Research Article

Pest Identification and Counting of Yellow Plate in Field Based on Improved Mask R-CNN

Minxi Rong, Zhizheng Wang, Bin Ban, and Xiaoli Guo

College of Mathematics and Information Science, Zhengzhou University of Light Industry, Zhengzhou 450000, China

Correspondence should be addressed to Xiaoli Guo; xlguo@zzuli.edu.cn

Received 30 December 2021; Accepted 18 February 2022; Published 15 March 2022

Academic Editor: Fahad Al Basir

Copyright © 2022 Minxi Rong et al. 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 identification is the basis of insect research and disaster control and is of great importance for the design of pest control strategies and the protection of beneficial insects. Due to human subjective limitations and the small size and uneven distribution of pests, traditional methods of distinguishing and counting pest types based on experience cannot quickly and accurately detect and identify pests. Therefore, this paper proposes an object detection algorithm based on the improved Mask R-CNN model, aiming to improve the accuracy and efficiency in pest identification and counting. The algorithm improves the FPN structure in the feature extraction network and increases the weight coefficient when fusing feature layers of different scales. Based on the task of target detection and recognition, weight coefficient is adjusted to a proper parameter so that the semantic information and positioning information can be made full use to achieve more accurate recognition and positioning. The results of the experimental analysis of 1000 sample images show that the improved Mask R-CNN model has a recognition and detection accuracy of 99.4%, which is 2.7% higher than that of the unimproved Mask R-CNN model. The main contribution of this method is to improve the detection speed, and at the same time, the recognition accuracy has been significantly improved. This algorithm provides technical support for pest detection in the agricultural field and makes a contribution to the intellectualization of agricultural management.

1. Introduction

Insects are a large and important member of nature’s biological chain, and these organisms are widely distributed, occupying about 75% of the animal kingdom species [1]. The vast majority of insects feed on plants, leading to a decrease in the quality and yield of crops, and pest infestation is one of the most important factors limiting agricultural production [2]. Therefore, it is very important to study the image recognition of insects and find out the available patterns for the design of pest control strategies [3].

Traditional insect identification is based on the identification of insect species by experts or technicians with extensive knowledge of insects [4]. However, the existing experts and technicians with experience in classification are far from meeting the needs of various practical scenarios that are expanding more and more nowadays. At the same time, insect image recognition is considered to be a rather difficult image recognition problem due to the complex texture, small features, and variable environment of insects themselves. The interspecies similarity between insect groups and the differences caused by various gestures and movements further increase the difficulty of the image recognition task. The large number and variety of insect species make it easy to miss key information in the human construction of screening features [5], which leads to incorrect classification.

The development of computer hardware allows more complex operations to be processed on computers, which makes it possible to replace the human brain with computers for processing common insect image data. At present, traditional image segmentation methods are still the mainstream of insect image segmentation, such as mathematical morphology-based segmentation methods and threshold-based segmentation methods [6]. Yifeng Fan et al. proposed an image segmentation technique based on the mathematical morphology method, which used erosion and
expansion methods to eliminate reflections and noise in insect images [7]. Mele et al. proposed an insect image segmentation method based on the combination of global thresholding and the local seed region growth method. These traditional image segmentation methods only use edge, texture, color, and other features and require high image quality, which are inefficient, poorly applicable, and cannot make full use of the key information of insects for the complex field of insect classification. The complexity of insect image backgrounds in practical application scenarios also makes the traditional segmentation task more difficult, and relying only on low-level features to classify them has significant drawbacks [8].

In recent years, some machine learning models have started to be promoted and applied in practical scenarios. Insect image recognition counting methods based on machine learning [9] can calculate the results accurately and efficiently while saving a lot of manpower [10, 11]. The target detection algorithm represented by YOLO is faster in detection, but less accurate [12]. The target detection algorithm represented by Faster R-CNN is able to identify classification targets with higher accuracy [13], and slower processing speed requires a bit more hardware.

In this paper, we propose to use the Mask R-CNN model [14] for recognition, classification, and counting of insect images. The backbone feature extraction network structure automatically screens and extracts the key features from insect images, avoiding the errors caused by the human construction of screening features. The combination of the skeleton network (ResNet 101) and the improved feature pyramid network maximizes the retention of semantic and localization information and ensures the delivery of key information.

This paper aims to solve the problems of high image requirements and inaccurate classification in traditional classifiers and propose a feasible solution for insect identification and detection in the field. The background environment requirement is reduced by using yellow plates to capture insects instead of laboratory environment; the insect image quality requirement is reduced by using handheld mobile devices instead of high-precision cameras. The main contribution is to improve the accuracy of insect identification with reduced requirements, possessing practical feasibility to a large extent.

This paper is organized as follows: the first part outlines the motivation of this paper, related works, and the basic idea of this paper. The second part describes the work related to the acquisition of data and the processing methods. The third part describes the theoretical model design as well as our targeted improvements and optimization. The fourth part describes the experiments and results as well as the summary outlook.

2. Related Work

2.1. Data Collection. The data for this experiment were collected from mobile communication equipment. The camera lens has 64 million high-definition pixels, the shooting distance is 10 cm, the shooting angle is parallel to the yellow board, and the images are stored in JPG format. All images were shot in natural light with no artificial supplementary lighting, matching actual farm scenes. The 30 yellow traps in the experiment were evenly dispersed on 2 hm² of agricultural crop planting soil.

2.2. Data Preprocessing. In this study, aphids and leaf miner flies, which were caught in high numbers, were chosen as the main targets for detection and counting, and seven other insect species, such as grasshoppers, were used to assist in testing the accuracy of the classification. In particular, the original dataset was captured with a larger image size of 3456 * 4608 pixels, which is much larger than the normal deep learning training image size, and it is likely that feature loss, scaling failure, or other unknown anomalies due to overflowing GPU video memory from too large data will occur during training. To facilitate training and labeling, the raw data are cropped to 512 * 512 pixel images for processing in this paper. The effect is shown in Figure 1.

In this paper, a series of enhancement operations such as geometric transformation, mosaic enhancement, and adjusting brightness is performed on the images to expand the image data to improve model robustness and prevent overfitting of the model. The Labelme 3.6.2 [15] annotation tool was used in this experiment to annotate the data with a number of 1000 annotated images. The effect of the final data is shown in Figure 2:

2.3. Lab Environment. The experimental environment of this article is the Ubuntu20.04 operating system, and the hardware uses ASUS RTX3080ti TUF graphics card (12 GB). The software compiling environment is Python 3.6, with the CUDA 11.5 graphics computing platform. The NVIDIA-Tensorflow1.15+Kares framework was used. The experiment uses image-centric training and dynamically allocates the number of images for the GPU according to the size of the image. Each image has N samples, and the ratio of positive and negative samples is 1:3. 160K iterations were performed, the learning rate is set to 0.02, and the learning rate is set to 0.003 for the last 40K iterations. The decay weight is 0.0001 [16] and the momentum is 0.9.

3. Theoretical Model and Methodology Design

Mask R-CNN is a target instance segmentation framework proposed by He Kaiming et al., and this has been awarded as the ICCV2017 best paper. This signals that single-task type networks are no longer in the limelight and are being replaced by more complex, integrated multitask network models [17].

3.1. Mask R-CNN Network Structure. On the basis of the Faster R-CNN, Mask R-CNN has added a Mask branch; i.e., it has added instance segmentation to the original classification regression for target detection in combination with FCN for mask prediction. The framework of the Mask R-CNN network model includes a ResNet [18] based...
backbone convolutional extraction feature, FPN [19–21]
multiscale fusion of feature extraction networks [22], region
proposal networks [23], target region alignment operations
[24], and fully convolutional network FCN and fully con-
nected layer FC.

The input image enters the feature extraction network
for feature extraction by the ResNet-FPN structure and
generates a multiscale feature map. The images correspond
to the generated feature maps passed into the region, sug-
gesting the generation network to get the region of interest
(ROI). The ROI Align [25] structure uses bilinear interpo-
lation to accurately map the ROI onto the feature map and
extract the corresponding target feature blocks. The feature
blocks are fed to the fully connected layer FC and the fully
convolutional network FCN to complete the target classi-
fication regression and instance segmentation tasks [26],
respectively. The overall process is shown in Figure 3.

3.2. Counting Algorithm. The insect counting algorithm uses
image processing techniques from the cv2 and scikit li-
braries. In order to accurately identify the number of pests
counted in the input image, this study crops the image to be
predicted into pixels of the same size as the dataset. Thus, the
statistical total number is then related to the matrix derived
from the original image size (len_x, len_y), per image
information (det_count), and can be derived as equations (1)–(3):

\[ m = \left\lfloor \frac{\text{len}_x}{512} \right\rfloor \text{(round up)}, \quad (1) \]

\[ n = \left\lfloor \frac{\text{len}_y}{512} \right\rfloor \text{(round up)}, \quad (2) \]

\[ \text{num} = \sum_{i=0}^{m} \sum_{j=0}^{n} \text{det}_\text{count}. \quad (3) \]

The blurred image is first thresholded to segment the smooth edges, thereby separating the foreground from the background [27]. The minimum confidence level is used to improve the accuracy of identifying targets. The best target bounding box is filtered by eliminating redundant bounding boxes using a nonmaximum suppression algorithm. The filtered target results are presented as masks and detection boxes, and the detection boxes are traversed one by one according to the information given by the classification branch. The category score \( K \) is compared to a predetermined threshold \( N \) [28]. When \( K > N \), this target instance segmentation classification result belongs to that category and this category number statistic plus one. When \( K \leq N \), the target score is not higher than a predetermined threshold, so the target can be considered as not belonging to such a category and is not involved in the statistics. The available formula (4) is expressed as

\[ S_i = \begin{cases} S_i + 1 & K > N, \\ S_i + 0 & K \leq N. \end{cases} \quad (4) \]

The accuracy of this counting method relies on the recognition accuracy of the model [29]. The higher the accuracy of the recognition, the more accurate the returned count results. In this paper, an improved model of the ResNet101-FPN improved feature extraction network is used for learning and training, which shows significant performance improvement compared to the original Mask R-CNN network.

### 3.3. Improved FPN Algorithm

As shown in Figure 4, three routes exist for the ResNet-101-FPN structure [30]: bottom-up, horizontally connected, and top-down. Bottom-up from the picture to the feature layer is a typical forward propagation process. When the size of the feature map changes, the neural layers it passes through are grouped into a stage. Saving the final output of each stage constitutes the feature pyramid. The FPN uses a top-down process to fuse the upsampled results with a layer of the feature pyramid of the same dimension and size.

FPN effectively solves the multiscale problem. But, there is a lack of customized handling for different instance partitioning models. For example, the need for semantic information for target recognition in the large target detection problem is not going to be exactly the same as the ratio needed for small target recognition.

In response, this paper proposes an improved FPN structure that adds a weight \( w \) to the feature fusion operation for lateral connections, blending the rich semantic information at the higher levels of the feature pyramid with the accurate localization of shallow features on demand. The semantic information and localization information are enhanced to produce high-quality feature maps by fusing semantic information from different feature layers. Appropriate weight parameters are learned as needed in different instance segmentation operations to increase the accuracy of target recognition. As shown in the red boxed range in Figure 2, C5 is upsampled to obtain a tensor variable A1 of the same size and dimension as C4. A1 is combined with C4 weighted according to the learned parameters \( w \) to obtain a new feature layer P4, which can be expressed as Equation (5). P3 and P2 are derived by the same process as above. Not limited to this experiment, the improved FPN structure can be extended to other networks such as its MobileNet to form a new feature pyramid extraction network that outputs feature maps.
3.4. Loss Function Calculation. Due to the addition of the mask branch, the loss function of Mask R-CNN can be expressed as

\[ L = L_{\text{cls}} + L_{\text{box}} + L_{\text{mask}}. \]  

Among them, \( L \) is the loss function of each sample ROI, which consists of three parts: \( L_{\text{cls}} \) is the classification calculation loss; \( L_{\text{box}} \) is the bounding box position regression loss; and \( L_{\text{mask}} \) is the mask calculation loss. The bounding box regression loss function can be written as shown in the following formula:

\[ L_{\text{box}} = \frac{1}{N_{\text{reg}}} \sum_{i} P_i^t L_{\text{reg}}(t_i, t_i^*), \]  

\[ L_{\text{reg}} = \sum t_i \text{smooth}_{\ell_1}(t_i - t_i^*), \]  

\[ \text{smooth}_{\ell_1}(x) = \begin{cases} 0.5x^2 & |x| \leq 1, \\ |x| - 0.5|x| & > 1. \end{cases} \]  

In formulas (7)–(9), \( N_{\text{reg}} \) represents the number of anchor positions. When \( P_i^t \) is a positive sample, it is 1; when it is a negative sample, it is \( c \). \( t_i^* \) represents the true bounding box of the \( i \)-th anchor. \( t_i \) represents the bounding box regression parameters. The predictive loss function of the classifier can be expressed as formula (10). \( P \) represents the softmax probability distribution predicted by the classifier. \( U \) corresponds to the real target category label.

\[ L_{\text{cls}}(p, u) = -\log P_u. \]  

For the mask branch, because the influence of the competition between classes is cancelled, only one class contributes to the mask loss, that is, the loss caused by the FCN pixel instance segmentation. It can be expressed by formula (11). \( s_i \) represents the predicted probability of the mask, and \( s_i^* \) represents the true probability of the mask.

\[ L_{\text{mask}}(s_i^*, s_i) = -(s_i^* \log(s_i) + (1 - s_i^*) \log(1 - s_i)). \]  

4. Experimental Analysis

To speed up the convergence of the network iterations, this paper uses migration learning for assisted training [31, 32]. Make use of limited data, apply previously learned knowledge on new tasks, and allow models to generalize well to new environments. This approach has been shown to achieve excellent results on a range of vision tasks as well as on some tasks that rely on images as input. After freezing the backbone network, the key parameters are optimized and the head parameters are fine-tuned as a way to achieve faster model training.

When the network is trained, the loss value of the latest model of the moment is used as a direct indicator to evaluate the fit of the model every 50 cycles of running. As shown in Table 1, as the network training period increases and the learning rate is adjusted, the network loss can decrease smoothly in the first 200 cycles, proving that the network model can obtain a very effective improvement in detection accuracy after improvement.

After 200 epochs of learning training from Table 1, it can be seen that the model has achieved a fit of 99.2%. Next, the
The generalization ability of the model is tested on the test set to evaluate the model’s performance in more detail [33, 34].

Figure 5 shows the prediction effect of the generative model for unknown images. The model’s regression box accurately frames out the target without being too large or too small or shifted in position. The classification information is identified with the detection probability information on the regression box, which is the final judgment result after the class branch selects the species with the highest probability. Example split masks are displayed using different colors to distinguish different targets from each other.

After the model has identified the insects in the image, the three branches pass the results generated by each into the counting module for the final counting statistics. Following the flow of the counting algorithm in Section 1.2, we count the identified species one by one and finally output them. Figure 6 shows how the image is displayed at the terminal after counting. All the insect species that can be identified are displayed sequentially on the terminal, with the initial value set to 0. Every time an instance is traversed, the corresponding number of species is accumulated. Eventually, when the traversal is over the species, the number of insects is also counted.

This experiment compares the model’s predictions with two staff members who are skilled at counting insect populations as a reference for accurately assessing model performance, where FPS is used as the unit of measure for model run speed. The true species and the number of insects on the 1000 images were recorded as a reference for evaluation. The number of images with correct manual counts, the number of images with incorrect insect counts, and the number of deviations from the true value that appear on them are evaluated against each other for a specified time. The results from the processing of the model counting method were then evaluated against the true values. A relationship matrix was created to compare and assess the differences between the two, as shown in Table 2.

After the algorithm counts are completed, we find the images with errors for analysis and judgment. All eight errors were missed due to failure to identify the target. Four of the objects were due to insects half stuck to the yellow plate, and the remaining parts were obscured by the yellow plate, resulting in unclear features. Three of these cases were

| Training period | Network loss | Detection accuracy (%) |
|-----------------|--------------|------------------------|
| 50              | 0.143        | 79.6                   |
| 100             | 0.092        | 87.4                   |
| 150             | 0.072        | 94.3                   |
| 200             | 0.063        | 99.2                   |

Figure 5: Predict the instance segmentation of the picture.
feature omissions due to image cropping. The last case of error originated from two insects adhering and overlapping badly, resulting in incorrect identification counts. For the first and third errors, sample collection is unreasonable and due to the small probability of occurrence, the resulting error is negligible for the overall sample, and we can still predict the infestation by the number of statistics. For the second type of error, the image cutting method with overlapping regions is used to avoid feature omission due to cutting images.

We compare Mask R-CNN with other target segmentation methods such as Faster R-CNN and YOLO, respectively, and evaluate the algorithm performance in terms of mAP and FPS, respectively, as shown in Table 3. The improved Mask R-CNN model has a significant improvement in the accuracy of the model compared to other methods, with a maximum mAP of 79.6%, and the mask holds the advantage with a more accurate recognition rate.

Table 2: Statistics and comparison of counting results.

| Number of images counted correctly | Error count image |   |
|-----------------------------------|------------------|---|
| Manual                           | 947              | 53 | 67 |
| Algorithm count                  | 994              | 6  | 8  |

Table 3: Comparison of the performance of each model.

| Model      | mAP (%) | FPS |
|------------|---------|-----|
| Faster R-CNN | 76.4    | 7   |
| YOLO       | 63.4    | 45  |
| SSD500     | 76.8    | 19  |
| Mask R-CNN | 79.6    | 4   |
| This article | 80.2    | 4   |

For a technology to spread in real life, it is not only theoretical accuracy and speed that are needed to achieve the desired results. There is a greater need for specific analysis in the context of the actual situation, combining various factors in production with the model to evaluate the practicality in an integrated manner. In Table 5, the performance of each aspect of the model is analyzed in relation to the actual scenario. The target segmentation model is capable of identifying images for counting at a consistent rate and the results are reproducible; manual counting is subject to subjective human limitations and the small and uneven distribution of pest individuals, which affect the quality of statistical results. On the other hand, for unknown species of insects, manual counting requires a lot of time to learn the characteristics of the new species and to consolidate and deepen the impression from time to time; the instance segmentation model only needs to collect a certain number of samples for the machine to learn to be able to identify the species accurately forever, and the machine learning algorithm is also superior for scalability. For day-to-day operational costs, manual counting costs a lot of time and is costly in terms of market value and human resources; the counting algorithm is cost-effective as it requires only normal office equipment to be able to perform computing counts after training is completed. It is clear that, for large-scale,
systemic farming methods, counting algorithms outperform manual counting methods from a variety of perspectives.

5. Conclusion

Identifying and counting the species and numbers of pests in the field is related to the ability to take timely measures to control the growth of the pest population, which affects the economic efficiency of the field crops. A feasible method for sorting and counting insects in the field was designed to address the small size and uneven distribution of pests. The problem of inaccurate small target recognition is solved by improving the FPN structure and combining semantic information with localization information more effectively. Its test results show that our model has the advantages of being faster, more accurate, less expensive, and more scalable than traditional counting methods.

6. Future Work

The importance and broad prospects of pest control in the digital and intelligent aspects of agriculture. In future work, we will focus on refining the dataset to tag the classification and mask labels more accurately. Also, to enhance practicality, design more convenient procedures: the processing system uses everyday mobile communication devices to capture images and upload them, processes them for recognition and counting in the form of remote links or remote servers; and presents the final results to the user in a combination of graphics and text using, for example, an app.

Data Availability

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

Table 4: Comparison of the two models under the coco dataset.

| Method               | Backbone     | AP  | AP50 | AP75 | APs | APm | APR |
|----------------------|--------------|-----|------|------|-----|-----|-----|
| Mask R-CNN           | ResNet-50-FPN| 33.1| 54.4 | 34.9 | 15.4| 36.3| 49.7|
| Improved Mask R-CNN  | ResNet-50-FPN| 33.9| 55.1 | 35.4 | 16.1| 36.5| 50.5|
| Mask R-CNN           | ResNet-101-FPN| 35.4| 57.9 | 37.7 | 15.2| 38.1| 52.3|
| Improved Mask R-CNN  | ResNet-101-FPN| 36.3| 58.6 | 38.2 | 16.3| 38.8| 52.9|

Table 5: Mask R-CNN and manual counting metric scoring.

|                          | Mask R-CNN | Manual counting |
|--------------------------|------------|-----------------|
| Accuracy                 | 99.4%      | Inversely proportional to task load and working time |
| Consistency              | Standard, repeatable | Inversely proportional to task load and working time |
| Operational requirements | No special skills required | Need to be familiar with the characteristics of a variety of insects |
| Spend time               | 3FPS       | 30s–120s variable depending on the picture condition |
| Scalability              | Can be trained to recognize other types of insects or extended to other application scenarios | Not applicable |
| Cost                     | Daily office equipment and negligible electricity bills | Human resources expenses |

Conflicts of Interest

The authors declare that they have no conflicts of interest to report regarding the present study.

Acknowledgments

The authors thank the National Natural Science Foundation of China (11601491).

References

[1] X. Yan, Y. Wang, and Y. Li, “A study on the origin and evolution of insects and the ambiguity of their causes,” Journal of Yan’an University (Natural Science Edition), vol. 22, no. 01, pp. 81–84, 2003.
[2] K. Wu, “Current situation and development strategies of agricultural entomology in China,” Plant Protection, vol. 36, no. 02, pp. 1–4, 2010.
[3] X. Chen, “Progress, problems and prospects of biological pest control research in China in the 21st century,” Insect Knowledge, vol. 47, no. 04, pp. 615–625, 2010.
[4] Q. Yao, J. Lv, and B. Yang, “Research progress of automatic recognition and counting of insects based on images,” China Agricultural Science, vol. 44, no. 14, pp. 2886–2899, 2011.
[5] Z. Liu, Research on Agricultural Pest Detection Method Based on Mapping Feature Analysis, Zhejiang University, Hangzhou, China, 2017.
[6] M. Hua, “Overview of deep learning based image semantic segmentation algorithms,” China’s Strategic Emerging Industries, vol. 4, p. 120, 2018.
[7] K. Mele, “Insect soup challenge: segmentation, counting, and simple classification,” in Proceedings of the 2013 IEEE International Conference on ComputerVision Workshops, pp. 168–171, IEEE, Sydney, NSW, Australia, December, 2013.
[8] H.-W. Pang, P. Yang, X. Chen, Y. Wang, and C.-L. Liu, “Insect recognition under natural scenes using R-FCN with anchor boxes estimation,” in Image and Graphics, pp. 689–701, Springer, Berlin, Germany, 2019.
[9] Z. Lv, Research on the Detection Method of Agricultural Pests Based on the Analysis of Atlas Characteristics, Zhejiang University, Hangzhou, China, 2017.

[10] C. Wang and L. Bu, “A survey of target detection algorithms based on convolutional neural networks,” Ship Electronic Engineering, vol. 41, no. 09, pp. 161–169, 2021.

[11] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” Communications of the ACM, vol. 60, no. 6, pp. 84–90, 2017.

[12] X. Hu, Y. Liu, and Z. Zhao, “Real-time detection of uneaten feed pellets in underwater images for aquaculture using an improved YOLO-V4 network,” Computers and Electronics in Agriculture, vol. 185, 2021.

[13] Y. Xing, “Research on insect small target detection based on faster-RCNN,” Journal of Kashgar University, vol. 42, no. 03, pp. 65–69, 2021.

[14] K. He, G. Gkioxari, P. Dollar, and R. Girshick, “Mask R-CNN,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 42, no. 02, pp. 386–397, 2020.

[15] B. C. Russell, A. Torralba, and K. P. Murphy, “LabelMe: a database and web-based tool for image annotation,” International Journal of Computer Vision, vol. 77, no. 1/2/3, pp. 157–173, 2008.

[16] S. Theodoridis, Stochastic Gradient descent, pp. 161–231, Elsevier, Amsterdam, 2015.

[17] Y. Li and L. Zhang, “Research on panorama segmentation method based on multi-task learning,” Journal of Testing Technology, vol. 34, no. 6, pp. 514–519, 2020.

[18] K. M. He, G. Gkioxari, and P. Dillar, “Mask R-CNN,” in Proceedings of the IEEE International Conference on Computer Vision, pp. 2961–2969, IEEE, Venice, Italy, October, 2017.

[19] A. Qin and Y. Zhang, “YOLOV3 traffic sign recognition and detection based on FPN improvement,” Scientific Journal of Intelligent Systems Research, vol. 2, no. 11, 2020.

[20] B. Chen, X. Ju, and Y. Gao, “A quaternion two-stream R-CNN network for pixel-level color image splicing localization,” Chinese Journal of Electronics, vol. 30, no. 06, pp. 1069–1079, 2021.

[21] R. Niu, Z. Zhao, and S. Zhang, “Optical remote sensing image ship detection based on FPN feature extraction,” Radio Engineering, vol. 51, no. 11, pp. 1296–1302, 2021.

[22] D. Bolya, C. Zhou, and F. Xiao, “YOLACT++ better real-time instance segmentation,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 99, p. 1, 2020.

[23] C. Yu and G. Zhang, “Research on traffic sign detection method based on faster R-CNN,” Information recording material, vol. 22, no. 10, pp. 72-73, 2021.

[24] K. Chen, Z. Zeng, and J. Yang, “A deep region-based pyramid neural network for automatic detection and multi-classification of various surface defects of aluminum alloys,” Journal of Building Engineering, vol. 43, 2021.

[25] H. A. O. zhenbang, “Automated tree-crown and height detection in a young forest plantation using mask region-based convolutional neural network (Mask R-CNN),” ISPRS Journal of Photogrammetry and Remote Sensing, vol. 178, pp. 112–123, 2021.

[26] S. Evan, L. Jonathan, and D. Trevor, “Fully convolutional networks for semantic segmentation,” IEEE transactions on pattern analysis and machine intelligence, vol. 39, no. 4, pp. 640–651, 2017.

[27] L. Zhu, M. Ma, and Z. Zhang, “Research on automatic segmentation method of Lepidopteran target image front background based on fully convolutional network,” Acta Entomologica Sinica, vol. 61, no. 2, pp. 255–262, 2018.

[28] L. D. Rong, “A deep learning approach to the screening of malaria infection: automated and rapid cell counting, object detection and instance segmentation using mask R-CNN,” Computerized Medical Imaging and Graphics, vol. 88, no. prepublish, Article ID 101845, 2021.

[29] H. Sukju, “Automatic pest counting from pheromone trap images using deep learning object detectors for matsucoccus thunbergianae monitoring,” Insects, vol. 12, no. 4, p. 342, 2021.

[30] D. Xiao, Y. Huang, and Y. Zhang, “Detection algorithm of field yellow board pests based on improved Faster R-CNN,” Journal of Agricultural Machinery, vol. 52, no. 06, pp. 242–251, 2021.

[31] H. Yang, X. Xiao, and Q. Huang, “Rice pest identification based on convolutional neural network and transfer learning,” Progress in Laser and Optoelectronics, vol. 13, 2021.

[32] K. Weiss, T. M. Khoshgoftaar, and D. Wang, “A survey of transfer learning,” Journal of Big Data, vol. 3, no. 1, pp. 1–40, 2016.

[33] J. Revaud, J. Almazan, and R. Rezende, “Learning with average precision: training image retrieval with a listwise loss,” in Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), pp. 5106–5115, IEEE, Seoul, Korea (South), November, 2019.

[34] A. J. Kapoor and H. S. Fan, “Intelligent detection using convolutional neural network,” Earth and Environmental Science, vol. 234, Article ID 12061, 2019.