

Research Article

Lightweight Field Insect Recognition and Classification Model Based on Improved Deep Learning under Complex Background

Feng Wu  and Yueying Li 

College of Information Engineering, Xinyang Agriculture and Forestry University, Xinyang 464000, China

Correspondence should be addressed to Yueying Li; liyueying@xyafu.edu.cn

Received 4 November 2022; Revised 16 February 2023; Accepted 22 February 2023; Published 7 April 2023

Academic Editor: Jiacheng Yang

Copyright © 2023 Feng Wu and Yueying Li. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Insect monitoring in the field is an extremely important part of the agricultural production system. Recent advances in computer technology have provided the technical foundation for automatic field insect monitoring. In insect automatic monitoring, insect recognition and classification based on images is one of the most active research areas. Rapid advancements in computer vision technology based on deep learning have provided new ideas for implementing automatic field insect monitoring. Firstly, the field insect images are preprocessed and input to the lightweight algorithm for feature extraction, and the prediction networks of different sizes are output by multiscale feature fusion; then, the joint cross-merge ratio is introduced for automatic identification and classification of field insects. Compared with other algorithms, the simulation results show that the proposed algorithm has higher accuracy, less time consumption, and stronger robustness. It effectively solves the insect accumulation and background interference problems and can identify field insects online in real time.

1. Introduction

Insects are closely connected to human life and play an important part in the functioning of the biosphere [1]. There are more than a million different species of insects on this planet [2]. There are some insect species that contribute to the development and reproduction of plants. However, there are also insect species that may cause significant damage to the development of agricultural and forestry crops and to the storage of agricultural and forestry products [3]. Therefore, accurate and timely insect detection is an important research direction in the field of insect research. Early insect detection is typically accomplished through manual classification and counting performed by professionals with the assistance of professional knowledge [4].

In recent years, due to the progress of computer vision, insect detection based on image processing technology has attracted the attention of a large number of experts and scholars [5]. The data that were used in the early stages of image-based research on insects were typically specimen data with a high resolution or insect data with a simple

background and a common posture. These images were captured by sophisticated equipment in the laboratory after the insects had been collected in the field. In most cases, there is just one insect specimen depicted in the image, and it takes the center stage [6]. Therefore, early insect detection was primarily a problem of classification. Researchers have begun to try to directly analyze insect data captured in field scenes, which typically have complex backgrounds and insufficient insect subjects [7]. This is possible due to the development of technology and the increase in demand. It is necessary to separate the insect subject from the background by performing a large number of complex manual or semimanual preprocessing steps. This significantly increases the amount of manual participation required in the recognition process and does not lend itself well to the scaling up and automating of insect detection. The traditional form of computer vision requires the steps of feature extraction to be completed. The concept is extracted from a group of image objects that share as many characteristics as possible. One example is the insect extraction task, which requires the insects' colour, shape, and texture features to be extracted, as

well as the incorporation of the extracted features into an artificial neural network [8], decision tree [9], or support vector machine classifier [10]. This feature extraction method for image classification must be selected manually, which means that the extraction and selection of features must be performed manually. Not only is this a more time-consuming process, but it is also more susceptible to being influenced by the subjective factors of researchers.

Computer vision methods that are based on deep learning are becoming increasingly popular and have made significant progress in a variety of fields, including the detection of pedestrians [11]. This is due in large part to the rapid development of computer hardware over the past few years. In particular, the performance of graphics computing unit devices has improved significantly. Deep learning introduces the concept of end-to-end learning, which means that researchers no longer need to manually extract features from images to achieve end-to-end location detection [12]. And, they only need to inform the machine about the specific object class to be learned [13]. The network was trained using deep learning, and it then automatically formulated the characteristics that were the most descriptive and important for each object [14]. To put it another way, the network was successful in identifying possible patterns across a variety of image types. Since deep learning can handle the complexity of farming environments, it provides new avenues for research and development in the field of computer vision-based techniques for agricultural insect detection and classification. Incorporating depth wise separable convolution into the feature extraction layer allows for a significant cut in the amount of calculation that must be performed. We introduced generalized intersection over union (GIoU) as a way to improve the accuracy of the network prediction target. This was accomplished by using multiscale feature fusion, which results in the output of prediction networks of varying sizes. These networks are designed to ensure the detection efficiency of insects of varying sizes. In conclusion, when contrasted with the existing classical algorithms, the findings demonstrate that the proposed algorithm achieves higher accuracy in the automatic identification and classification of field insects, accelerates the efficiency of automatic identification and classification of field insects, and has the potential to be applied in a diverse range of contexts. This opens the door to a number of new avenues of research and development.

2. Literature Review

Traditional insect monitoring work is generally carried out by plant protection workers to conduct field surveys and classify insects by relying on their own experience or consulting professional books [15]. This type of work has high labour intensity and poor timeliness, which cannot meet the current needs of pest occurrence monitoring and hinders rapid decision-making in the process of agricultural pest control [16]. Therefore, the development of some automated insect identification and counting methods is helpful to the accuracy and effectiveness of insect monitoring so as to reduce the loss of agricultural economy caused by insect

pests every year and further improve the implementation of Internet identification and conversion [17]. The progress of information technology has promoted the technical level of the agricultural production system. The automatic detection and classification of insects based on images are one of the research hotspots in this field, and scholars have begun to combine sensor and computer technology for automatic insect classification [18]. Early automatic recognition and counting of insect images are cumbersome, as shown in Figure 1.

It is mainly divided into several steps: insect image acquisition, insect image preprocessing, insect target segmentation, insect feature extraction, insect target classification, insect recognition, and counting [19]. Among them, image processing includes image enhancement, edge detection, and image content segmentation. Image enhancement is extracting interesting information through image transformation to avoid poor image quality caused by unreasonable illumination and angle. The main process is as follows: first, the collected insect images are subjected to image processing and background-foreground separation, and then category analysis is performed using image recognition methods. Computer-based insect image segmentation mainly relies on traditional image segmentation methods, such as threshold-based segmentation methods, edge flow-based segmentation methods, and segmentation methods based on wavelet analysis (see Figure 2).

Abdul et al. used multiresolution segmentation and other methods to realize the identification and counting of pests, such as rice longitudinal leaf whorl [20]. In this stage of feature extraction, a variety of insect features, such as morphological features [21], texture features [22], and local features [23] need to be selected. Wen et al. effectively realized the classification of various fruit tree pests by combining local features and global features and establishing a model [24]. The steps of an insect target detection algorithm based on traditional computer vision are cumbersome. Xie et al. adopted multitask sparse representation and multiple kernel learning and comprehensively used insect texture, colour, shape, HOG histogram, and other features to realize the classification of a variety of different field insects [25]. Yang et al. used image retrieval technology [26] to establish a binomial search table based on the morphological taxonomic identification features of insects. Then, MySQL was used to establish a taxonomic database of common vector insects to realize the taxonomic identification of insects in the field. Although these methods have made some progress, there is still a certain distance from the practical application due to the complex interference of the agricultural production environment. As shown in Figure 3, researchers need to manually extract the dominant features of insects, such as colour, shape, and texture, and then classify insects by BP neural network, support vector machine (SVM), pattern recognition, binary tree recognition, and other methods.

In recent years, object detection algorithms based on deep learning have made significant progress in other fields, so many researchers have tried to apply deep learning to the field of insect recognition. Yang et al. used a model based on

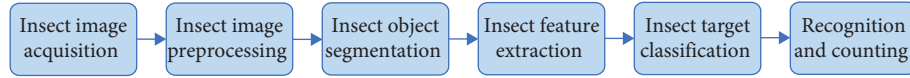


FIGURE 1: Traditional insect identification and counting workflow.

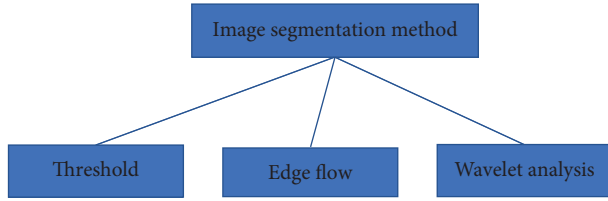


FIGURE 2: Background and target insects' segmentation.

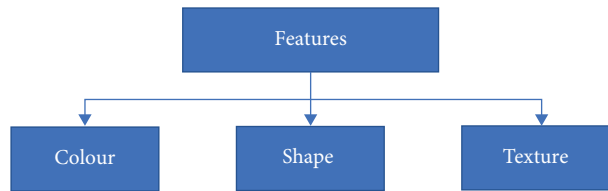


FIGURE 3: Features which need manual extraction.

a deep residual network [27] to classify ten insects in a complex context and obtained 98.67% accuracy, while the SVM-based classification method obtained only 44.00% accuracy in the same dataset [28]. In practical applications, since a variety of insects appear in the picture simultaneously, it is necessary to use deep learning based on object detection algorithms. The object detection algorithm based on deep learning provides an end-to-end training method without manual feature extraction, which dramatically simplifies the process of insect identification and counting. Zhong et al. used sensors to shoot insects on sticky plates. Due to the small number of samples, the YOLO target detection algorithm and SVM classifier were combined to complete the detection and classification of insects, respectively, which had certain advantages compared with the pure YOLO algorithm, but the speed was significantly slower [29]. The discriminative information of insect images often exists in a very fine region, making it difficult to extract the desired features from the whole image level. Chen and Chen proposed a bilinear pooling convolutional neural network based on feature fusion to classify insect images [30]. This method reduces the level of detail in insect images and can effectively extract higher-order features. However, the model suffers from slow parameter convergence and does not take into account the influence of other classes of training sample distribution, therefore, the recognition results of this method are often unsatisfactory. Compared with traditional computer vision methods, the object detection algorithm based on deep learning does not need to manually extract insect features, so it can avoid the interference of human subjective factors and greatly simplify the process of automatic identification and counting of insects. In the above works, deep learning-based methods usually achieve better results than traditional computer vision methods. With the

reduction of hardware cost and the maturity of software ecology, the method based on deep learning has become the mainstream research direction in the vision-based automatic identification and classification field.

3. Lightweight Deep Learning-Based Field Insects Recognition and Classification Model

3.1. Architecture of the Model. Since their inception, the YOLO series of object detection algorithms have maintained a solid reputation for both their lightning-fast detection speed and their high level of accuracy. YOLO and YOLOv2 served as the foundation upon which YOLOv3 was built. In the convolutional network, the external network concentrates on information that is more specific, while the deep network concentrates on information that is more specific to its meaning. Deep semantic information is helpful for accurately detecting the target; however, shallow detail information has the potential to improve detection accuracy. If only a small amount of in-depth information is used, detection performance may suffer as a result. In order to achieve this goal, this article proposes the design of an automatic identification and classification algorithm for field insects that is based on a lightweight deep learning model. The architecture of this model can be seen in Figure 4.

Firstly, the image was input into the algorithm. Secondly, the feature extraction network was used to output three feature maps of different sizes. Then, the idea of feature pyramid networks (FPN) was used to predict the feature layers of different sizes in the feature extraction network, and upsample and feature fusion were used to fuse the feature information of multiple scales together. Finally, the detection was performed independently on the fusion feature maps of multiple scales.

3.2. Model Compression Strategy. In practical applications, deep neural networks are computationally expensive and therefore difficult to deploy. This article is based on YOLOv3 algorithm for target recognition, but the size of YOLOv3 weight file is generally above 200 MB. A large number of computations are mainly from convolutional operations. The purpose of the convolution operation is to extract more features, and the deeper the network level is, the more features can be mined. For tasks that require fewer target categories, or insect data sets with small samples, there is a lot of “redundancy” in convolutional operations. Therefore, it is necessary to reduce the computation time by reducing the model and memory consumption without affecting the computational accuracy and reducing the amount of operations. In this article, we perform model compression by sparsifying the network at different structural levels and then by a channel pruning strategy based on BN layers.

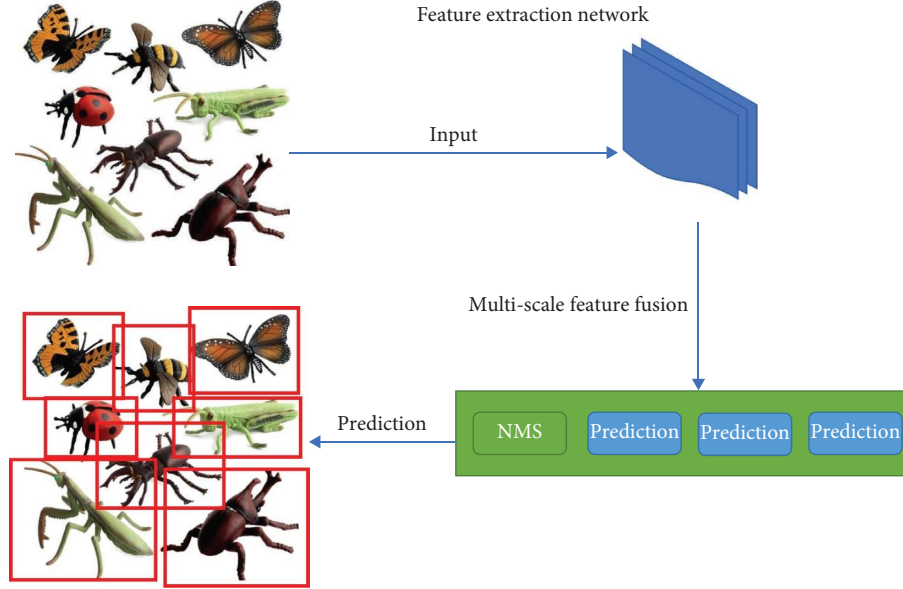


FIGURE 4: Framework of lightweight insect detection and classification model.

The total compression scheme is shown in Figure 5. Initial pretraining was first performed to converge to a high accuracy, and then sparse training is performed by scaling factors in the BN layer. The model measures the importance of each channel by learning. Subsequently, the unimportant channels, i.e., the convolution kernel and the corresponding feature maps, were pruned according to the set pruning ratio, and the model width was narrowed. In this article, the initial setting of 70% pruning ratio is iterated to finally reach 97% pruning ratio. Based on the channel pruning, the importance of the residual layers was measured by the sum of the scale factors of each residual layer, and a certain number of residual layers are pruned out. The number of model forward inference layers is reduced and finally fine-tuned to recover the accuracy. The whole process can be iterated in a loop. The pruning of the residual layers in each iteration is optional and is shown as a dashed line in the Figure 5.

3.3. Lightweight Feature Extraction Network. In order to reduce the model performance overhead, this article redesigned the feature extraction network based on depth wise separable convolution (DSC). DSC is based on the following assumptions: The mapping of cross-channel correlation and spatial correlation in the convolutional neural network feature map can be separated. The principle of depth-separable convolution is shown in Figure 6. There are two cycles in the DSC network. Deep convolutional separation can significantly reduce the calculation of the number of parameters. Assume that the input $D_F \times D_F \times M$ characteristics of layer, including M for channel number, size of convolution kernels for $D_K \times D_K \times N$. When the output feature size is $(D_P \times D_P \times M)$, the computation cost in a conventional convolutional network is $(D_P \times D_P \times M) \times D_K \times D_K \times N$. The convolution operation is divided into two steps in a deeply separable network. In

the first step, deep convolution is performed, and the input features of each layer are convolved by using the convolution of size $D_K \times D_K \times 1$. At this stage, the network computation amount is $(D_P \times D_P \times 1) \times D_K \times D_K \times M$, and the depth of the feature map remains unchanged. The second step is the point convolution operation. The convolution kernel size is $1 \times 1 \times M$, and the output feature map size is $(D_P \times D_P \times N)$, and the calculation amount is $D_K \times D_K \times N \times (1 \times 1 \times M)$.

When $D_K = 3$, the depth of separable convolution of 8 to 9 times less than the standard convolution computation, on the one hand, speeds up the detection speed, on the other hand, reduces the memory footprint.

In the default configuration of YOLOv3, the network resolution is 416 by 416. Because the size of the image used in this investigation is predetermined, it must be scaled and filled in a manner that is adaptable to the network resolution. This process leaves some areas of the image with blank areas. As a result, the input resolution of the feature extraction network has been designed to be 608 by 480, and there is no need to fill too many blank areas after the image has been scaled.

3.4. Generalized Intersection over Union. Intersection over Union (IoU) can be used to measure the similarity of two bounding frames and is an important metric used in the field of target detection to evaluate the performance of target detectors. The higher the overlap between the prediction frame and the real frame, the larger the IoU value, which is calculated as shown in the following equation:

$$\text{IoU} = \frac{A \cap B}{A \cup B}, \quad (1)$$

where A and B represent the prediction frame and the real frame, respectively.

If IoU is used directly as the bounding frame loss, IoU cannot measure the distance between two bounding frames

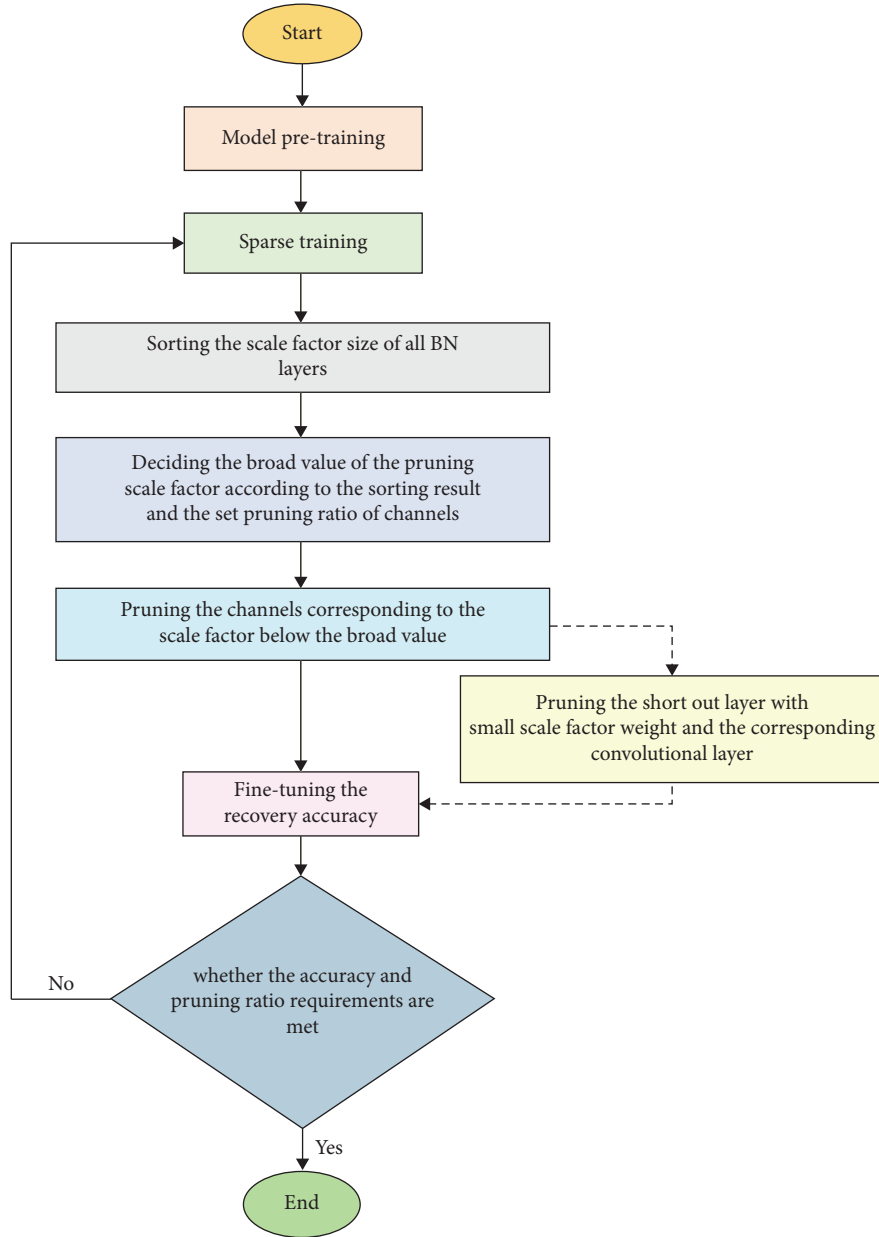


FIGURE 5: Model compression process.



FIGURE 6: Principle of deep separable convolution.

when there is no overlap between the prediction frame and the real frame. When the regions do not overlap, IoU is 0, which cannot reflect the relationship between regions. As shown in Figure 7, IoU is 0 in both (a) and (b), but the distance between the prediction frame and the real frame in Figure 7(a) is obviously closer, and its prediction is better.

Based on this, the GIoU proposed in this article makes full use of the advantages of IoU with scale invariance and can be used as the distance between two frames. At the same time, it overcomes the shortcomings of IoU when the

prediction frame does not overlap with the real frame, and can better reflect the overlap between the prediction frame and the real frame. As shown in Figure 8, the white frame range is the representation area of C .

The GIoU is calculated as shown in the following equation:

$$GIoU = IoU - \frac{|C \setminus (A \cup B)|}{|C|}. \quad (2)$$

Where A and B are the prediction frame and the true frame, respectively, and C is the minimum closed frame containing both.

$$GIoU = -1 + \frac{A \cup B}{C}. \quad (3)$$



FIGURE 7: Different scenarios when IoU is 0. (a) The prediction frame is far away from the real frame. (b) The prediction frame is closer to the real frame.

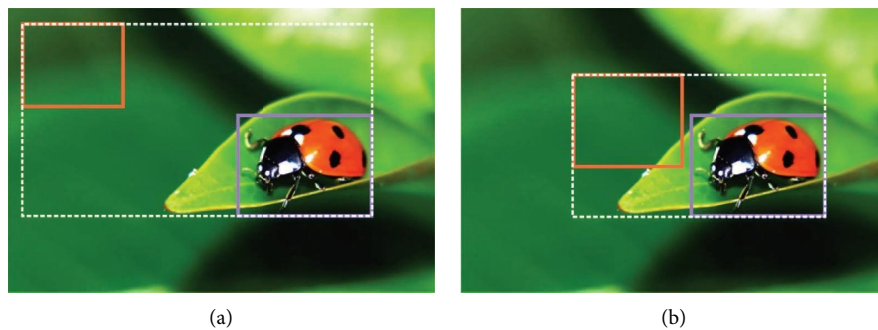


FIGURE 8: Representation area of C in GIoU. (a) The prediction frame is far away from the real frame. (b) The prediction frame is closer to the real frame.

From equations (2) and (3), the GIoU varies in the range $(-1, 1]$ and $\text{GIoU} = 1$ when the prediction frame overlaps with the real frame. When the prediction frame does not coincide with the real frame, i.e., $\text{IoU} = 0$, the formula for calculating GIoU can be transformed into an equation.

The farther the prediction frame is from the real frame, the closer $A \cup B \setminus C$ is to 0 and the closer GIoU is to -1 , and the closer the prediction frame is to the real frame, the closer $A \cup B \setminus C$ is to 1 and the closer GIoU is to 0. Therefore, compared to IoU, GIoU can not only reflect the relationship between nonoverlapping regions, but also better evaluate the two boundary frame's overlap degree.

To show the superiority of GIoU, two different prediction results are evaluated using IoU and GIoU, respectively, where Figure 9(a) predicts that the frame is offset from the center of the real frame, Figure 9(b) predicts that the frame is completely within the real frame and the distribution is close to the actual target. When IoU is used as the criterion, both prediction results score is 0.56, which cannot reflect the difference. When GIoU is used as the criterion, the center of the prediction frame in Figure 9(b) is closer to the true value and the boundary is not beyond the true target, so the model tends to choose the prediction result in Figure 9(b) more.

4. Results and Discussion

4.1. Experimental Environment. Tensorflow was used as the deep learning framework in the experiment, and an image

processor was used to accelerate the training process. The hardware configuration was an I7-11700 processor, 64G memory, and Nvidia GTX3080 graphics, as shown in Table 1.

The operating environment is Ubuntu 19.10, Python3.7, and Tensorflow1.15, CUDA 10.1, cuDNN 7.6.5. Table 2 shows the information on the operating environment.

4.2. Data Acquisition and Annotation. Even though there are some datasets for agricultural insect identification that are available to the public, it can be difficult to use these datasets to complete the task of multitaxonomic insect identification. This is especially true in complex natural environments due to the wide variety of insects and the interference caused by a large number of nontarget insects. For the purpose of this investigation, insect data were collected in the central region of China using the plant protection trap light. In order to prevent the activity from having an effect on the imaging process, the far infrared heating equipment was used to kill the insects that were trapped. After positioning the insects so that they were illuminated by the same light and set against the same background, high-definition photographs of the insects were taken from directly above. There are multiple insects of a variety of species in each picture, and their placement within the frame is completely random. The algorithm's reliability is ensured by the fact that it allows insects of varying body sizes to be stacked together.

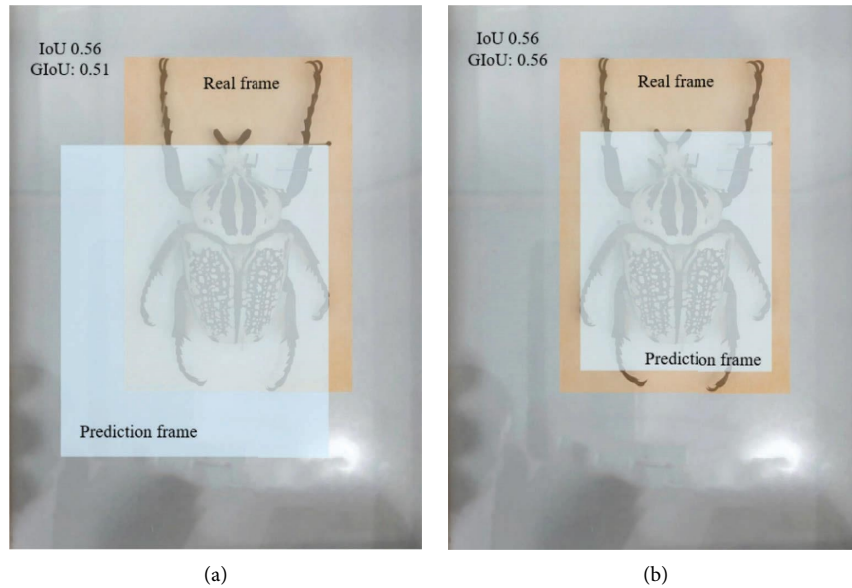


FIGURE 9: Comparison of predicted results for IoU and GIoU.

TABLE 1: Information on the experimental environment.

Graphics	GTX 3080
RAM	64G
Processor	I7-11700

TABLE 2: Information on the operating environment.

Operating system	Ubuntu 19.10
Programming language	Python3.7
Model framework	Tensorflow1.15
CUDA toolkit	CUDA 10.1
Library	cuDNN 7.6.5

Visual object tagging tool, which was developed by Microsoft and stored in Pascal VOC format, was used to label and categorize the vast majority of the insects contained within the data that was acquired with the assistance of agricultural specialists. The number of samples of certain insect species is low due to the characteristics of the biodiversity found in nature as well as the limited time available for sampling. In order to make the experiment run more smoothly, certain insects were chosen to participate in this study. Insects that were found in low numbers or caused low levels of damage were grouped together into the same classification based on their biological characteristics. This classification was then further broken down into eight distinct categories.

In this experiment, 5000 images were selected as the training set and 300 images as the test set, and Table 3 shows the number of insect samples and targets for each classification in the training set.

4.3. Training Results of Several Algorithms. In order to determine how well the algorithm works, it is compared to a number of well-known target networks. This is done so that

TABLE 3: Sample and target quantity of each classification picture.

Insect categories	Sample size	Target size
Chilo suppressalis	805	1685
Cicadellidae	621	1782
Coleoptera	1253	2234

the algorithm's effectiveness can be gauged. YOLOv3 has been reduced to its simplest form, known as Yolov3-tiny, which consists of only two output layers of varying scales. It does so at the expense of accuracy in order to increase detection speed and reduce resource occupancy, which results in improved performance in real time. In this experiment, the feature extraction layer of Faster R-CNN is represented by VGG16. Faster R-CNN is considered to be one of the more traditional two-stage object detection algorithms. The same test set was used for the evaluation of each algorithm's detection capabilities, and the same training set was used for each algorithm's training. Table 4 presents the findings of the study. The Faster R-CNN algorithm is a two-stage object detection algorithm. The number of parameters for this algorithm is as high as 130 million, and the size of the generation algorithm is 540 MB, which is significantly higher than the size of the generation algorithm for other algorithms. The average amount of time needed to detect each image is 336 milliseconds, which is a very slow rate. The parameters of YOLOv3 are approximately half of those of Faster R-CNN, which are up to 60 million, and the volume of the algorithm is also half of that of Faster R-CNN. On the other hand, the detection speed of YOLOv3 is significantly higher than that of Faster R-CNN, and the detection time of each photo is approximately 51 milliseconds, resulting in good real-time performance.

The YOLOv3-TINY algorithm has the fewest number of parameters out of all the algorithms, with only 9 million, and it also has the smallest volume. However, the loss of accuracy

TABLE 4: Effectiveness test results of different algorithms.

Models	mAP	Speeds	Number of parameters	Algorithm size
YOLOv3	70.25	51	60	241
YOLOv3-TINY	53.63	31	9	33
Faster R-CNN	69.10	336	130	540
Our method	74.15	50	9	31

is very high, with a mAP of 53.63 percent, which is the lowest of all algorithms. Although the algorithm is only 33 MB large, it has breakneck detection speed and good real-time performance. The proposed algorithm only has 9 million parameters and a volume of only 31 MB, which is slightly lower than YOLOv3-Tiny but much lower than Faster R-CNN and YOLOv3 and is approximately one-eighth the size of YOLOv3. There is no apparent advantage in speed when compared with YOLOv3, but the detection accuracy is 3.9 percent higher than that of YOLOv3, which achieves the highest result among all algorithms. This is because the larger input size used in the feature extraction layer was used.

In the comparative analysis of the test results between YOLOv3 and the algorithm presented in this study, it was observed that both algorithms exhibit similar performance in scenarios with less dispersed background interference and minimal target clutter. However, in areas characterized by high target density or overlapping regions, the proposed algorithm demonstrated significantly superior results compared to YOLOv3. This improvement may be attributed to the enhanced resolution of feature extraction facilitated by a more comprehensive network architecture, which captures a greater level of detail. In addition to this, GIoU causes the algorithm to select the prediction frame that is located closer to the center of the target. This helps to prevent prediction frames from receiving a low filtering score due to the overlap of prediction frames that occurs when the target is close.

5. Conclusion and Future Research

In order to improve the accuracy of automatic field insect identification and classification, an algorithm for automatic field insect identification and classification based on a lightweight deep learning model is proposed. The model is based on YOLOv3 target detection framework with integrated DSC and GIoU, which has low resource consumption and high level of accuracy at the same time. Experiments were conducted to train the algorithm using 5000 insect photos obtained under trap lights, and the results show that the algorithm in this article has good recognition ability and good robustness for a wide range of insects, and can solve the problems of insect stacking and background interference. Compared with other algorithms, the algorithm in this article has a higher correction rate in automatic identification and classification of insects in the field.

Although the algorithm in this article achieves good results, there is a lot of randomness in the actual environment during the sampling process, resulting in an unbalanced number between classifications. Moreover, the lack

of comprehensiveness in terms of the insect species covered in this article and the number and quality of images used for training the model needs to be continuously supplemented in the subsequent work. The prediction results of the automatic detection and classification model will be compared with the survey data from the local agricultural department in collaboration with the local plant protection department at a later stage. Support will be provided to further validate the accuracy and effectiveness of the model and to identify as well as replace traditional survey methods.

Data Availability

The labeled data set used to support the findings of this study is available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Authors' Contributions

Feng Wu wrote the manuscript and analyzed the data. Yueyin Li supervised the work and designed the study. All authors have read and agreed the final version to be published.

Acknowledgments

This study was supported by the Xinyang Agriculture and Forestry University.

References

- [1] J. Walker and C. Granjou, "MELiSSA the minimal biosphere: human life, waste and refuge in deep space," *Futures*, vol. 92, pp. 59–69, 2017.
- [2] H. C. J. Godfray, O. T. Lewis, and J. Memmott, "Studying insect diversity in the tropics," *Philosophical Transactions of the Royal Society B*, vol. 354, pp. 87–100, 2000.
- [3] S. Skendžić, M. Zovko, I. P. Živković, V. Lešić, and D. Lemić, "The impact of climate change on agricultural insect pests," *Insects*, vol. 12, no. 5, p. 440, 2021.
- [4] H. Tian, T. Wang, Y. Liu, X. Qiao, and Y. Li, "Computer vision technology in agricultural automation—a review," *Information Processing in Agriculture*, vol. 7, no. 1, pp. 1–19, 2020.
- [5] J. Liu and X. Wang, "Plant diseases and pests detection based on deep learning: a review," *Plant Methods*, vol. 17, no. 1, p. 22, 2021.
- [6] A. Fuentes, S. Yoon, S. Kim, and D. Park, "A robust deep-learning-based detector for real-time tomato plant diseases and pests recognition," *Sensors*, vol. 17, no. 9, p. 2022, 2022.
- [7] M. Martineau, D. Conte, R. Raveaux, I. Arnault, D. Munier, and G. Venturini, "A survey on image-based insect classification," *Pattern Recognition*, vol. 65, pp. 273–284, May 2017.
- [8] D. Xia, P. Chen, B. Wang, J. Zhang, and C. Xie, "Insect detection and classification based on an improved convolutional neural network," *Sensors*, vol. 18, no. 12, p. 4169, 2018.
- [9] Q. Luo, W. Xin, and X. Qiming, "Identification of pests and diseases of *Dalbergia hainanensis* based on EVI time series and classification of decision tree," in *Proceedings of the IOP*

- Conference Series: Earth and Environmental Science*, vol. 69, Article ID 012162, Chengdu, China, June 2017.
- [10] G. Pattnaik and K. Parvathi, "Automatic detection and classification of tomato pests using support vector machine based on HOG and LBP feature extraction technique," in *Advances in Intelligent Systems and Computing*, vol. 1199, pp. 49–55, Springer Science and Business Media Deutschland GmbH, Singapore, 2021.
- [11] L. Liu, W. Ouyang, X. Wang et al., "Deep learning for generic object detection: a survey," *International Journal of Computer Vision*, vol. 128, no. 2, pp. 261–318, 2020.
- [12] J. Yang, J. Man, M. Xi, X. Gao, W. Lu, and Q. Meng, "Precise measurement of position and attitude based on convolutional neural network and visual correspondence relationship," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 6, pp. 2030–2041, 2020.
- [13] M. Kulin, T. Kazaz, I. Moerman, and E. de Poorter, "End-to-End learning from spectrum data: a deep learning approach for wireless signal identification in spectrum monitoring applications," *IEEE Access*, vol. 6, Article ID 18484, 2018.
- [14] M. S. Norouzzadeh, A. Nguyen, M. Kosmala et al., "Automatically identifying, counting, and describing wild animals in camera-trap images with deep learning," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 115, no. 25, Article ID E5716, 2018.
- [15] I. Potamitis, P. Eliopoulos, and I. Rigakis, "Automated remote insect surveillance at a global scale and the Internet of things," *Robotics*, vol. 6, no. 3, p. 19, 2017.
- [16] S. Zhang, J. Liang, and S. Tang, "Optimal application timing of pest control tactics in nonautonomous pest growth model," *Abstract and Applied Analysis*, vol. 2014, pp. 1–12, 2014.
- [17] B. Wang, "Identification of crop diseases and insect pests based on deep learning," *Scientific Programming*, vol. 2022, Article ID 9179998, 10 pages, 2022.
- [18] J. Yang, C. Wang, B. Jiang, H. Song, and Q. Meng, "Visual perception enabled industry intelligence: state of the art, challenges and prospects," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 3, pp. 2204–2219, 2021.
- [19] P. Boissard, V. Martin, and S. Moisan, "A cognitive vision approach to early pest detection in greenhouse crops," *Computers and Electronics in Agriculture*, vol. 62, no. 2, pp. 81–93, 2008.
- [20] R. M. S. Abdul, Y. A. Yap, T. H. Wong, S. Mansor, and R. Mispan, "Automated identification and counting of pests in the paddy fields using image analysis," in *Proceedings of the Computers in Agriculture and Natural Resources*, p. 759, Orlando, FL, USA, July 2006.
- [21] T. Kasinathan, D. Singaraju, and S. R. Uyyala, "Insect classification and detection in field crops using modern machine learning techniques," *Information Processing in Agriculture*, vol. 8, no. 3, pp. 446–457, 2021.
- [22] Z. Juan and C. Xiao-Ping, "Field pest identification by an improved Gabor texture segmentation scheme," *New Zealand Journal of Agricultural Research*, vol. 50, no. 5, pp. 719–723, 2007.
- [23] C. Wen, D. E. Guyer, and W. Li, "Local feature-based identification and classification for orchard insects," *Bio-systems Engineering*, vol. 104, no. 3, pp. 299–307, 2009.
- [24] C. Wen, D. Guyer, and W. Li, "Automated insect classification with combined global and local features for orchard management," in *Proceedings of the 2009 Reno, Nevada*, vol. 2, p. 1, Reno, Nevada, June 2009.
- [25] C. Xie, J. Zhang, R. Li et al., "Automatic classification for field crop insects via multiple-task sparse representation and multiple-kernel learning," *Computers and Electronics in Agriculture*, vol. 119, pp. 123–132, 2015.
- [26] J. Yang, B. Jiang, and H. Song, "A distributed image-retrieval method in multi-camera system of smart city based on cloud computing," *Future Generation Computer Systems*, vol. 81, pp. 244–251, 2018.
- [27] J. Yang, Y. Zhu, B. Jiang, L. Gao, L. Xiao, and Z. Zheng, "Aircraft detection in remote sensing images based on a deep residual network and super-vector coding," *Remote Sensing Letters*, vol. 9, no. 3, pp. 228–236, 2018.
- [28] X. Cheng, Y. Zhang, Y. Chen, Y. Wu, and Y. Yue, "Pest identification via deep residual learning in complex background," *Computers and Electronics in Agriculture*, vol. 141, pp. 351–356, 2017.
- [29] Y. Zhong, J. Gao, Q. Lei, and Y. Zhou, "A vision-based counting and recognition system for flying insects in intelligent agriculture," *Sensors*, vol. 18, no. 5, p. 1489, 2018.
- [30] S. Chen and J. Chen, "Classification for fruit fly images using convolutional neural networks," *Information Technology and Control*, vol. 51, no. 2, pp. 376–389, 2022.