

## Article

# Recommending Advanced Deep Learning Models for Efficient Insect Pest Detection

Wei Li <sup>1,\*</sup> , Tengfei Zhu <sup>1</sup>, Xiaoyu Li <sup>1</sup>, Jianzhang Dong <sup>2</sup> and Jun Liu <sup>3</sup>

<sup>1</sup> School of Instrument Science and Engineering, Southeast University, Nanjing 210096, China; 220203584@seu.edu.cn (T.Z.); 220203613@seu.edu.cn (X.L.)

<sup>2</sup> College of Software Engineering, Southeast University, Suzhou 215123, China; 220215781@seu.edu.cn

<sup>3</sup> Institute of Agricultural Facilities and Equipment, Jiangsu Academy of Agricultural Sciences, Nanjing 210014, China; liujun@jaas.ac.cn

\* Correspondence: li-wei@seu.edu.cn

**Abstract:** Insect pest management is one of the main ways to improve the crop yield and quality in agriculture and it can accurately and timely detect insect pests, which is of great significance to agricultural production. In the past, most insect pest detection tasks relied on the experience of agricultural experts, which is time-consuming, laborious and subjective. In recent years, various intelligent methods have emerged for detection. This paper employs three frontier Deep Convolutional Neural Network (DCNN) models—Faster-RCNN, Mask-RCNN and Yolov5, for efficient insect pest detection. In addition, we made two coco datasets by ourselves on the basis of Baidu AI insect detection dataset and IP102 dataset, and compared these three frontier deep learning models on the two coco datasets. In terms of Baidu AI insect detection dataset whose background is simple, the experimental results strongly recommend Yolov5 for the insect pest detection, because its accuracy reaches above 99% while Faster-RCNN's and Mask-RCNN's reach above 98%. Meanwhile, Yolov5 has the faster computational speed than Faster-RCNN and Mask-RCNN. Comparatively speaking, with regard to the IP102 dataset whose background is complex and categories are abundant, Faster-RCNN and Mask-RCNN have the higher accuracy, reaching 99%, than Yolov5 whose accuracy is about 97%.

**Keywords:** insect pest detection; deep learning; Yolov5; Faster-RCNN; Mask-RCNN



**Citation:** Li, W.; Zhu, T.; Li, X.; Dong, J.; Liu, J. Recommending Advanced Deep Learning Models for Efficient Insect Pest Detection. *Agriculture* **2022**, *12*, 1065. <https://doi.org/10.3390/agriculture12071065>

Academic Editors: Nen-Fu Huang and Ho-Hsien Chen

Received: 23 June 2022

Accepted: 13 July 2022

Published: 21 July 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

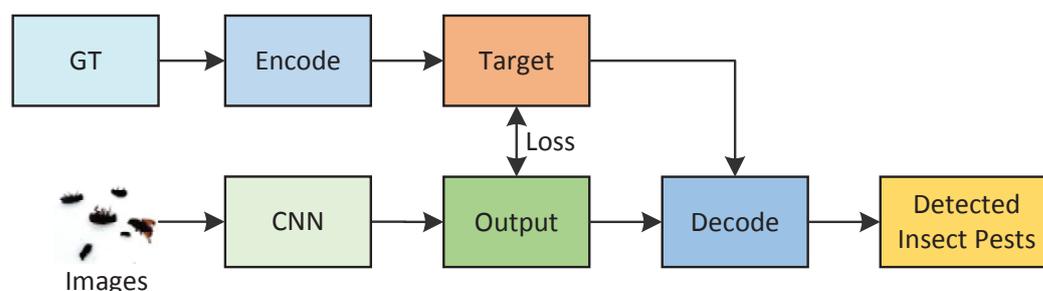
In agricultural production management, the detection of insect pests has always been a critical problem [1]. Insect pests are responsible for 20% of annual crop losses worldwide [2]. Therefore, the control of insect pests plays a crucial role in agricultural production, which has an important impact on agricultural development, grain production and farmers' income increase [3]. Insect pest detection is a very challenging task in agricultural image processing [4]. Traditional insect pest detection methods have the drawback of requiring well-trained taxonomists to detect insect pests based on morphological features accurately [5]. With the development of computer science, intelligent technology has won growing attention in agricultural applications [6]. The scientific and effective management through intelligent algorithms can not only become a technical way to replace the traditional manual detection to improve the efficiency [7], but also avoid the spread of insect pests in a large area [8], thus improving the quality of crop products. In particular, deep learning technology is pushing forward the agriculture from the traditional style to the intelligent one.

The insect pest detection in agriculture is an important prerequisite for the prediction, prevention and control of them, so as to improve the yield in prediction and control actions [9]. Image detection methods have the advantages of high efficiency, low cost and easy operation for detecting the insect pests. These methods can timely and accurately detect the types of insect pests and provide necessary information for farmers to take

measures to prevent and control the spread of insect pests [10]. Hence, image-based detection methods have become research hotspots of insect pest control in the past few years [11]. Generally speaking, there are a lot of traditional image detection methods suitable for insect pest detection, such as the detection methods based on the threshold [12], edge [13], region test [14] and graph theory [15]. However, these traditional detection methods are appropriate for a small number of samples or small scope of detection, and the detection accuracies are not very stable [16]. When the number of samples to be detected is large and the types of samples are complex, the accuracy rate will decrease sharply.

Deep learning can overcome the disadvantages of traditional image detection methods and thus it has drawn growing attention in the field of detecting the agricultural insect pests [17]. Deep learning can alleviate the problems of low detection and recognition accuracy in the case of detecting multiple complex types of samples due to its advantages of high precision and strong adaptability [18]. Broadly speaking, one-stage algorithms can directly extract features from images in an end-to-end framework to predict object location and classification [19]. The accuracies of the earlier one-stage algorithms are not as high as the two-stage algorithms [20]. However, the one-stage algorithms have a faster computational speed than the two-stage algorithms [21]. The two-stage algorithms usually generate candidate areas containing the objects to be inspected in the first stage and then implement further fine-grained object detection in the second stage [22]. Therefore, the two-stage algorithms have relatively slower speed than the one-stage algorithms, but have higher accuracy. However, with the development of one-stage algorithms, some one-stage algorithms have surpassed the two-stage algorithms in terms of accuracy [23].

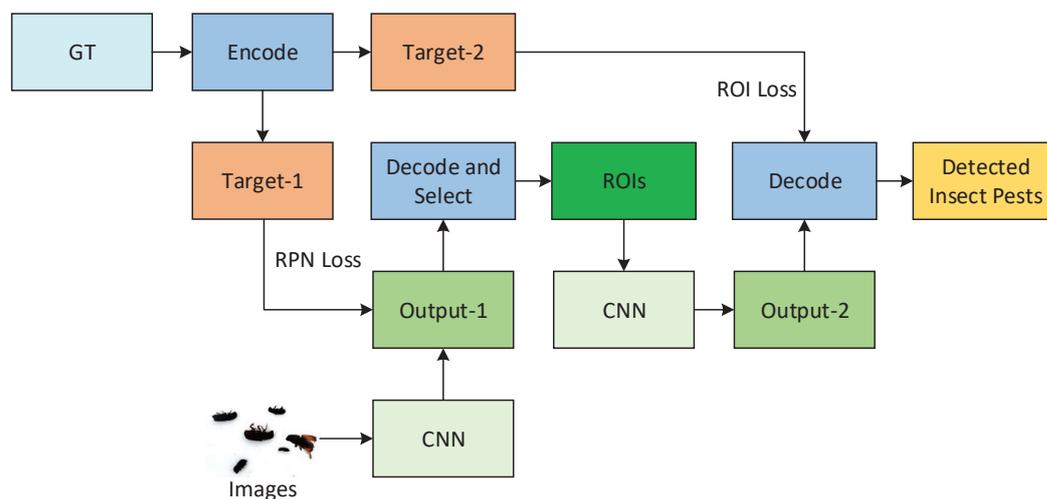
The one-stage algorithms can directly generate the category probability and coordinates of objects for detection. Comparing with the two-stage algorithms, the one-stage algorithms do not need the region proposal stage, and therefore the overall process is relatively simple [24]. As shown in Figure 1, during training process, output can be generated through CNN, and the corresponding detected images can be generated after decoding process. In addition, during test process, Ground Truth (GT) needs to be encoded to put into the output to calculate the corresponding losses.



**Figure 1.** Flow chart of one-stage algorithms.

For the two-stage algorithms, CNN has the functions of feature extraction, feature selection and feature classification [25]. The typical two-stage algorithms for detection include Faster-RCNN and Mask-RCNN, whose components include window sliding, region generation, image classification of candidate regions, and post-processing. In the two-stage algorithms, the first stage takes charge of detecting the position of the object, and the second stage is responsible for further refining the results of the first stage for each candidate region.

As shown in Figure 2, during training process, output-1 can be generated through CNN in the first phase, and then the decoding and selecting process generates candidate areas, and obtains appropriate Region of Interests (ROIs) based on the output-1. Then the output-2 is generated in the second phase for further elaboration. At last, the corresponding detected images are generated after decoding process. In addition, during test process, GT needs to be encoded to put into the output to calculate the corresponding losses.



**Figure 2.** Flow chart of two-stage algorithms.

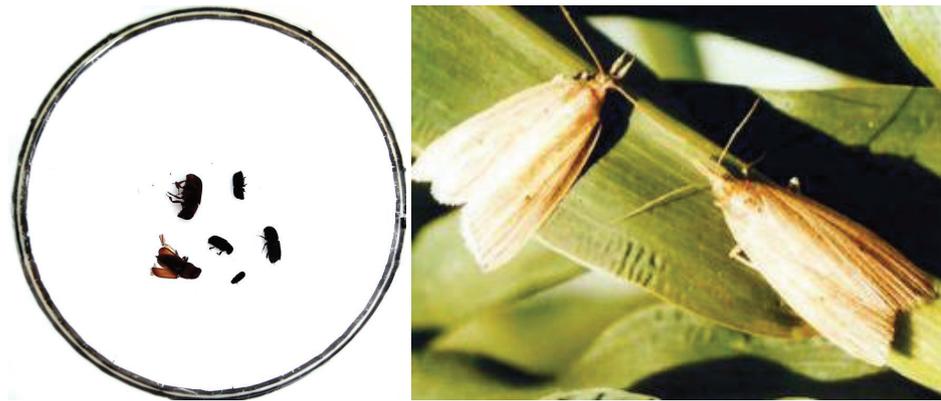
In this paper, we investigate the advanced deep learning approaches to uncover their potentials for the issue of insect pest detection. The experiments mainly include detecting insect pests based on the two coco datasets: the one is to judge whether insect pests are small or normal according to the size of the individual based on Baidu AI insect detection dataset [26]; the other is to judge the categories of insect pests based on IP102 dataset [27]. The experimental results are then compared to determine which algorithm is the most suitable for insect pest detection on the different datasets.

The goal of this paper is to recommend the most suitable advanced deep learning models for accurate and fast insect pest detection in different image data situations. To this end, we probe into their efficiency on different datasets. On the basis of Baidu AI insect detection dataset and IP102 dataset, we made the two coco datasets by ourselves [28]. We evaluate and compare the advanced algorithms—Faster-RCNN ResNet50 [29], Mask-RCNN ResNet50 [30] and Yolov5 Darknet53 [31] for the two coco datasets. In simple terms, the three algorithms on the basis of Baidu AI insect detection dataset have been fine-tuned to adapt to our dataset and finally achieve the accuracy of 98.1%, 98.4% and 99.5%, respectively. In addition, the computational speed of Yolov5 is the fastest. Hence, it has huge potential in the application of real-time insect pest detection. While for the IP102 dataset, Yolov5 has the fastest speed among the three algorithms [32]. Meanwhile, the accuracies of Faster-RCNN and Mask-RCNN are higher, both reaching 99% while Yolov5's is lower, reaching 97%. This research can contribute to achieving the United Nations' 2030 Agenda for Sustainable Development.

## 2. Materials and Methods

### 2.1. Collection of Insect Pests

It is common for the sample collection to capture the images of insect pests under remote type of photographed lamp on automation equipment in agriculture, such as Baidu AI insect detection dataset [33]. In addition, using camera directly to photograph insect pests in the field is also a common way, such as IP102 dataset. Figure 3 illustrates the images of insect pests obtained by the two methods mentioned above.



**Figure 3.** Two images of insect pests captured under a photographed lamp and photographed by cameras directly in the field successively.

## 2.2. Datasets

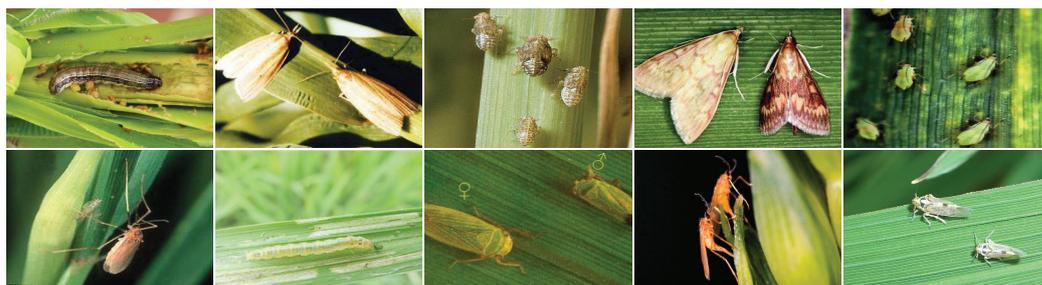
### 2.2.1. Insect Pest Detection

High-precision detection of the pest size and category lays a foundation for higher pesticide use efficiency, which is of great significance to the realization of precision agriculture. We choose two datasets for the detection of sizes and categories of insect pests. In Baidu AI insect detection dataset, the pest size is significantly different. According to the distribution of pest size, pesticide can be used more accurately in insect pest control. In IP102 dataset, we focus on ten specific categories: army worm, asiatic rice borer, brown plant hopper, corn borer, English grain aphid, rice gall midge, rice leaf roller, rice leafhopper, wheat blossom midge and white backed plant hopper. The reason why we choose these categories is that they are mainly in rice, wheat and corn fields and can represent typical insect pests in common crops.

Figure 4 shows the pest sample images with different sizes selected from Baidu AI insect detection dataset. Figure 5 shows sample images including ten categories selected from IP102 dataset.



**Figure 4.** Two images selected from Baidu AI insect detection dataset.



**Figure 5.** Images including ten categories selected from IP102 dataset. The first gives include the examples of army worm, asiatic rice borer, brown plant hopper, corn borer and English grain aphid; the second row gives the examples of rice gall midge, rice leaf roller, rice leafhopper, wheat blossom midge and white backed plant hopper.

### 2.2.2. Insect Pest Samples Acquisition

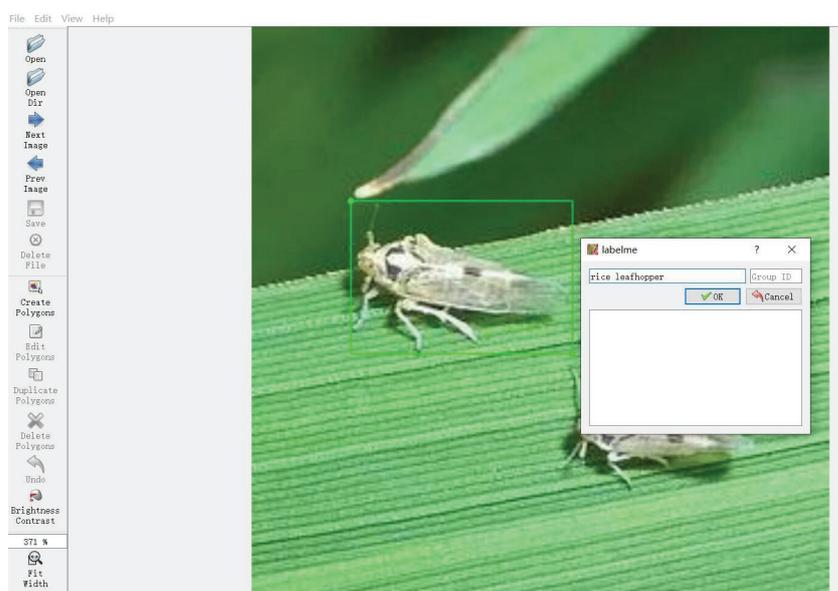
In Baidu AI insect detection dataset whose background is simple, each image contains different sizes of insect pests. We randomly selected 1000 images from this dataset for method evaluation and comparison. Among these 1000 images, we randomly chose 800 images to create the coco dataset for training and validation and utilized remaining images for testing.

Besides, in IP102 dataset whose background is complex and categories are abundant, we chose 10 categories of insect pests, mainly in the rice, wheat and corn fields. We selected 2000 images from this dataset. In these 2000 images, 1600 images were used for training and validation and the remaining images were used for model testing.

### 2.2.3. Data Labeling

In order to train Faster-RCNN, Mask-RCNN and Yolov5, we manually labeled the images in the datasets by Labelme. Labelme is an image tagging tool developed by MIT's Computer Science and Artificial Intelligence Laboratory that creates image tagging tasks. Through Labelme, a boundary box is drawn around each insect pest in the image.

As shown in Figure 6, the boundary box can effectively mark the insect pests. According to the way of marking selection, a file is generated after we mark each image, and the file stores the coordinate and category information of the insect pests. Figure 6 is an example of making a coco dataset by Labelme.



**Figure 6.** We adopt Labelme to annotate each insect pest in an image by marking a rectangle.

### 2.3. Model Evaluation

In order to verify the experimental efficacy of the coco datasets of insect pests made by ourselves on the basis of Baidu AI insect detection dataset and IP102 dataset in different DCNN models, we use three DCNN models to verify the coco datasets. This paper mainly studies two kinds of algorithms—one-stage algorithm and two-stage algorithm. The one-stage algorithm used in this paper is Yolov5, the two-stage algorithms used in this paper are Faster-RCNN and Mask-RCNN.

#### 2.3.1. Model 1: Yolov5 Darknet53

Yolov5 is a one-stage algorithm. Our study employed the Yolov5 Darknet53 model. Because there are 53 convolutional layers in the network, it is called Darknet53. As shown in Figure 7, the network structure diagram of Yolov5 is mainly divided into four parts: Input, Backbone, Neck and Prediction [34]. The general process is briefly described in the following. The images flow into the input layer for preprocessing, and then flow into the backbone layer for feature extraction. After that, the features flow into the neck layer for feature integration. Finally, the bounding boxes (bbox) are generated by the prediction layer.

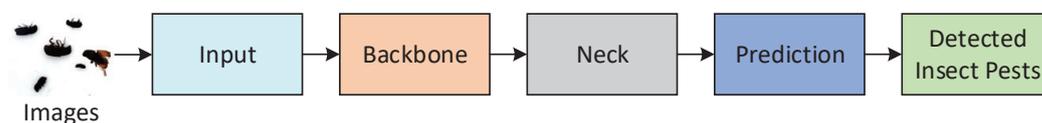


Figure 7. Yolov5 Darknet53 model.

Generally speaking, the detection accuracy of one-stage algorithms is lower than two-stage algorithms. However, with the development of one-stage algorithms, such as Yolov5, its detection accuracy has greatly improved [35]. The advantages of the methods used in each part of Yolov5 are also described.

(1) Input: mosaic data enhancement.

Mosaic data enhancement method is to cut four images randomly and then splice them onto an image as training data. Thus, the background will be enriched and small targets will be easier to detect [36].

(2) Backbone: CSP structure.

The main process of Cross Stage Partial (CSP) is that before entering a block, the input is divided into two parts: the one is evaluated through the block and the other is concatenated directly through a shortcut. This process can reduce the calculation amount and ensure the accuracy rate. There are two kinds of CSP structures in Yolov5. In terms of Yolov5s, the CSP1\_X structure is applied to the Backbone network. In CSP1\_X, the inputs are divided into two parts to enter two branches: the one goes through CBL, Res Unit and convolution layer step by step in one branch and the other directly goes through the convolution layer in the other branch; the two inputs will go into Concat block for merge operation, and then go through Batch Normalization (BN), Leaky Rectified Linear Units (ReLU), and CBL blocks. The CSP2\_X structure is applied to the Neck network. The difference between CSP2\_X and CSP1\_X is that CSP2\_X replaces the Res Unit block with the CBL block. CBL is a combination of Convolution, BN and Leaky ReLU. Res Unit, which is designed based on the inspiration from the residual learning [37], consists of two CBL modules and one skip connection [38]. The flow charts of CSP1\_X and CSP2\_X have been illustrated in Figure 8.

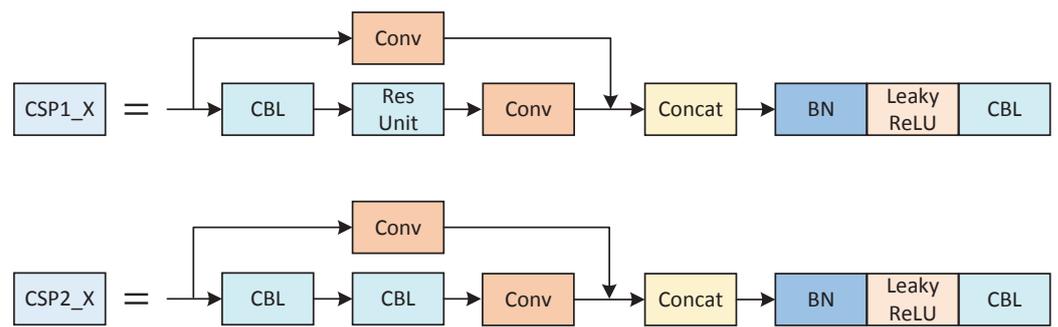


Figure 8. Flow charts of CSP1\_X and CSP2\_X.

(3) Neck: CSP2\_X structure.

In the Yolov5’s Neck structure, CSP2\_X is used for reference. CSP2\_X can strengthen the ability of feature fusion and solve the problem of gradient information repetition to ensure the inference speed and accuracy. Figure 9 shows the neck of Yolov5.

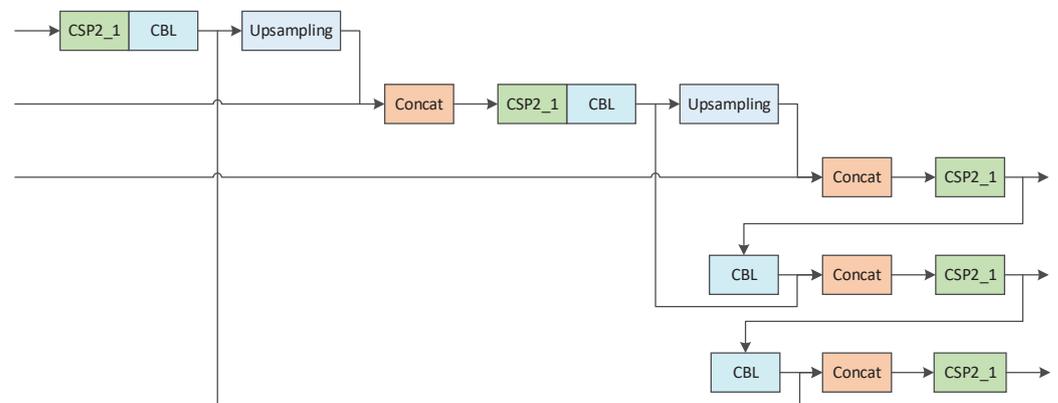


Figure 9. Neck of Yolov5.

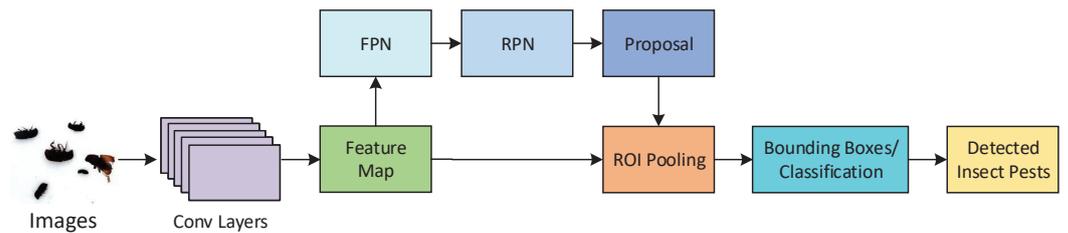
(4) Prediction: GIOU Loss.

In Yolov5, Generalized Intersection Over Union (GIOU) is used for the computation of the bbox regression loss to solve the problem that the boundary boxes do not coincide because GIOU introduces the minimum external box for optimization to solve the problem that loss is 0 when there is no overlap between the detection box and the real box.

According to different depth of network, there are 4 versions of the networks for Yolov5, including Yolov5s, Yolov5m, Yolov5l and Yolov5x models. We use the most basic version, Yolov5s on behalf of the others, for evaluation and comparison, because the other versions are just modelled by widening or deepening it. In our opinion, the effectiveness of Yolov5s also represents the feasibility of the other three models.

2.3.2. Model 2: Faster-RCNN ResNet50

Faster-RCNN is a two-stage algorithm. Our study employed the Faster-RCNN ResNet50 model. ResNet50 is a variant of ResNet which has 53 convolutional layers. The process of Faster-RCNN has been illustrated in Figure 10. The general process is that it scales the input images and puts them into the convolution layers to extract the feature maps. Further, the feature maps are sent to Feature Pyramid Network (FPN) and Region Proposal Network (RPN) step by step to generate the candidate boxes [39]. Then the original feature maps and all the candidate boxes go into the ROI pooling layer [40]. Finally, feature maps will be calculated, and sent to the fully-connected layer for target classification and coordinate regression.



**Figure 10.** Faster-RCNN ResNet50 model.

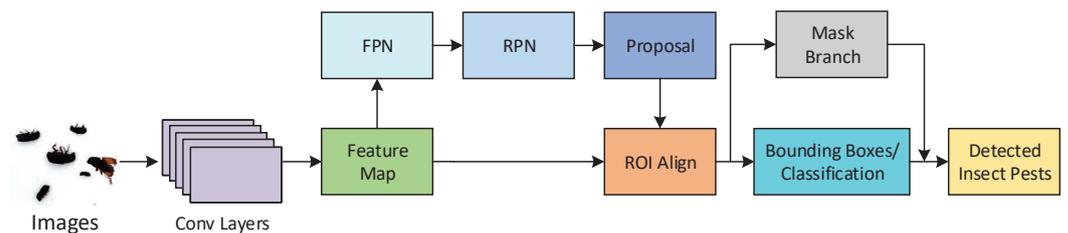
The main advantages of Faster-RCNN are as follows.

RPN generates region proposals by using the feature maps after CNN operation. It replaces Selective Search (SS) and thus improves the speed significantly.

From Fast-RCNN to Faster-RCNN, the four basic steps of candidate region generation, feature extraction, classification, and position refinement are finally unified into one deep framework [41]. The end-to-end framework is beneficial for learning the features effective for classification.

### 2.3.3. Model 3: Mask-RCNN ResNet50

The main process of Mask-RCNN is shown in Figure 11. Firstly, the images are fed into the convolutional layers to obtain the feature maps. Further, the feature maps are sent to FPN and RPN step by step to generate the candidate boxes. Then the original feature maps and all the candidate ROIs flow into the ROI Align layer. Finally, classification, bbox regression and mask generation will be accomplished.

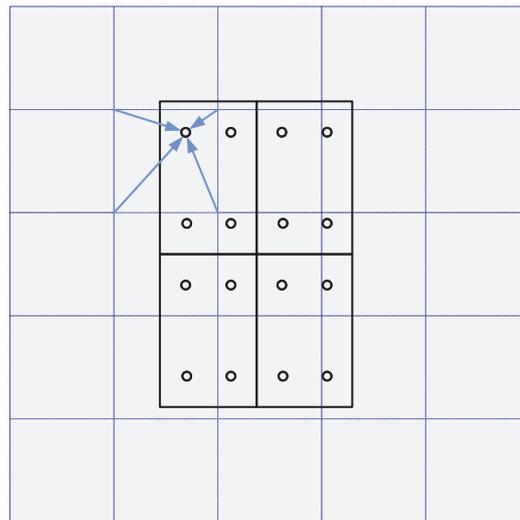


**Figure 11.** Mask-RCNN ResNet50 model.

The main advantages of Mask-RCNN are as follows.

Mask-RCNN is an algorithm based on the Faster-RCNN and its detection tasks include not only classification and bbox regression but also mask generation. For each ROI prediction, the binary mask is used to determine whether the pixel belongs to the target [42].

The quantization operation of ROI pooling in Faster-RCNN leads to misalignment when ROIs of different sizes are converted into ROI of fixed sizes via ROI Pooling. In Mask-RCNN, the ROI Align layer uses bilinear interpolation instead of the quantization operation to obtain pixels of the image in order to solve this problem. Thus, the pixels in the original image and the feature map are completely aligned without deviation, which will improve the detection accuracy. As shown in Figure 12, the area surrounded by the dotted lines represents the feature map, and the area surrounded by the solid lines represents an ROI. The ROI is divided into  $2 \times 2$  cells. If the sampling points are 4, we divide each cell into four small squares, and the center of each small square is regarded as the sampling point. As indicated by the four arrows, the value of the pixel is obtained via bilinear interpolation of the sample point pixels. Then the max pooling is performed on the four sampling points in each cell to obtain the final result [43].



**Figure 12.** In Mask-RCNN, ROI Align calculates the value of each sampling point by bilinear interpolation from the nearby grid points on the feature map.

### 3. Results and Discussion

#### 3.1. Experimental Setting

We conducted the experiments on a workstation equipped with a CPU named AMD Ryzen 5 5600X 6-Core Processor, a GPU named NVIDIA GeForce RTX 3070, and a RAM of 32 GB. To implement the DCNN models, we utilized Python 3.7.9 and Pytorch 1.6.0.

#### 3.2. Performance Metrics

Accuracy and speed are the most common performance indicators for insect pest detection. In addition, precision and recall are also very important. To fully evaluate the model performance, we take advantage of the four metrics, Accuracy (Acc), Speed, Precision (Pre) and Recall (Rec), for method evaluation. Acc, Pre and Rec can be calculated based on True Positives (TP), False Positives (FP), True Negatives (TN) and False Negatives (FN). Acc measures the ratio of correct predictions over the total number of predictions (see Equation (1)), speed means the computational time for method implementation, Pre is the fraction of TP among the detected positives (see Equation (2)), and Rec is the fraction of TP among the correct predictions (see Equation (3)).

$$Acc = \frac{TP + FN}{TP + FP + TN + FN} \quad (1)$$

$$Pre = \frac{TP}{TP + FP} \quad (2)$$

$$Rec = \frac{TP}{TP + FN} \quad (3)$$

#### 3.3. Implementation Details

The details of method implementations are provided below.

(1) Yolov5 Darknet53. The losses for Yolov5 include the classification loss, the bbox regression loss and the objectness loss. We adopted the learning rate as 0.00125, the momentum as 0.9, the weight attenuation as 0.0001 and the batch size as 1.

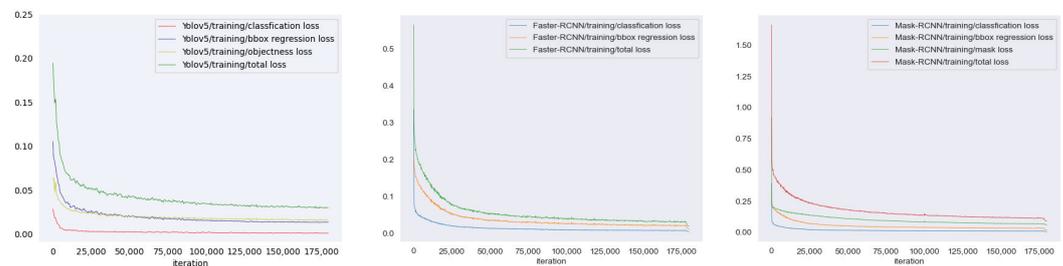
(2) Faster-RCNN ResNet50. The losses for Faster-RCNN involve the classification loss and the regression loss, i.e., the ROI classification loss, the ROI position regression loss and the GT position regression loss. We adopted the learning rate as 0.00125, the momentum as 0.9, the weight attenuation as 0.0001 and the batch size as 1.

(3) Mask-RCNN ResNet50. The losses for Mask-RCNN contain the classification and regression boxes. In addition, Mask-RCNN also outputs a binary mask for each ROI and defines a multi-task loss function for each ROI during training. They are respectively the classification loss, the regression frame loss and the mask loss. We adopted the learning rate as 0.00125, the momentum as 0.9, the weight attenuation as 0.0001 and the batch size as 1.

### 3.4. Model Evaluation Based on Baidu AI Insect Detection Dataset

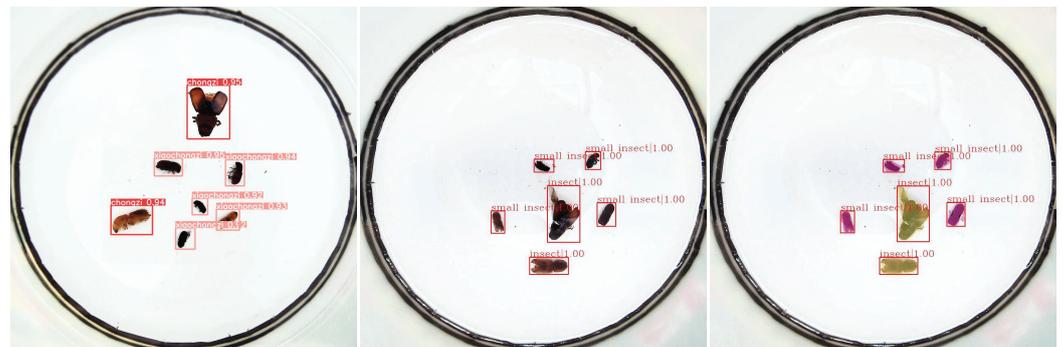
On Baidu AI insect detection dataset, we evaluate the insect pest detection performance by utilizing three deep learning models, including Yolov5, Faster-RCNN and Mask-RCNN.

In the training process of insect pest detection, the variation tendency of losses for these three models has been shown in Figure 13. Obviously, the three models all show fast convergence with low losses, which indicates that the models have good ability for learning representative features to distinguish insect pests.



**Figure 13.** The loss tendency of Yolov5, Faster-RCNN and Mask-RCNN models for insect pest detection in training.

In addition, Some examples of testing results have been shown in Figure 14, which demonstrates that insect pests can be correctly detected and the sizes can be well identified by the three algorithms.



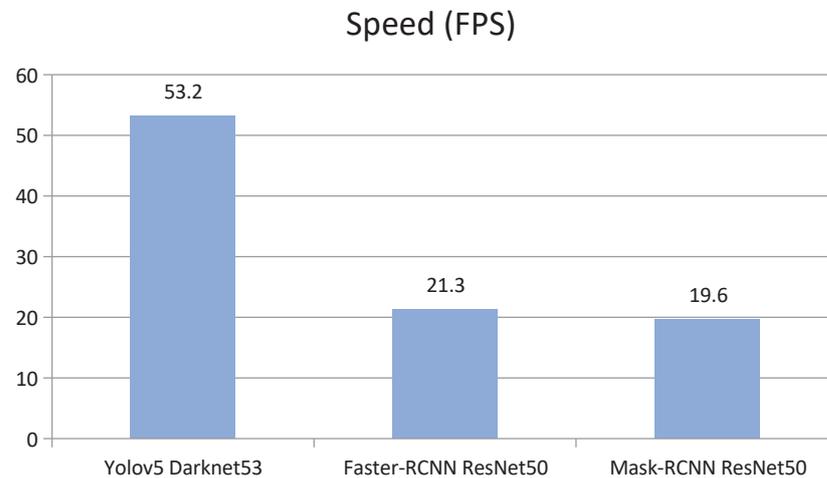
**Figure 14.** Some examples of the testing results of Yolov5, Faster-RCNN and Mask-RCNN models for insect pest detection.

According to the results in Table 1, the three algorithms show a good ability to detect different sizes of insect pests, and their accuracy can reach above 98%, among which the accuracy of Yolov5 is the highest and reach 99.54%.

**Table 1.** Performances of Yolov5, Faster-RCNN and Mask-RCNN based on Baidu AI insect detection dataset.

Models	Accuracy	Precision	Recall
Yolov5 Darknet53	99.54%	99.34%	99.62%
Faster-RCNN ResNet50	98.13%	99.52%	99.56%
Mask-RCNN ResNet50	98.41%	99.64%	99.86%

Figure 15 illustrates another performance indicator in insect pest detection, namely the average inference speed of the three models. From the figure, we can find that not only the accuracy but also the detection speed of Yolov5 is the fastest, reaching 53.2 Frames Per Second (FPS). Meanwhile, the speed of Faster-RCNN is 21.3 FPS and Mask-RCNN's is 19.6 FPS. Hence, Yolov5 is about 2.5 times as fast as Faster-RCNN and Mask-RCNN.

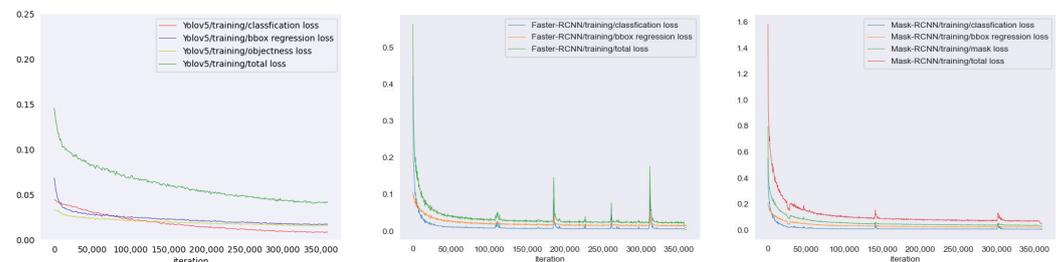


**Figure 15.** The speed of Yolov5, Faster-RCNN and Mask-RCNN models for insect pest detection.

### 3.5. Model Evaluation Based on IP102 Dataset

Similarly, we adopt three deep learning models, Yolov5, Faster-RCNN and Mask-RCNN to conduct insect pest detection experiments on IP102 dataset.

In the training process of insect pest detection, the variation tendency of losses for these three models has been shown in Figure 16.



**Figure 16.** The loss tendency of Yolov5, Faster-RCNN and Mask-RCNN models for insect pest detection.

In addition, some examples of testing results have been shown in Figure 17. Insect pests in the image can be correctly detected and the categories can be well identified via the three algorithms.



**Figure 17.** Some examples of testing results of Yolov5, Faster-RCNN and Mask-RCNN models for insect pest detection.

As with Baidu AI insect detection dataset, these models show fast convergence with low losses, contributing to learning representative features well for insect pest detection. In addition, we find that the losses of Faster-RCNN and Mask-RCNN are slightly lower than those of Yolov5. In terms of accuracy, the performance of Faster-RCNN and Mask-RCNN on IP102 dataset is clearly better than that of Yolov5, which is contrary to our results on Baidu AI insect detection dataset. Specifically, the accuracy of Yolov5, Faster-RCNN and Mask-RCNN models for insect pest detection is 97.62%, 99.43% and 99.68%, respectively.

Beyond that, because there are some small and overlapping targets in the selected images, the precision and recall of Faster-RCNN are the lowest among the three algorithms, 89.04% and 90.21%, respectively. Therefore, we conduct some additional experimental settings on the Faster-RCNN model to improve the precision and recall of this model. The experimental steps are as follows. Firstly, we add the mosaic data enhancement into Faster-RCNN for image preprocessing because the mosaic data enhancement method is utilized to cut four images randomly to splice onto an image, so that the background can be enriched and small insect pests can be more easily detected. Secondly, we employ the Complete IOU (CIOU) loss instead of the L1 loss for the calculation of bbox regression and classification losses. CIOU considers the intersection ratio of the two frames, the distance between the center point and the width and height ratio of the two frames, which can facilitate the detection of the target. Thirdly, we adopt soft Non Maximum Suppression (NMS) instead of NMS. Soft NMS attenuates the scores of boxes that overlap with the box with the highest score instead of setting them to 0 and thus the overlapping objects can be well detected. Finally, we use ResNet101 which is a variant of ResNet and has 101 layers, instead of ResNet50. The deeper network has bigger capacity to capture the image characteristics on these datasets. Attributing to these improvements, the precision and recall of Faster-RCNN increase by nearly 2.5 percentage points, 91.42% and 92.84%. All the results have been provided in Table 2.

In terms of computational velocity, Yolov5 has the fastest speed among the three algorithms, which is consistent with our conclusion on Baidu AI insect detection dataset.

**Table 2.** Performances of Yolov5, Faster-RCNN and Mask-RCNN for the detection of insect pests based on IP102 dataset.

Models	Accuracy	Precision	Recall
Yolov5 Darknet53	97.62%	92.54%	93.43%
Faster-RCNN ResNet101	99.43%	91.42%	92.84%
Mask-RCNN ResNet50	99.68%	90.36%	92.15%

In view of the above results, we can decide which algorithm can be recommended for the efficient detection of insect pests according to different data situations. We recommend Yolov5 for the relatively simple images with few classes (like Baidu AI insect detection dataset), while Faster-RCNN and Mask-RCNN for the more complex images with abundant classes (like IP102 dataset).

Furthermore, we compare the recommended deep models with the related advanced approaches to demonstrate the superiorities of the recommended models in insect pest detection. The compared approaches include SqueezeNet [44], Single Shot MultiBox Detector (SSD) [45], Capsule Network (CapsNet) [46], Combination of DCNN and Transfer Learning (DCNNT) [47], Yolov2 [48], Multi-Scale Capsule Network (MS-CapsNet) [49], ResNet50 [50], GoogleNet [51], Bio-inspired Model [52], Faster-RCNN ResNet50 [53], CNN [54], Faster-RCNN VGG16 [55], AlexNet [56] and Multi-Scale Convolution-Capsule (MSCC) [57]. It is not difficult to find that all of these approaches belong to the methodological family of deep learning. In the investigation, there is a majority of relevant research which takes advantage of IP102 dataset for method validation, but hardly any work resorts to Baidu AI insect detection dataset. Therefore, we follow this research and utilize IP102 dataset for method comparison. According to the results in Table 3, we can easily find that our recommended deep models, Mask-RCNN and Faster-RCNN, inspiringly outperform the latest competitive

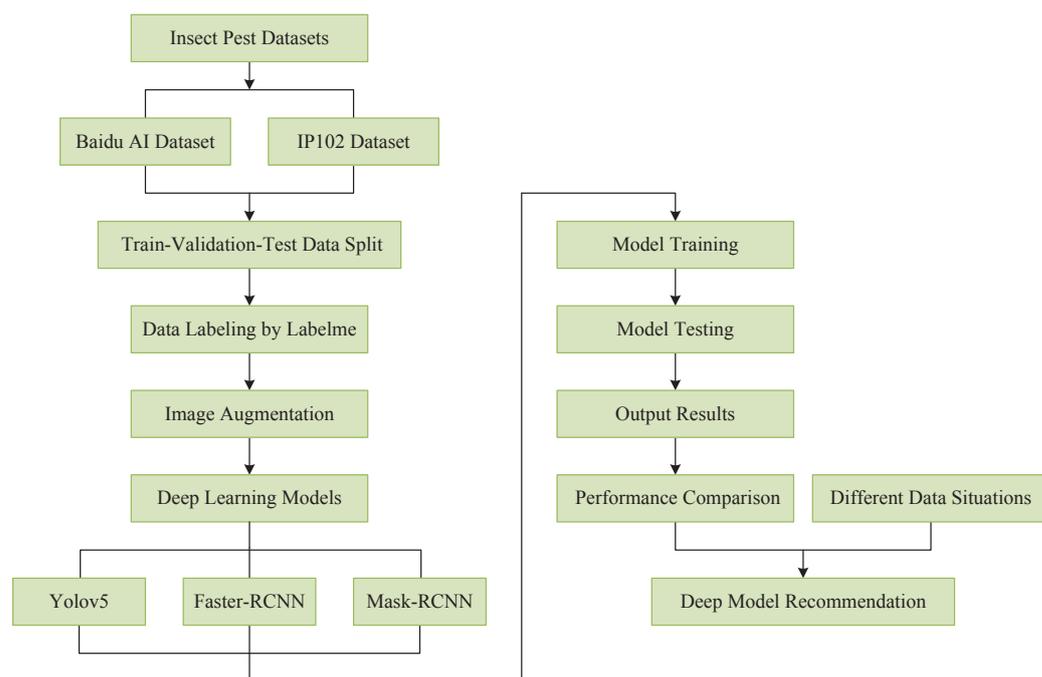
approaches in terms of accuracy, although they both face a heavier classification burden from not less insect pest classes than these approaches in classification.

**Table 3.** Method comparison on IP102 dataset.

Research	Methods	Accuracy	Classes
Iandola et al. (2016) [44]	SqueezeNet	67.51%	8
Ning et al. (2017) [45]	SSD MobileNet	92.12%	8
	SSD Inception	93.47%	8
Li et al. (2018) [46]	CapsNet	82.4%	9
Thenmozhi and Reddy (2019) [47]	DCNNT	84.7%	9
Cui et al. (2019) [48]	Yolov2	87.66%	8
Wang et al. (2019) [49]	MS-CapsNet	89.6%	9
Yan et al. (2020) [50]	ResNet50	85.5%	9
Khalifa et al. (2020) [51]	GoogleNet	88.80%	8
Nanni et al. (2020) [52]	Bio-inspired Model	92.4%	10
Ramalingam et al. (2020) [53]	Faster-RCNN ResNet50	96.06%	8
Kasinathan et al. (2021) [54]	CNN	91.5%	9
		93.9%	5
Karar et al. (2021) [55]	Faster-RCNN VGG16	98.9%	5
Chen et al. (2022) [56]	AlexNet	80.3%	9
Xu et al. (2022) [57]	MSCC	92.4%	9
Ours	Mask-RCNN ResNet50	99.6%	10
	Faster-RCNN ResNet101	99.4%	10
	Yolov5 Darknet53	97.6%	10

Deep learning has sparked a boom in the development of artificial intelligence. In the post era of deep learning, various network models burgeon, mushroom and flourish. These deep models sweep across diverse application areas, overwhelm various conventional machine learning techniques, and lead the technological trend of many topics in these areas. However, in the method pool of deep learning, which model can efficiently detect the insect pests remains a controversial topic nowadays. In this work, the evaluation results in Tables 1 and 2 support the data-driven point of view to resolve this dispute: different models can be recommended for different data situations. Moreover, although any small change of the deep model structure may result in a big difference in the final performance, the comparison results in Table 3 confirm the advantage of the recommended models over existing deep networks for this issue. At the same time, these results can also enlighten us to further explore new and more powerful technology on the basis of the mechanism of recommended models for insect pest detection in agriculture.

In the end, the proposal in this paper displays the overall flow chart of our work in Figure 18.



**Figure 18.** The overall flow chart of this work.

#### 4. Conclusions

In our work, we have studied three DCNN models—Yolov5, Faster-RCNN and Mask-RCNN in order to recommend relatively the most suitable one for efficient insect pest detection with respect to the image data situation. By experimental demonstration, we find that when the background of insect pest images is simple, such as Baidu AI insect detection dataset, Yolov5 can more efficiently detect the insect pests, with an accuracy above 99% and a speed about 2.5 times as fast as Faster-RCNN and Mask-RCNN; when the background of images is more complex and the insect categories are more abundant, such as IP102 dataset, Faster-RCNN and Mask-RCNN are more suitable for insect pest detection, with an accuracy above 99% and a normally fast speed. Furthermore, by method comparison, we have also demonstrated the advantage of the recommended deep learning models over the related recent approaches for insect pest detection.

Although we are able to recommend the most suitable deep network according to the image data situation, there still exists weakness for our proposal. The weakness of the recommended deep learning models for insect pest detection lies in the lack of high adaptability to different image data situations, which results in the sensitive performances case by case. Because the suitable deep network needs to be selected automatically or has to be chosen manually to suit the actual condition of the sample data, the method applicability will be more or less limited for the sample data with dynamic complexity and changeable scale. To cope with it, in future, our ongoing work will include but is not limited to designing an improved deep learning scheme based on the transfer learning strategy. This scheme will integrate the merits of the three deep learning methods into one framework and can hopefully adapt to different image data complexities and scales for accurate and fast insect pest detection in agriculture.

**Author Contributions:** Conceptualization and methodology, W.L. and T.Z.; software, validation, and original draft preparation, W.L. and T.Z.; review and editing, supervision, W.L., X.L., J.D. and J.L. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported in part by the Jiangsu Agricultural Science and Technology Innovation Fund under Grant CX(20)3071, and in part by the Fundamental Research Funds for the Central Universities under Grant 2242021R41094.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The datasets used and/or analyzed during the current study are available from the corresponding author on reasonable request.

**Conflicts of Interest:** The authors declare that they have no competing interest.

## References

1. Amiri, A.N.; Allahbakhsh. An Effective Pest Management Approach in Potato to Combat Insect Pests and Herbicide. *3 Biotech* **2019**, *9*, 16. [[CrossRef](#)] [[PubMed](#)]
2. Fernández, R.; Petek, M.; Gerasymenko, I.; Juterek, M.; Patron, N.J. Insect Pest Management in The Age of Synthetic Biology. *Plant Biotechnol. J.* **2021**, *20*, 25–36. [[CrossRef](#)] [[PubMed](#)]
3. Nomura, Y.; Shigemura, K. Development of Real-Time Screening System for Structural Surface Damage Using Object Detection and Generative Model Based on Deep Learning. *J. Soc. Mater. Sci. Jpn.* **2019**, *68*, 250–257. [[CrossRef](#)]
4. Sütő, J. Embedded System-Based Sticky Paper Trap with Deep Learning-Based Insect-Counting Algorithm. *Electronics* **2021**, *10*, 1754. [[CrossRef](#)]
5. Lima, M.C.F.; Leandro, M.E.D.D.A.; Valero, C.; Coronel, L.C.P.; Bazzo, C.O.G. Automatic Detection and Monitoring of Insect Pests—A Review. *Agriculture* **2020**, *10*, 161. [[CrossRef](#)]
6. Dca, A.; Jg, B.; Hp, C.; Ad, A. Methods of Insect Image Capture and Classification: A Systematic Literature Review. *Smart Agric. Technol.* **2021**, *1*, 100023.
7. Wu, L.; Liu, Z.; Bera, T.; Ding, H.; Xu, J. A Deep Learning Model to Recognize Food Contaminating Beetle Species Based on Elytra Fragments. *Comput. Electron. Agric.* **2019**, *166*, 105002. [[CrossRef](#)]
8. Patel, P.P.; Vaghela, D.B. Crop Diseases and Pests Detection Using Convolutional Neural Network. In Proceedings of the 2019 IEEE International Conference on Electrical, Computer and Communication Technologies, Coimbatore, India, 20–22 February 2019; pp. 1–4.
9. Ngugi, L.C.; Abelwahab, M.; Abo-Zahhad, M. Recent Advances in Image Processing Techniques for Automated Leaf Pest and Disease Recognition—A Review. *Inf. Process. Agric.* **2021**, *8*, 27–51. [[CrossRef](#)]
10. Gayathri, A.G.; Remya Ajai, A.S. VLSI Implementation of Improved Sobel Edge Detection Algorithm. In Proceedings of the 2021 International Conference on Communication, Control and Information Sciences, Idukki, India, 16–18 June 2021; pp. 1–6.
11. Lu, F.; Xie, F.; Shen, S.; Yang, J.; Huang, L. The One-Stage Detector Algorithm Based on Background Prediction and Group Normalization for Vehicle Detection. *Appl. Sci.* **2020**, *10*, 5883. [[CrossRef](#)]
12. Fan, C.; Wang, Q. Research on Image Segmentation Method Using A Structure-preserving Region Model-based MRF. *Clust. Comput.* **2019**, *22*, 15329–15334. [[CrossRef](#)]
13. Pan, R.; Zhang, Z.; Fan, Y.; Cao, J.; Lu, K.; Yang, T. Multi-objective Optimization Method for Learning Thresholds in A Decision-theoretic Rough Set Model. *Int. J. Approx. Reason.* **2016**, *71*, 34–49. [[CrossRef](#)]
14. Zelazo, D.; Mesbahi, M.; Belabbas, M.A. Graph Theory in Systems and Controls. In Proceedings of the 2018 IEEE Conference on Decision and Control, Miami, FL, USA, 17–19 December 2018; pp. 6168–6179.
15. Junyan, B. Research on The Technology of Artificial Intelligence in Computer Network under The Background of Big Data. In Proceedings of the 2020 International Conference on Computer Communication and Network Security, Xi'an, China, 21–23 August 2020; pp. 51–54.
16. Arshad, T.; Jia, M.; Guo, Q.; Gu, X.; Liu, X. Fruit Classification Through Deep Learning: A Convolutional Neural Network Approach. In *International Conference in Communications, Signal Processing, and Systems*; Liang, Q., Wang, W., Liu, X., Na, Z., Jia, M., Zhang, B., Eds.; Springer: Singapore, 2020; pp. 2671–2677.
17. Li, W.; Wang, D.; Li, M.; Gao, Y.; Wu, J.; Yang, X. Field Detection of Tiny Pests from Sticky Trap Images Using Deep Learning in Agricultural Greenhouse. *Comput. Electron. Agric.* **2021**, *183*, 106048. [[CrossRef](#)]
18. Asefpour Vakilian, K.; Massah, J. Performance Evaluation of A Machine Vision System for Insect Pests Identification of Field Crops Using Artificial Neural Networks. *Arch. Fr. Pflanzenschutz* **2013**, *46*, 1262–1269. [[CrossRef](#)]
19. Wu, M.; Lu, Z.; Chen, Q.; Zhu, T.; Lu, E.; Lu, W.; Liu, M. A Two-Stage Algorithm of Locational Marginal Price Calculation Subject to Carbon Emission Allowance. *Energies* **2020**, *13*, 2510. [[CrossRef](#)]
20. Hu, J.; Fang, J.; Du, Y.; Liu, Z.; Ji, P. A Security Risk Plan Search Assistant Decision Algorithm Using Deep Neural Network Combined with Two-stage Similarity Calculation. *Pers. Ubiquitous Comput.* **2019**, *23*, 541–552. [[CrossRef](#)]
21. Gao, M.; Bai, Y.; Li, Z.; Li, S.; Zhang, B.; Chang, Q. Real-time Jellyfish Classification and Detection Based on Improved Yolov3 Algorithm. *Sensors* **2021**, *21*, 8160. [[CrossRef](#)]
22. Tc, A.; Ning, W.; Rw, A.; Hong, Z.A.; Gz, C. One-stage CNN Detector-based Benthonic Organisms Detection with Limited Training Dataset. *Neural Netw.* **2021**, *144*, 247–259.
23. Srivastava, S.; Divekar, A.V.; Anilkumar, C.; Naik, I.; Pattabiraman, V. Comparative Analysis of Deep Learning Image Detection Algorithms. *J. Big Data* **2021**, *8*, 66. [[CrossRef](#)]

24. Li, M.; Wang, H.; Yang, L.; Liang, Y.; Wan, H. Fast Hybrid Dimensionality Reduction Method for Classification Based on Feature Selection and Grouped Feature Extraction. *Expert Syst. Appl.* **2020**, *150*, 113277. [[CrossRef](#)]
25. Shanmuganathan, V.; Yesudhas, H.R.; Khan, M.S.; Khari, M.; Gandomi, A.H. R-CNN and Wavelet Feature Extraction for Hand Gesture Recognition with EMG Signals. *Neural Comput. Appl.* **2020**, *32*, 16723–16736. [[CrossRef](#)]
26. BaiDu Company, N.F.U. Baidu AI Insect Detection Dataset. Available online: <https://aistudio.baidu.com/aistudio/datasetdetail/19638/> (accessed on 14 January 2020).
27. Wu, X.; Zhan, C.; Lai, Y.K.; Cheng, M.M.; Yang, J. IP102: A Large-scale Benchmark Dataset for Insect Pest Recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 8779–8788.
28. Liu, B.; Zhao, W.; Sun, Q. Study of Object Detection Based on Faster-RCNN. In Proceedings of the 2017 Chinese Automation Congress, Jinan, China, 20–22 October 2017; pp. 6233–6236.
29. Du, Y. A Crop Image Segmentation and Extraction Algorithm Based on Mask RCNN. *Entropy* **2021**, *23*, 1160.
30. Popescu, D.E. An Integrated Approach for Monitoring Social Distancing and Face Mask Detection Using Stacked ResNet-50 and Yolov5. *Electronics* **2021**, *10*, 2996.
31. Javed, H.; Iqbal, J.; Khan, T.M. Studies on Population Dynamics of Insect Pest of Safflower, *Carthamus Tinctorius* L. *Pak. J. Zool.* **2013**, *45*, 213–217.
32. Jin, S.; Sun, L. Application of Enhanced Feature Fusion Applied to YOLOv5 for Ship Detection. In Proceedings of the Chinese Control and Decision Conference, Kunming, China, 22–24 May 2021; pp. 7242–7246.
33. Ding, W.; Taylor, G. Automatic Moth Detection from Trap Images for Pest Management. *Comput. Electron. Agric.* **2016**, *123*, 17–28. [[CrossRef](#)]
34. Wang, J.; Chen, Y.; Gao, M.; Dong, Z. Improved YOLOv5 network for real-time multi-scale traffic sign detection. *arXiv* **2021**, arXiv:2112.08782v2.
35. Sung, J.Y.; Yu, S.B.; Korea, S.h.P. Real-time Automatic License Plate Recognition System using YOLOv4. In Proceedings of the IEEE International Conference on Consumer Electronics–Asia, Seoul, Korea, 1–3 November 2020; pp. 1–3.
36. Luo, J.; Fang, H.; Shao, F.; Zhong, Y.; Hua, X. Multi-scale Traffic Vehicle Detection Based on Faster-RCNN with NAS Optimization and Feature Enrichment. *Def. Technol.* **2021**, *17*, 1542–1554. [[CrossRef](#)]
37. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
38. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. *arXiv* **2018**, arXiv:1804.02767.
39. Mansoor, A.; Porras, A.R.; Linguraru, M.G. Region Proposal Networks with Contextual Selective Attention for Real-time Organ Detection. In Proceedings of the 2019 IEEE 16th International Symposium on Biomedical Imaging, Venice, Italy, 8–11 April 2019; pp. 1193–1196.
40. Wang, K.; Liu, M.Z. Object Recognition at Night Scene Based on DCGAN and Faster-RCNN. *IEEE Access* **2020**, *8*, 193168–193182. [[CrossRef](#)]
41. Su, H.; Wei, S.; Yan, M.; Wang, C.; Shi, J.; Zhang, X. Object Detection and Instance Segmentation in Remote Sensing Imagery Based on Precise Mask-RCNN. In Proceedings of the IEEE International Geoscience and Remote Sensing Symposium, Yokohama, Japan, 28 July–2 August 2019; pp. 1454–1457.
42. Chu, P.; Li, Z.; Lammers, K.; Lu, R.; Liu, X. Deep Learning-based Apple Detection Using A Suppression Mask-RCNN. *Pattern Recognit. Lett.* **2021**, *147*, 206–211. [[CrossRef](#)]
43. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask R-CNN. In Proceedings of the International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2980–2988.
44. Iandola, F.N.; Han, S.; Moskewicz, M.W.; Ashraf, K.; Dally, W.J.; Keutzer, K. SqueezeNet: AlexNet-level Accuracy with 50x Fewer Parameters and <0.5 MB Model Size. *arXiv* **2016**, arXiv:1602.07360.
45. Ning, C.; Zhou, H.; Song, Y.; Tang, J. Inception Single Shot MultiBox Detector for Object Detection. In Proceedings of the 2017 IEEE International Conference on Multimedia & Expo Workshops, Hong Kong, China, 10–14 July 2017; pp. 549–554.
46. Li, Y.; Qian, M.; Liu, P.; Cai, Q.; Li, X.; Guo, J.; Yan, H.; Yu, F.; Yuan, K.; Yu, J.; et al. The Recognition of Rice Images by UAV Based on Capsule Network. *Clust. Comput.* **2018**, *22*, 9515–9524. [[CrossRef](#)]
47. Thenmozhi, K.; Reddy, U.S. Crop Pest Classification Based on Deep Convolutional Neural Network and Transfer Learning. *Comput. Electron. Agric.* **2019**, *164*, 104906. [[CrossRef](#)]
48. Cui, J.; Zhang, J.; Sun, G.; Zheng, B. Extraction and Research of Crop Feature Points Based on Computer Vision. *Sensors* **2019**, *19*, 2553. [[CrossRef](#)]
49. Wang, D.; Xu, Q.; Xiao, Y.; Tang, J.; Bin, L. Multi-scale Convolutional Capsule Network for Hyperspectral Image Classification. In *Chinese Conference on Pattern Recognition and Computer Vision*; Springer International Publishing: Cham, Switzerland, 2019; pp. 749–760.
50. Yan, P.; Su, Y.; Tian, X. Classification of Mars Lineament and Non-lineament Structure Based on ResNet50. In Proceedings of the 2020 IEEE International Conference on Advances in Electrical Engineering and Computer Applications, Dalian, China, 25–27 August 2020; pp. 437–441.
51. Khalifa, N.E.; Loey, M.; Taha, M. Insect Pests Recognition Based on Deep Transfer Learning Models. *J. Theor. Appl. Inf. Technol.* **2020**, *98*, 60–68.

52. Nanni, L.; Maguolo, G.; Pancino, F. Insect Pest Image Detection and Recognition Based on Bio-inspired Methods. *Ecol. Inform.* **2020**, *57*, 101089. [[CrossRef](#)]
53. Ramalingam, B.; Mohan, R.E.; Pookkuttath, S.; Gómez, B.F.; Sairam Borusu, C.S.C.; Wee Teng, T.; Tamilselvam, Y.K. Remote Insects Trap Monitoring System Using Deep Learning Framework and IoT. *Sensors* **2020**, *20*, 5280. [[CrossRef](#)]
54. Kasinathan, T.; Singaraju, D.; Uyyala, S.R. Insect Classification and Detection in Field Crops Using Modern Machine Learning Techniques. *Inf. Process. Agric.* **2021**, *8*, 446–457. [[CrossRef](#)]
55. 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](#)]
56. Chen, H.C.; Widodo, A.M.; Wisnujati, A.; Rahaman, M.; Lin, J.C.W.; Chen, L.; Weng, C.E. AlexNet Convolutional Neural Network for Disease Detection and Classification of Tomato Leaf. *Electronics* **2022**, *11*, 951. [[CrossRef](#)]
57. Xu, C.; Yu, C.; Zhang, S.; Wang, X. Multi-Scale Convolution-Capsule Network for Crop Insect Pest Recognition. *Electronics* **2022**, *11*, 1630. [[CrossRef](#)]