applied sciences

MDPI

Article

Deep Learning-Based Image Recognition of Agricultural Pests

Weixiao Xu [1], Lin Sun [2], Cheng Zhen [3], Bo Liu [1,*], Zhengyi Yang [4] and Wenke Yang [5]

[1] School of Mechanical Engineering, North University of China, Taiyuan 030051, China
[2] School of Life Science, Southwest Forestry University, Kunming 650233, China
[3] School of Social Development, Yangzhou University, Yangzhou 225002, China
[4] School of Computer Science and Engineering, UNSW Sydney, Sydney 2052, Australia
[5] Enmotech Data AU, Sydney 2113, Australia
* Correspondence: liubozb@nuc.edu.cn; Tel.: +86-13753118579

Abstract: Pests and diseases are an inevitable problem in agricultural production, causing substantial economic losses yearly. The application of convolutional neural networks to the intelligent recognition of crop pest images has become increasingly popular due to advances in deep learning methods and the rise of large-scale datasets. However, the diversity and complexity of pest samples, the size of sample images, and the number of examples all directly affect the performance of convolutional neural networks. Therefore, we designed a new target-detection framework based on Cascade RCNN (Regions with CNN features), aiming to solve the problems of large image size, many pest types, and small and unbalanced numbers of samples in pest sample datasets. Specifically, this study performed data enhancement on the original samples to solve the problem of a small and unbalanced number of examples in the dataset and developed a sliding window cropping method, which could increase the perceptual field to learn sample features more accurately and in more detail without changing the original image size. Secondly, combining the attention mechanism with the FPN (Feature Pyramid Networks) layer enabled the model to learn sample features that were more important for the current task from both channel and space aspects. Compared with the current popular target-detection frameworks, the average precision value of our model (mAP@0.5) was 84.16%, the value of (mAP@0.5:0.95) was 65.23%, the precision was 67.79%, and the F1 score was 82.34%. The experiments showed that our model solved the problem of convolutional neural networks being challenging to use because of the wide variety of pest types, the large size of sample images, and the difficulty of identifying tiny pests.

Keywords: pest recognition; convolutional neural networks; sliding window cropping method; data enhancement; attention mechanisms

Citation: Xu, W.; Sun, L.; Zhen, C.; Liu, B.; Yang, Z.; Yang, W. Deep Learning-Based Image Recognition of Agricultural Pests. *Appl. Sci.* **2022**, *12*, 12896. https://doi.org/10.3390/app122412896

Academic Editor: Górnicki Krzysztof

Received: 21 November 2022
Accepted: 3 December 2022
Published: 15 December 2022

1. Introduction

Object detection is essentially the localization and classification of objects, which uses rectangular borders to localize the objects to be detected and to classify them accurately. Object detection is one of the critical areas of computer vision. It is widely used in real life and industrial production, and includes face detection [1], text detection [2,3], pedestrian detection [4,5], marker detection [6,7], video detection [8,9], vehicle detection [10,11], medical image detection [12], and so on.

Pests are one of the most important factors affecting crop yields and impact regional agricultural economic stability and food security [13]. According to incomplete statistics [14], 20–30% of global agricultural production is affected by pests and weed infestation, causing global economic losses of USD 70 billion [15]. Therefore, accurately identifying and counting pests is essential for pest management. Traditional pest-identification technology detects and identifies pests manually, which wastes a lot of human and financial resources, and the accuracy of identification and counting can be affected by subjective factors relating to personnel.

Pests and diseases have always been a complex problem in agricultural production, as they cause large economic losses every year. To effectively prevent and control pests and diseases in farm fields, pest and disease information needs to be collected and analyzed [16,17]. Owing to the diversity of pests on farmland and the complexity of available information types, the traditional pest-monitoring methods through manual observation and statistics can no longer meet the needs of modern large-scale agricultural production for pest control work [18–20]. Therefore, applying deep-learning methods to the intelligent recognition of crop pest images has become a critical research hotspot [16–24].

Researchers have recently combined deep learning to develop many object-detection frameworks for pest identification. Yang et al. [16] achieved the localization and identification of 23 key pest species in tea plantations using salient maps and a CNN (Convolutional Neural Network). Xie Chengjun et al. [17] identified 35 insect species using a sparse coding pyramid model. Sun Peng et al. [18] used an attentional convolutional neural network to identify soybean pests. Cheng et al. [19] introduced an AlexNet-based residual block to construct a new pest-identification network to identify ten pests in the natural environment. Fuentes et al. combined the excellent parts of previous models to detect nine diseases and insect pests on tomato plants [20] and achieved good results. Lin et al. [21] developed a new target-detection model that uses fast R-CNN as the terminal model to classify 24 pest categories. Sabanci et al. [22] proposed a new convolution and recursion hybrid network that combined AlexNet and bidirectional short-term memory (BiLSTM) to identify wheat grains damaged by pests. Gambhir et al. [23] developed an interactive Android and web interface based on CNN to diagnose problems and diseases in crops. Li et al. [24] developed a new object-detection framework based on faster R-CNN, which could conduct online monitoring of plant diseases and pest identification. However, these deep-learning models usually perform resize operations on samples, leading to the loss of sample features for pest samples of considerable size, and the excellent performance of deep-learning models is determined mainly by the number of examples in the dataset. Therefore, pest datasets with a small number of samples and too-large image sizes are undoubtedly a great challenge for target-detection models aiming to perform well.

This paper proposes a model for detecting farmland pests in response to the problems of large and unbalanced sample categories, large image sizes, and a small number of samples in farmland pest datasets. Specifically, this study uses a sliding window cropping method to learn the sample features inside larger images carefully. Secondly, this paper introduces an attention mechanism inside the model to focus more on learning the sample features that are more important for the task at hand. Finally, data augmentation is performed on a smaller number of samples in the category. In this way, the model could learn sample features in a more comprehensive and detailed way to improve the efficiency of pest detection.

The main contributions of this paper are as follows.

1. We propose a new pest-identification model that aims to achieve good performance, even with a small number of samples, unbalanced categories, and large sample image sizes;
2. We devise a new sliding window cropping method that aims to increase the perceptual field to learn sample features more carefully and comprehensively, which may be missed due to large image sizes;
3. We perfectly integrate the attention mechanism with the FPN layer in the model to make the model more focused on sample features that are more useful for the task at hand;
4. We augment the data for small numbers of sample categories as well as for unbalanced samples to prevent their adverse effects.

2. Related Work

This section will provide a detailed introduction to some of the modules involved in our pest-detection model to set the stage for the model we adopt later.

2.1. Existing Object-Detection Frameworks

Owing to the progress of Internet technology and the excellent performance of neural networks, deep convolutional neural networks (DCNNs) have gradually attracted the attention of researchers because they can learn the intrinsic characteristics of images, from shallow to deep, and their robustness is excellent [25–27]. With the improvement of computational power and the increase in the number of available data sets [28], the application of object detection based on DCNNs is becoming more and more extensive. In 2012, A. Krizhevsky et al. proposed a new neural network named AlexNet DCNN [29], which won the ILSVRC-2012 competition (the error rate of the top five was 15.3%). This work stimulated the re-study of applying deep convolution neural networks to object-detection models. In 2014, R. Girshick et al. proposed a target-detection framework named RCNN [30], which was a milestone in applying DCNN-based methods to object detection. Redmon et al. proposed a neural network-based object-detection system, a YOLO scheme (You Only Look Once: Unified, Real-Time Object Detection) [31], and presented it at CVPR 2016. The Fast R-CNN proposed by Ross B. Girshick et al. [32] is an excellent solution to the drawback of inputting Region Proposal regions into CNN networks separately in traditional R-CNN. The Cascade RCNN proposed by Zhaowei Cai et al. [33] further improved the accuracy of detecting objects. The DCNN-based target-detection method has advantages over traditional target-detection methods, and is becoming more and more popular nowadays.

2.2. The Cbam Attention Mechanism

The attention mechanism [34–36] has received attention in recent years for its ability to focus on information relevant to the task and ignore irrelevant information. As shown in Figure 1, the cbam attention mechanism [35] consists of two parts: spatial attention and channel attention.

The part in the blue box in the figure is channel attention, and the part in the red box is spatial attention. Specifically, the input feature map F is subjected to maximum global pooling and global average pooling based on the width and height, respectively, to obtain two $1 \times 1 \times C$ feature maps. Then, they are integrated into a two-layer neural network (MLP). The number of neurons in the first layer is C/r (r is the reduction rate), the activation function is Relu, and the number of neurons in the second layer is C. This two-layer neural network is shared. Then, the output features are summed point by point, and sigmoid activation is performed to generate the final channel attention feature, $M_c(F)$. Finally, it is multiplied element by element to create the input element F' needed for spatial attention. The formula of channel attention is as follows:

$$\begin{aligned} M_c(F) &= \sigma(\mathrm{MLP}(\mathrm{AvgPool}(F)) + \mathrm{MLP}(\mathrm{MaxPool}(F))) \\ &= \sigma\left(W_1\left(W_0(F^c_{avg})\right) + W_1(W_0(F^c_{max}))\right) \end{aligned} \tag{1}$$

Then, the feature map F' output by the channel attention module is used as the input feature map of the spatial attention module. First, we operate the channel-based global maximum pool and global average pool to obtain two $H \times W \times 1$ feature maps, and then we splice the two feature maps on the channel. We convolute it by 7×7 (7×7 is better than 3×3) to reduce the dimension to one track, i.e., $H \times W \times 1$. Then, we produce a sigmoid to generate a spatial attention feature, namely M_s. Finally, we multiply this feature by the input feature of the module to obtain the final generated feature, F''.

The equation of spatial attention is as follows:

$$M_s(F) = \sigma\left(f^{7\times7}([\mathrm{AvgPool}(F); \mathrm{MaxPool}(F)])\right) = \sigma\left(f^{7\times7}\left(\left[F^s_{avg}; F^s_{max}\right]\right)\right) \tag{2}$$

The feature F' after channel attention is:

$$F' = \mathrm{M_c(F)} \otimes F \tag{3}$$

The final feature F'' obtained after the whole cbam attention is:

$$F'' = \mathrm{M_s(F)} \otimes F' \tag{4}$$

At this point, we consider that F'' contains features that are more important to the model's current task.

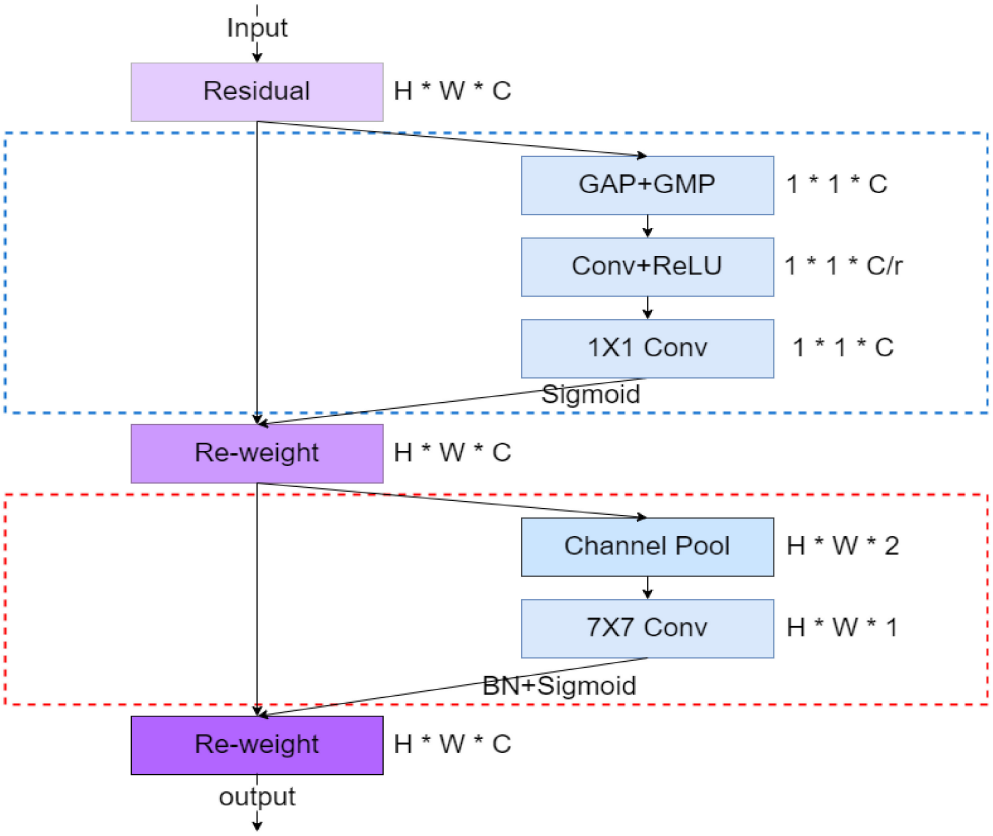

Figure 1. Schematic diagram of the cbam attention mechanism module.

2.3. The Architecture of ResNet and ResNeXt

The core idea of ResNet [37], proposed by He et al., is to pass the original input information directly to the subsequent network. The residual block is designed based on the above idea, and its structure is shown in the left panel of Figure 2. This design of the residual block makes the neural network deeper and less prone to gradient disappearance, because the original input in the residual block can be propagated forward faster by constant mapping. ResNeXt [38] improves ResNet. First, it retains the design concept of residual blocks in ResNet and improves the ResNet residual blocks by introducing group convolution to obtain the ResNeXt residual blocks, as shown in the right panel of Figure 2. Second, it widens the basis of the network by referring to the split–transform–merge concept in Inception [39]. Finally, it adopts the concise design principle of VGG [40] to overcome the drawback that the convolutional kernel parameters of Inception [39] are too complex and difficult to control.

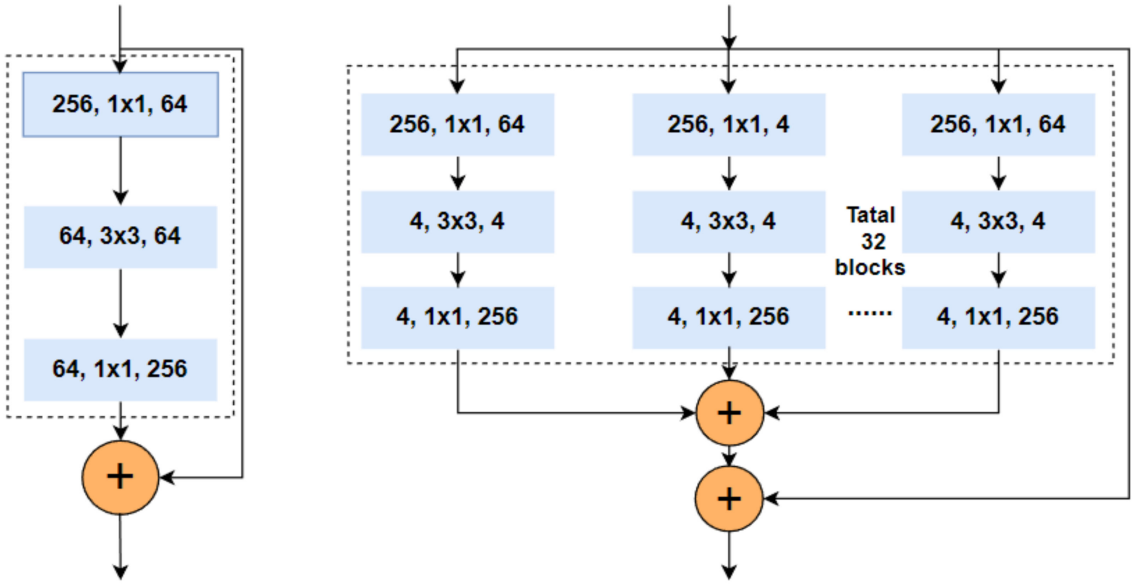

Figure 2. The left and right figures show the structural frameworks of ResNet and ResNeXt, respectively.

3. Method

In this section, we present the general structure of the pest-identification model. As shown in Figure 3, our model consists of four main modules, including data preprocessing, feature extraction, attention-based FPN, and cascaded structure.

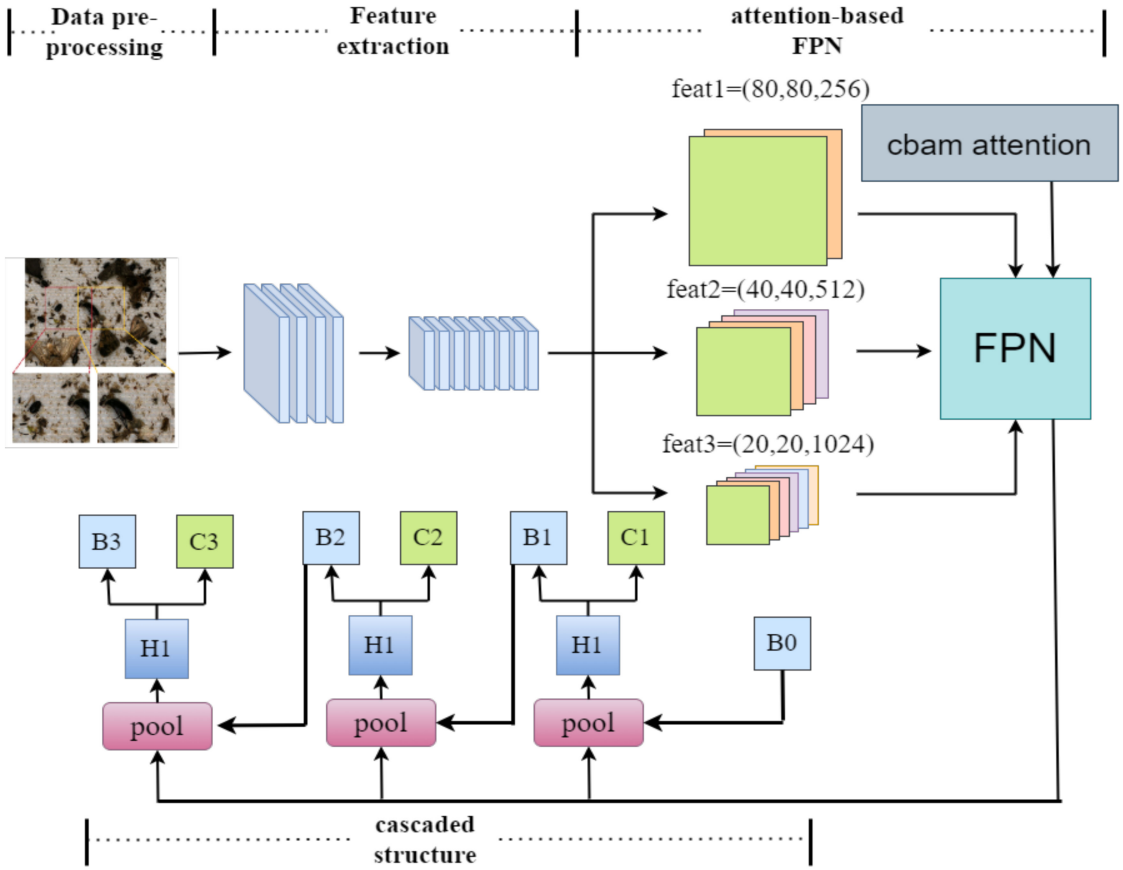

Figure 3. The general framework of the image-recognition model for farmland pests.

3.1. Data Preprocessing

3.1.1. Sliding Window Cropping Method

The image size of the pest dataset was much larger than that of other datasets generally used for target detection (the size of the images in the pest dataset was 5472 × 3648). If the original image is the input and the image is directly resized, then the operation will lose a lot of sample information, which may lead to the neural network not learning useful sample information, especially the sample information for tiny target pests. Therefore, we improved the sliding window cropping method in YOLT [41], as shown in Figure 4, and our approach could arbitrarily crop images to a specified size (e.g., 1280 × 1280) as the input of the model (the cropped images are called blocks in this paper); adjacent blocks will have β% area overlap.

$$\beta = \frac{k_1 \cap k_2}{k_1} \times 100 \quad (5)$$

where k_1 and k_2 represent the area of the current sliding window and the scope of the next sliding window, respectively, and β is a super parameter used to control the size of the overlapping regions to ensure that each area of the original image is ultimately detected. Although this may result in some parts of an image being repeatedly seen, the NMS algorithm can remove these detected regions [42]. In general, each pest sample image was cut into dozens of blocks of a specified size, these blocks were examined one by one, and then the results were combined to obtain the final detection results of these pest sample images.

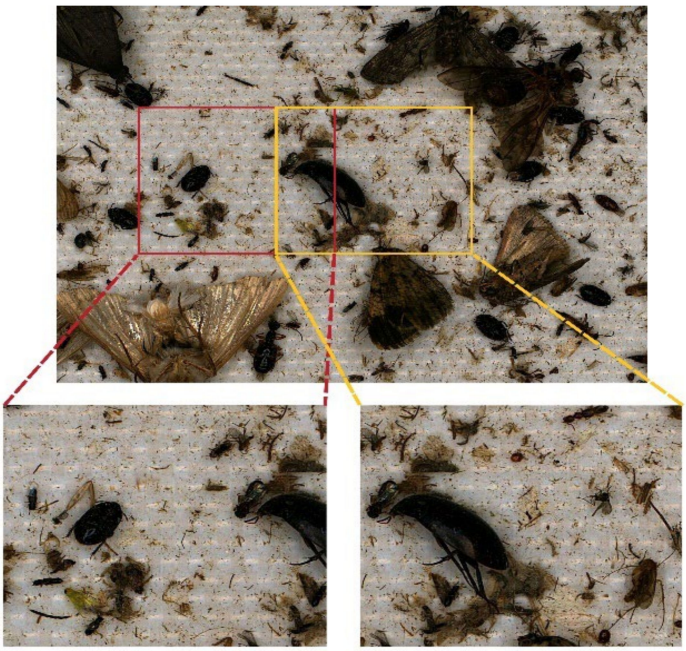

Figure 4. Schematic diagram of our sliding window cropping method.

3.1.2. Data Enhancement

Due to the small number of samples and unbalanced categories in the pest dataset, we introduced the albumentations data-enhancement library to perform data enhancement on these samples, aiming to complement the unstable examples and the small number of samples with different strategies, such as flipping, random radial transformation, blurring, combination, and random selection. Figure 5 shows images of the samples after the unexpected data enhancement of the original pest samples.

All the sample preprocessing processes for this data set can be completed online, which significantly reduces the time of data set reconstruction and makes it easier for us to

make other modifications to the data set. For example, we could cut the sliding window size to any size at will without cutting the entire data set and reloading it offline.

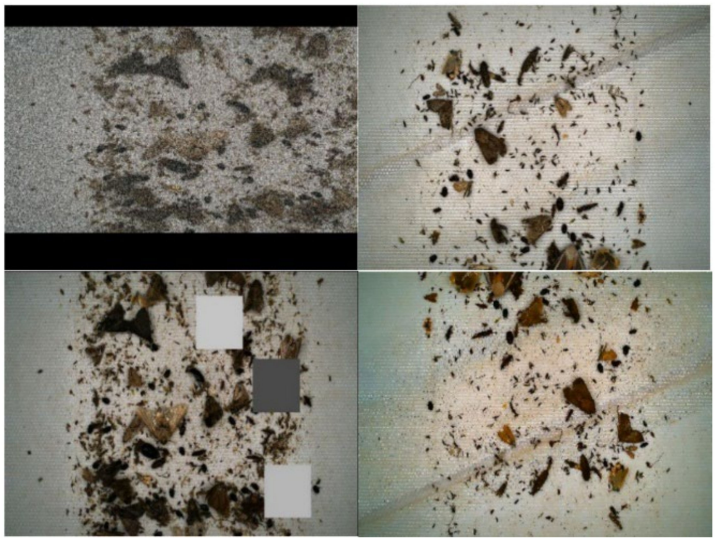

Figure 5. Images obtained after the original pest samples were enhanced by the albumentations data-enhancement method.

3.2. Feature Extraction

Due to the diversity and complexity of pest images, it is difficult for traditional machine learning algorithms with basic convolutional neural networks to classify them. Therefore, the model's accuracy needs to be prioritized, so we compared the experimental results of several current mainstream target-detection models, as shown in Table 1. It can be seen that the YOLOX [43] model with YOLOX-x as the backbone network had the highest Box-AP value, but its required video memory (Men) was nearly four times higher than that of the Cascade R-CNN model with X-101-32×4d-FPN as the backbone network; therefore, in this paper, we improved on the Cascade R-CNNN model and set ResNeXt-101 (32 × 4d) as the backbone network.

Table 1. Comparison of the experimental results of some mainstream target-detection models.

Model	Backbone	Men(GB)	Box-AP
Cascade Rcnn	R-50-FPN	4.2	40.1
Cascade Rcnn	R-101-FPN	6.2	42.3
Cascade Rcnn	X-101-32×4d-FPN	7.6	43.7
YOLOX	YOLOX-s	7.6	40.5
YOLOX	YOLOX-l	19.9	49.4
YOLOX	YOLOX-x	28.1	50.9
Faster Rcnn	R-50-FPN	4.0	37.4
Faster Rcnn	R-101-FPN	6.0	39.4
Faster Rcnn	X-101-32×4d-FPN	7.2	41.2

In Figure 2, the basic structure of the ResNet network is shown on the left, and the basic framework of the ResNeXt network is shown on the right. With the same depth, ResNeXt-101 (32 × 4d) had the same parameters as ResNet101, but its accuracy was higher, so ResNeXt-101 (32 × 4d) was chosen as the backbone network of the model.

3.3. Attention-Based FPN

Attention mechanisms have recently become increasingly popular and can be combined with convolutional neural networks to make models autonomously emphasize sample features that are more useful for the task at hand. We compared the effects of the

attention modules of se [34], cbam [35], and eca [36], and found that the overall model loss decreased faster and the prediction map increased by 0.162 after introducing the cbam attention mechanism to the FPN part of our model for the same epoch.

Specifically, the input image was generally extracted using ResNeXt-101 (32 × 4d) features, which would output three feature layers at different locations of the backbone network: the middle layer, the lower layer, and the bottom layer. When the input image was (640, 640, 3), the three feature layers were feat1 = (80, 80, 256), feat2 = (40, 40, 512), and feat3 = (20, 20, 1024), corresponding to small, medium, and large targets, respectively. Then, the FPN layer could fuse the features of the three previous feature layers with different shapes. In contrast, we introduced the cbam attention mechanism to the PFN layer, as shown in Figure 6, where we included the attention module after the 2D convolution, upsampling, and downsampling, respectively. This allowed the model to learn the sample features P3_out, P4_out, and P5_out from the three feature layers, which were more meaningful for the current task, facilitating future pest-detection tasks.

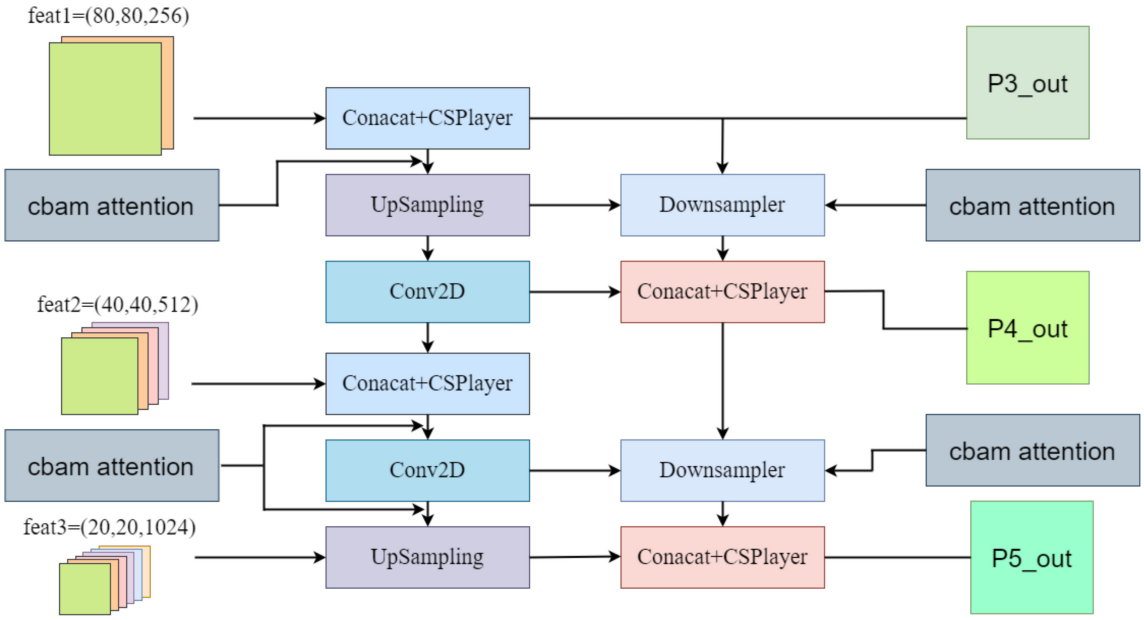

Figure 6. Diagram of the FPN layer adding the cbam attention mechanism.

3.4. Cascaded Structure

Most of the proposals by RPN in Faster RCNN [32] are not of high quality, which leads to there being no way to directly use the high-threshold detector. While the cascade R-CNN [33] uses cascade regression as a resampling mechanism to increase the IOU value of proposals by stage, the resampled recommendations from the previous step can be adapted to the next stage with a higher threshold.

As the IoU threshold selection of R-CNN in Faster R-CNN [16] dramatically influenced the quality of bbox detection, Cascade R-CNN proposed a cascade R-CNN structure, as shown in Figure 7. The average values of C1, C2, and C3 were the classification results of images, and the detection frame output by B3 was the final result. Different stages used different IoU thresholds to recalculate positive and negative samples and sampling strategies to gradually improve the detection frame's quality.

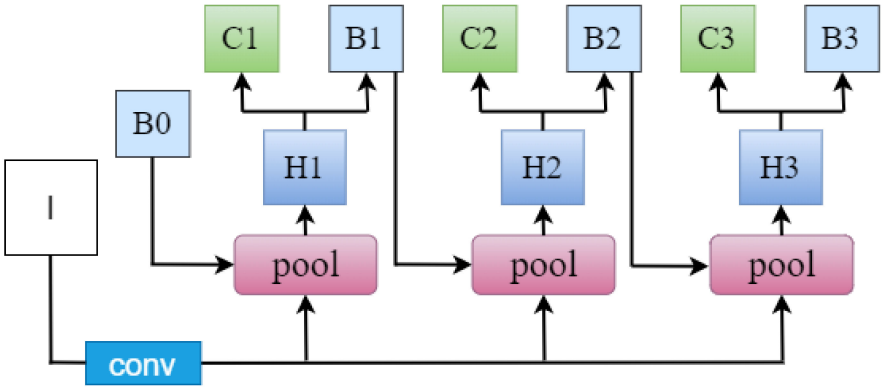

Figure 7. The cascade structure in the Cascade R-CNN network model.

4. Experiment

4.1. Dataset Processing and Partitioning

The dataset used in this paper was from the 10th Teddy Cup Data Mining Challenge, which contained 3015 images with 28 pest types, and some samples of the dataset are shown in Figure 8. Some pest types in the dataset contain too few samples compared with other pest types, so to avoid category imbalance, we used the image enhancement method to extend the data appropriately, as shown in Figure 5. For the division of the dataset, we used a training set:test set ratio of 8.5:1.5.

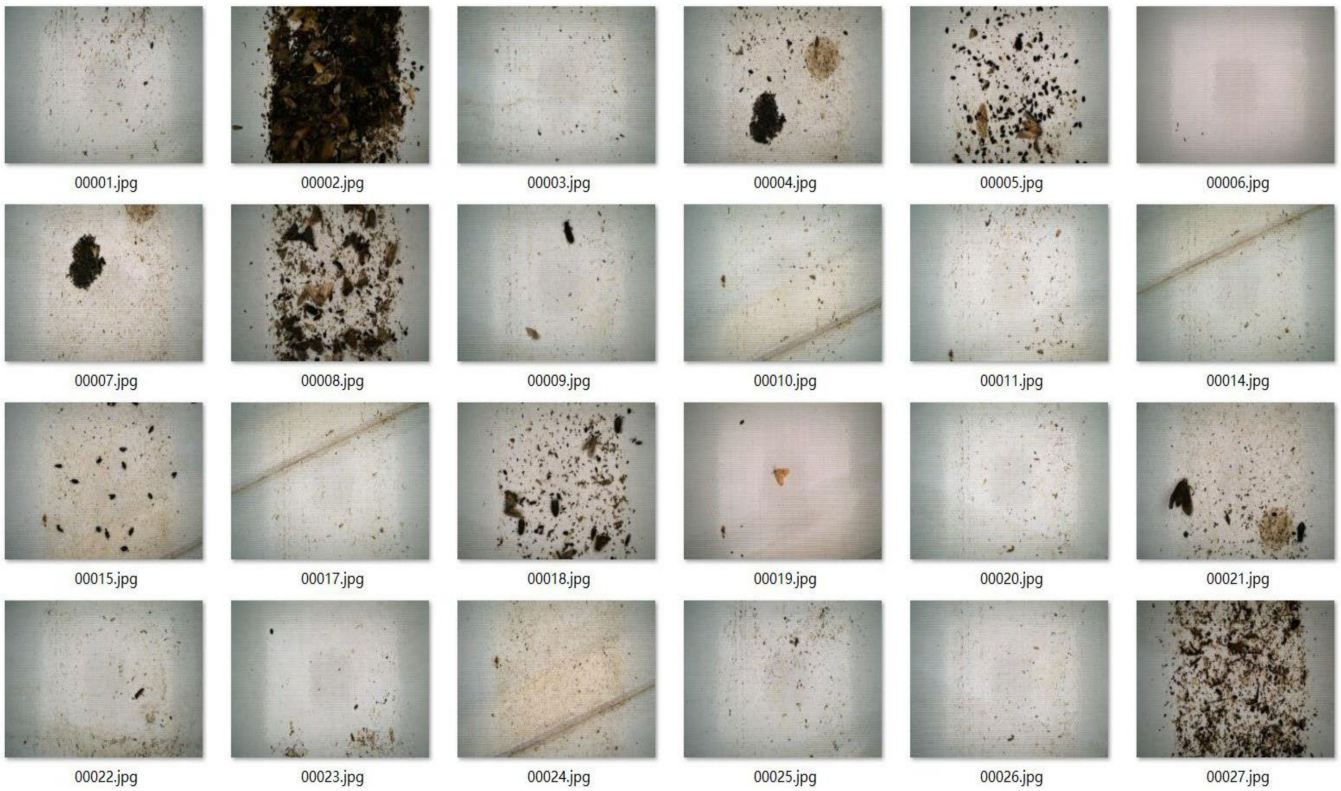

Figure 8. Some of the 10th Teddy Cup Data Mining Challenge Farmland Pest Dataset samples.

In order to make a fair comparison between our model and the previous model, the experimental setup parameters are shown in Table 2.

Table 2. Experimental parameter setting details.

Model	Backbone	Optimizer	Learning Rate	Momentum	Image-Size	Batch-Size	Epochs
Cascade Rcnn	R-50-FPN	SGD	0.005	0.9	1280 × 1280	64	100
Cascade Rcnn	R-101-FPN	SGD	0.005	0.9	1280 × 1280	64	100
Cascade Rcnn	X-101-32×4d-FPN	SGD	0.005	0.9	1280 × 1280	64	100
YOLOX	YOLOX-s	SGD	0.005	0.9	1280 × 1280	64	100
YOLOX	YOLOX-l	SGD	0.005	0.9	1280 × 1280	64	100
YOLOX	YOLOX-x	SGD	0.005	0.9	1280 × 1280	64	100
Faster Rcnn	R-50-FPN	SGD	0.005	0.9	1280 × 1280	64	100
Faster Rcnn	R-101-FPN	SGD	0.005	0.9	1280 × 1280	64	100
Faster Rcnn	X-101-32×4d-FPN	SGD	0.005	0.9	1280 × 1280	64	100
Ours	X-101-32×4d-FPN	SGD	0.005	0.9	1280 × 1280	64	100

4.2. Evaluation Indicators

Unlike the classification problem, target detection not only aims to accurately classify the target, but also to accurately localize the target. The most common evaluation metrics used in target detection algorithms for detection accuracy are Precision, Recall, F1-Score, Average Precision (AP), Mean Average Precision (mAP), and Intersection Over Union (IOU), defined as follows.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{6}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{7}$$

$$\text{AP}_\text{j} = \int_0^1 \text{p(r)dr} \tag{8}$$

$$\text{mAP} = \frac{1}{\text{m}} \sum_\text{m} \text{AP}_\text{j} \tag{9}$$

$$\text{F1} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{10}$$

$$\text{IOU} = \frac{\text{Area of Overlap}}{\text{Area of Union}} \tag{11}$$

4.3. Experimental Results

Figure 9 shows that the scores of mAP@0.5 and mAP@0.5:0.95 in our model for the pest dataset were 0.84 and 0.65, respectively. We believe that, when the IOU threshold gradually increased, some pest types were difficult to assess because of the poor learning effect of their sample characteristics due to the small number of samples, leading to this result.

It can be seen from Figure 10 that our model could accurately localize as well as classify most of the insects; however, the detection performance of the network was not so good when many bugs were stacked together or when the number of samples of certain classes of bugs was small.

We also compared the experimental results of different models on pest datasets. As shown in Table 3, under the same experimental environment, our proposed method had advantages over other models, and its mAP_0.5 was 11.81% higher than that of the Cascade R-CNN model with ResNeXt-101 as the backbone network without any change.

To investigate the impact of the proposed sliding window cropping method and the attention-based FPN, we conducted experiments on them separately. From Table 4, we can see that the sliding window cropping method could solve the problem of semantic information loss in large image input networks and improve the model's prediction accuracy. In addition, adding the attention module to the FPN layer could also enhance the model's performance.

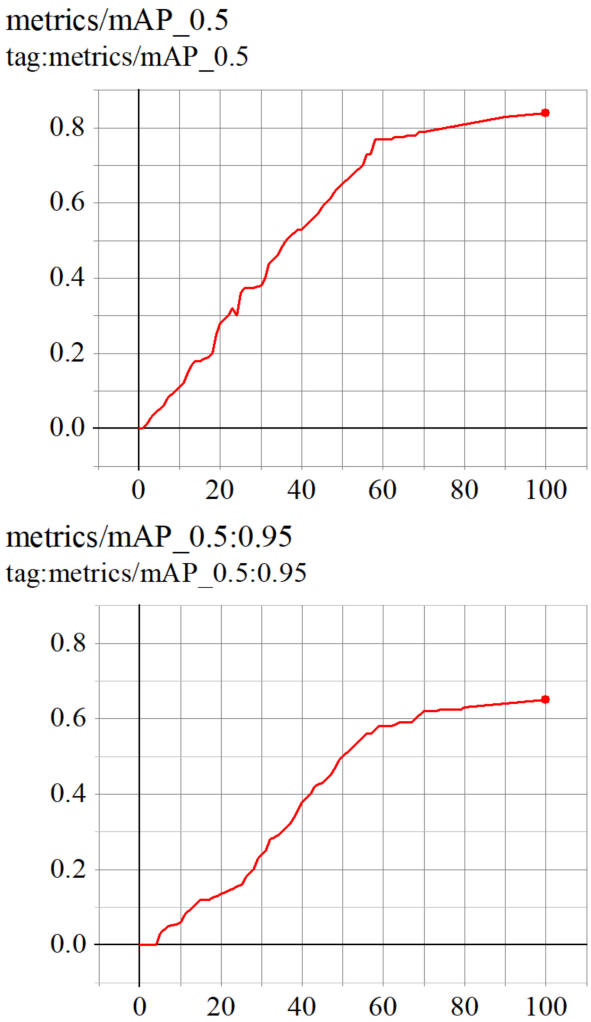

Figure 9. Performance comparison of mAP_0.5 and mAP_0.5:0.95.

Table 3. Experimental results of different models on the test set of the pest dataset.

Method	Backbone	mAP_0.5	mAP_0.5:0.95	Precision	F1-Score
Cascade Rcnn	R-101-FPN	70.62	42.35	45.36	67.64
Cascade Rcnn	X-101-32×4d-FPN	72.35	45.12	68.54	71.21
YOLOX	YOLOX-s	66.39	40.16	64.37	63.87
YOLOX	YOLOX-l	79.82	61.82	75.42	77.15
Faster Rcnn	R-101-FPN	65.45	49.67	51.56	63.24
Faster Rcnn	X-101-32×4d-FPN	68.16	51.84	57.63	66.83
Ours	X-101-32×4d-FPN	84.16	65.23	67.79	82.34

Table 4. Effect of the sliding window cropping method and attention-based FPN on the model.

Method	Backbone	Sliding Window Cutting	Add Attention to the FPN	mAP_0.5	mAP_0.5:0.95	Precision	F1-Score
Ours	X-101-32×4d-FPN	×	×	72.35	45.12	68.54	71.21
Ours	X-101-32×4d-FPN	×	√	75.64	47.17	69.03	76.36
Ours	X-101-32×4d-FPN	√	×	76.38	46.58	67.52	77.49
Ours	X-101-32×4d-FPN	√	√	84.16	65.23	67.79	82.34

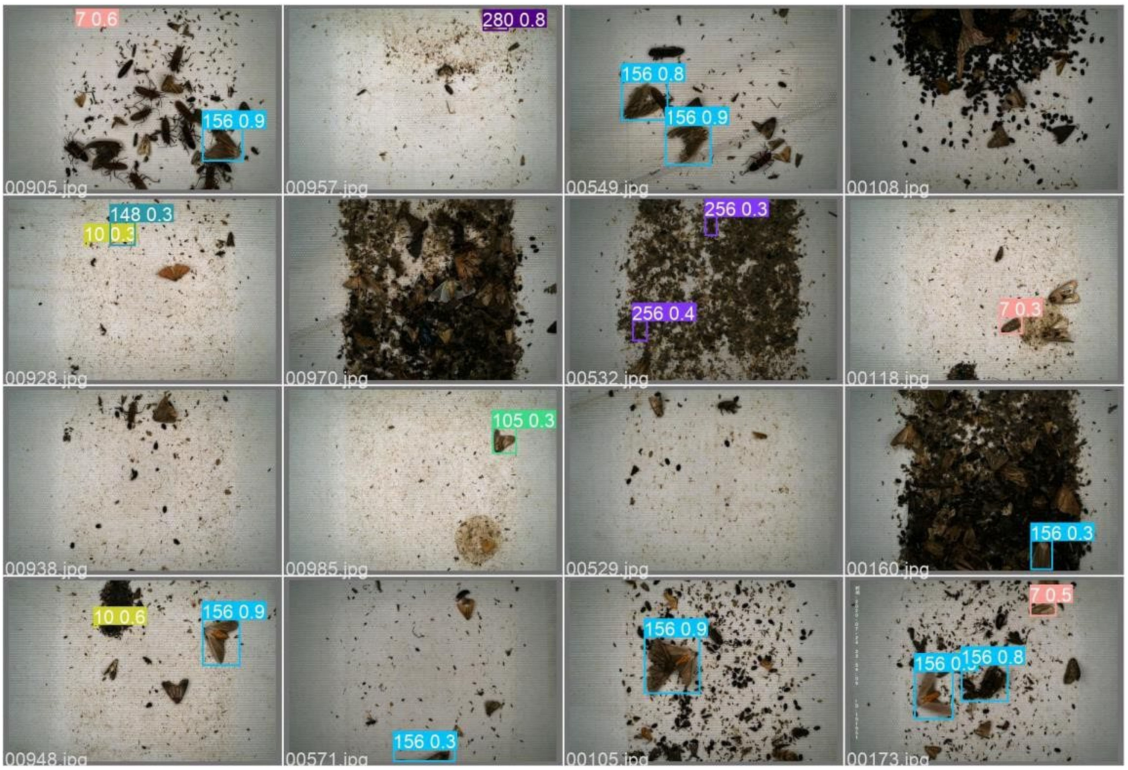

Figure 10. Prediction results for some of the pests in our model.

5. Conclusions

We propose a new target-detection framework based on the Cascade RCNN to address the problems of large sample image sizes, a small number of samples, and class imbalance in the pest sample dataset. Specifically, we developed a sliding window cropping method that could learn sample features more accurately and in detail without changing the original input image size. Secondly, the attention mechanism was also perfectly combined with the FPN layer in the model, aiming to enable it to learn sample features that are more important to the task at hand. Experiments showed that our model could localize and classify pests more accurately than the current mainstream target-detection models. On the pest data set, the model's average accuracy (mAP@0.5) was 84.16%, (mAP@0.5:0.95) was 65.23%, the accuracy was 67.79%, and the F1 score was 82.34%. This demonstrates that our proposed sliding window cropping method and attention-based FPN layer could be well-integrated with other target-detection models to better locate and identify samples in the pest dataset.

In future work, to apply our pest detection system to real life, we could embed this model into modern mobile devices, which would be very helpful in promoting agricultural production. In addition, we could add images of pest species from time to time to increase the number of datasets in order to detect pests more accurately.

Author Contributions: Conceptualization, W.X., L.S. and C.Z.; methodology, W.X., L.S. and B.L.; software, C.Z. and Z.Y.; validation, W.X. and C.Z.; writing—original draft preparation, W.X., L.S. and W.Y.; writing—review and editing, Z.Y. and W.Y. 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: Not applicable.

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

References

1. Taigman, Y.; Yang, M.; Ranzato, M.A.; Wolf, L. Deepface: Closing the gap to human-level performance in face verification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 1701–1708.
2. Huang, W.; Qiao, Y.; Tang, X. Robust scene text detection with convolution neural network induced mser trees. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; Springer: Cham, Switzerland, 2014; pp. 497–511.
3. Zhang, Z.; Zhang, C.; Shen, W.; Yao, C.; Liu, W.; Bai, X. Multi-oriented text detection with fully convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 4159–4167.
4. Ouyang, W.; Wang, X. Joint deep learning for pedestrian detection. In Proceedings of the IEEE International Conference on Computer Vision, Sydney, Australia, 1–8 October 2013; pp. 2056–2063.
5. Zhang, L.; Lin, L.; Liang, X.; He, K. Is faster R-CNN doing well for pedestrian detection. In Proceedings of the European Conference on Computer Vision, Las Vegas, NV, USA, 27–30 June 2016; Springer: Cham, Switzerland, 2016; pp. 443–457.
6. Hoi, S.C.; Wu, X.; Liu, H.; Wu, Y.; Wang, H.; Xue, H.; Wu, Q. Logo-net: Large-scale deep logo detection and brand recognition with deep region-based convolutional networks. *arXiv* **2015**, arXiv:1511.02462.
7. Kleban, J.; Xie, X.; Ma, W.Y. Spatial pyramid mining for logo detection in natural scenes. In Proceedings of the 2008 IEEE International Conference on Multimedia and Expo, Hannover, Germany, 26 April–23 June 2008; IEEE: Piscataway, NJ, USA, 2008; pp. 1077–1080.
8. Kang, K.; Li, H.; Yan, J.; Zeng, X.; Yang, B.; Xiao, T.; Zhang, C.; Wang, Z.; Wang, R.; Wang, X.; et al. T-cnn: Tubelets with convolutional neural networks for object detection from videos. *IEEE Trans. Circuits Syst. Video Technol.* **2017**, *28*, 2896–2907. [CrossRef]
9. Kang, K.; Ouyang, W.; Li, H.; Wang, X. Object detection from video tubelets with convolutional neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 817–825.
10. Chen, X.; Xiang, S.; Liu, C.L.; Pan, C.H. Vehicle detection in satellite images by parallel deep convolutional neural networks. In Proceedings of the 2013 2nd IAPR Asian Conference on Pattern Recognition, Washington, DC, USA, 5–8 November 2013; IEEE: Piscataway, NJ, USA, 2013; pp. 181–185.
11. Fan, Q.; Brown, L.; Smith, J. A closer look at Faster R-CNN for vehicle detection. In Proceedings of the 2016 IEEE Intelligent Vehicles Symposium (IV), Gothenburg, Sweden, 19-22 June 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 124–129.
12. Litjens, G.; Kooi, T.; Bejnordi, B.E.; Setio, A.A.A.; Ciompi, F.; Ghafoorian, M.; Van Der Laak, J.A.; Van Ginneken, B.; Sánchez, C.I. A survey on deep learning in medical image analysis. *Med. Image Anal.* **2017**, *42*, 60–88. [CrossRef] [PubMed]
13. 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]
14. Boedeker, W.; Watts, M.; Clausing, P.; Marquez, E. The global distribution of acute unintentional pesticide poisoning: Estimations based on a systematic reviez. *BMC Public Health* **2020**, *20*, 1875. [CrossRef]
15. Hu, Z.; Xu, L.; Cao, L.; Liu, S.; Luo, Z.; Wang, J.; Li, X.; Wang, L. Application of non-orthogonal multiple access in wireless sensor networks for smart agriculture. *IEEE Access* **2019**, *7*, 87582–87592. [CrossRef]
16. Yang, G.; Bao, Y.; Liu, Z. Localization and identification of pests in tea plantations based on image saliency analysis and convolutional neural network. *Trans. Chin. Soc. Agric. Eng.* **2017**, *33*, 156–162.
17. Xie, C.; Li, R.; Dong, W.; Song, L.; Zhang, J.; Chen, H.; Chen, T. Image recognition of farmland pests based on sparse coding pyramid model. *Trans. Chin. Soc. Agric. Eng.* **2016**, *32*, 144–150.
18. Sun, P.; Chen, G.; Chao, L. Image recognition of soybean pests based on attentional convolutional neural network. *China J. Agric. Mech.* **2020**, *41*, 171–176.
19. Cheng, X.; Zhang, Y.; Chen, Y.; Wu, Y.; Yue, Y. Pest identification via deep residual learning in complex background. *Comput. Electron. Agric.* **2017**, *141*, 351–356. [CrossRef]
20. Fuentes, A.; Yoon, S.; Kim, S.C.; Park, D.S. A robust deep-learning-based detector for real-time tomato plant diseases and pests recognition. *Sensors* **2017**, *17*, 2022. [CrossRef]
21. Jiao, L.; Dong, S.; Zhang, S.; Xie, C.; Wang, H. AF-RCNN: An anchor-free convolutional neural network for multi-categories agricultural pest detection. *Comput. Electron. Agric.* **2020**, *174*, 105522. [CrossRef]
22. Sabanci, K.; Aslan, M.F.; Ropelewska, E.; Unlersen, M.F.; Durdu, A. A Novel Convolutional-Recurrent Hybrid Network for Sunn Pest–Damaged Wheat Grain Detection. *Food Anal. Methods* **2022**, *15*, 1748–1760. [CrossRef]
23. Gambhir, J.; Patel, N.; Patil, S.; Takale, P.; Chougule, A.; Prabhakar, C.S.; Managanvi, K.; Raghavan, A.S.; Sohane, R.K. *Deep Learning for Real-Time Diagnosis of Pest and Diseases on Crops*; Intelligent Data Engineering and Analytics; Springer: Singapore, 2022; pp. 189–197.
24. Li, D.; Wang, R.; Xie, C.; Liu, L.; Zhang, J.; Li, R.; Wang, F.; Zhou, M.; Liu, W. A recognition method for rice plant diseases and pests video detection based on deep convolutional neural network. *Sensors* **2020**, *20*, 578. [CrossRef]
25. Bengio, Y.; Courville, A.; Vincent, P. Representation learning: A review and new perspectives. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*, 1798–1828. [CrossRef]

26. Kavukcuoglu, K.; Sermanet, P.; Boureau, Y.L.; Gregor, K.; Mathieu, M.; Cun, Y. Learning convolutional feature hierarchies for visual recognition. *Adv. Neural Inf. Process. Syst.* **2010**, *23*, 1090–1098.
27. Oquab, M.; Bottou, L.; Laptev, I.; Sivic, J. Learning and transferring mid-level image representations using convolutional neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Zurich, Switzerland, 6–12 September 2014; pp. 1717–1724.
28. Rumelhart, D.E.; Hinton, G.E.; Williams, R.J. Learning representations by back-propagating errors. *Nature* **1986**, *323*, 533–536. [CrossRef]
29. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Commun. ACM* **2017**, *60*, 84–90. [CrossRef]
30. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Zurich, Switzerland, 6–12 September 2014; pp. 580–587.
31. 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), Gothenburg, Sweden, 19–22 June 2016; pp. 779–788.
32. Girshick, R. Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–12 December 2015; pp. 1440–1448.
33. Cai, Z.; Vasconcelos, N. Cascade r-cnn: Delving into high quality object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 6154–6162.
34. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7132–7141.
35. Woo, S.; Park, J.; Lee, J.Y.; Kweon, I.S. Cbam: Convolutional block attention module. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 3–19.
36. Wang, Q.; Wu, B.; Zhu, P.; Li, P.; Zuo, W.; Hu, Q. Supplementary material for 'ECA-Net: Efficient channel attention for deep convolutional neural networks. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; IEEE: Seattle, WA, USA, 2020; pp. 13–19.
37. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Gothenburg, Sweden, 19–22 June 2016; pp. 770–778.
38. Xie, S.; Girshick, R.; Dollár, P.; Tu, Z.; He, K. Aggregated residual transformations for deep neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1492–1500.
39. Chen, X.; Sun, Y.; Zhang, Q.; Liu, F. Two-stage grasp strategy combining CNN-based classification and adaptive detection on a flexible hand. *Appl. Soft Comput.* **2020**, *97*, 106729. [CrossRef]
40. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
41. Van Etten, A. You only look twice: Rapid multi-scale object detection in satellite imagery. *arXiv* **2018**, arXiv:1805.09512.
42. Neubeck, A.; Van Gool, L. Efficient non-maximum suppression. In Proceedings of the 18th International Conference on Pattern Recognition (ICPR'06), Washington, DC, USA, 20–24 August 2006; IEEE: Piscataway, NJ, USA, 2006; Volume 3, pp. 850–855.
43. Ge, Z.; Liu, S.; Wang, F.; Li, Z.; Sun, J. Yolox: Exceeding yolo series in 2021. *arXiv* **2021**, arXiv:2107.08430.