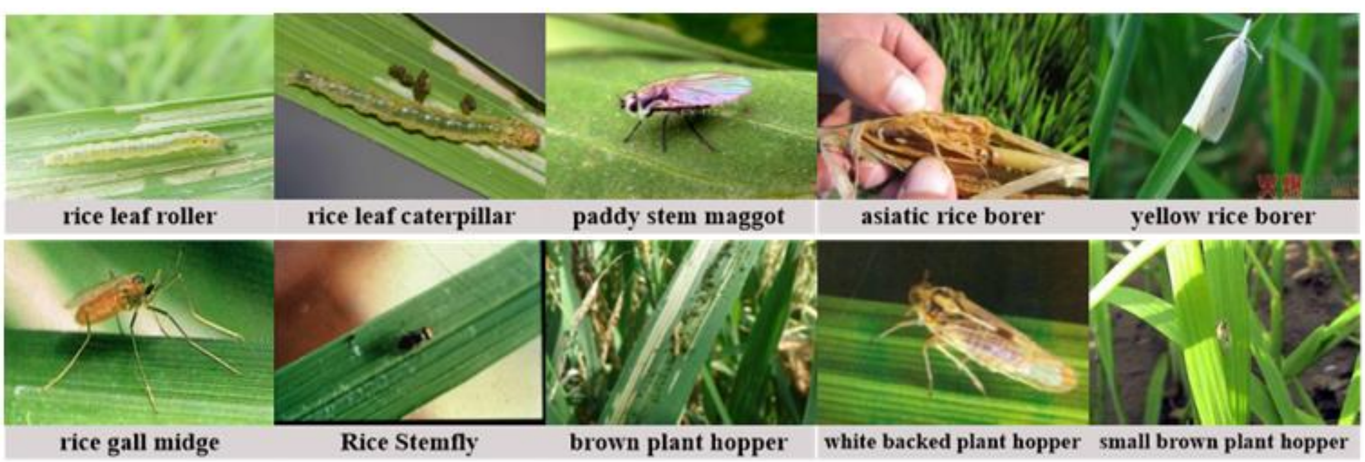

Figure 2. Some Representative Samples from IP102 Dataset [44].

3.2. YOLO Object Detection Models for Insect Pest Detectio N

3.2.1. YOLOv3

YOLOv3 [56] uses logistic regression to calculate the target detection score. It awards points to all targets within each bounding box. YOLOv3 can perform multilabel classification because it employs a logistic classifier for each class rather than the softmax class used in YOLOv2. YOLOv3 has a Darknet-53 backbone with 53 convolutional layers. These layers have more depth than the Darknet-19 used in YOLOv2. Darknet-53 primarily includes 3x3 and 1x1 filters, as well as skip links [57], [56]. The advantage of YOLOv3 over YOLOv2 is that a modified number is included in the error function and, for objects of significant size, occurs on three scales. The multi-glass problem has turned into a multi-label problem, and performance has been significantly improved when recognizing small-sized objects.

3.2.2. YOLOv4

Since the announcement of YOLOv3, no new version has been announced for about two years. Up to this point, EfficientDet [58] has achieved higher mAP performance than YOLOv3. EfficientDet uses EfficientNet [59] as the backbone for feature extraction and automatic neural network design, which makes it obsolete to develop models under human supervision. However, EfficientDet requires huge GPU power, making it difficult to win object detection research competitions, unless the participant is a large company. YOLOv4 [60] is an important advance in YOLO history, launching in April 2020. Based on YOLOv3, YOLOv4 can be trained and put to practical use with just one GPU, creating a faster model with the same accuracy as EfficientDet, created by researchers like Google who have huge computing power.

3.2.3. YOLOv5

YOLOv5 [61] was released by Utralytics, in October 2021, just a few weeks after the release of YOLOv4. YOLOv5 is a native extension of the PyTorch YOLOv3 implementation. YOLOv5 is a lightweight object detection architecture that is easy to modify, infers fast, and performs well. The YOLOv5 architecture has the following improvements: In the Input part, adaptive anchors using K-means can automatically compute anchor boxes that match the training dataset. All link boxes are automatically learned in YOLOv5 to custom data. The self-adjusting image scale adds black edges at least adaptively to the original image to speed up inference. For Backbone, Focus CSPDarknet53 is used, where the focusing mechanism combines higher resolution features with lower features by stacking into different channels rather than spatial locations. The focus mechanism is better for the model to learn the features of small subjects. For the Neck part, a cross stage partial (CSP) connection [62] is included in the FPN to shorten the feature extraction network for higher speed. For the Head part is the same as the YOLOv4 model. YOLOv5 is easier to use for a developer to implement object detection into an application than other object detection frameworks due to the following qualities: Positive pattern augmentation using an anchor pattern matching strategy adjacent positive; flexible variable configuration parameters, different levels of modeling can be obtained; improve overall performance through built-in hyperparameter optimization strategies. YOLOv5 currently has four different versions, which are labeled according to the model's size and complexity as small (S), medium (M), large (L), and extra large (X).

3.2.4. YOLOX

YOLOX [63], one of the most accurate detectors accessible, employs a more effective data enhancement method to pre-process the data. With some new advanced recognition methods, such as decoupled head, anchor-free, and advanced label giving strategy, YOLOX gets a better trade-off between speed and accuracy than competitors across all model sizes. YOLOX uses SimOTA as the label assignment strategy to to achieve state-of-the-art results across a large scale range of detection models. It is also an anchor-free frame-based detector that builds a high-performance detector by avoiding the issue of unbalanced positive and negative data with the anchor frame method. In terms of precision and efficiency, the use of decoupled heads for categorization and regression jobs outperforms other detectors. The general model architecture of YOLOX is shown in **Error! Reference source not found.**. The operating principle of insect identification using YOLO models is shown in

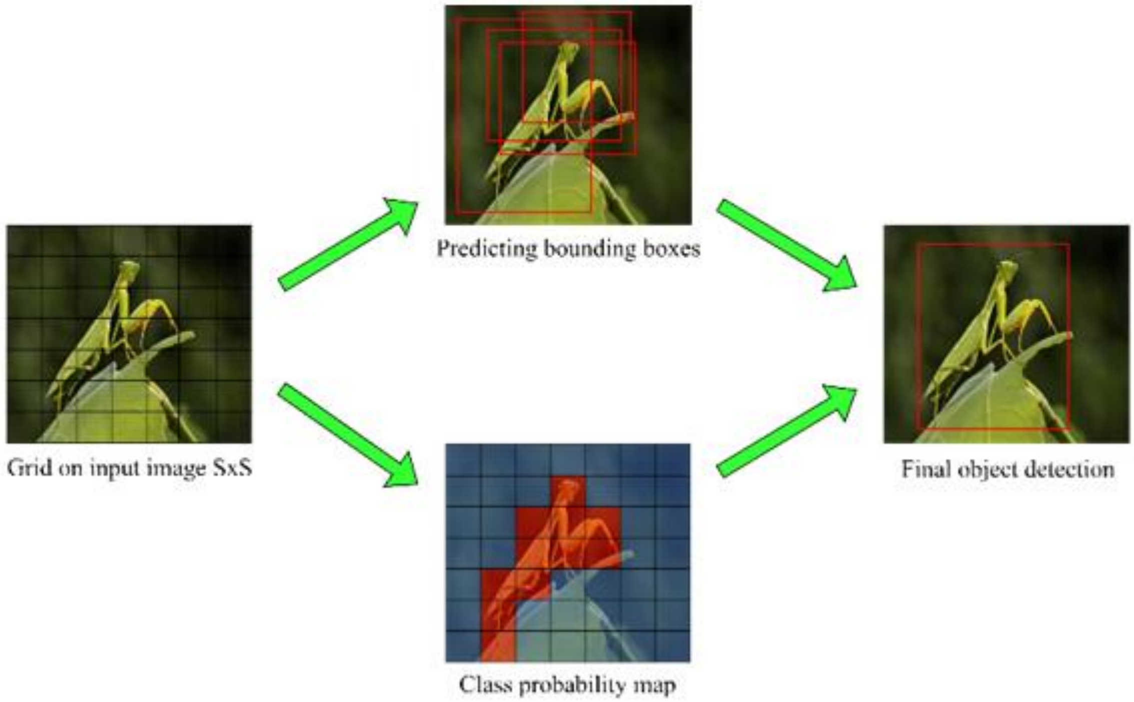
Predicting bounding boxes
Grid on input image SxS
Final object detection
Class probability map

Figure 3.

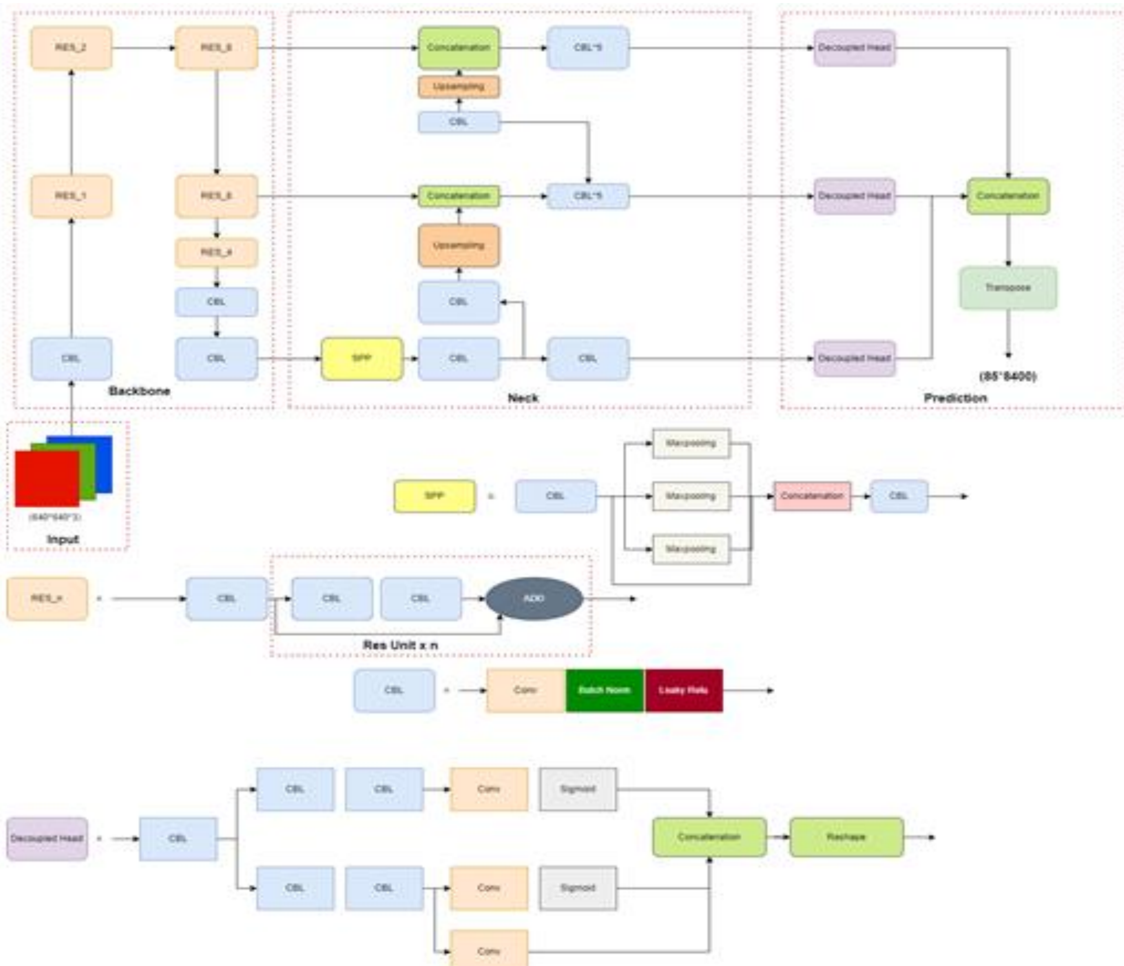

Figure 6. The General Architecture of YOLOX [63].

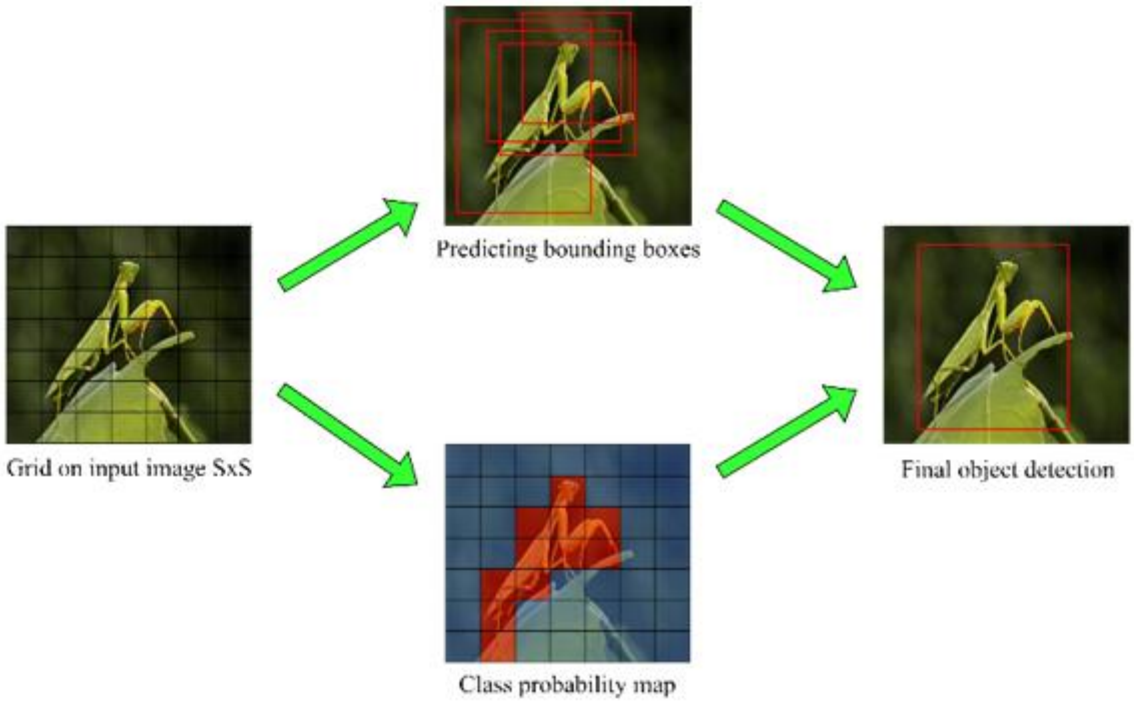

Figure 3. The Operating Principle of Insect Identification using YOLO Models.

3.3. Transfer Learning

Transfer learning [64] is a machine learning method where a model developed for a task is reused as the starting point for a model on a second task. It is a popular approach in deep learning where pre-trained models are used as the starting point on computer vision tasks given the vast compute and time resources required to develop neural network models on these problems and from the huge jumps in skill that they provide on related problems. Transfer learning is one of the most efficient methods when the training data is small. Transfer learning can help increase model accuracy while also shortening training time. This technique proved to be very useful in practical applications. Instead of training the entire model, transfer learning on the trained models is used, which uses fewer resources and cuts training time in half. Transfer learning employs the optimized parameters of a pre-trained network as well as the training of a few extra layers based on the requirements of an underlying task. In this study, pretrained weights trained on the ImageNet dataset [65] are utilised to continue training YOLO models to recognize insect objects.

3.4. The Process of Real-Time Insect Detection System

Our proposed system consists of five consecutive steps (Figure 4). In the first stage, the insect pest images are collected for CNN model training and evaluation. Second, the entire dataset is strictly preprocessed through annotation and augmentation to increase the number of samples in the insect datasets. Image data augmentation is a method for artificially increasing the size of a training dataset by modifying existing images slightly based on certain parameters. Third, the YOLO object detection models are trained on insect datasets. The detection performance of the fine-tuned models is evaluated by using the dataset split. Finally, the best YOLO model was chosen for a practical farming application.

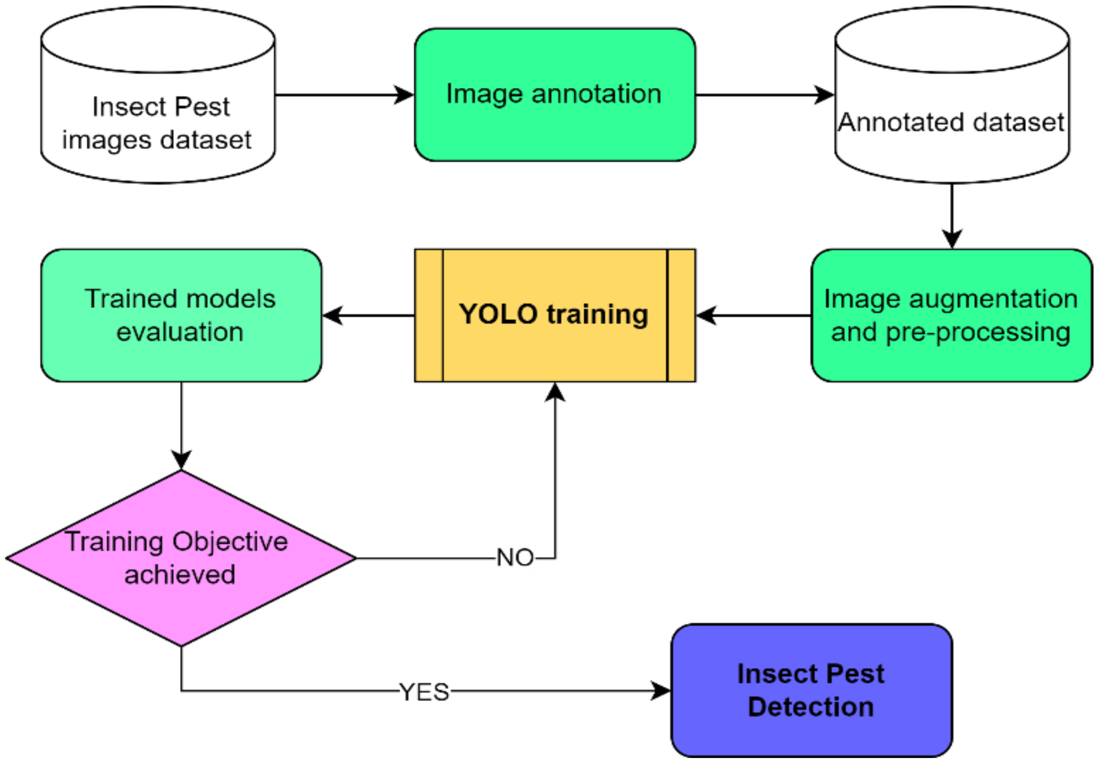

Figure 4. Schematic Flowchart of the Research Approach.

4. RESULTS AND DISCUSSION

4.1. Experimental Setup and Training

The experiments were conducted on a Tesla K80 12 GB (Gigabyte) GPU in the Google COLAB. For neural network training, on Insect10, 1,634 images were used to train each underlying model for object recognition. On IP102, 45,095 images were used to train large-scale object detectors. Regularization method was performed from the BN layer to update the weight of the model. For training the network, the learning rate was set at 0.01, the momentum factor was set at 0.937, the decay rate of the weights was set at 0.0005, the initial vector and IoU threshold were set at 0.01, and the gain coefficients of hue (H), saturation (S), and lightness (L) were set at 0.015, 0.7, and 0.4, respectively. Stochastic Gradient Descent (SGD) [66] was used as the optimization algorithm [67], and each model was created using Pytorch [68]. The final output of the detection model was the location bounding box of the target insect pest categories (the prediction box of the position), and the probability of a particular class. **Error! Reference source not found.** and **Error! Reference source not found.** show the detailed training strategies of each model for YOLOv5 and YOLOX respectively.

Table 2. The Detailed Training Strategies of Underlying Yolov5 Models.

Dataset	Model	Model Size	Batch-Size	Epochs	GPU Usage (GB)	Training Time (h)
Insect10	YOLOv5-N	1.8	16	150	4.98	0.82
	YOLOv5-S	7.2	16	150	7.11	1,23
	YOLOv5-M	21.2	16	150	7.45	2,11
	YOLOv5-L	46.5	16	150	8.93	3,62
	YOLOv5-X	86.7	8	150	10.22	6,37

IP102	YOLOv5-N	1.8	16	150	11.03	8.90
	YOLOv5-S	7.2	16	150	20.34	10.97
	YOLOv5-M	21.2	16	150	20.82	22.09
	YOLOv5-L	46.5	8	150	22.33	35.13
	YOLOv5-X	86.7	8	150	23.24	67.15

Table 3. The Detailed Training Strategies of Underlying YOLOX Models.

Dataset	Model	Model Size	Batch-Size	Epochs	GPU Usage (GB)	Training Time (h)
Insect10	YOLOX-Nano	0.91	16	150	4.11	3,77
	YOLOX-Tiny	5.06	16	150	4.09	3.67
	YOLOX-S	9.0	16	150	4.89	7,51
	YOLOX-M	25.3	16	150	5.09	10,88
	YOLOX-L	54.2	8	150	6.22	19.36
IP102	YOLOX-Nano	0.91	16	150	22.12	44,12
	YOLOX-Tiny	5.06	16	150	21.82	40,34
	YOLOX-S	9.0	16	150	23.34	80,49
	YOLOX-M	25.3	8	150	23.5	109.46
	YOLOX-L	54.2	4	150	23.6	184.95

4.2. Evaluation Metrics of the Trained Models

In order to evaluate the performance of YOLO models on insect datasets, several comprehensive metrics were employed for rigorous evaluation. Mean Average Precision (mAP) with Intersection over Union (IoU) in [.50:.05:.95], $mAP^{.50}$ and $mAP^{.75}$ is ultilized to evaluate insect pest detection performance of the models. In which, IoU is a number that quantifies the degree of overlap between two boxes (Figure 5). In the case of object detection and segmentation, IoU evaluates the overlap of the Ground Truth and Prediction region. mAP represents the average value of AP, which is used to measure the overall detection accuracy of object detection. For object detection, AP and mAP are the best indicators to measure the detection accuracy of the model [69]. Besides, Precision and Recall are also two major metrics employed in our insect datasets, which describe the false positive reduction and misdetection rate respectively. The formula is shown in Equations (1) - (3):

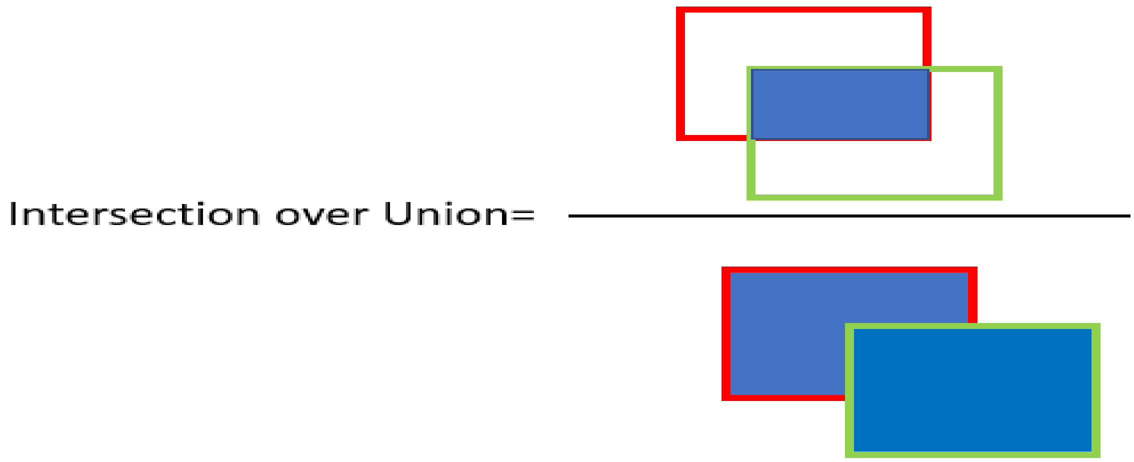

Figure 5. The Predicted Bounding Box is Drawn in Red While the Ground-Truth Bounding Box is Drawn in Green. Intersection Over Union (IOU) is the Intersection Over Union between These Bounding Box.

In the following formulas, *TP* is the number of true positives, the samples that are correctly identified as insect; *FN* is the number of false negatives, the samples that are incorrectly identified as the background; *TN* is the number of true negatives, the samples that are correctly identified as the background; and *FP* is the number of false positives, the samples that are incorrectly identified as insect. The precision *P* in formula (1) measured the classification ability of the model by calculating the ratio of the number of correctly detected targets to the overall number of detected targets. Recall *R* in formula (2) is a measure of the model's detection capability, which is obtained by calculating the ratio of the number of correctly detected targets to the total number oftargets. *AP* in formula (3) is the average precision, which measures the detection performance of a model by calculating the area under the Precision-Recall curve. The 10-class or 102-class mean of the above indicators was calculated, and the mean average precision (*mAP*), mean Precision (*mPrecision*), and mean Recall (*mRecall*) were obtained.

$$Precision\ (P) = \frac{TP}{TP + FP} \quad (1)$$

$$Recall\ (R) = \frac{TP}{TP + FN} \quad (2)$$

$$Average\ Precision\ (AP) = \frac{1}{\#classes} \sum_{k=1}^{\#classes} P \quad (3)$$

4.3. Results

This section presents the results when training YOLOv5, YOLOX models using transfer learning on the Insect10 and IP102 datasets. The images are selected at random from the test set to better illustrate the results of our experiments. The performances of YOLOv5 and YOLOX were verified by experiments. The comparison results of the models obtained in the ablation experiments are shown in
Table **1** and

Table **2.**

4.3.1. Training Results on Insect10

The training process of YOLOv4, YOLOv5, YOLOX models on the Insect10 dataset was evaluated through the mAP indicator with IoU thresholds of 0.5 and 0.5:0.95, mPrecission and mRecall. Figure 6 shows the training results of different YOLOX model architectures on the Insect10 dataset. It can be seen that there is not a significant difference between the models, and the average accuracy indicators increase with the size of the model variations because the dataset is quite small and has an even distribution between classes. However, the models have been unstable since the 20th loop and show signs of increasing in the last loops. As shown in

Table **1**, compared to the baseline model YOLOv4, the mAP of YOLOv5 and YOLOX do not show better performance. It is because the number of images and the classes of Insect10 are still small.

Table 1. Training Results of Yolo Models using Transfer Learning on Dataset Insect10.

Methods	**mAP^{test}**	**mAP^{test} 0.5**	**mAP^{test} 0.5:0.95**	**mPrecision**	**mRecall**
YOLOv4	84.94	84.90	63.19	81.00	80.00
YOLOv4-tiny	64.35	64.40	48.34	69.00	58.00
YOLOv5-N	70.10	70.10	40.20	77.00	64.70
YOLOv5-S	70.50	70.50	35.90	73.90	67.30
YOLOv5-M	76.60	76.60	42.70	78.70	73.00
YOLOv5-L	78.90	78.90	46.80	81.80	73.40
YOLOv5-X	73.00	73.00	40.90	73.60	71.00
YOLOX-Nano	77.43	77.43	51.89	-	-
YOLOX-Tiny	77.54	77.54	53.17	-	-
YOLOX-S	84.84	84.84	58.50	-	-
YOLOX-M	82.34	82.34	61.92	-	-
YOLOX-L	84.01	84.06	65.04	-	-

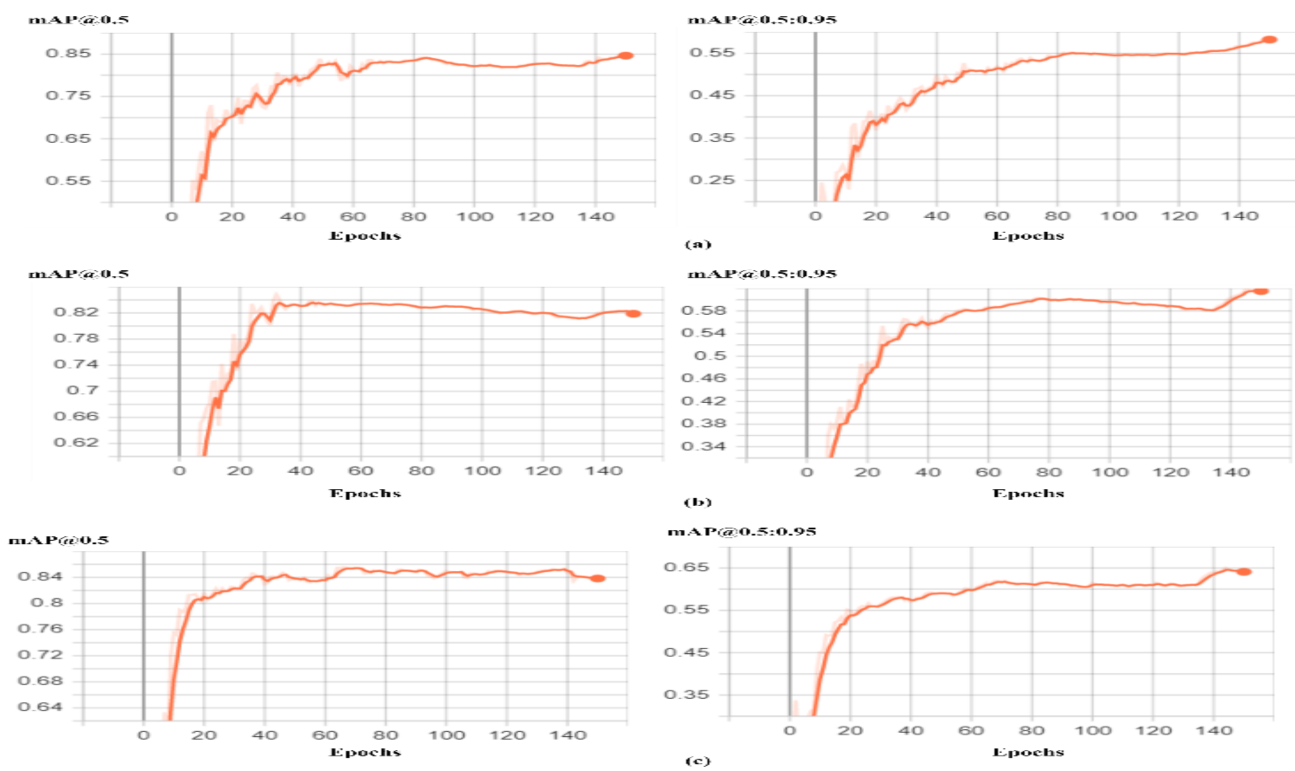

Figure 6. The Training Results Of Different YOLOX Model Architectures on Insect10 Dataset (mAP@IoU:0.5, mAP@IoU:0.5:0.95) with (a) YOLOX-S, (b) YOLOX-M, (c)YOLOX-L.

The results of the YOLOX-S model for detecting *Acalymma vittatum* with many individuals in a single image from the Insect10 dataset are shown in Figure 7.

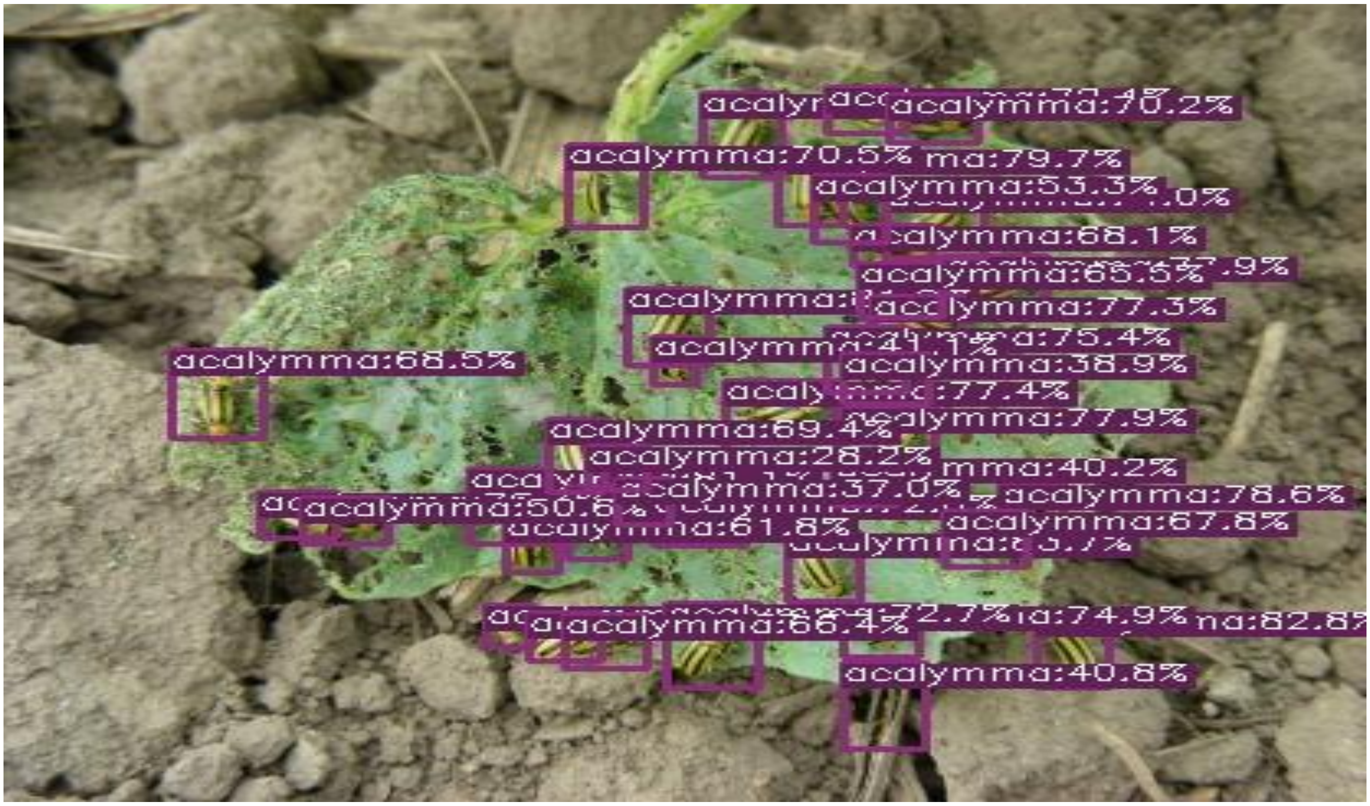

Figure 7. The results of the YOLOX-S Model for Detecting Acalymma Vittatum with Many Individuals in a Single Image from the Insect10 Dataset.

4.3.2. Training Results on IP102

The process of training different architectures of the YOLOX model on the IP102 dataset evaluated by the indicators mAP.05 and mAP.50:.95, is detailed in Figure 8. During the first 20 iterations, the mAP of YOLO models is gradually increasing and stable ; however, the YOLOX-M and YOLOX-L models have a decrease in accuracy in the last iterations. The mAP index increased sharply at loop 130 but not significantly, the proportion of mAP accuracy assessments of the model variants increased with the size of those variants. As shown in

Table **2**, compared to the baseline model YOLOv3, the mAP of YOLOv4, YOLOv5 and YOLOX have much better results than the YOLOv3 model in the article [44], 54.19% mAP versus 25.67% mAP, increased by 28.52%. Further, compared to the model YOLOv4, the mAP of the YOLOv5 and YOLOX achieve 54.01%, 54.19% respectively, an increased by 27.69%.

Table 2. The training Results of YOLOV5, YOLOX on ip102.

Methods	**mAP^{test}**	**mAP^{test} 0.5**	**mAP^{test} 0.5:0.95**	**mPrecision**	**mRecall**
FRCNN [70]	21.05	47.87	15.23	-	-
FPN [71]	28.10	54.93	23.30	-	-
SSD300 [72]	21.49	47.21	16.57	-	-
RefineDet [73]	22.84	49.01	16.82	-	-

Methods	**mAP^{test}**	**mAP^{test} 0.5**	**mAP^{test} 0.5:0.95**	**mPrecision**	**mRecall**
YOLOv3 [44]	25.67	50.64	21.79	-	-
YOLOv4	26.50	52.70	22.06	-	-
YOLOv5-S	42.90	42.90	24.00	44.20	50.80
YOLOv5-M	47.41	47.41	27.90	50.30	51.20
YOLOv5-L	50.10	50.10	29.91	50.60	51.50
YOLOv5-X	54.01	54.01	32.52	50.10	58.10
YOLOX-Nano	49.38	49.38	32.19	-	-
YOLOX-Tiny	50.00	49.96	33.05	-	-
YOLOX-S	52.30	52.30	31.10	-	-
YOLOX-M	**54.19**	54.19	35.08	-	-
YOLOX-L	53.93	53.93	34.71	-	-
I-YOLOX-S	52.27	52.27	34.14	-	-
I-YOLOX-M	**54.19**	54.20	35.08	-	-
I-YOLOX-L	53.93	53.93	34.71	-	-

The average accuracy of classes on dataset IP102 with the YOLOX-S model is illustrated in **Error! Reference source not found.**. In particular, the YOLOX-S has high AP recognition accuracy on some insect species, such as *Cicadellidae* (89.40%), *Unaspis yanonensis* (87.30%), *Icerya purchasi Maskell* (90.70%), *parathrene regalis* (100%). However, there are still some insects that the system can not recognize such as *therioaphis maculata Buckton* (0%), *Phyllocoptes oleiverus ashmead* (0%). Figure 9 depicts the real-time detection results of the YOLOX-S model for English or aphid insects with many individuals from the IP102 dataset. The detection results of the YOLOX-S model have also been tested for many insect species, many individuals on the same image as in the IP102 dataset shown in Figure 10.

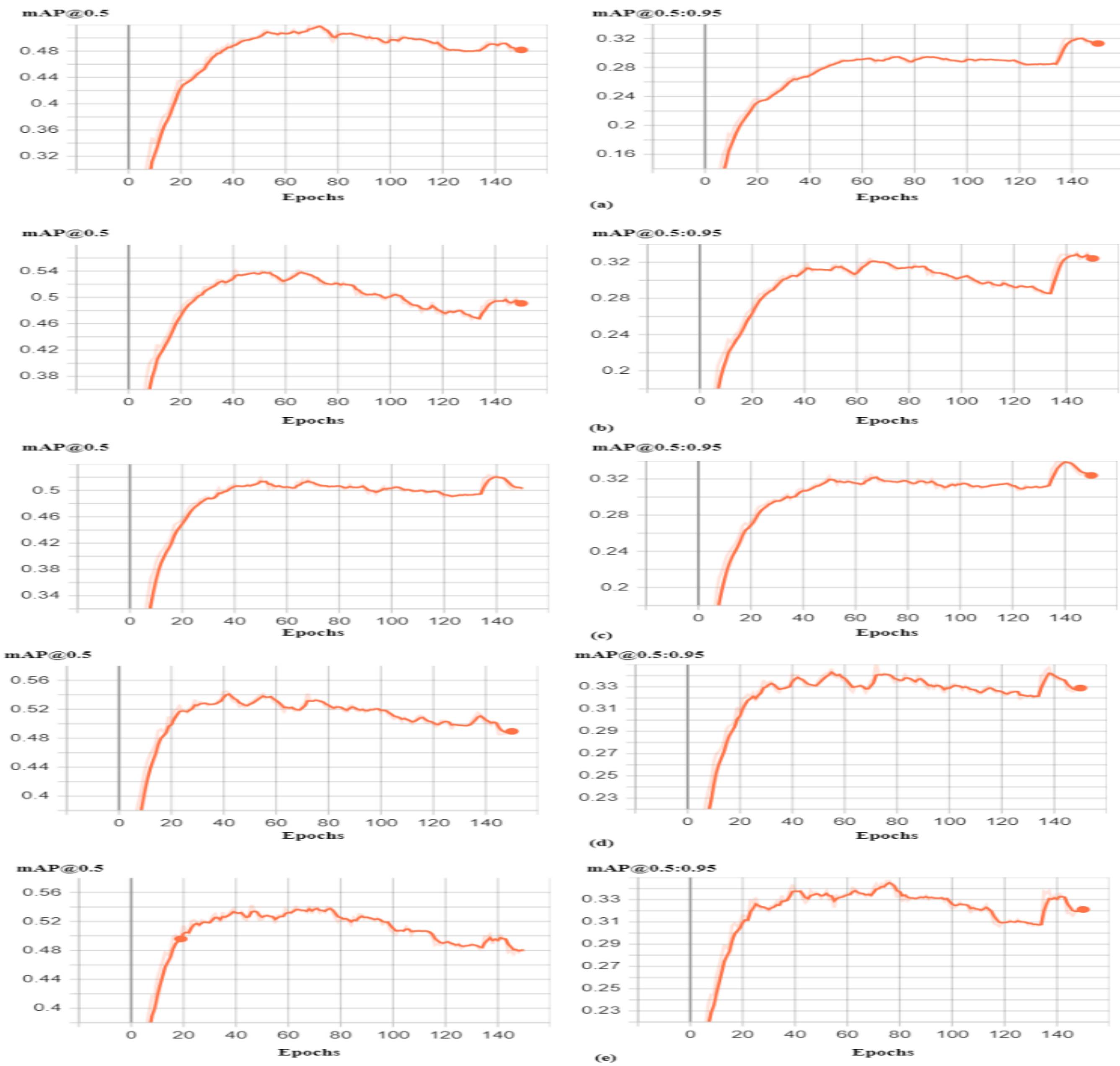

Figure 8. The Training Results of Different YOLOX Model Architectures on IP102 (mAP@IoU:0.5 and mAP@IoU:0.5:0.95) with (a) YOLOX-Nano, (b) YOLOX-tiny, (c) YOLOX-S, (d) YOLOX-M, (e) YOLOX-L.

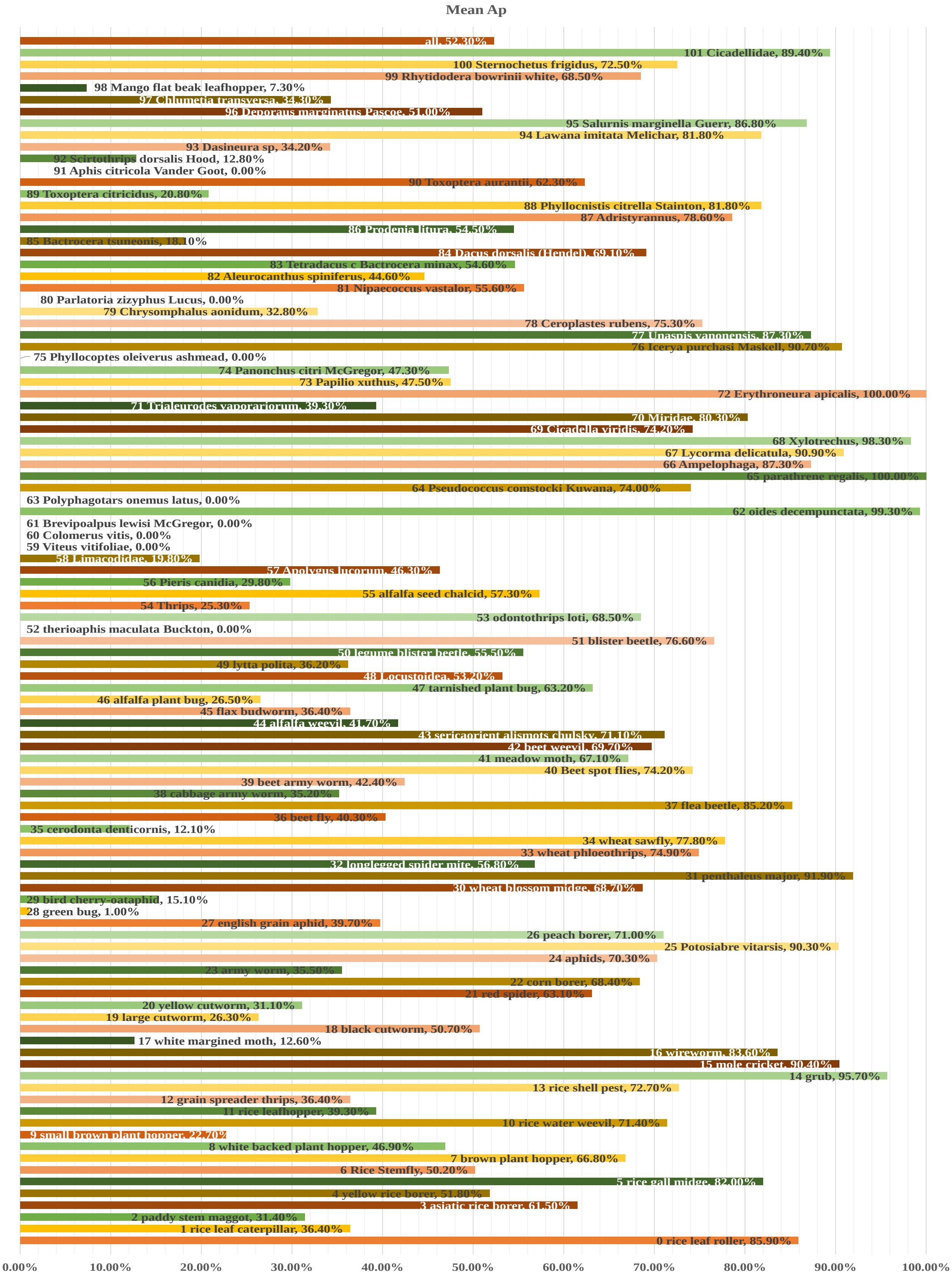

Figure 13. Average Accuracy of Classes on Dataset IP102 with YOLOX-S Model.

Figure 9. The Real-Time Detection Results of the YOLOX-S Model for English or Aphid Insects with many Individuals from the IP102 Dataset.

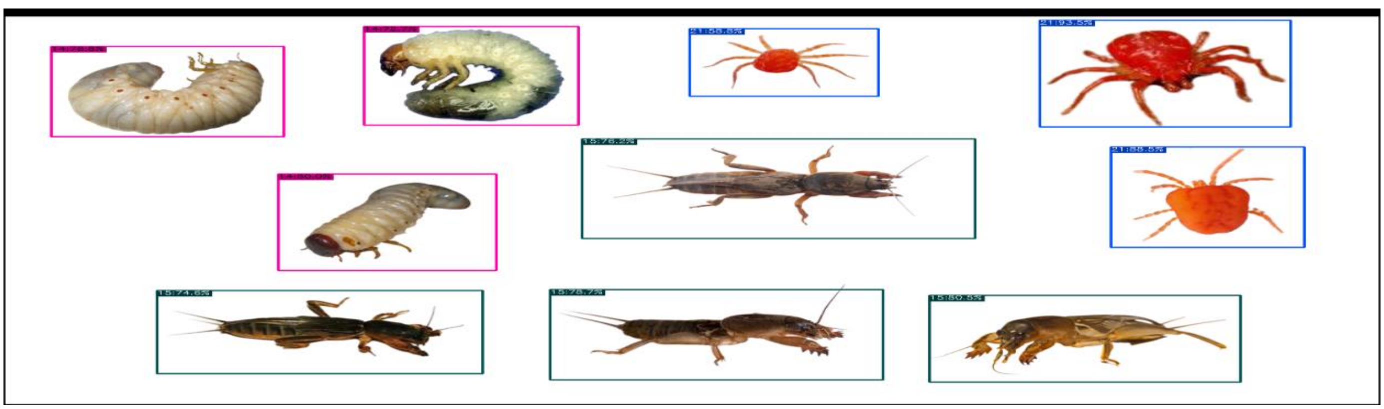

Figure 10. The Detection Results of YOLOX-S Model of many Species, Many Individuals on the Same Image as in IP102.

5. DISCUSSION

Among the trained and validated models with transfer learning, the YOLOX models were the most successful models. These YOLOX models were studied in terms of real-time object recognition. It is found that YOLOX-L can recognizes efficiently 10 different categories of insect against different backgrounds with 84.84%, and YOLOX-M can recognizes efficiently 102 different categories of insect with 54.19% on IP102 datasets. As shown in

Table **1**, the mAP, Precision, and Recall on the Insect10 are usually very high and there is not a significant difference between the models of the architectures. This is a small data set, but there is a uniform distribution of the number of input images between the classes, which explains the rather high mAP parameters between the models. However, it is difficult to judge that the architectures of YOLOv4 have an average accuracy performance that outperforms the architectures of the remaining models. Due to the different complexity between models, YOLOv4 needs more iterations, namely Class * 2000

iterations to get the best results. While the rest of the models are only 150 loops. Meanwhile, the evaluation indicators on the IP102 dataset in

Table **2** are only achieved with two architectures, YOLOv5 and YOLOX, due to the technology and training time being too large, not feasible when implemented on YOLOv4. In general, the mAP of the models is still quite high when trained with a large data set and there is no uniform distribution between classes like IP102. The mAP accuracy ranges from 40% to 54% within the 0.5 IoU threshold and increases with the size of the YOLO models.

The YOLOv4, YOLOv5, and YOLOX architectures are relatively good at correctly identifying insects. On the Insect10, averages ranged from 70.10% mAP to 84.94% mAP, and on the IP102 dataset, averages ranged from 26.50% mAP to 54.19% mAP. However, on the Insect10 dataset, YOLOv4 has a relatively good accuracy performance like other YOLO models, while on IP102 YOLOv5 has a lower accuracy rate than YOLOX on the same input image but not significantly. For the ability to identify insects on video, the accuracy between the two YOLO architectures still has good similarities on the Insect10 dataset. However, on a large-scale dataset like IP102, there is a certain difference between the detection models, specifically the YOLOv5 model has a lower mAP in recognition than the YOLOX model (from 10% to 17%). From the analysis of the recognition results of the model, it can be seen that the system and our collected insect dataset have high accuracy, identify insect pest objects in many situations and are suitable for implementation on mobile terminal devices, mobile applications, and websites. The models are still quite accurate and promise to improve the evaluation indicators to better match real projects.

6. CONCLUSION AND FUTURE WORK

Deep learning models are commonly used to detect insect in plants. However, a major problem is low accuracy when real-world images are presented to the model. In this study, after conducting a comprehensive comparison of the several detection frameworks from deep learning, we built an object detection system that can detect insect in digital images in real-time using the single-stage object detection architectures including YOLOv4, YOLOv5 with five different scales (N, S, M, and L), and YOLOX with four different scales (N, S, M, L, and X). In which, the YOLOX-L model achieved the highest accuracy of 54.19% mAP on the IP102 dataset. The next step in the research is to collect and add other common insect. Then more efficient insect detection models will be studied to implemented the system on high-performance mobile terminal devices.

7. ACKNOWLEDGMENT

This study was supported by technical staff and agricultural experts from An Giang University, and Vietnam National University in Ho Chi Minh City, Vietnam.

REFERENCES

[1] M. Carvajal-Yepes *et al.*, “A global surveillance system for crop diseases,” *Science (80-.).*, vol. 364, no. 6447, pp. 1237–1239, 2019, doi: 10.1126/science.aaw1572.

[2] W. Boedeker, M. Watts, P. Clausing, and E. Marquez, “The global distribution of acute unintentional pesticide poisoning: estimations based on a systematic review,” *BMC Public Health*, vol. 20, no. 1, p. 1875, 2020, doi: 10.1186/s12889-020-09939-0.

[3] 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.

[4] Z. Hu *et al.*, “Application of Non-Orthogonal Multiple Access in Wireless Sensor Networks for Smart Agriculture,” *IEEE Access*, vol. 7, pp. 87582–87592, 2019, doi: 10.1109/ACCESS.2019.2924917.

[5] L. C. Ngugi, M. Abelwahab, and M. Abo-Zahhad, “Recent advances in image processing techniques for automated leaf pest and disease recognition – A review,” *Inf. Process. Agric.*, vol. 8, no. 1, pp. 27–51, 2021, doi: 10.1016/j.inpa.2020.04.004.

[6] F. A. de Assis, J. A. P. V. de Moraes, G. A. de Assis, and F. J. T. Parolin, “Induction of Caterpillar Resistance in Sunflower Using Silicon and Acibenzolar-S-Methyl,” *J. Agric. Sci. Technol.*, vol. 17, pp. 543–550, 2015.

[7] C. M. M. Mols and M. E. Visser, “Great Tits (Parus major) Reduce Caterpillar Damage in Commercial Apple Orchards,” *PLoS One*, vol. 2, 2007.

[8] F. Sulvai, B. J. M. Chaúque, and D. L. P. Macuvele, “Intercropping of lettuce and onion controls caterpillar thread, Agrotis ípsilon major insect pest of lettuce,” *Chem. Biol. Technol. Agric.*, vol. 3, pp. 1–5, 2016.

[9] P. T. T. Thuy, S. Van Geluwe, V. A. Nguyen, and B. Van der Bruggen, “Current pesticide practices and environmental issues in Vietnam: management challenges for sustainable use of pesticides for tropical crops in (South-East) Asia to avoid environmental pollution,” *J. Mater. Cycles Waste Manag.*, vol. 14, pp. 379–387, 2012.

[10] N. P. Duy, T. Y. Đan, and V. T. T. Dương, “Valuing health risk due to pesticides use in agricultural production in Tam Binh district, Vinh Long province: A choice experiment approach,” *Can Tho Univ. J. Sci.*, p. 164, 2018.

[11] D. Hughes *et al.*, "Pesticides use and health impacts on farmers in Thailand, Vietnam, and Lao PDR: Protocol for a survey of knowledge, behaviours and blood acetyl cholinesterase concentrations," *PLoS One*, vol. 16, 2021.

[12] A. Grube, D. Donaldson, T. Kiely, and L. Wu, "Pesticides Industry Sales and Usage: 2006 and 2007 Market Estimates," *U.S. Environ. Prot. Agency*, pp. 1–41, 2011, [Online]. Available: http://nepis.epa.gov/Adobe/PDF/3000659P.pdf.

[13] S. T. Narenderan, S. N. Meyyanathan, and B. Babu, "Review of pesticide residue analysis in fruits and vegetables. Pre-treatment, extraction and detection techniques.," *Food Res. Int.*, vol. 133, p. 109141, 2020.

[14] M. A. J. Al-Sammarraie and N. A. Jasim, "DETERMINING THE EFFICIENCY OF A SMART SPRAYING ROBOT FOR CROP PROTECTION USING IMAGE PROCESSING TECHNOLOGY," *INMATEH Agric. Eng.*, 2021.

[15] E. Onler, "REAL TIME PEST DETECTION USING YOLOv5," *Int. J. Agric. Nat. Sci. E*, vol. 14, no. 3, pp. 232–246, 2021, [Online]. Available: https://www.researchgate.net/publication/357516853.

[16] C. L. McCarthy, S. Rees, and C. P. Baillie, "Machine vision-based weed spot spraying: a review and where next for sugarcane?," 2010.

[17] R. Oberti *et al.*, "Selective spraying of grapevines for disease control using a modular agricultural robot," *Biosyst. Eng.*, vol. 146, pp. 203–215, 2016.

[18] M. Lippi, N. Bonucci, R. F. Carpio, M. Contarini, S. Speranza, and A. Gasparri, "A YOLO-Based Pest Detection System for Precision Agriculture," in *2021 29th Mediterranean Conference on Control and Automation (MED)*, 2021, pp. 342–347, doi: 10.1109/MED51440.2021.9480344.

[19] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015, doi: 10.1038/nature14539.

[20] C. B. Murthy, M. F. Hashmi, N. D. Bokde, and Z. W. Geem, "Investigations of object detection in images/videos using various deep learning techniques and embedded platforms-A comprehensive review," *Appl. Sci.*, vol. 10, no. 9, 2020, doi: 10.3390/app10093280.

[21] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2016-Decem, pp. 779–788, 2016, doi: 10.1109/CVPR.2016.91.

[22] B. Zhang, M. Zhang, and Y. Chen, "Crop pest identification based on spatial pyramid pooling and deep convolution neural network," *Nongye Gongcheng Xuebao/Transactions Chinese Soc. Agric. Eng.*, vol. 35, no. 19, pp. 209–215, 2019, doi: 10.11975/j.issn.1002-6819.2019.19.025.

[23] S. Yang *et al.*, "Maize-YOLO : A New High-Precision and Real-Time Method for Maize Pest Detection," *Insects*, 2023.

[24] J. Liu and X. Wang, "Plant diseases and pests detection based on deep learning: a review," *Plant Methods*, vol. 17, no. 1, pp. 1–18, 2021, doi: 10.1186/s13007-021-00722-9.

[25] M. S. Hossain, R. M. Mou, M. M. Hasan, S. Chakraborty, and M. A. Razzak, "Recognition and detection of tea leaf's diseases using support vector machine," *2018 IEEE 14th Int. Colloq. Signal Process. \& Its Appl.*, pp. 150–154, 2018.

[26] E. M. . Alegre and J. K. M. . Labajo, "The Impact of Work Stress on the Psychological Well-being of Public Elementary School Teachers", International Journal of Membrane Science and Technology, vol. 10, no. 2, pp. 719-727, 2023.

[27] N. Vinushree, B. Hemalatha, and V. K. Kaliappan, "Efficient Kernel-Based Fuzzy C-Means Clustering for Pest Detection and Classification," in *2014 World Congress on Computing and Communication Technologies*, 2014, pp. 179–181, doi: 10.1109/WCCCT.2014.61.

[28] A. Martin, D. Sathish, C. Balachander, T. K. Hariprasath, and G. Krishnamoorthi, "Identification and counting of pests using extended region grow algorithm," *2015 2nd Int. Conf. Electron. Commun. Syst.*, pp. 1229–1234, 2015.

[29] Y. Kumar, A. K. Dubey, and A. Jothi, "Pest detection using adaptive thresholding," *2017 Int. Conf. Comput. Commun. Autom.*, pp. 42–46, 2017.

[30] D. Čirjak, I. Miklečić, D. Lemić, T. Kos, and I. P. Živković, "Automatic Pest Monitoring Systems in Apple Production under Changing Climatic Conditions," *Horticulturae*, vol. 8, no. 6, 2022, doi: 10.3390/horticulturae8060520.

[31] V. Agnihotri, "Machine Learning based Pest Identification in Paddy Plants," in *2019 3rd International conference on Electronics, Communication and Aerospace Technology (ICECA)*, 2019, pp. 246–250, doi: 10.1109/ICECA.2019.8822047.

[32] B. Rajesh, M. V. Sai Vardhan, and L. Sujihelen, "Leaf Disease Detection and Classification by Decision Tree," in *2020 4th International Conference on Trends in Electronics and Informatics (ICOEI)(48184)*, 2020, pp. 705–708, doi: 10.1109/ICOEI48184.2020.9142988.

[33] A. Kamilaris and F. X. Prenafeta-Boldú, "Deep learning in agriculture: A survey," *Comput. Electron. Agric.*, vol. 147, no. February, pp. 70–90, 2018, doi: 10.1016/j.compag.2018.02.016.

[34] N. T. Nam and P. D. Hung, "Pest detection on Traps using Deep Convolutional Neural Networks," *Proc. 1st Int. Conf. Control Comput. Vis.*, 2018.

[35] T.-L. Lin, H.-Y. Chang, and K.-H. Chen, "Pest and Disease Identification in the Growth of Sweet Peppers using Faster R-CNN," in *2019 IEEE International Conference on Consumer Electronics - Taiwan (ICCE-TW)*, 2019, pp. 1–2, doi: 10.1109/ICCE-TW46550.2019.8991893.

[36] D. Li *et al.*, "A recognition method for rice plant diseases and pests video detection based on deep convolutional neural network," *Sensors (Switzerland)*, vol. 20, no. 3, 2020, doi: 10.3390/s20030578.

[37] J. Gambhir *et al.*, "Deep Learning for Real-Time Diagnosis of Pest and Diseases on Crops," 2022, pp. 189–197.

[38] Y. Zhong, J. Gao, Q. Lei, and Y. Zhou, "A vision-based counting and recognition system for flying insects in intelligent agriculture," *Sensors (Switzerland)*, vol. 18, no. 5, 2018, doi: 10.3390/s18051489.

[39] M. Zha, W. Qian, W. Yi, and J. Hua, "A Lightweight YOLOv4-Based Forestry Pest Detection Method Using Coordinate Attention and Feature Fusion," *Entropy (Basel).*, vol. 23, no. 12, p. 1587, Nov. 2021, doi: 10.3390/e23121587.

[40] W. Zhang, H. Huang, Y. Sun, and X. Wu, "AgriPest-YOLO: A rapid light-trap agricultural pest detection method based on deep learning," *Front. Plant Sci.*, vol. 13, no. December, pp. 1–16, 2022, doi: 10.3389/fpls.2022.1079384.

[41] C. Wen *et al.*, "Pest-YOLO: A model for large-scale multi-class dense and tiny pest detection and counting," *Front. Plant Sci.*, vol. 13, no. October, 2022, doi: 10.3389/fpls.2022.973985.

[42] I. Ahmad *et al.*, "Deep Learning Based Detector YOLOv5 for Identifying Insect Pests," *Appl. Sci.*, vol. 12, no. 19, 2022, doi: 10.3390/app121910167.

[43] J. Huang, Y. Huang, H. Huang, W. Zhu, J. Zhang, and X. Zhou, “An Improved YOLOX Algorithm for Forest Insect Pest Detection,” *Comput. Intell. Neurosci.*, vol. 2022, 2022, doi: 10.1155/2022/5787554.

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

[45] R. K. Samanta and I. Ghosh, “Tea Insect Pests Classification Based on Artificial Neural Networks,” *Int. J. Comput. Eng. Sci.*, vol. 2, no. 6, pp. 1–13, 2012.

[46] J. Wang, C. Lin, L. Ji, and A. Liang, “A new automatic identification system of insect images at the order level,” *Knowledge-Based Syst.*, vol. 33, pp. 102–110, 2012, doi: https://doi.org/10.1016/j.knosys.2012.03.014.

[47] K. Venugoban and A. Ramanan, “Image Classification of Paddy Field Insect Pests Using Gradient-Based Features,” *Int. J. Mach. Learn. Comput.*, no. February, pp. 1–5, 2014, doi: 10.7763/ijmlc.2014.v4.376.

[48] C. Xie *et al.*, “Automatic classification for field crop insects via multiple-task sparse representation and multiple-kernel learning,” *Comput. Electron. Agric.*, vol. 119, pp. 123–132, 2015.

[49] Z. Liu, J. Gao, G. Yang, H. Zhang, and Y. He, “Localization and Classification of Paddy Field Pests using a Saliency Map and Deep Convolutional Neural Network,” *Sci. Rep.*, vol. 6, no. February, 2016, doi: 10.1038/srep20410.

[50] C. Xie *et al.*, “Multi-level learning features for automatic classification of field crop pests,” *Comput. Electron. Agric.*, vol. 152, no. July, pp. 233–241, 2018, doi: 10.1016/j.compag.2018.07.014.

[51] A. A. Alfarisy, Q. Chen, and M. Guo, “Deep learning based classification for paddy pests & diseases recognition,” *Proc. 2018 Int. Conf. Math. Artif. Intell.*, 2018.

[52] L. Deng, Y. Wang, Z. Han, and R. Yu, “Research on insect pest image detection and recognition based on bio-inspired methods,” *Biosyst. Eng.*, vol. 169, pp. 139–148, 2018.

[53] E. Ayan, H. Erbay, and F. Varçın, “Crop pest classification with a genetic algorithm-based weighted ensemble of deep convolutional neural networks,” *Comput. Electron. Agric.*, vol. 179, Dec. 2020, doi: 10.1016/j.compag.2020.105809.

[54] A. Mikołajczyk and M. Grochowski, “Data augmentation for improving deep learning in image classification problem,” in *2018 International Interdisciplinary PhD Workshop, IIPhDW 2018*, 2018, pp. 117–122, doi: 10.1109/IIPHDW.2018.8388338.

[55] C. Shorten and T. M. Khoshgoftaar, “A survey on Image Data Augmentation for Deep Learning,” *J. Big Data*, vol. 6, no. 1, 2019, doi: 10.1186/s40537-019-0197-0.

[56] J. Redmon and A. Farhadi, “YOLOv3: An Incremental Improvement,” *ArXiv*, vol. abs/1804.0, 2018.

[57] J. Redmon and A. Farhadi, “YOLO9000: Better, Faster, Stronger,” 2017, pp. 6517–6525, doi: 10.1109/CVPR.2017.690.

[58] M. Tan, R. Pang, and Q. V. Le, “EfficientDet: Scalable and Efficient Object Detection,” Nov. 2019, [Online]. Available: http://arxiv.org/abs/1911.09070.

[59] T. Elsken, J. H. Metzen, and F. Hutter, “Neural Architecture Search: A Survey,” *ArXiv*, vol. abs/1808.0, 2018.

[60] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, “YOLOv4: Optimal Speed and Accuracy of Object Detection,” 2020, [Online]. Available: http://arxiv.org/abs/2004.10934.

[61] Ultralytics, “YOLOv5: Train Custom Data.” 2021, [Online]. Available: https://github.com/ultralytics/yolov5/wiki/Train-Custom-Data.

[62] 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.

[63] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, “{YOLOX:} Exceeding {YOLO} Series in 2021,” *CoRR*, vol. abs/2107.0, 2021, [Online]. Available: https://arxiv.org/abs/2107.08430.

[64] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, “How transferable are features in deep neural networks?,” *Adv. Neural Inf. Process. Syst.*, vol. 4, no. January, pp. 3320–3328, 2014.

[65] O. Russakovsky *et al.*, “ImageNet Large Scale Visual Recognition Challenge,” *Int. J. Comput. Vis. Vol.*, Sep. 2014, [Online]. Available: http://arxiv.org/abs/1409.0575.

[66] S. Ruder, “An overview of gradient descent optimization algorithms,” *arXiv Prepr. arXiv1609.04747*, 2016.

[67] U. Nepal and H. Eslamiat, “Comparing YOLOv3, YOLOv4 and YOLOv5 for Autonomous Landing Spot Detection in Faulty UAVs,” *Sensors*, vol. 22, no. 2, 2022, doi: 10.3390/s22020464.

[68] A. Chaudhary, K. Chouhan, J. Gajrani, and B. Sharma, “Deep Learning With PyTorch,” 2020.

[69] N. Khasawneh, M. Fraiwan, and L. Fraiwan, “Detection of K-complexes in EEG signals using deep transfer learning and YOLOv3,” *Cluster Comput.*, 2022, doi: 10.1007/s10586-022-03802-0.

[70] S. Ren, K. He, R. B. Girshick, and J. Sun, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks,” *CoRR*, vol. abs/1506.0, 2015, [Online]. Available: http://arxiv.org/abs/1506.01497.

[71] T.-Y. Lin, P. Dollár, R. B. Girshick, K. He, B. Hariharan, and S. J. Belongie, “Feature Pyramid Networks for Object Detection,” *CoRR*, vol. abs/1612.0, 2016, [Online]. Available: http://arxiv.org/abs/1612.03144.

[72] W. Liu *et al.*, “SSD: Single Shot MultiBox Detector,” *CoRR*, vol. abs/1512.0, 2015, [Online]. Available: http://arxiv.org/abs/1512.02325.

[73] S. Zhang, L. Wen, X. Bian, Z. Lei, and S. Z. Li, “Single-Shot Refinement Neural Network for Object Detection,” *CoRR*, vol. abs/1711.0, 2017, [Online]. Available: http://arxiv.org/abs/1711.06897.

DOI: https://doi.org/10.15379/ijmst.v10i2.1306