agriculture

MDPI

Article

YOLO-Based Light-Weight Deep Learning Models for Insect Detection System with Field Adaption

Nithin Kumar [1,*], Nagarathna [2] and Francesco Flammini [3,*]

[1] Department of Computer Science & Engineering, Vidyavardhaka College of Engineering, Mysuru 570002, India
[2] Department of Computer Science & Engineering, PES College of Engineering, Mandya 571401, India
[3] IDSIA USI-SUPSI, University of Applied Sciences and Arts of Southern Switzerland, 6928 Manno, Switzerland
[*] Correspondence: nithingowda021@vvce.ac.in (N.K.); francesco.flammini@supsi.ch (F.F.)

Abstract: The most incredible diversity, abundance, spread, and adaptability in biology are found in insects. The foundation of insect study and pest management is insect recognition. However, most of the current insect recognition research depends on a small number of insect taxonomic experts. We can use computers to differentiate insects accurately instead of professionals because of the quick advancement of computer technology. The "YOLOv5" model, with five different state of the art object detection techniques, has been used in this insect recognition and classification investigation to identify insects with the subtle differences between subcategories. To enhance the critical information in the feature map and weaken the supporting information, both channel and spatial attention modules are introduced, improving the network's capacity for recognition. The experimental findings show that the F1 score approaches 0.90, and the mAP value reaches 93% through learning on the self-made pest dataset. The F1 score increased by 0.02, and the map increased by 1% as compared to other YOLOv5 models, demonstrating the success of the upgraded YOLOv5-based insect detection system.

Keywords: YOLOv5; insect pest; classification; deep learning; object detection

check for updates

Citation: Kumar, N.; Nagarathna; Flammini, F. YOLO-Based Light-Weight Deep Learning Models for Insect Detection System with Field Adaption. *Agriculture* **2023**, *13*, 741. https://doi.org/10.3390/agriculture13030741

Academic Editor: József Sütő

Received: 7 February 2023
Revised: 18 March 2023
Accepted: 20 March 2023
Published: 22 March 2023

1. Introduction

A zoology subfield called entomology covers insect-related research [1]. We discovered that a more thorough investigation is required to identify the species level of insects due to the vast number of harmful insect populations. The aim to stop insects from damaging people, animals, plants, and farms has led to a surge in entomology research in recent years [2,3]. Entomology research is crucial because it opens new avenues and benefits for chemistry, medicine, engineering, and pharmaceuticals inspired by insects and nature [4]. Insects rob and obliterate a third of the world's harvests, resulting in the loss of numerous products, and businesses suffer losses. Quick and accurate identification of insects is essential to avoid financial losses and progress the study of entomology. Scientists are also inspired by insects while developing robotics, sensors, mechanical structures, aerodynamics, and intelligent systems [5]. Although an estimated 1.5 million insect species are on the planet, only about 750 thousand are identified and classified species [6]. However, it is uncommon for scientists to keep finding and naming new species. Some bug species are wiped out covertly due to destruction and forest fires [7]. These factors make scholarly research on insect detection crucial for demonstrating biodiversity. Identifying the order level an insect belongs to is vital in determining insects [8]. Knowing the order level is necessary to differentiate the type. Scientific research has led to the identification of 31 insect orders in the natural world [9]. Insect orders dating back to 2002 were discovered. About 21 factors have to be considered, such as the number of wings, body type, legs, and head shape [8,10]. Traditional insect testing methods are time-consuming as many criteria must be met to avoid diagnostic errors. No decision support system was found to classify at this level when examining the insect recognition literature at the order level. It was also pointed out that no thorough deep-learning comparative study identifies and classifies insects at eye level. Research shows that several classification approaches based on deep learning, artificial intelligence, or machine learning are required to consistently and accurately classify and recognize insects.

Several studies on pest taxonomy and identification have recently been published, but this area of research has yet to be fully explored [10,11]. The most popular technique for classifying and identifying insect pests in current research is transfer learning using support vector machines and DL frameworks that have already been trained [12–17]. However, the SVM machine learning technique needs more time to prepare with more enormous datasets [18]. The constraints in transfer learning that cause the most worry are overfitting and negative transfer. Ding et al. [19] conducted an experiment to classify moths in 24 classes of insects from the internet using VGG16 and achieved a classification accuracy of 89.22%. Shi et al. [20] conducted an experiment to detect and classify stored grain insects in eight classes of stored grain insects using DenseNet-121 and achieved a classification and detection accuracy of 88.06%. Mamdouh et al. [21] conducted an experiment to classify olive fruit flies using MobileNet and achieved a classification accuracy of 96.89%. Even if we consider any latest research work, researchers are working on only one insect species, which is harmful to any crop or ecosystem because of a lack of data availability. Our specific research objective of this study is to generalize insect detection and classification for real-time insect monitoring devices by:

- developing faster and more accurate YOLO-based detection and a classification model with an attention mechanism to classify the insect order and bring down the classification to the species level using the same model;
- comparing the discovered model's performance with other state-of-the-art insect detection techniques.

The rest of the manuscript is organized as follows: Section 2 briefly describes related works in insect detection and classification. Section 3 provides a description of the dataset used with YoloV5 and CBAM implementation. Section 4 consists of a discussion of achieved results. Finally, the manuscript concludes with a conclusion and aims for future enhancement.

2. Related Work

Pest detection techniques come in three flavors: semi-automated, automatic, and manual. The flies that emerge from the traps are counted by teams of professional workers employed in manual pest detection methods. There are manual detection methods that are labor-intensive, cumbersome, and error-prone. E-traps are used in automatic and semi-automatic systems [14,22]. A wireless transceiver, controller and camera are included in the E-Trap [14]. In a semi-automatic detection system, flies are identified and counted by experts using photographs collected by E-traps and transferred to a server [14,22]. However, a human expert is still required to operate the semi-automated device. Non-expert users can use a semi-automated approach that takes pictures of insect wings, thorax, and abdomen as input to help identify the most likely candidate species for trapped insects [22]. This is not conceivable in the wild since insects are captured on sticky traps or in attractant-baited liquids. However, the automatic detecting systems run entirely on their own. The automated detection systems use pictures, optoacoustic methods, and spectroscopy to find and count insect pests. When applying picture-based detection methods, deep learning, machine learning, and image processing are used.

2.1. Opto-Acoustic Techniques

Potamitis et al. [23] developed an optoacoustic spectrum analysis-based method for identifying olive fruit flies. The optoacoustic spectrum analysis can determine the species by examining the patterns of the insects' wingbeats. The authors looked at its temporal and frequency domains to better understand the recorded signal. The qualities that were discovered in the time and frequency parts are given as input into the random forest classifier. The recall, F1-score, and precision of the random forest classifier were all 0.93. Optoacoustic detection is ineffective in determining the difference between peaches and figs. Additionally, sunshine can also affect sensor readings. The trap is also more susceptible to shocks or other unexpected impacts that could cause false alarms on windy days.

2.2. Image Processing Techniques

Image processing is the foundation of the detection technique [24–26]. Although image-processing methods are simpler to employ than machine learning or deep learning algorithms, their performance is lighting-dependent, and they only have a moderate accuracy (between 70 and 80 percent). An image processing technique was created by Doitsidis et al. [24] to find olive fruit flies; the algorithm uses auto-brightness adjustment to remove the impact of varying lighting and weather conditions. The borders are then made sharper using a coordinate logic filter to emphasize the contrast between the dark insect and the bright background. The program then uses a noise reduction filter after a circular Hough transforms to establish the trap's boundaries and achieve a 75% accuracy. Tirelli et al. [25] developed a Wireless Sensor Network (WSN) to find pests in greenhouses; the algorithm used to analyze photographs first takes the effects of variations in lighting out of the picture before denoising it and then looking for blobs. Insect image processing, insect segmentation, and sorting were all part of the Sun et al. [26] proposal for "soup" photographs of insects. The insects appear floating on the liquid surface in images of bug "soup". The algorithm was tested with 19 pictures of soup, and it worked well for most of the pictures. Philimis et al. [27] were able to detect olive fruit flies and medflies in the field with the help of McPhail traps and WSNs. The creation of WSNs, which are networks of sensors that gather data, allows for the processing and transmission of that data to humans. Actuators that respond to certain events may also be present in WSNs.

2.3. Machine Learning Techniques

To classify 14 species of butterflies, Kaya et al. [28] developed a classifier using machine learning. The authors extracted the properties for texture and color features; the recovered features were fed into a three-layer neural network as inputs and achieved a classification accuracy of 92.85%.

2.4. Deep Learning Techniques

Using a multi-class classifier based on deep learning, Zhong et al. [29] categorize and count six different kinds of flying insects. The detection and coarse counting processes are built on the YOLO approach [30]; the authors modify the photographs by scaling, flipping, rotating, translating, changing contrast, and adding noise to expand the dataset size. Additionally, they used a YOLO network that had already been trained and then modified the network parameters using the dataset for insect classification. Kalamatianos et al. [31] used the DIRT dataset; the authors examined various iterations of a deep learning detection algorithm called Faster Region Convolutional Neural Networks (Faster-RCNN) [31]. Before classification, convolutional neural networks with region suggestions, or RCNNs, indicate the regions of objects. Faster-RCNN was successful since it attained an mAP of 91.52%, the highest average precision for various recall settings. Although the detection accuracy of grayscale and RGB images is roughly the same, the authors demonstrated that the size of the image strongly influences detection. Due to the computationally demanding nature of Faster RCNN, each e-trap sends its periodically collected image to a server for processing, devising a method for spotting codling moths [32]. Translation, rotation, and reversal all enhance the visuals. An algorithm for color correction is used to equalize the average brightness of the red, green, and blue channels during the pre-processing of the images. The moths in the photos are then found using a sliding window [32]. CNNs are supervised learning systems that apply filters with predetermined weights on picture pixels utilizing an extensive deep-learning neural network [19] for detecting 24 different insect species in agricultural fields; a trained VGG-19 network is used to extract the necessary characteristics. Then, the bug's location is determined via the "Region Proposal Network (RPN) [33]", and an mAP of 89.22% was attained.

3. Materials and Methods

The proposed methodology consists of five sequential steps. We collected early-stage pest pictures for DL model training and evaluation. Next, we preprocessed the entire dataset by annotating and augmenting it to add more examples to the dataset. Image data augmentation uses pre-existing photographs that have been considerably edited based on several parameters to enhance the training data set's size artificially. Third, we trained the YOLO object recognition model using the insect dataset. We analyzed the results and validated the improved model's detection performance using a shared validation dataset. Finally, we determined the best model for practical use in field adaptation.

3.1. Dataset Collection

We used a publicly available dataset in our experimental study: Taxonomy Orders Object Detection Dataset [34]. The Taxonomy Orders Object Detection Dataset—Insects from seven distinct orders are represented in the dataset by more than 15,000 photos as shown in Figure 1, including members of the Coleoptera, Araneae, Hemiptera, Hymenoptera, Lepidoptera, Odonata, and Diptera families. The photographs were gathered from the iNaturalist and BugGuide websites, and they were categorized according to family and order. The training, testing, and validation portions of the dataset we employed were separated. For training, 12,299 images were used; 1538 images were used for validation, followed by 1538 for testing. The original dataset's test and validation folders were combined for the study and utilized to verify the trained models. The most effective model in terms of mAP was discovered after training and evaluating many model versions. We used photos with diverse backgrounds to examine the generalizability of our approach and its potential for identifying various insect pests.

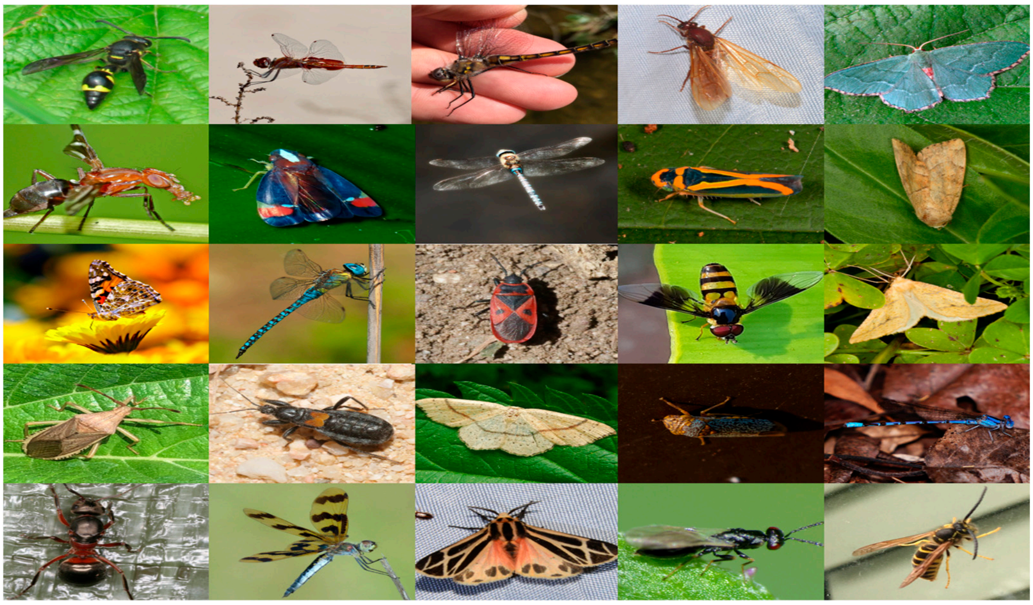


Figure 1. Images of Insect pests used in training and validation of object detection models.

Once the practical model in terms of mAP was discovered, the same model was used to bring down the classification to the species level by manually labeling the images present in the following classes—Diptera class (Bactrocera dorsalis, house fly, honey bee, mosquito), Odonata class (Dragon fly), Lepidoptera (butterfly) and Araneae (spider).

3.2. Data Annotation

The objects detected in each image were marked in a rectangle during annotation. Many objects can be marked in the image. A text file with the same name as the associated image contains information about what was identified in each image. Each object in this text file has its line. Rectangle coordinates are normalized between 0 and 1 to be independent of image size [30]. YOLO v5 requires annotations in the .txt file, with each line describing the bounding box of each image. The image text file will not exist if the object is not included. Each row in the text file includes the following details shown in Equation (1):

$$\text{(id\_class, x\_centre, y\_centre, width, height)} \quad (1)$$

Images of various sizes can therefore be utilized in the object detection system because each image will be evaluated in accordance with its size.

3.3. Data Augmentation

There was no need to perform additional operations to increase the dataset samples because the dataset contained 15,374 photos. More data generally improves the performance of DL models [35]. Nevertheless, gathering a lot of data for training is a complex undertaking. As a result, the need for more data frequently arises during data analysis. In order to overfit and generalize the DL model, more training data might be created using available images through flipping, rotation (horizontal and vertical), blur, and saturation.

3.4. YOLOv5

The YOLO algorithms divide the image into N grids [30], with an equal-sized S × S region in each grid. Each of these N grids is responsible for discovering and locating the object it contains, as seen in Figure 2.

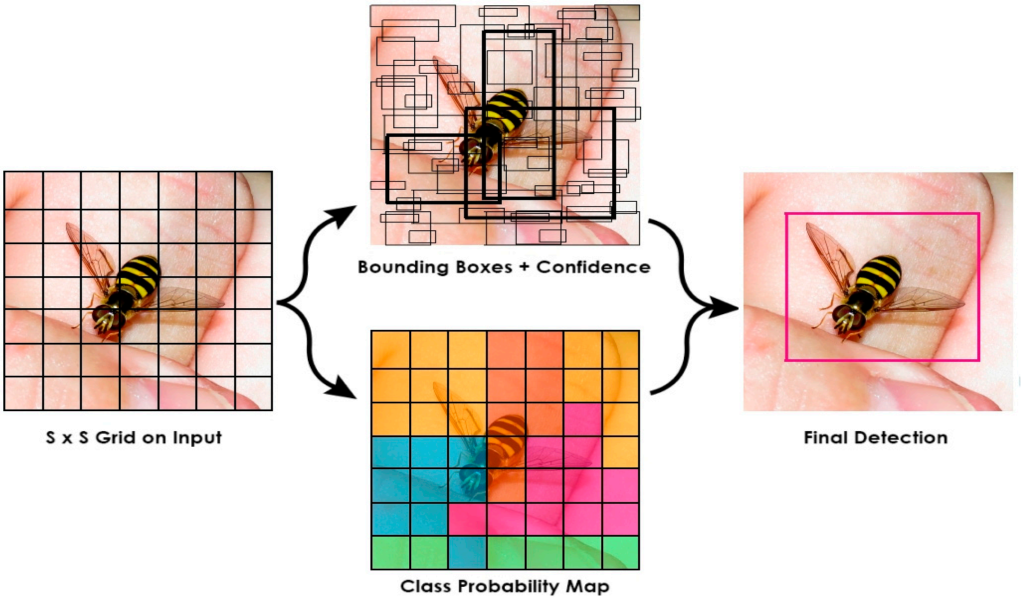


Figure 2. Object Detection in YOLO Framework.

Only inputs with the exact resolution as the training image are supported by the YOLO training model [30]. Further, only one object can be detected at a time, even when each detected grid contains many objects. A single-stage real-time object detection model is called YOLOv2 or YOLO9000 using Darknet-19 as a foundation with anchor boxes to forecast bounding boxes, batch normalization, a high-resolution classifier, and other features; it outperforms YOLOv1 in a number of ways [36]. The YOLOv3 architecture was influenced by the ResNet and FPN designs [37]. The feature extractor, also known as Darknet-53 with

52 convolutions, analyses the image at various spatial compressions using three prediction heads similar to FPN and skip connections similar to ResNet. When comparing performance, YOLOv4 outperforms EfficientDet by a factor of two [38]. Additionally, the average precision and frames per second increased by 10% and 12%, respectively, compared to YOLOv3. YOLOv5 is nearly ninety percent smaller than YOLOv4, requires only 27 MB, and has accuracy comparable to YOLOv4. This demonstrates YOLOv5's potent skills [38].

As shown in Figure 3, four components make up the YOLOv5 network structure—backbone, neck, head (prediction), and input [30,39]. In contrast to YOLOv4, YOLOv5 uses mosaic data augmentation as its input method for image enhancement. The training procedure for mosaic data improvement involves randomly cutting four images and creating one new image. Backbone uses focus structure and CSP structure to combine visual feature data from various image granularities into a convolutional neural network [39]. A set of network layers that mix and combine image features are added to the neck as part of the FPN + PAN (path aggregation network) structure. Finally, these image features are then sent to the prediction layer [38].

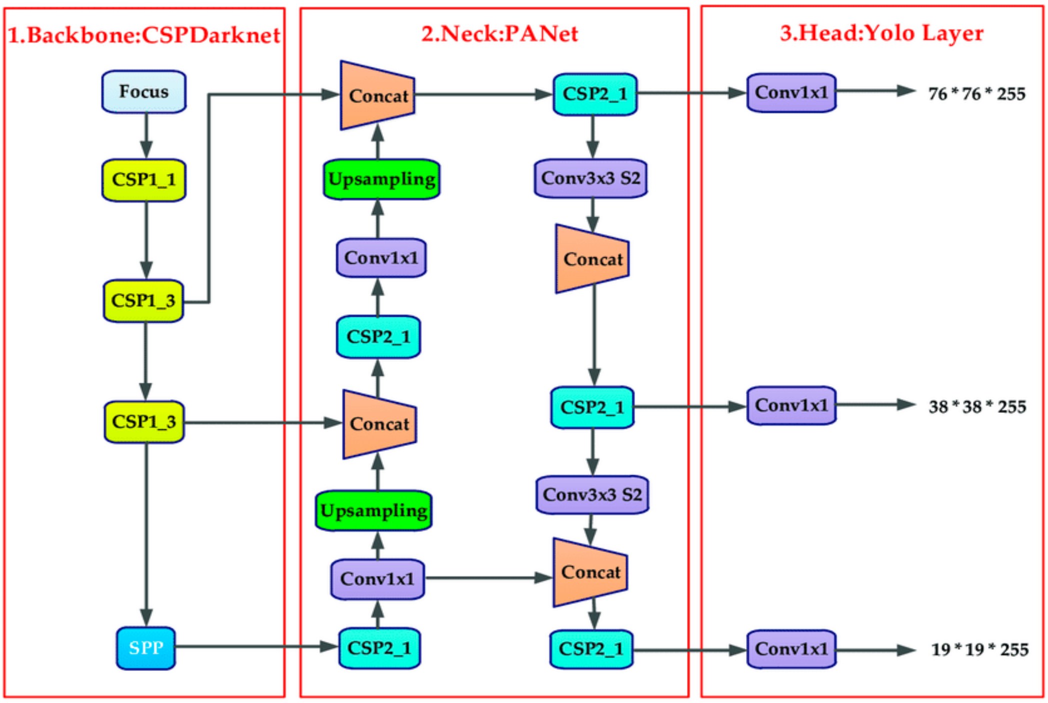


Figure 3. YOLO v5 architecture [39].

YOLOv5 contains four models in total [38]. Starting from YOLOv5s (smallest model) to YOLOv5x (extra largest model). YOLOv5s records 7,027,720 parameters with 272 layers; it is the family's smallest model and is best for using the CPU for inference. YOLOv5m records 20,879,400 parameters in 369 layers. This is a medium size model. It is a good balance between speed and accuracy, so it may be the best model for training on many datasets. YOLOv5l records 46,149,064 parameters in 468 layers. The large model of the YOLOv5 family will work perfectly on the datasets where we need to find small objects. YOLOv5x, both in terms of size and mAP, it is the largest of the four models. However, it has 86.7 million parameters and is slower than the other models of YOLOv5.

3.5. Attention Mechanism

The neural network can learn through training which portions in each new image require attention, forming attentiveness. More researchers have been using deep learning in recent years, and they have shown that the attention mechanism can enhance the model's performance. Jaderberg et al. [40] proposed the spatial transformation network (STN) in

2015. STN has been added as a learnable module to CNN. The spatial invariance can be altered and increased by the feature map's characteristics. The model can still recognize and identify feature abilities even if the input is altered and somewhat modified. Hou et al. [41] suggested squeeze-and-excitation networks (SENet) in 2017. It is split into two sections: the compression section and the excitation section. The purpose of the compression component is to generally compress the input feature map's channel direction into one dimension. The excitation part's job is to forecast each channel's importance, apply the right channel from the preceding feature-map once the importance of various channels has been determined, and then carry out further operations. Past experience [40–42] has demonstrated that the attention mechanism can help the network focus more on the key elements, increasing recognition accuracy and improving the effect. Therefore, it makes sense to incorporate an attention mechanism into YOLOv5.

An attention mechanism module called CBAM is integrated into YOLOv5's CONV in order to give the network its attention mechanism capability. As an illustration, the new network structure of the CBL module in YOLOv5 is represented in Equation (2).

$$\text{CBL} = \text{CONV} + \text{BN} + \text{ReLU} \tag{2}$$

As shown in Figure 4, to achieve channel attention and spatial attention, CONV is combined with the CBL module. The channel attention module and the spatial attention module take the role of the original CONV [42], enabling the network to reinforce important information while reducing the interference from irrelevant information [40].

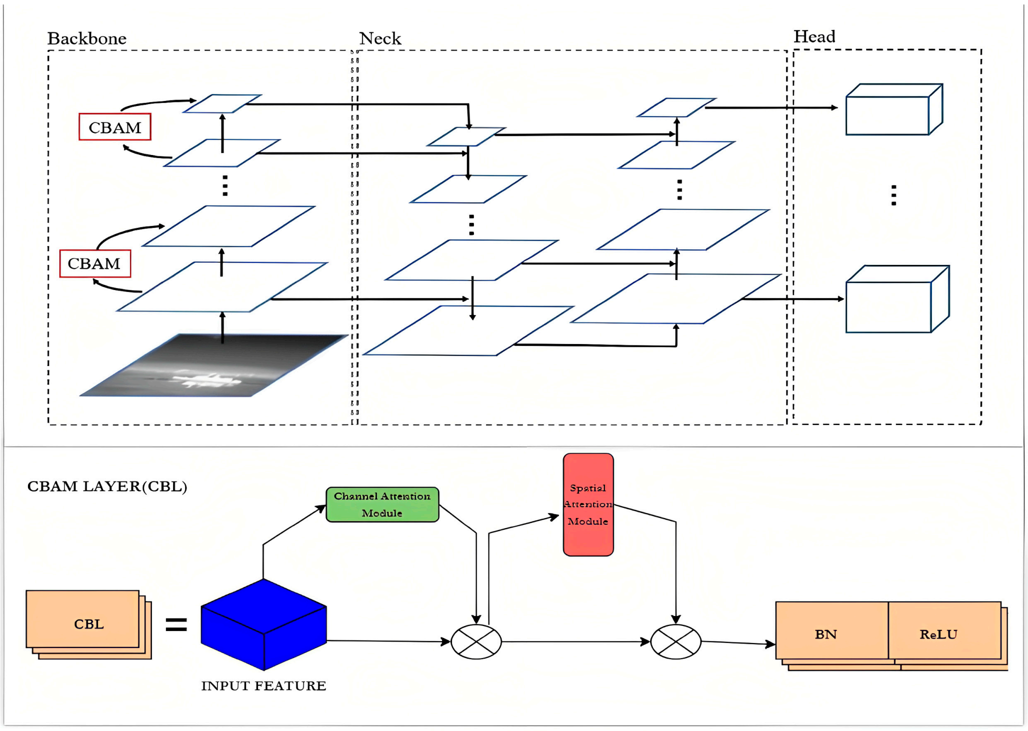


Figure 4. Modified Structure of YoloV5 with CBL.

3.6. Experimental Setup

Each underlying model for insect recognition was trained using 12,299 photos of 7 classes of insects such as Araneae, Coleoptera, Diptera, Hemiptera, Hymenoptera, Lepidoptera, and Odonata throughout the training phase. The stack size was set to 4 for YOLOv5x, 16 for YOLOv5s, YOLOv5m, and YOLOv5l, and images were reduced to 640 × 640 pixels. Initially, the annotations were in PASCAL VOC format in XML files. It was changed to YOLO format in text files. YOLOv5 was cloned from the GitHub repository. The dataset was augmented, and several transforms were performed, such as random vertical and horizontal flips, brightness, and sharpness corrections. These data were initially trained on YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x models.

First, the models were trained for ten epochs. However, the output graph of these ten epochs looked like the model needed to be more balanced. The loss curves gradually descended, and the precision and recall gradually increased; this showed the opportunity to increase the epochs from 10 to 20. The output of 20 had better results compared to 10 epochs. Another experiment of training the model to 25 epochs was performed. However, this resulted in model overfitting. As a result, a 20-epoch model was finalized.

In the modified YOLOv5x architecture, CBAM layers were introduced in the backbone of YoloV5 architecture. For this, the yolov5x.yaml configuration file was modified by introducing the CBAM layers. Additionally, the CBAM function was added in the common.py file in the YOLOv5/models directory. The training process is the same as the regular model, but the configuration file path must be changed in the training argument. Optimizer was used in SGD by default, but it was tested on Adam. However, it did not yield good results, and the learning rate used in training was "1×10^{-3}". To fine-tune and modify the weights, the model runs for 20 epochs. Pytorch was used to construct each model. The model's final output included the probability of a specific class as well as the location bounding box of the target insect pest categories (the prediction box of the position). The system configuration for the training is shown in Table 1, and each model's specific training methods are shown in Table 2. In order to assess the effectiveness of the underlying models, we employed a validation set of 1537 pictures. We used a total of 1538 unexposed pictures for insect pest detection in order to test the models.

Table 1. Experiment environment.

Configuration	Version
CPU	v4 CPU
RAM	16 GB
GPU	Nvidia Tesla T4
GPU Memory	16 GB
Python Version	3.8.10
Dataset Size	12GB

Table 2. Various versions of YOLOv5 models used in the training.

YOLO Version	No. of Layers	No. of Parameters	Training Time (in h)	Inference Time (in s)
Version-5s	272	7,027,720	4	0.15
Version-5m	369	20,879,400	6.5	0.23
Version-5l	468	46,149,064	7.8	0.33
Version-5x	567	83,365,852	8.1	0.59
New Version-5x	641	94,246,049	8.97	0.50

4. Results and Discussions

We used an evaluation metric called intersection over union (IoU) to evaluate the accuracy of the object detector on a given dataset [43]. This scoring metric is commonly used in object detection problems.

As seen in Figure 5, intersection over union (IoU) calculates how much two boxes overlap. In the context of object identification and segmentation, IoU assesses the overlap of ground truth and prediction regions [43]. To compare the performance of the models, we utilized mAP@IoU = 0.5 and mAP@IoU = 0.5:0.95, precision, recall, and F1 score. If the IoU is higher than the specified threshold, it is calculated as TP (true positive). If it is low, it is calculated as FP (false positive). Using the TP, FP, TN (true negative), FN (false negative) values, we calculated precision, recall, and mAP using Equations (3)–(6). The threshold value is 0.5 at mAP@IoU:0.5 and has taken ten different values in steps of 0.05 between 0.5 and 0.95 at mAP@IoU:0.5:0.95. The classification results can be divided into four categories—"true positive (TP)” indicates that the outcome is the accurate categorization of the insects in the image, “true negative (TN)” indicates that the insects in the image do not fall into a particular category, “false positive (FP)” indicates that the outcome is the incorrect identification of the insects from the particular class as the insect is from the other class, and “false negative (FN)” indicates that the outcome is the incorrect categorization of the insects belonging to a particular class is misclassified as not belonging to that particular class.

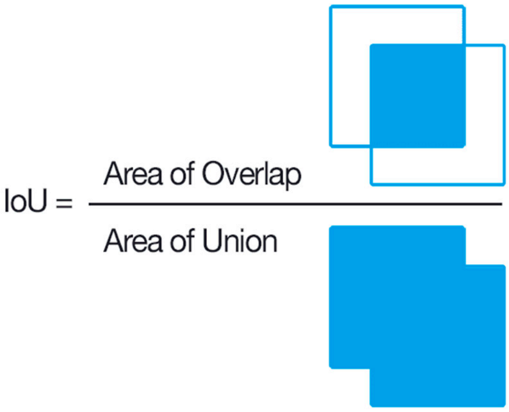


Figure 5. IoU calculation.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{3}$$

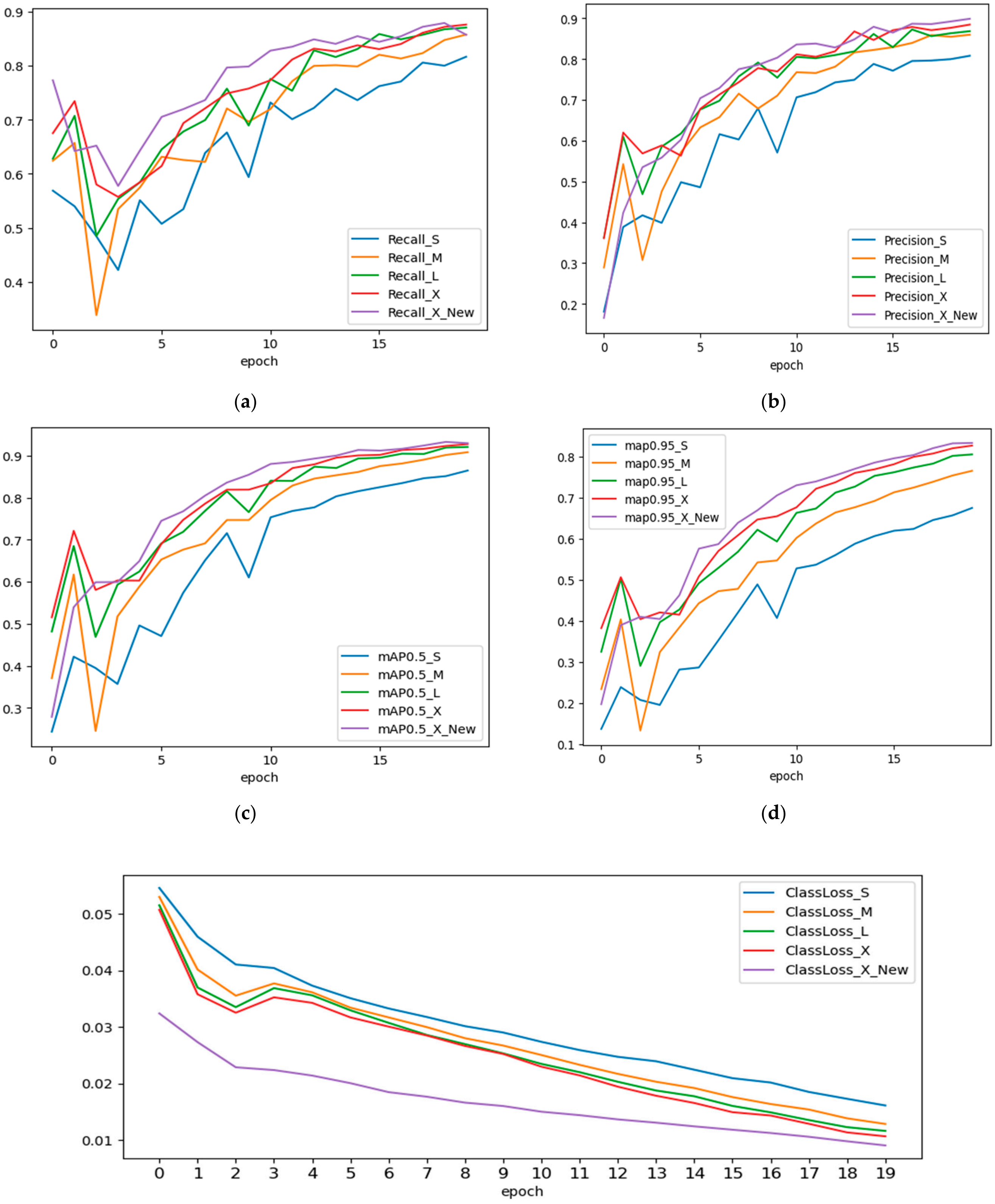
$$\text{Recall} = \frac{TP}{TP + FN} \tag{4}$$

$$\text{mAP} = \frac{1}{N} \sum\nolimits_{i=1}^{N} AP\text{i} \tag{5}$$

$$\text{F1} = 2 \times \frac{Precision * Recall}{Precision + Recall} \tag{6}$$

As a result of the training, we examined the precision, recall, mAP@IoU:0.5, and mAP@IoU:0.5:0.95 measures. The models’ training was restarted from scratch, all model metrics are shown to fluctuate, as shown in Figure 6a–d, notably during the first ten epochs of training. Increasing the number of epochs will help to lessen these swings. After the 20th epoch, we also noticed the drop in precision and recall values in the YOLOv5s and YOLOv5m models. This is due to overfitting, causing it to perform poorly on the test data by a drop in precision and recall values after the 20th epoch. The output of five different model iterations is shown in Figure 7.

To examine the experimental outcomes of YOLOv5 (s, m, l, x, and modified x), 1538 unseen photos were used as a test set. Table 3 presents the results of the statistical indicators of the recognition performance of five different YOLO architectures used in the experimental study. On the same insect dataset, all five models were trained. When comparing the models, the hyperparameters were held constant. The comparison findings demonstrate that the performance of YOLOv5s was less accurate than the other models with precision, recall, a minimum of mAP@0.5, mAP@0.5:0.95, and F1 score.



(**a**) (**b**)

(**c**) (**d**)

(**e**)

Figure 6. (**a**) Precision, (**b**) recall, (**c**) mAP@0.5 (**d**) mAP@0.95 and (**e**) loss curves of models used in experimental study.

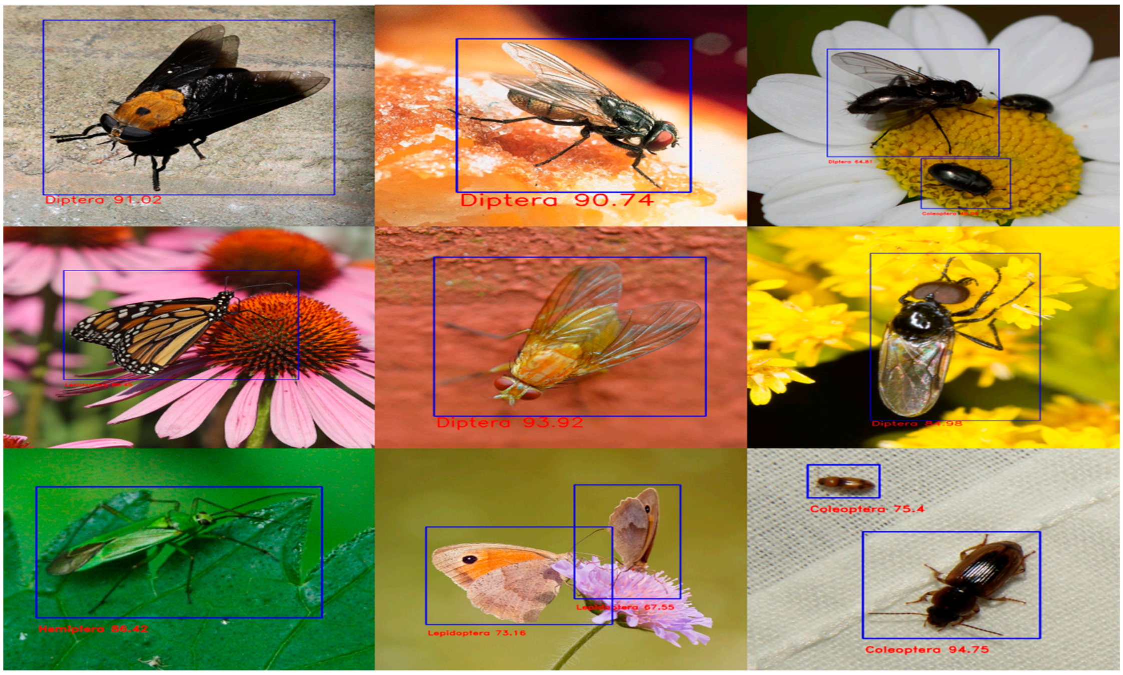


Figure 7. Detection results of YOLOv5x on different insect orders.

Table 3. Performance comparison of all five models used in the experimental study using statistical indicators.

Model	**Training**				**Validation**				**Testing**			
	P	**R**	**F1**	**mAP**	**P**	**R**	**F1**	**mAP**	**P**	**R**	**F1**	**mAP**
Version-5s	0.808	0.816	0.811	0.864	0.806	0.821	0.813	0.865	0.812	0.799	0.850	0.851
Version-5m	0.859	0.857	0.857	0.908	0.861	0.857	0.858	0.908	0.836	0.855	0.865	0.889
Version-5l	0.868	0.870	0.869	0.920	0.870	0.869	0.869	0.921	0.865	0.854	0.859	0.908
Version-5x	0.884	0.875	0.880	0.928	0.883	0.876	0.879	0.928	0.882	0.846	0.863	0.912
New Version-5x	0.899	0.857	0.877	0.930	0.898	0.856	0.876	0.930	0.868	0.886	0.878	0.922

Based on the statistical data shown in Table 3, 'YOLOv5x' developed under the trained model provides excellent results in terms of recognition accuracy. This makes it suitable for real-time pest detection with field adaption.

We randomly picked pictures from the test set to demonstrate our experiments' outcomes better. As seen in Figure 7, we discovered that the model could distinguish between various insect species with multiple backgrounds, plant types, and lighting conditions. Once the practical model in terms of mAP was discovered, i.e., YoloV5x with CBAM, the same model was used to bring down the identification to species level, as shown in Figure 8, by manual labelling the 3000 images present in the following selected insects from various classes—Diptera class (Bactrocera Dorsalis, house fly, honey bee, mosquito), Odonata class (dragon fly), Lepidoptera class (butterfly) and Araneae (spider) of the same dataset [34].

The confusion matrix is a method for summarizing a classification algorithm's performance. The diagonal line shows the significance of the prediction outcomes in the confusion matrix; the horizontal and vertical lines represent false negatives and false positives, respectively. Figure 9 illustrates the confusion matrix of five YOLO models used in the experimental study conducted on seven classes of insects.

The YOLOv5x-based insect detection and classification model is lightweight because we employed effective methods such as mixed-precision training, gradient accumulation, and reduced batch sizes to minimize memory usage during training. Remember that some of these methods can cause a very tiny drop in accuracy [44]. The outcomes of the insect recognition paradigm are contrasted with those of other models in Table 4. The developed model can be employed in insect monitoring devices to maximize its efficacy in a particular field or location by raising the capture rate of target insect species while reducing the capture of non-target species. The YOLOv5x-based insect detection system can assist in decreasing the usage of pesticides and other hazardous insect control techniques [45].

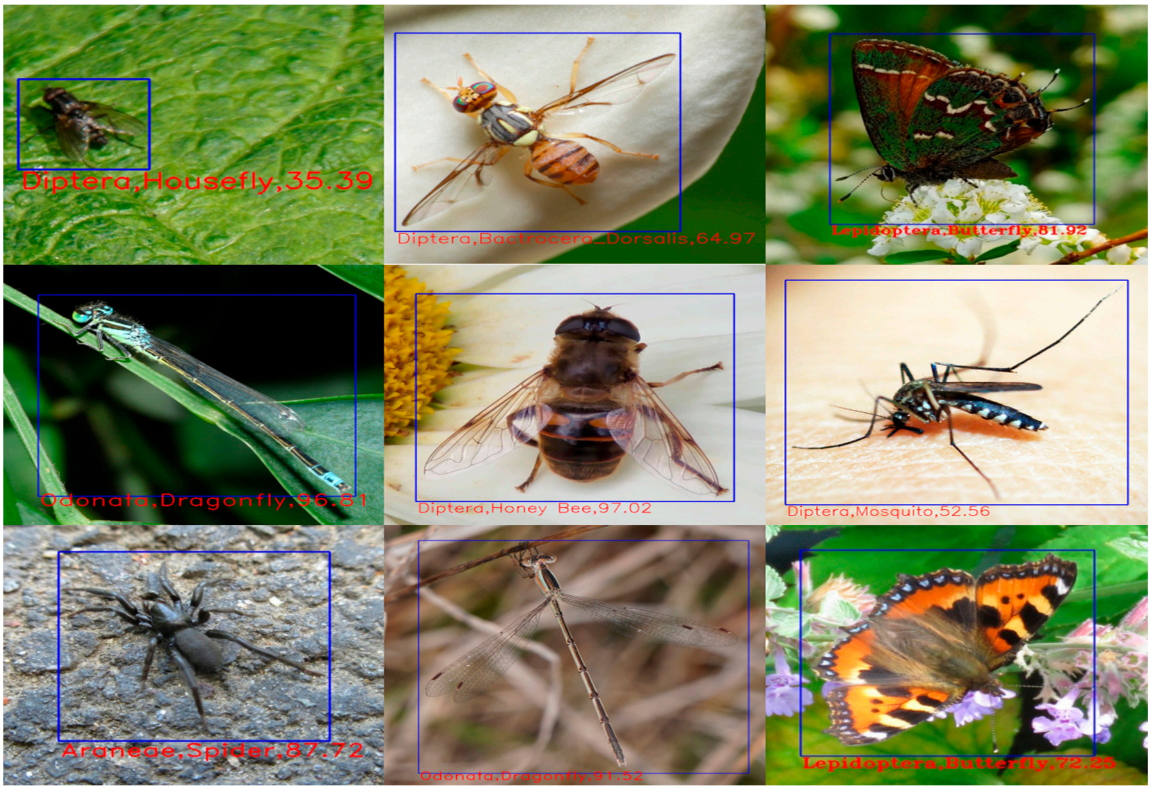
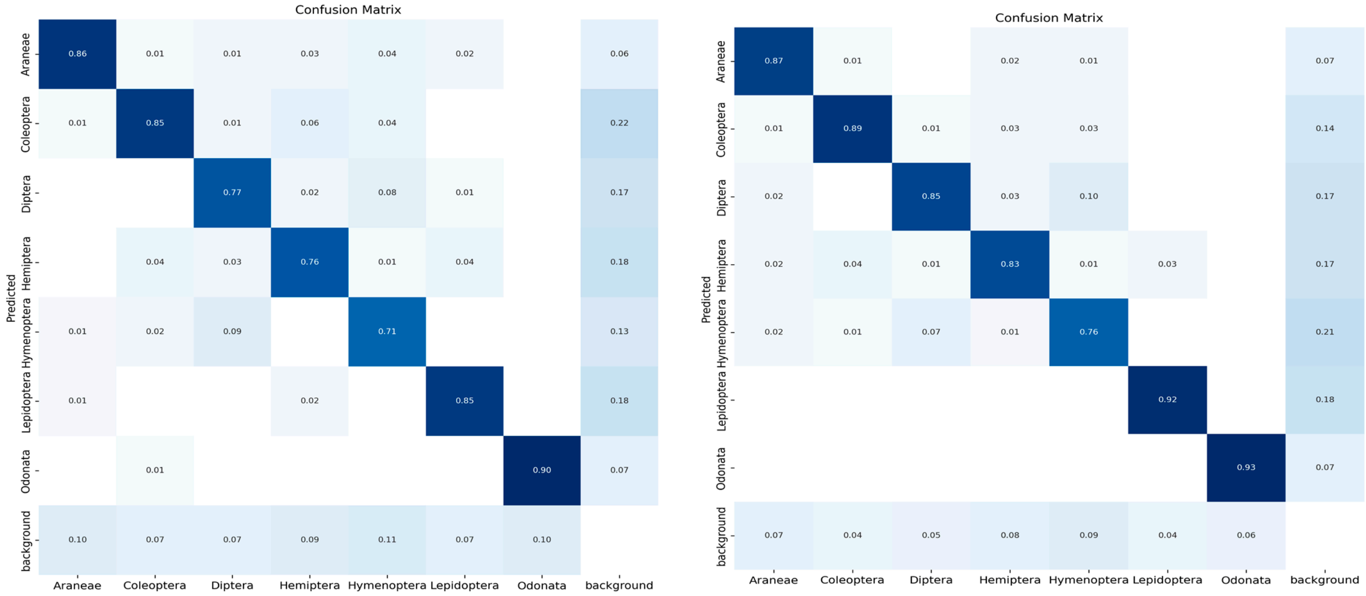
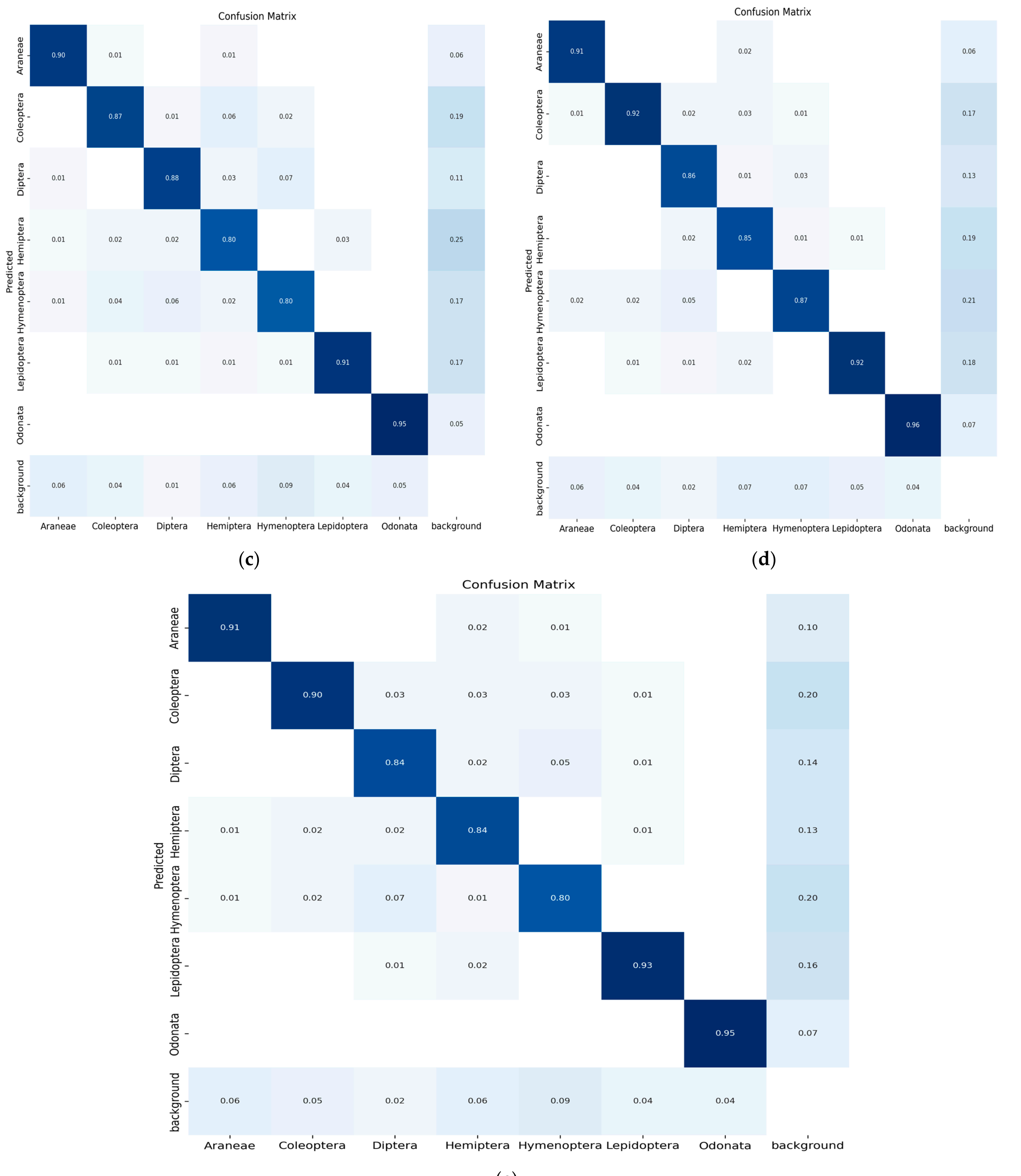


Figure 8. Detection results of YOLOv5x on different insect species.



(**a**) (**b**)

Figure 9. *Cont.*



(**c**) (**d**)

(**e**)

Figure 9. (**a**) Confusion matrix (YOLO-v5s), (**b**) confusion matrix (YOLO-v5m), (**c**) confusion matrix (YOLO-v5l), (**d**) confusion matrix (YOLO-v5x) and (**e**) confusion matrix (modified YOLO-v5x).

Table 4. Comparative table on various deep learning model used for insect classification and detection.

Reference	Dataset	Model	No. of Insects in One Picture	Purpose	Accuracy
[21]	24 classes of insects from internet	VGG19	One Insect	Classification	89.22
[44]	8 classes of stored grain insects	DenseNet-121	Multiple Insect	Classification and Detection	88.06
[46]	5 classes of insects	CNN	One Insect	Classification	90
[47]	5 classes of insects	Faster R-CNN	One Insect	Classification	98.9
[42]	DIRT Dataset (Olive fruit fly)	MobileNet	One Insect	Classification	96.89
Proposed Model	7 classes of flying insects	YOLOv5 + CBAM	Multiple Insect	Classification and Detection	93

5. Conclusions

The YOLOv5 single-stage object detection architecture with five variations (s, m, l, x, and modified-x) was used in this study to create an order and species identification system that can identify insect pests in image files. We used a Kaggle dataset of pictures captured in actual outdoor settings. Using an Nvidia Tesla T4 16GB GPU, we trained, validated, and tested the system on the AWS cloud. Through learning on a self-created pest dataset, the YOLOv5x and modified YOLOv5x models among all YOLO models produced more accurate findings; the F1 score is close to 0.90, and the mAP value is 93%. Insect detection systems based on YOLOv5x and modified YOLOv5x are suitable for field adaption in real-time pest recognition systems; the F1 score of modified YOLOv5x increased by 0.02 and the mAP by 1% compared to YOLOv5x. In comparison to other YOLO models, YOLOv5x is a cutting-edge insect identification model that achieves excellent accuracy and real-time performance in the field of computational entomology. In the future, it may be thought to combine MobileNet with YOLOv5x to increase the detection speed significantly. The developed model can be employed by efficient training on harmful insect data in insect monitoring devices to reduce the usage of pesticides and other hazardous insect control techniques.

Author Contributions: N.K., conceptualization, methodology, software, validation and writing; N., supervision and writing; F.F., conceptualization—knowledge discovery, review. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Available online: https://www.kaggle.com/mistag/arthropod-taxonomy-orders-object-detection-dataset (accessed on 1 December 2022).

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

DL	Deep Learning
ML	Machine Learning
SVM	Support Vector Machine
CNN	Convolutional neural network
WSN	Wireless Sensor Network
SPP	Spatial Pyramid Pooling
CBAM	Convolutional Block Attention Module
CBL	CBAM layer
CSP	Cross Stage Partial
BN	Batch Normalization
ReLU	Rectified Linear Unit
mAP	Mean Average Precision
FPN	Feature Pyramid Network

References

1. Cheng, D.F.; Wu, K.M.; Tian, Z.; Wen, L.P.; Shen, Z.R. Acquisition and analysis of migration data from the digitised display of a scanning entomological radar. *Comput. Electron. Agric.* **2002**, *35*, 63–75. [CrossRef]
2. Martineau, M.; Conte, D.; Raveaux, R.; Arnault, I.; Munier, D.; Venturini, G. A survey on image-based insect classification. *Pattern Recognit.* **2017**, *65*, 273–284. [CrossRef]
3. Thenmozhi, K.; Reddy, U.S. Crop pest classifcation based on deep convolutional neural network and transfer learning. *Comput. Electron. Agric.* **2009**, *164*, 104906. [CrossRef]
4. Ngô-Muller, V.; Garrouste, R.; Nel, A. Small but important: A piece of mid-cretaceous burmese amber with a new genus and two new insect species (odonata: Burmaphlebiidae & 'psocoptera': Compsocidae). *Cretac. Res.* **2020**, *110*, 104405. [CrossRef]
5. Serres, J.R.; Viollet, S. Insect-inspired vision for autonomous vehicles. *Curr. Opin. Insect Sci.* **2018**, *30*, 46–51. [CrossRef] [PubMed]
6. Fox, R.; Harrower, C.A.; Bell, J.R.; Shortall, C.R.; Middlebrook, I.; Wilson, R.J. Insect population trends and the IUCN red list process. *J. Insect Conserv.* **2019**, *23*, 269–278. [CrossRef]
7. Krawchuk, M.A.; Meigs, G.W.; Cartwright, J.M.; Coop, J.D.; Davis, R.; Holz, A.; Kolden, C.; Meddens, A.J. Disturbance refugia within mosaics of forest fire, drought, and insect outbreaks. *Front. Ecol. Environ.* **2020**, *18*, 235–244. [CrossRef]
8. Gullan, P.J.; Cranston, P.S. *The Insects: An Outline of Entomology*; Wiley: Hoboken, NJ, USA, 2014.
9. Wheeler, W.C.; Whiting, M.; Wheeler, Q.D.; Carpenter, J.M. The Phylogeny of the Extant Hexapod Orders. *Cladistics* **2001**, *17*, 113–169. [CrossRef]
10. Kumar, N.; Nagarathna. Survey on Computational Entomology: Sensors based Approaches to Detect and Classify the Fruit Flies. In Proceedings of the 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Kharagpur, India, 1–3 July 2020; pp. 1–6. [CrossRef]
11. Amarathunga, D.; Grundy, J.; Parry, H.; Dorin, A. Methods of insect image capture and classification: A Systematic literature review. *Smart Agric. Technol.* **2021**, *1*, 100023. [CrossRef]
12. Yang, Z.; He, W.; Fan, X.; Tjahjadi, T. PlantNet: Transfer learning based fine-grained network for high-throughput plants recognition. *Soft Comput.* **2022**, *26*, 10581–10590. [CrossRef]
13. Rehman, M.Z.U.; Ahmed, F.; Khan, M.A.; Tariq, U.; Jamal, S.S.; Ahmad, J.; Hussain, L. Classification of citrus plant diseases using deep transfer learning. *Comput. Mater. Continua* **2022**, *70*, 1401–1417. [CrossRef]
14. Xiao, Z.; Yin, K.; Geng, L.; Wu, J.; Zhang, F.; Liu, Y. Pest identification via hyperspectral image and deep learning. *Signal Image Video Process.* **2022**, *16*, 873–880. [CrossRef]
15. Amrani, A.; Sohel, F.; Diepeveen, D.; Murray, D.; Jones, M. Deep learning-based detection of aphid colonies on plants from a reconstructed Brassica image dataset. *Comput. Electron. Agric.* **2023**, *205*, 107587. [CrossRef]
16. Wang, Q.J.; Zhang, S.Y.; Dong, S.F.; Zhang, G.C.; Yang, J.; Li, R.; Wang, H.Q. Pest24: A large-scale very small object data set of agricultural pests for multi-target detection. *Comput. Electron. Agric.* **2020**, *175*, 105585. [CrossRef]
17. Amrani, A.; Sohel, F.; Diepeveen, D.; Murray, D.; Jones, M. Insect detection from imagery using YOLOv3-based adaptive feature fusion convolution network. *Crop Pasture Sci.* **2023**. [CrossRef]
18. Zhang, J.P.; Li, Z.W.; Yang, J. A parallel SVM training algorithm on large-scale classification problems. In Proceedings of the 2005 International Conference on Machine Learning and Cybernetics, Guangzhou, China, 18–21 August 2005. [CrossRef]
19. Xia, D.; Chen, P.; Wang, B.; Zhang, J.; Xie, C. Insect detection and classification based on an improved convolutional neural network. *Sensors* **2018**, *18*, 4169. [CrossRef]
20. Shi, Z.; Dang, H.; Liu, Z.; Zhou, X. Detection and Identification of Stored-Grain Insects Using Deep Learning: A More Effective Neural Network. *IEEE Access* **2020**, *8*, 163703–163714. [CrossRef]
21. Mamdouh, N.; Khattab, A. YOLO-Based Deep Learning Framework for Olive Fruit Fly Detection and Counting. *IEEE Access* **2021**, *9*, 84252–84262. [CrossRef]
22. Sciarretta, A.; Tabilio, M.R.; Amore, A.; Colacci, M.; Miranda, M.A.; Nestel, D.; Papadopoulos, N.T.; Trematerra, P. Defining and evaluating a decision support system (DSS) for the precisepest management of the Mediterranean fruit fly, *Ceratitis capitata*, at the farm level. *Agronomy* **2019**, *9*, 608. [CrossRef]
23. Potamitis, I.; Rigakis, I.; Tatlas, N.A. Automated surveillance of fruitflies. *Sensors* **2017**, *17*, 110. [CrossRef]
24. Doitsidis, L.; Fouskitakis, G.N.; Varikou, K.N.; Rigakis, I.I.; Chatzichristo, S.A.; Papafilippaki, A.K.; Birouraki, A.E. Remote monitoring of the *Bactrocera oleae* (Gmelin) (Diptera: Tephritidae) population using an automated McPhail trap. *Comput. Electron. Agric.* **2017**, *137*, 69–78. [CrossRef]
25. Tirelli, P.; Borghese, N.A.; Pedersini, F.; Galassi, G.; Oberti, R. Automatic monitoring of pest insects traps by Zigbee-based wireless networking of image sensors. In Proceedings of the 2011 IEEE International Instrumentation and Measurement Technology Conference, Hangzhou, China, 10–12 May 2011; pp. 1–5. [CrossRef]
26. Sun, C.; Flemons, P.; Gao, Y.; Wang, D.; Fisher, N.; La Salle, J. Automated image analysis on insect soups. In Proceedings of the 2016 International Conference on Digital Image Computing: Techniques and Applications (DICTA), Gold Coast, QLD, Australia, 30 November–2 December 2016; pp. 1–6. [CrossRef]
27. Philimis, P.; Psimolophitis, E.; Hadjiyiannis, S.; Giusti, A.; Perelló, J.; Serrat, A.; Avila, P. A centralised remote data collection system using automated traps for managing and controlling the population of the Mediterranean (*Ceratitis capitata*) and olive (*Dacus oleae*) fruit flies. In Proceedings of the First International Conference on Remote Sensing and Geoinformation of the Environment (RSCy2013), Paphos, Cyprus, 8 April 2013. [CrossRef]

28. Kaya, Y.; Kayci, L. Application of artificial neural network for automatic detection of butterfly species using color and texture features. *Vis. Comput.* **2014**, *30*, 71–79. [CrossRef]
29. Zhong, Y.; Gao, J.; Lei, Q.; Zhou, Y. A vision-based counting and recognition system for flying insects in intelligent agriculture. *Sensors* **2018**, *18*, 1489. [CrossRef] [PubMed]
30. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788. [CrossRef]
31. Kalamatianos, R.; Karydis, I.; Doukakis, D.; Avlonitis, M. DIRT: The dacus image recognition toolkit. *J. Imaging* **2018**, *4*, 129. [CrossRef]
32. Ding, W.; Taylor, G. Automatic moth detection from trap images for pest management. *Comput. Electron. Agric.* **2016**, *123*, 17–28. [CrossRef]
33. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
34. Kaggle. Available online: https://www.kaggle.com/mistag/arthropod-taxonomy-orders-object-detection-dataset (accessed on 1 December 2022).
35. Patrício, D.I.; Rieder, R. Computer vision and artificial intelligence in precision agriculture for grain crops: A systematic review. *Comput. Electron. Agric.* **2018**, *153*, 69–81. [CrossRef]
36. Murthy, C.B.; Hashmi, M.F.; Muhammad, G.; AlQahtani, S.A. YOLOv2pd: An efficient pedestrian detection algorithm using improved YOLOv2 model. *Comput. Mater. Contin.* **2021**, *69*, 3015–3031. [CrossRef]
37. Wang, Y.M.; Jia, K.B.; Liu, P.Y. Impolite pedestrian detection by using enhanced YOLOv3-Tiny. *J. Artif. Intell.* **2020**, *2*, 113–124. [CrossRef]
38. Wu, Z. Using YOLOv5 for Garbage Classification. In Proceedings of the 2021 4th International Conference on Pattern Recognition and Artificial Intelligence (PRAI), Yibin, China, 20–22 August 2021; pp. 35–38. [CrossRef]
39. Han, S.; Dong, X.; Hao, X.; Miao, S. Extracting Objects' Spatial–Temporal Information Based on Surveillance Videos and the Digital Surface Model. *ISPRS Int. J. Geo-Inf.* **2022**, *11*, 103. [CrossRef]
40. Jaderberg, M.; Simonyan, K.; Zisserman, A. Spatial transformer networks. *Adv. Neural Inf. Process. Syst.* **2015**, *28*, 2017–2025. [CrossRef]
41. Hou, G.; Qin, J.; Xiang, X.; Tan, Y.; Xiong, N.N. AF-net: A medical image segmentation network based on attention mechanism and feature fusion. *Comput. Mater. Contin.* **2021**, *69*, 1877–1891. [CrossRef]
42. Fukui, H.; Hirakawa, T.; Yamashita, T. Attention branch network: Learning of attention mechanism for visual explanation. In Proceedings of the CVPR, Long Beach, CA, USA, 15–20 June 2019; pp. 10705–10714. [CrossRef]
43. Rezatofighi, H.; Tsoi, N.; Gwak, J.; Sadeghian, A.; Reid, I.; Savarese, S. Generalized Intersection Over Union: A Metric and a Loss for Bounding Box Regression. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 658–666. [CrossRef]
44. Redmon, J.; Farhadi, A. YOLO9000: Better, faster, stronger. In Proceedings of the CVPR, Honolulu, HI, USA, 21–26 July 2017; pp. 7263–7271. [CrossRef]
45. Ahmad, I.; Yang, Y.; Yue, Y.; Ye, C.; Hassan, M.; Cheng, X.; Wu, Y.; Zhang, Y. Deep Learning Based Detector YOLOv5 for Identifying Insect Pests. *Appl. Sci.* **2022**, *12*, 10167. [CrossRef]
46. Kasinathan, T.; Singaraju, D.; Uyyala, S.R. Insect classification and detection in field crops using modern machine learning techniques. *Inform. Process. Agric.* **2020**, *8*, 446–457. [CrossRef]
47. Karar, M.E.; Alsunaydi, F.; Albusaymi, S.; Alotaibi, S. A new mobile application of agricultural pests recognition using deep learning in cloud computing system. *Alex. Eng. J.* **2021**, *60*, 4423–4432. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.