Hindawi
Journal of Sensors
Volume 2023, Article ID 4039179, 15 pages
https://doi.org/10.1155/2023/4039179

Research Article

Rapid and Accurate Identification of Grass Seedlings in Agricultural Fields Based on Optimized YOLOX Model

Jie Kang,[1] **Yi Gu**,[2] **Zhi Yuan Wang**,[3] **and Xing Yu Lu**[1]

[1]*College of Mechanical & Electrical Engineering, Sanjiang University, Nanjing, China*
[2]*School of Mechanical Engineering, Jiangnan University, Wuxi, China*
[3]*College of Engineering, Nanjing Agricultural University, Nanjing, China*

Correspondence should be addressed to Jie Kang; kang_jie@sju.edu.cn

Received 11 February 2023; Revised 12 October 2023; Accepted 16 October 2023; Published 6 November 2023

Academic Editor: Sana Ullah Jan

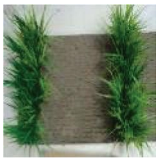
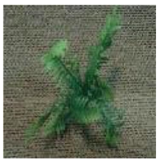
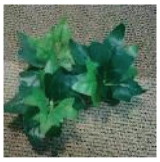
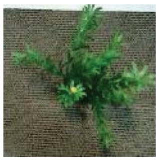
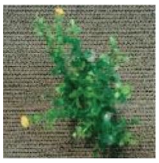
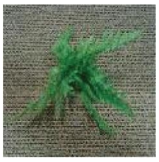
Traditional agricultural cultivation is labor-intensive and vulnerable to natural climate conditions, such as heavy rainfall and drought. Concerns over food safety have also brought attention to the growth of weeds and the misuse of agricultural chemicals, which can have a serious negative impact on crop growth and safety. We investigated the feasibility of the YoloX model in the field of agricultural weed identification to address the problem of weed handling in growing crops. In order to overcome the effects of climate and environment, we chose a purchased weed model for our study. We used a binocular vision camera for image acquisition and created a database containing 6,000 samples and enhanced the original database of 1,000 samples with data. In order to address the complex background of the weed images, the changing lighting environment of the binocular camera-acquired images, and the noise interference, we performed histogram equalization, image denoising, and background processing on the dataset. These processing measures aim to improve the overall learning efficiency of the model in order to improve the accuracy of deep learning on weeds. Target detection platforms based on the TensorFlow and PyTorch frameworks were established, respectively, and the mainstream target detection models Faster R-CNN and YoloX series target detection models were trained with the same dataset for comparative analysis using the longitudinal comparison method. The results show that the training under the PyTorch framework yields better models than the training under the TensorFlow framework. YoloX-x has higher recognition accuracy, faster recognition speed, and more stable compared to the Faster R-CNN model, with an average recognition rate of 97.07% and an average recognition time of 0.062 s. Moreover, the optimization of YoloX by incorporating an attention-focusing mechanism resulted in an improved accuracy rate with a decrease in recognition time, with an average recognition rate of 97.70% and an average recognition time of 0.029 s.

1. Introduction

In the 21st century, with the rapid development of science and technology, human beings have entered the 5G era, and various intelligent devices have emerged one after another, making our life more convenient. Traditional agricultural cultivation is greatly affected by weed growth, which affects the growth and yield of crops. The widespread use of agricultural chemicals has had a great negative impact on the natural environment and human health and has also caused quality and food safety problems. It takes a lot of human and material resources to solve such problems and is greatly affected by the natural climate, such as heavy rainfall, drought, etc. [1]. The problem of food quality and food safety has been caused by the negative impact of product quality and food safety. Therefore, in the field of agriculture, the use of precision control variables and standardized intelligent agricultural operations not only to achieve the use of herbicides to accurately spray weeds to complete the task of weed control but also for the growth characteristics of different crops to accurately control the amount of watering and fertilization, so as to reduce the use of agricultural chemicals, environmental problems and food safety problems, and the cost of agricultural operations. It has become a popular research topic [2].

Subeesh A pointed out [3] that traditional weed management is not very efficient and has a negative impact on the

Table 1: Illustration of seedling and weed varieties.

YM	JC	HC	MCX	JJC	BNH
					

integration of smart agricultural machinery. Therefore, adopting automatic weed detection has a positive role in solving weed problems and increasing crop yield. Consequently, a computer vision-based intelligent targeted spraying system was adopted to study weed recognition in bell pepper fields based on deep learning using RGB images. By comparing the feasibility of Alexnet, GoogLeNet, Inception V3, and Xception networks and continuously adjusting parameters for optimal performance, it was ultimately found that Inception V3 demonstrated superior performance in 30 sets of datasets and 16 subdatasets, with accuracy rates of 97.7%, 98.5%, and a recall rate of 97.8%. Xiaojun et al. [4] proposed a detection method that identifies crops and defines all other green plants as weeds. In comparison to YOLOv3, CenterNet, and Faster R-CNN for vegetable weed detection, Faster R-CNN had a significantly longer computation time than YOLOv3 and CenterNet. YOLOv3 and CenterNet had similar computation times, with YOLOv3 achieving an accuracy of 97.1% and a recall rate of 97.0%. Among these three models, YOLOv3 and CenterNet exhibited high accuracy and computational efficiency [4].

In the study of companion weeds in cornfields, Wang et al. [5] proposed a fine-grained weed recognition method using Bilinear CNN. Comparing nine general image classification models such as AlexNet and VGG-16, it was found that using VGGNet-19 and ResNet-50 as backbone networks to extract weed features and applying transfer learning to train the model on the dataset resulted in higher recognition rates and speeds. The final results indicated that network models incorporating high-order information had a higher accuracy of 98.5% compared to single models, and the use of Bilinear CNN and lightweight feature extraction networks effectively improved model recognition speed [5]. To address weed problems during the growth of cotton seedlings in Xinjiang, Yan et al. [6] used Xinjiang cotton and weeds as experimental subjects. They established a Faster R-CNN recognition model based on VGG16 training and analyzed the reasons for the low weed recognition rate. The conclusion was that the overlapping and varying density of weeds and cotton growth positions in cotton fields led to a low recognition rate. Through comparative analysis, VGG16 was determined to be the best feature extraction network, resulting in a model with high recognition accuracy and robustness, with an average recognition rate of 91.49% and an average recognition time of 0.262 s [6]. In the study of weed leaf age recognition in farmland by Quan et al. [7], they proposed the use of Mask R-CNN to obtain plant leaf age. By comparing different weather conditions (sunny, cloudy, and after rain) and shooting angles (30° and 45°) for slant and top views, the results showed that using the NMS algorithm and ResNet-101 in the Mask R-CNN model, under cloudy conditions and at a 30° viewing angle, achieved the highest recognition rate of 91.50% with an average processing time of 0.5683 s [7].

YOLOX and Faster R-CNN are both outstanding algorithms in the field of object detection, each with its own characteristics and advantages. By comparing their performances, we can better understand their differences, providing references and foundations for choosing the most suitable algorithm. The YOLOX algorithm is known for its high real-time performance, accuracy, lightweight nature, excellent scalability, and wide applicability. It has become one of the highly regarded algorithms in the object detection domain. On the other hand, the Faster R-CNN algorithm is a deep learning-based object detection framework that employs a two-stage detection method. It generates candidate boxes using the region proposal network (RPN) and performs detection using a classification network. Its performance on multiple datasets is excellent, making it one of the standard algorithms in the field of object detection.

As computer vision and deep learning technologies continue to evolve, attention mechanisms have gained widespread attention as a core component of object detection algorithms. Late, attention mechanisms have also been extensively studied and applied in the field of image processing. Therefore, applying attention mechanisms to the YOLOX algorithm can play a crucial role in further enhancing algorithm performance and application effectiveness.

(1) The main objective of this project is to conduct research on image recognition and processing for a mobile parallel multifunctional agricultural robot. The project focuses on rice field environments and involves identifying two categories of objects: the first category consists of weeds such as seedlings (YM), rotala indica (JJC), fernwort (JC), fat hen (HC), purslane (MCX), and flixweed (BNH). Seedlings and weed species are illustrated in Table 1; the second category encompasses crops, with young seedlings being the primary focus. The project involves tasks such as image capture, image processing, and utilizing neural networks to achieve the recognition of weeds and crops in the images. The main contents of the project are as follows: image collection using a stereo camera to create a dataset and image annotation using labeling.

(2) Due to challenges in sample data collection and significant environmental constraints, offline data augmentation will be performed on the collected dataset. This augmentation includes noise reduction, image transformations, contrast and brightness adjustments, etc. Subsequently, image preprocessing will be conducted, mainly focusing on noise elimination, highlighting details, and background processing.

(3) Establishing Faster R-CNN and YOLOX object detection platforms based on TensorFlow and PyTorch neural network frameworks. The preprocessed weed dataset will be used for learning and training. The Faster R-CNN and YOLOX object detection platforms will be trained, and a comparison will be made between their recognition accuracy and detection times. This step aims to identify the most suitable object detection model based on the project's dataset.

(4) Establishing an optimized YOLOX object detection platform with the inclusion of the CBAM mixed attention mechanism based on the PyTorch neural network framework. This optimized platform will be compared against the object detection model derived from step (3), analyzing the advancements brought by the enhanced YOLOX model.

2. Related Work

2.1. Image Acquisition. Image acquisition camera is the core part of the image source. The important thing for image acquisition is the clarity of the acquisition of detailed features, so the choice of 1.3 million pixels, the choice of USB communication, convenient and rapid, easy to communicate, the final use of Huibo VisionJet technology company's binocular camera HBV-1714-2 S2.0, 1.3 million pixels, manual zoom.

2.2. Dataset Creation. The dataset includes three major parts: training set, validation set, and test set, and the establishment of the dataset is one of the most core parts in the establishment of deep learning models. In order to make the deep learning model achieve more accurate recognition effect, it needs to provide rich dataset for sample training [8]. Based on the main direction of this research is weed recognition, and the research time of the project is winter, it is extremely difficult to obtain natural weeds, so we use the simulation plant model of online purchase weeds, including seedlings, ashwagandha, sowing artemisia, fern, horsetail, knapweed, with corrugated cardboard, simulate the real weed growth environment, in order to reduce the similarity of the image, use the random method of weed location's, number, and species randomly placed. The original size of the image was 720 × 960, and the collected samples were cropped and transformed to the final sample size of 800 mm × 400 mm, as shown in Figure 1, and a total of 1,000 samples were collected to make the dataset.

2.3. Data Enhancement. In order to increase the training samples of the images, improve the feature information of the dataset, and increase the image features for the subsequent deep learning, this project adopts data enhancement techniques, according to the actual situation encountered during sample acquisition, such as shooting angle, light intensity, distance between the camera and the target object, etc., by adding noise processing to the images, mainly adding Gaussian noise and pretzel noise [9]. The image is transformed, including horizontal, vertical, and equal scale flip; the brightness and contrast map of the image are adjusted. After image data enhancement, the original 1,000 datasets can be expanded to 6,000.

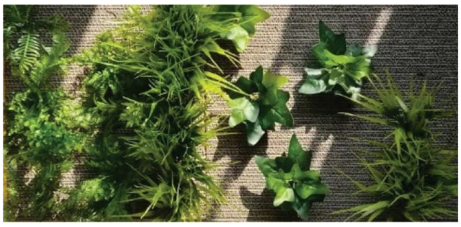


Figure 1: Sample images.

2.4. Image Preprocessing. To improve the accuracy and reduce the learning time during deep learning, three methods are cited for preprocessing. First, image denoising is performed to eliminate Gaussian noise as well as pretzel noise in the image by median filtering [10]. Second, the image contrast is improved by histogram equalization to highlight more feature information of the target object. Finally, because of the complex background in the image, only weed and crop features are retained in the image as much as possible by preserving the green color gamut in HSV space and black masking the rest of the color gamut. Finally, by utilizing the "shutil" module in Python, a dataset partitioning program was developed. Through a random combination approach, the dataset was divided into training, validation, and testing sets with a ratio of 7 : 2 : 1.

2.5. Convolutional Neural Network. Convolutional neural network is a neural network structure containing convolutional layer, pooling layer, and fully connected layer, which plays a crucial role in the field of computer vision [11]. Weed images have rich feature information, such as shape features, texture features, color features, and location space features [12], through the learning and training process of the convolutional neural network, the network is able to identify and classify different weed images, so as to achieve good classification performance. Among them, the convolutional layer uses a filter to perform convolutional operations on the image, which can extract high-level feature information from the image; the pooling layer downsamples the convolutional features to reduce the size and complexity of the data and increase the robustness of the features; and the fully-connected layer maps the pooled features to the final classification results. In convolutional neural network, each convolutional layer, pooling layer, and fully connected layer consists of multiple neurons, and during the learning process, the network updates the network parameters by back propagation algorithm to get more accurate classification results [13]. The code of this

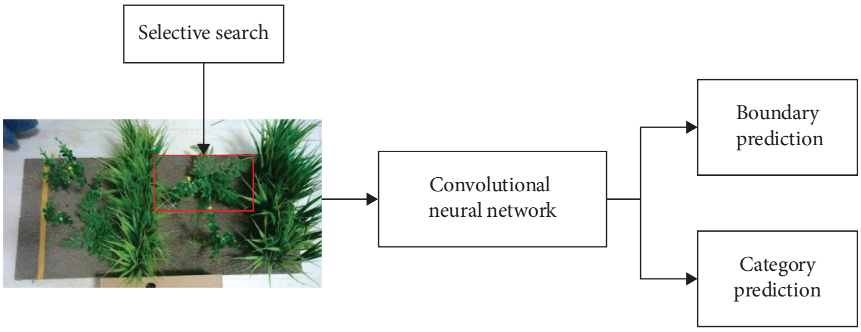


Figure 2: Structure of R-CNN model.

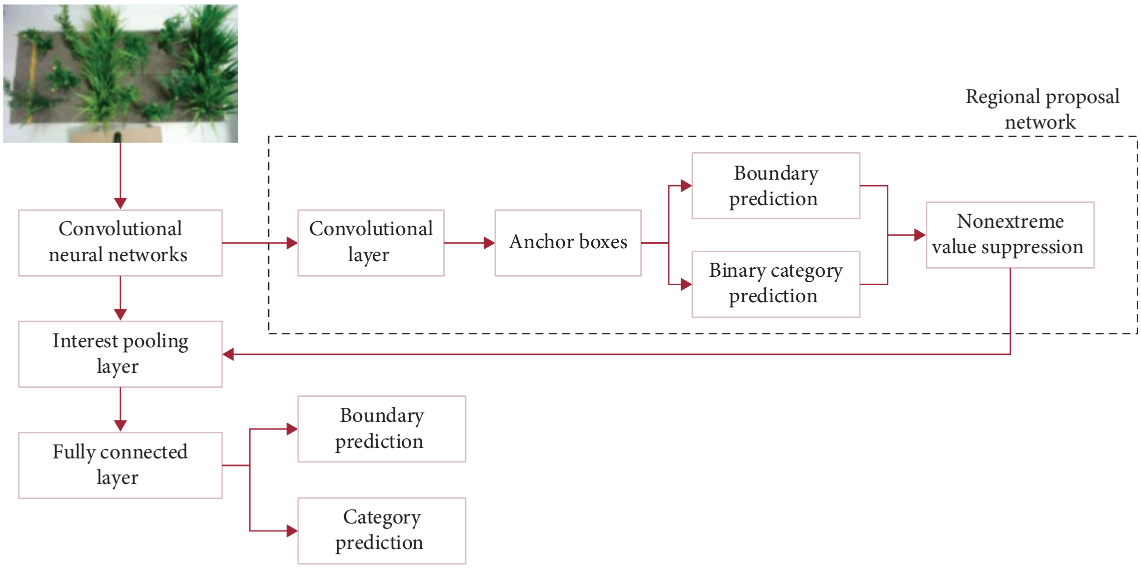


Figure 3: Structure of Faster R-CNN model.

project is developed under Win10 system environment, GPU is MX150, CPU is intel i5 8th Gen, Pycharm integrated development environment, utilizing OpenCV, a computer vision library based on Python language.

2.5.1. Introduction of Faster R-CNN Algorithm. Region-based convolutional neural networks (R-CNNs) are among the pioneers that apply deep learning to object detection, and their family has been continuously refined to include Fast R-CNN, Faster R-CNN, and Mask R-CNN [14]. This project focuses on Faster R-CNN for object detection. R-CNN serves as the foundation for many object detection algorithms, and its model is shown in Figure 2. R-CNN selects multiple proposed regions, resulting in several forward computations of the convolutional neural network, which makes the model computationally expensive and slow.

Faster R-CNN is optimized based on R-CNN and Fast R-CNN. As shown in Figure 3, the model uses the entire image for convolution. Faster R-CNN is based on two main components: the RPN and the classifier. RPN generates candidate regions for the target objects, while the classifier classifies and locates these regions [15].

RPN calculates feature maps for the entire image through convolutional neural network convolution and generates candidate regions using sliding windows. For each candidate region, RPN calculates its degree of matching with the real region and returns the most matched candidate as input for the next step. RPN can quickly generate a large number of candidate regions using convolutional neural networks and adapt to different sizes and aspect ratios of target objects using sliding windows. The classifier is designed based on R-CNN. It takes the candidate regions generated by RPN as input and performs convolutional calculations for each region to obtain feature vectors. Then, the classifier classifies these feature vectors to determine whether each region contains a target object and locates it. The classifier adopts a multilayer convolutional neural network structure that can perform fine classification and localization for each candidate region, effectively reducing model computation and time [16].

Faster R-CNN achieves efficient and accurate object detection by introducing RPN and optimizing the design of R-CNN. Its emergence provides strong support for the development of deep learning in the field of object detection. It is therefore presented and used as a YoloX comparison model.

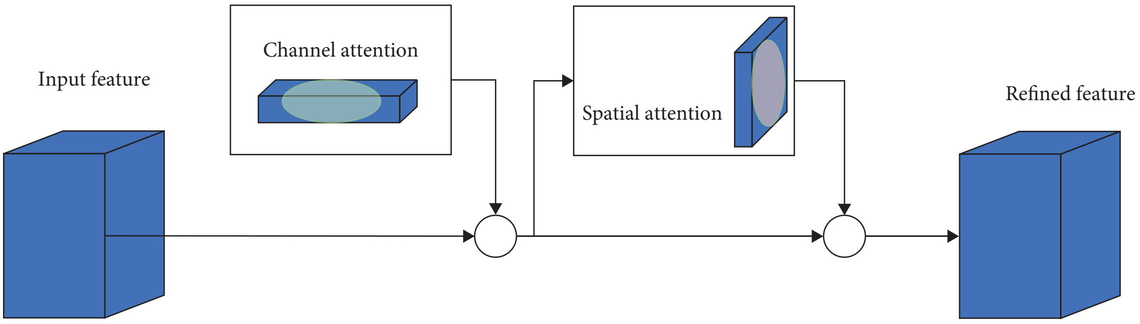


Figure 4: Structure of CBAM attention mechanism.

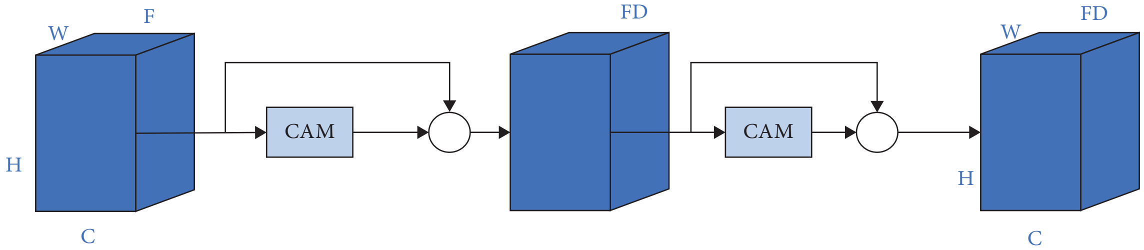


Figure 5: Structure of channel attention mechanism flowchart.

2.5.2. Introduction to the YoloX Algorithm. YoloX is the latest algorithm of the Yolo series (You Only Look Once) proposed by Kuang at the end of 2021, with YoloX-s, YoloX-l, YoloX-m, YoloX-x, etc. YoloX is based on YoloV3 and Darknet 53 and continues to use the Focus backbone in YoloV5 and the mosaic data enhancement in YoloV4. The network structure of YoloV5 and the mosaic data enhancement of YoloV4, the innovative proposal of using multiround classification regression layer and anchor-free anchorless mechanism, reducing the complexity of the detection head and the number of predictions per sample [17]. Compared with the previous Yolo series, Yolo used to implement classification and regression in a 1×1 convolutional kernel, but in YoloX, it is implemented separately first, and then integrated in the final prediction stage. The use of SimOTA advanced label assignment makes the computation faster and reduces additional hyperparameters [18].

2.5.3. CBAM Hybrid Attention Mechanism. The convolutional block attention module (CBAM) hybrid attention mechanism is a deep learning model structure used for computer vision tasks. As shown in Figure 4, it consists of two attention modules: the channel attention module and the spatial attention module. The channel attention module is employed to weight the feature maps of each channel, enhancing the importance of useful information while reducing the weight of irrelevant information. The spatial attention module, on the other hand, focuses on weighting each spatial position of the feature map, giving higher importance to relevant spatial positions and diminishing the importance of irrelevant ones. These two attention modules can be combined to form the CBAM hybrid attention mechanism, further improving model performance and enhancing weed species recognition. We will now elaborate on these two types of attention mechanisms.

The channel attention mechanism directs the model's focus onto specific channels within the feature map, thereby compressing spatial information. The specific steps involve: applying both max-pooling (MaxPool) and average-pooling (AvgPool) to the input feature map, resulting in two $1 \times 1 \times C$ vectors (C being the number of channels). Subsequently, these vectors are fed into fully connected layers (shared multilayer perceptrons, MLP) with shared parameters to produce feature outputs. The output features are then element-wise summed, activated through a sigmoid function, and multiplied element-wise with the original input feature map, yielding the channel-attentive weighted feature map needed for the spatial attention module. The process is illustrated in Figure 5, and the formula is represented by Equation (1), where *W* denotes the weights of the shared MLP.

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

The spatial attention mechanism directs the model's focus onto the spatial plane of the feature map, identifying the spatial regions within the recognition plane that require attention, thereby reducing the channel dimensionality. The steps involve performing global average pooling and global max pooling separately across all channels at each pixel position of the input feature map, resulting in two $H \times W$ matrices. Following this, a convolution operation with a 7×7 kernel is

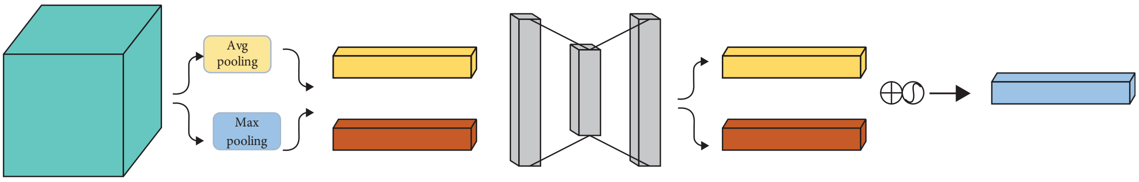


Figure 6: Structure of spatial attention mechanism flowchart.

performed to increase receptive field size, followed by a sigmoid activation function. Finally, the spatial attention weights are multiplied element-wise with the channel-attentive weighted feature map, resulting in the spatial-attentive weighted feature map. The process is shown in Figure 6, and the computational formula is Equation (2). First, the input feature matrix F' is subjected to average pooling and maximum pooling operations to obtain the corresponding feature matrices F'avg and F'max. Then, these two feature matrices are convolved with a 7 × 7 convolution kernel $f^{7\times7}$. Finally, the convolution results are processed by a sigmoid function to obtain the final output matrix M_s.

$$\begin{aligned} M_s(F') &= \mathrm{sigmoid}\left(f^{7\times7}([\mathrm{AvgPool}(F'); \mathrm{MaxPool}\,(F')])\right) \\ &= \mathrm{sigmoid}\left(f^{7\times7}\left(\left[F^s_{\mathrm{avg}\sim}; F^s_{\mathrm{max}\sim}\right]\right)\right). \end{aligned} \tag{2}$$

In summary, the spatial and channel attention mechanisms focus on the spatial positions and channel relationships of the input feature map, respectively. In this study, we introduce these mechanisms into the YOLOX object detection architecture to enhance the model's recognition performance.

2.5.4. YOLOX Object Detection Architecture with Attention Mechanism Module. Based on the aforementioned research analysis, we conclude that the CBAM hybrid attention mechanism significantly aids in feature extraction for the YOLOX recognition model. In this section, we will detail how the CBAM module is utilized to provide weed-specific feature extraction capabilities and how it is integrated into the YOLOX detection architecture. This integration aims to achieve a comparative analysis of recognition rates between the improved YOLOX algorithm with CBAM and the original YOLOX algorithm on the same dataset.

To achieve this, we introduce a class label vector into the input layer of both the CBAM channel attention module and the spatial attention module. This enables the selective application of attention mechanisms based on the class label vector. For instance, the attention mechanism is applied when the corresponding class of the feature map is related to weed seedlings, while it is suppressed for nonweed seedling classes. This approach is implemented within the Forward propagation function of the CBAM module. The Forward function defines how inputs for both healthy and unhealthy tomato conditions flow through the CBAM module, involving a series of calculations and operations to yield the output with weighted features.

To enhance the accuracy of weed category detection, we leverage the CBAM module to boost the model's ability to extract features related to weed categories. We then incorporate the CBAM module into the YOLOX architecture after the CspLayer within its CSP module. In this study, the YOLOX structure with the embedded CBAM module remains unchanged. The improved YOLOX network structure, with the CBAM module added, is depicted in Figure 7. It allows for easy adjustment of parameters for both the channel attention submodule and the spatial attention submodule within the code.

3. Methods

The evaluation metrics for target detection mainly include precision, recall, average precision, all-category average precision, F1 score, and logarithmic average miss rate [19]. By testing the Faster R-CNN target detection model and YoloX-x's target detection model under TensorFlow and PyTorch network framework, AP, mean average precision (mAP), time, F1 score, and related test data are derived to further analyze the models. Subsequently, the YoloX target detection model optimized by adding CBAM hybrid attention mechanism based on PyTorch neural network framework was established and tested, resulting in AP, mAP, time, F1 scores, and related test data for comparison and analysis with the unoptimized model4.

4. Experiment

4.1. Testing the Faster R-CNN Model. The Faster R-CNN object detection algorithm was trained and tested using both the TensorFlow and PyTorch frameworks. In the initial training round, weights pretrained on the voc07 + 12 dataset from the Pascal competition were utilized. The training parameters for this round are presented in Table 2, while the Faster R-CNN model's specific parameters are outlined in Table 3. Training was conducted for 100 epochs using the TensorFlow framework, resulting in 100 sets of weight files. The model was further fine-tuned with the weights from the 100th training epoch, achieving a loss function value of 0.621. Similarly, in the PyTorch framework, training was carried out for 100 epochs, yielding 100 sets of weight files, and the final fine-tuning was performed using the weights from the 100th epoch, resulting in a loss function value of 0.700. Following the completion of training, the model was subjected to prediction testing, which encompassed both image and video testing.

The trained Faster R-CNN model underwent image testing, where a single input image was passed through the

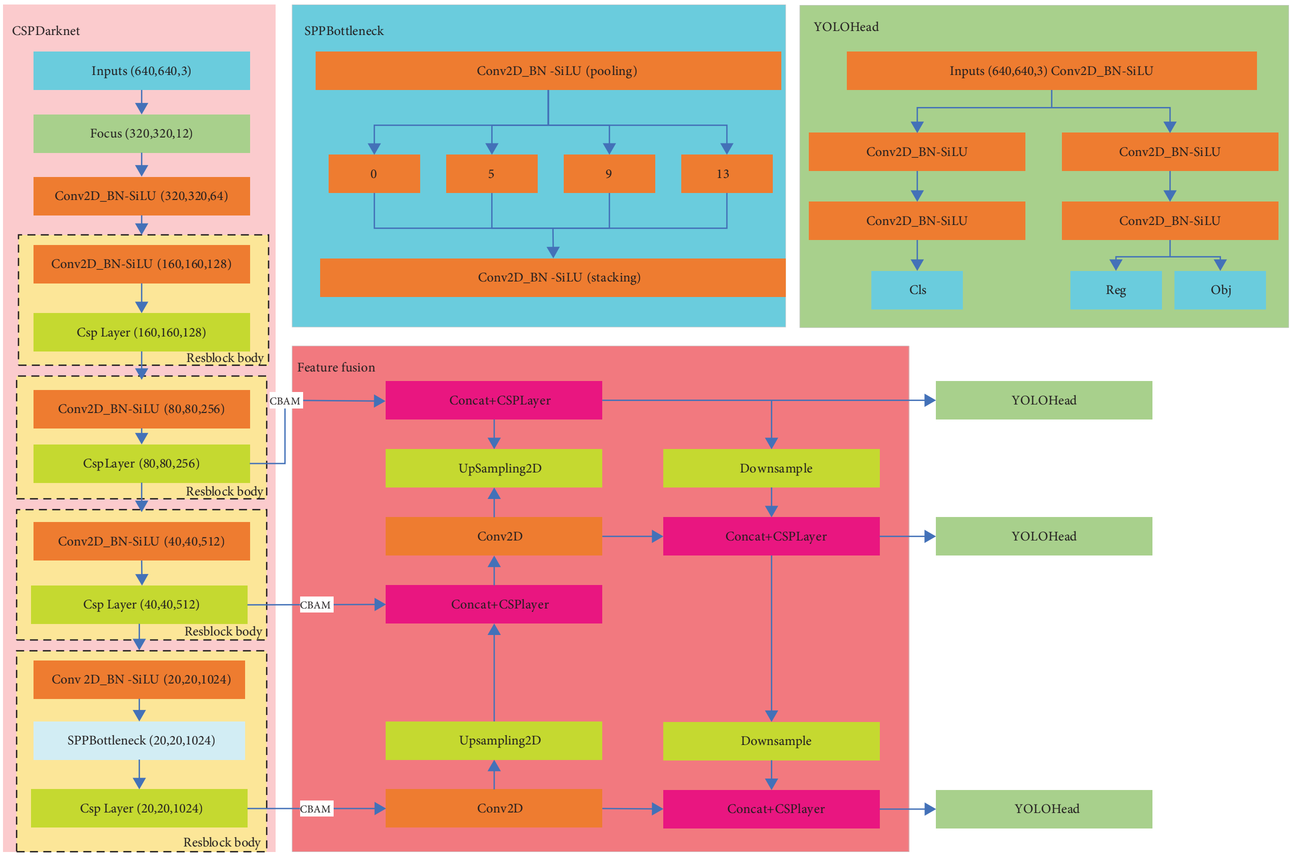


Figure 7: Improved YOLOX detection architecture.

Table 2: Training parameter settings.

Parameter name	Set value
input_shape	416 × 416
Backbone	Resnet50
Anchors_size	64, 256, 512
Freeze_Epoch	50
Freeze_batch_size	2
UnFreeze_Epoch	100
Unfreeze_batch_size	1
num_workers	1

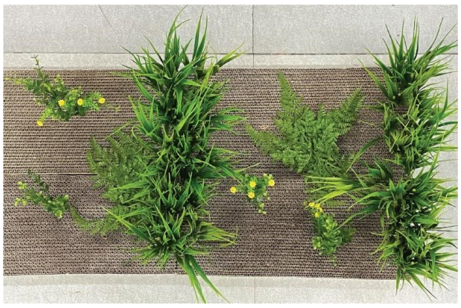


Figure 8: Test sample image.

Table 3: Faster R-CNN model parameter settings.

Parameter name	Set value
Trunk extraction network backbone	resnet50
Confidence	0.5
Nonextreme suppression value nms_iou	0.8
A priori box anchors_size	128, 256, 512

Faster R-CNN model for object detection. Bounding boxes were drawn around detected objects using different colors, and the names of the detected objects were annotated on each bounding box. The final output consists of the image with the annotated bounding boxes and a text (txt) file containing the coordinates of the detected objects. Please refer to Figures 8–10 for visual representations. For video testing, the dataset was compiled into a single video to simulate real-world detection scenarios. This video was input to the YOLOX model for object detection, with a set frames per second (FPS) of 25.0. The ultimate output is a video containing bounding boxes around detected objects. The visual representation of the results can be observed in Figure 10.

4.2. Testing YOLOX Model and Optimized YOLOX Model. In this section, we primarily test the YOLOX algorithm in different sizes, namely YOLOX-s, YOLOX-l, YOLOX-m, and YOLOX-x, using both the TensorFlow and PyTorch frameworks. During the training phase, the parameters are set according to Table 4, while the YOLOX model's specific

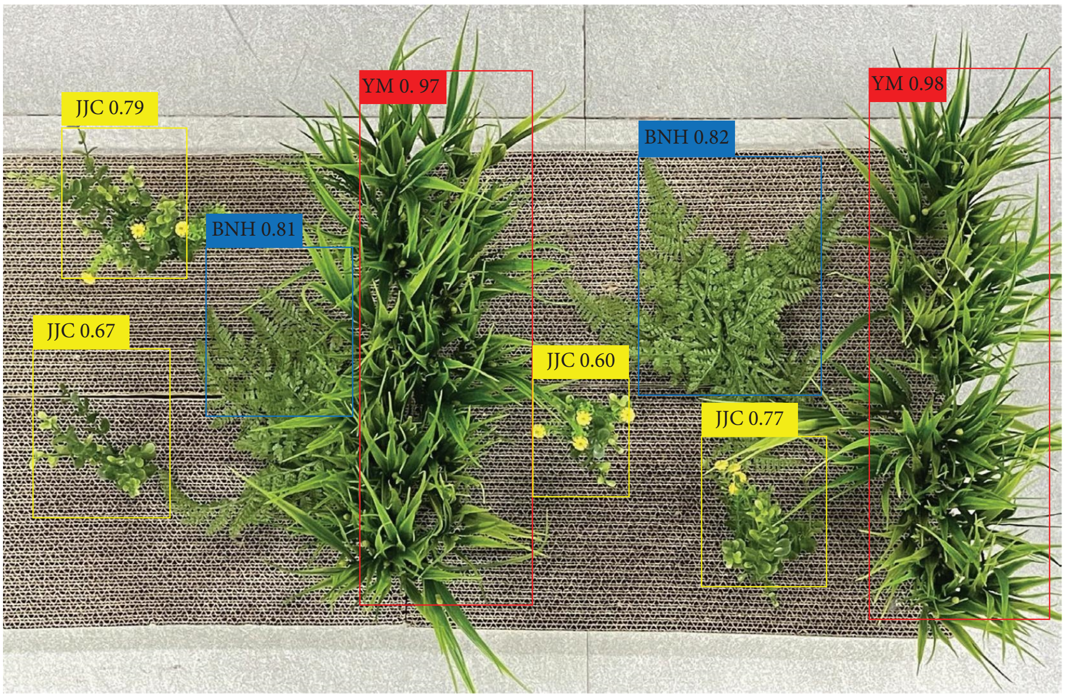


Figure 9: Predicted results.

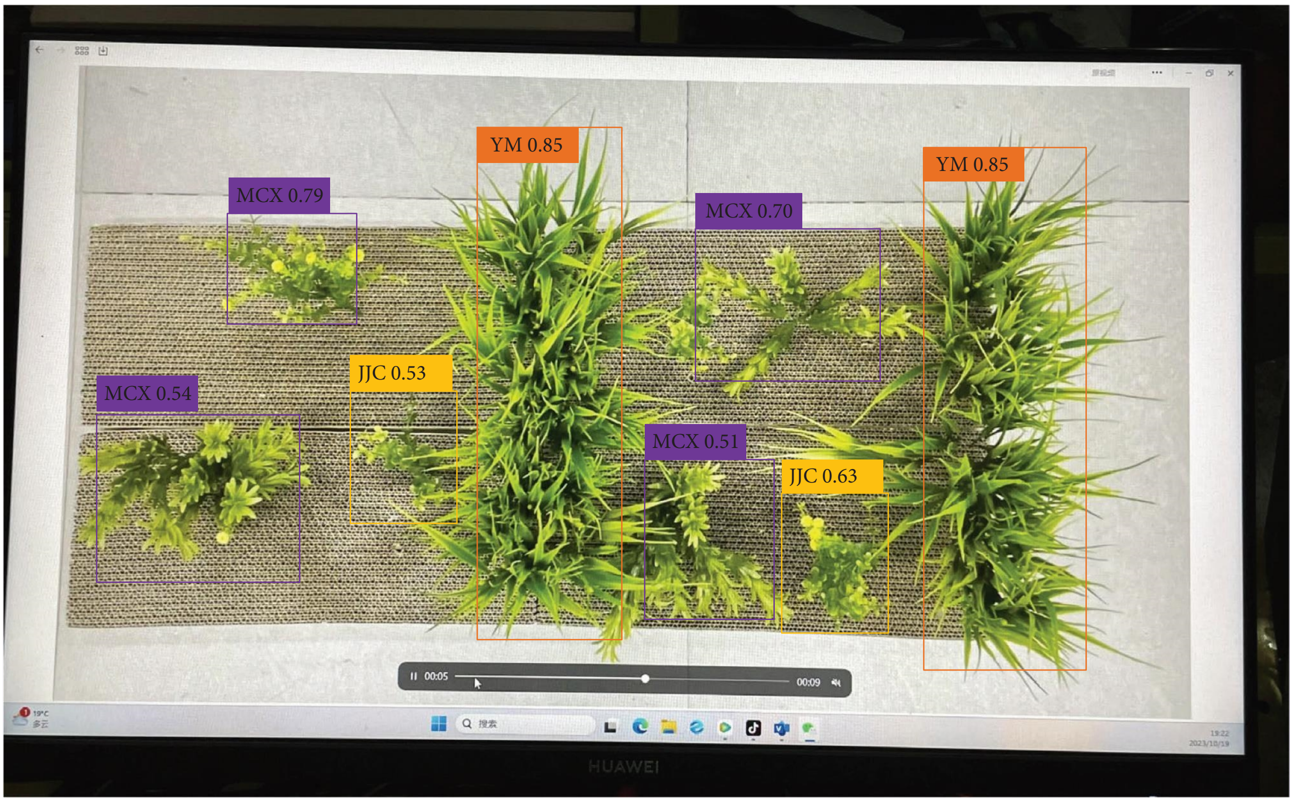


Figure 10: Video testing results.

parameters are outlined in Table 5. In the initial training round, weights pretrained on the COCO dataset provided by the Microsoft team were utilized. Training was conducted for 100 epochs, resulting in 100 sets of weight files. The final fine-tuning was performed using the weights from the 100th epoch. Following the completion of training, the models underwent prediction testing, which encompassed both image and video testing.

Table 4: YoloX training parameter settings.

Parameter name	Set value
input_shape	416 × 416
mosaic	False
Cosine_scheduler	False
Freeze_Epoch	50
Freeze_batch_size	2
UnFreeze_Epoch	100
Unfreeze_batch_size	1
num_workers	1

Table 5: YoloX model parameter settings.

Parameter name	Set value
Input image size	416 × 416
Confidence score	0.5
Nonmaximum suppression value	0.8
Maximum predicted boxes	100

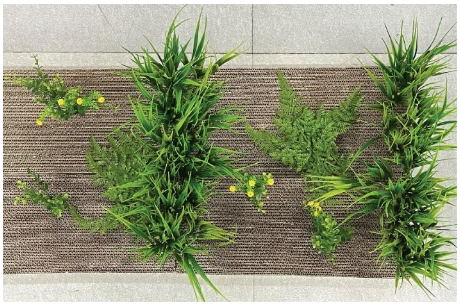


Figure 11: Test sample image.

The trained YOLOX model and the YOLOX model optimized with the CBAM mechanism underwent image testing. For each input single image, the YOLOX model was utilized for object detection. Bounding boxes were drawn around detected objects using various colors, and the names of the detected objects were annotated on each bounding box. Please refer to Figures 11–13 for visual representations. To simulate real-world detection scenarios, the dataset was compiled into a single video and input to the YOLOX model for object detection. The video's FPS were set to 25.0. The final output is a video containing bounding boxes around detected objects. The visual representation of the results can be observed in Figure 13.

5. Experimental Data Analysis

5.1. Analysis of AP Metrics. Due to the specific requirements of the project where recognition speed is important, we primarily focus on accuracy and F1 score as the main evaluation metrics. In the PyTorch framework, we conducted object detection on different types of weeds or crops using four versions of Faster R-CNN and YOLOX. The results are depicted in Figures 14 and 15. Despite all algorithms achieving an average detection time of under 500 ms, causing minimal impact on system performance, it is evident from the graphs that the YOLOX-X model exhibits the best performance in both frameworks. Consequently, we choose to perform a comparative analysis between the YOLOX-X algorithm and Faster R-CNN. In the PyTorch framework, significant improvements in AP values are observed for Faster R-CNN, YOLOX, and the optimized YOLOX-X algorithm. Upon comparing the data from Figures 15 and 16, it is observed that the AP values for seedlings in Faster R-CNN and YOLOX algorithms are nearly identical, differing by only 0.23%. The AP value for the improved YOLOX-X reaches 100%. For the recognition of "Portulaca" and "*Amaranthus blitum*," the AP values obtained from YOLOX-X training are higher by 1.7% and 2.06%, respectively, compared to Faster R-CNN training. The recognition rate of "Portulaca" with the improved YOLOX-X reaches 100%. In the case of "*Polygonum orientale*" and "*Stellaria media*," Faster R-CNN training yields higher AP values by 1.24% and 0.42%, respectively, than YOLOX-X training, and the improved YOLOX-X outperforms Faster R-CNN with a 1.31% higher AP value for "*Polygonum orientale*" and a 4.21% lower AP value for "*Stellaria media*."

5.2. Analysis of mAP and Recognition Time. Under both the TensorFlow and PyTorch frameworks, the model YOLOX-X achieves the highest mAP values for all categories: 93.57% and 97.07%, respectively. Regarding recognition time, YOLOX-s demonstrates the fastest recognition time at 0.038 and 0.027 s, respectively.

In the TensorFlow framework, when comparing the data from Figure 15, it is apparent that YOLOX-s boasts the fastest recognition time of 0.038 s, but its mAP is only 84.81%. Accuracy is a crucial metric for assessing the performance of machine learning models. In practical projects, even with fast recognition times, lack of improved accuracy would limit substantial progress. Comparing the mAP and recognition time between Faster R-CNN and YOLOX-x, the mAP value of YOLOX-x is only 0.86% higher than that of Faster R-CNN. In terms of recognition time, YOLOX-x is 0.566 s faster than Faster R-CNN, representing an 8.45-fold improvement.

In the PyTorch framework, the training times for all models remain under 0.1 s. YOLOX-s exhibits the fastest recognition time at 0.027 s, accompanied by an mAP of 96.13%. Comparing the mAP and recognition time between Faster R-CNN and YOLOX-x, YOLOX-x's mAP value is 0.43% higher than that of Faster R-CNN. Regarding recognition time, YOLOX-x is 0.007 s faster than Faster R-CNN, as depicted in Figure 16. Consequently, in the PyTorch framework, the mAP and recognition time differences between YOLOX-s, YOLOX-x, and Faster R-CNN are minimal, warranting further analysis based on F1 score.

Figures 17–19 indicate that in the TensorFlow framework, both the Faster R-CNN and YOLOX-X algorithms exhibit excellent recognition performance for seedlings. They achieve high AP and F1 scores, demonstrating their accuracy in seedling recognition. Both the Faster R-CNN and YOLOX-X models perform well in recognizing most plants, with each model potentially having a slight advantage

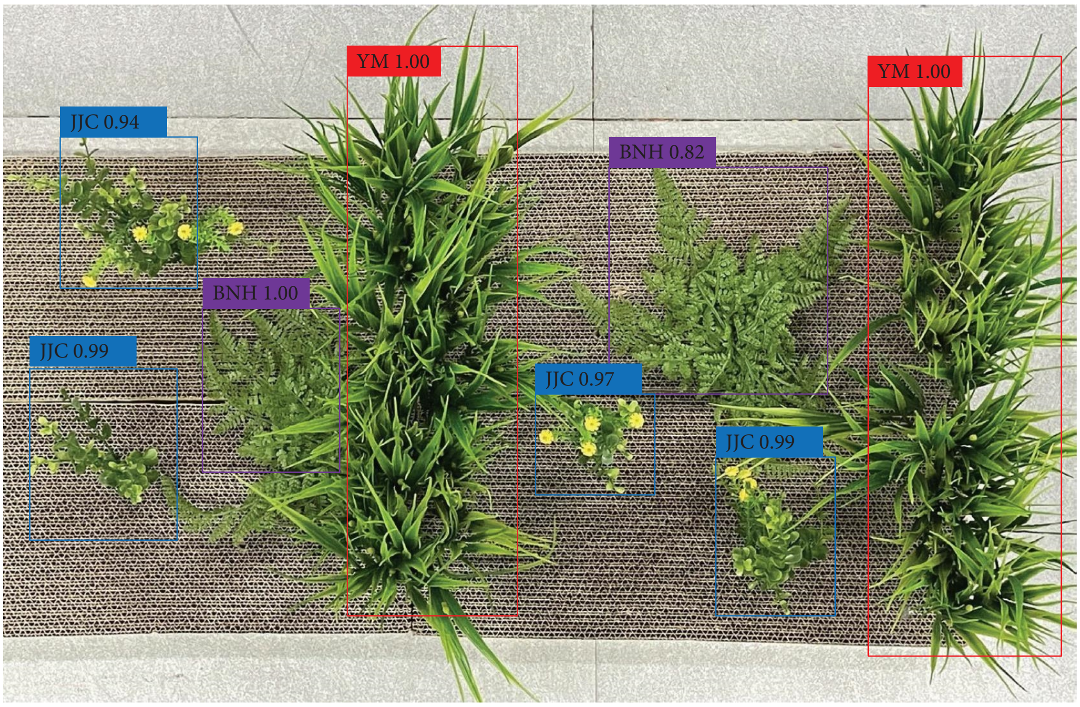


Figure 12: Predicted results.

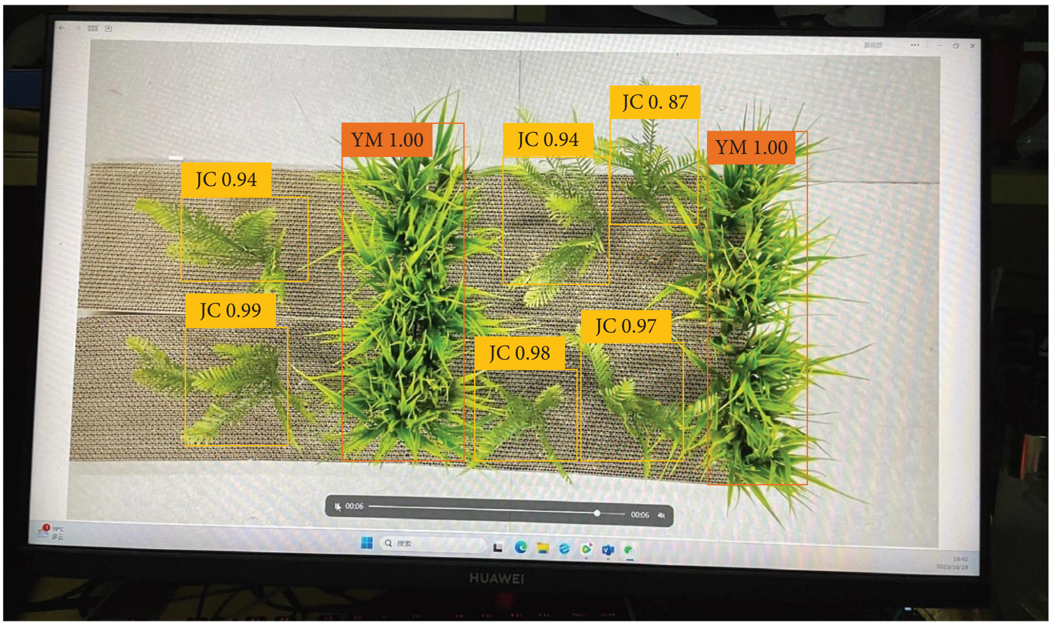


Figure 13: Video testing results.

in specific plant recognition. However, considering mAP and time comprehensively, YOLOX-X is superior.

In object detection, higher values for both recall and precision are ideal. However, these values are often negatively correlated in practice. The F1 score is a metric commonly used in statistics to measure the accuracy of binary classification algorithms. In object detection, it balances precision and recall, providing a good evaluation of algorithm performance.

From the figures, it can be observed that both YOLOX-X and Faster R-CNN have their strengths in the TensorFlow framework. However, in the PyTorch framework, all algorithms exhibit significantly improved F1 scores, indicating greater stability. Analyzing the AP values and F1 scores of YOLOX-X and Faster R-CNN, as shown in Figures 20 and 21, it is evident that the F1 score is positively correlated with the AP value; as the AP value increases, the F1 value also increases. Nevertheless, looking at the F1 score trendlines,

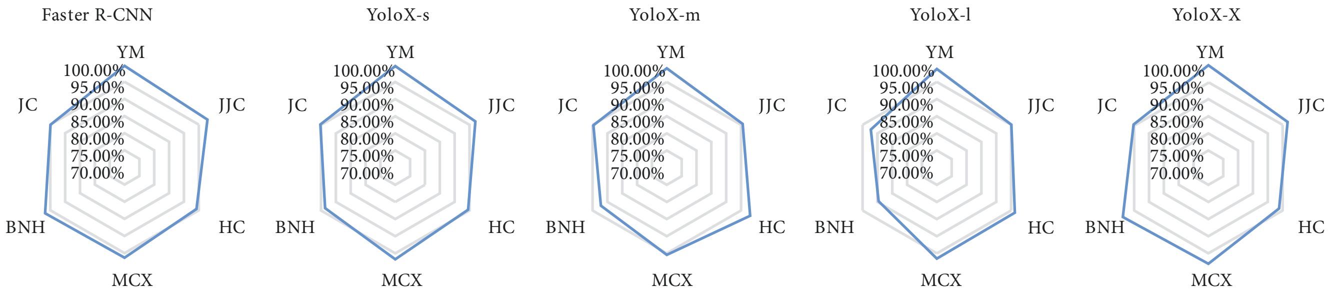


Figure 14: Radar chart of AP values for Faster R-CNN and YOLOX-x in the PyTorch framework.

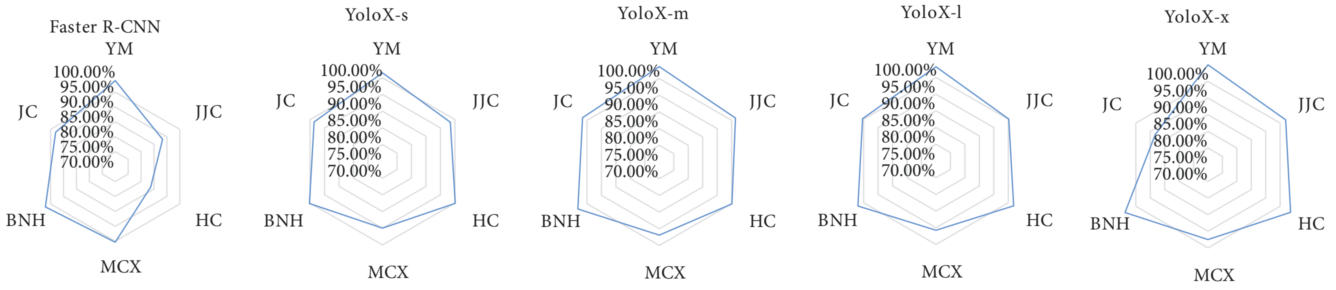


Figure 15: Radar chart of AP values for improved YOLOX-x in the PyTorch framework.

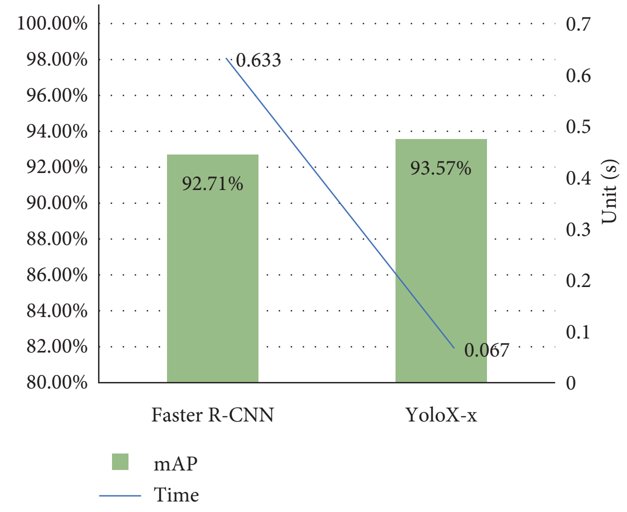


Figure 16: Mixed comparison of mAP and time for Faster R-CNN and YOLOX series in TensorFlow framework.

Faster R-CNN's F1 score has more variation, while YOLOX-X's F1 score shows smoother changes, indicating better stability in YOLOX-x's object detection algorithm.

Figures 20 and 21 illustrate that the improved YOLOX algorithm demonstrates significant improvements in both recognition time and accuracy. In the PyTorch framework, the improved YOLOX algorithm performs optimally.

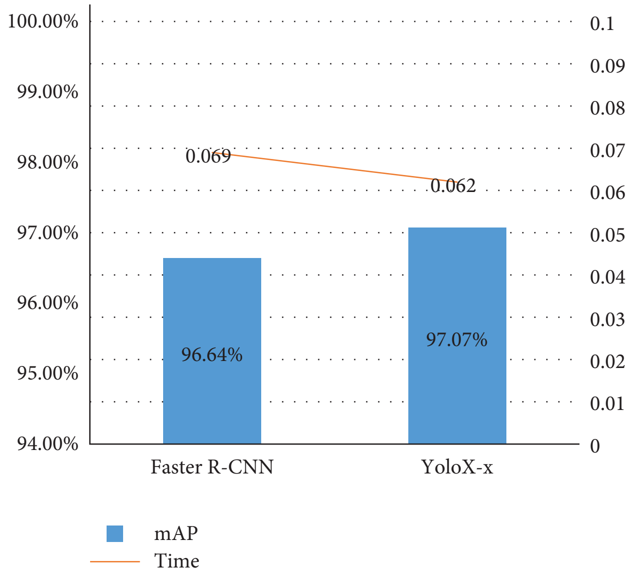


Figure 17: Mixed comparison of mAP and time for Faster R-CNN and YOLOX series in PyTorch framework.

5.3. Analysis of F1 Score. In object detection, higher values of both recall and precision are desired, but in practice, these values often exhibit a negative correlation. The F1 score is a metric used in statistics to measure the accuracy of binary classification models [20]. In object detection, it strikes a balance between precision and recall, providing a comprehensive evaluation of a model's performance. Figures 20 and 21 illustrate the radar charts of F1 scores for Faster R-CNN and YOLOX series in both the TensorFlow and PyTorch frameworks.

In Figure 20, we can observe that in the TensorFlow framework, YOLOX-X exhibits the most balanced F1 score within the YOLOX series. This indicates that the YOLOX-X model possesses good stability. Compared to YOLOX-X,

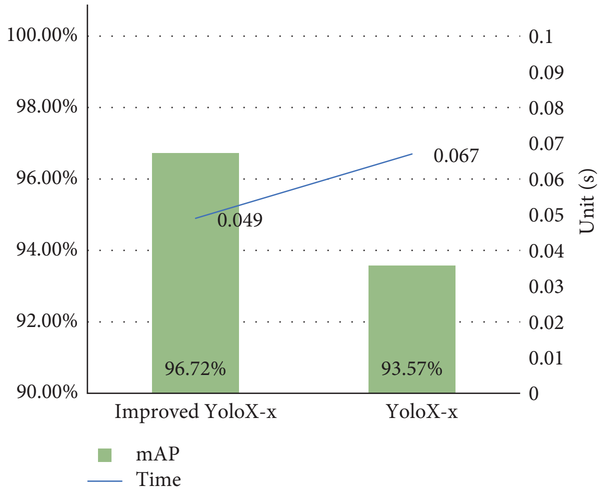


Figure 18: Mixed comparison of mAP and time for improved YOLOX series and YOLOX series in TensorFlow framework.

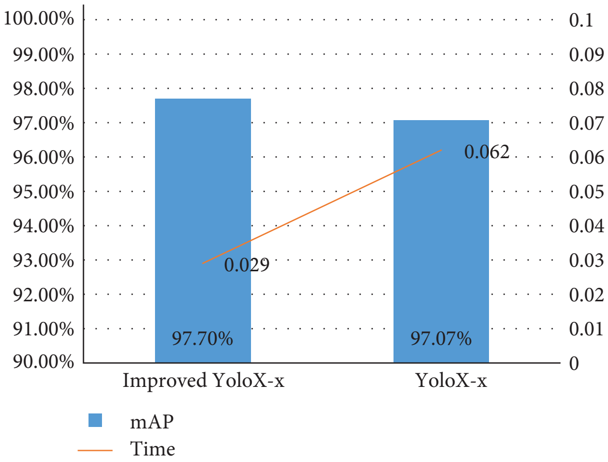


Figure 19: Mixed comparison of mAP and time for improved YOLOX and YOLOX series in PyTorch framework.

both YOLOX-X and Faster R-CNN have their respective strengths. As shown in Figure 21, under the PyTorch framework, all models show a significant improvement in F1 values, suggesting increased stability across the models. When comparing the AP values and F1 scores of YOLOX-X and Faster R-CNN in Figures 20 and 21, a positive correlation between F1 score and AP value is evident. Figures 22–25 indicate higher AP values correspond to higher F1 scores. However, examining the F1 score trend lines for YOLOX-X and Faster R-CNN, we notice that Faster R-CNN's F1 score fluctuates more significantly, while YOLOX-X's F1 score changes more gradually. This observation indicates that YOLOX-X's object detection model offers better stability.

5.4. Summary and Analysis. The main tasks in this section involve testing object detection algorithms and conducting data analysis using a cross-comparison approach. The Faster R-CNN and YOLOX object detection algorithms are separately tested under the TensorFlow and PyTorch frameworks. Evaluation metrics such as AP values, mAP values, time, and F1 scores are obtained. Additionally, the YOLOX algorithm with the inclusion of the CBAM mixed attention mechanism is tested, and similar evaluation metrics are derived. In terms of data analysis, a horizontal analysis approach is initially used to comprehensively compare Faster R-CNN and YOLOX-X, revealing varying evaluation metrics when faced with different recognition targets. Subsequently, a vertical analysis approach is employed to compare the YOLOX algorithm with the optimized YOLOX algorithm, analyzing the advancements brought by the Improved YOLOX algorithm.

The CBAM mixed attention mechanism is an adaptive weighting algorithm for input features, helping the algorithm better capture crucial information in images. Incorporating the CBAM mixed attention mechanism into the YOLO-X algorithm results in the following optimizations:

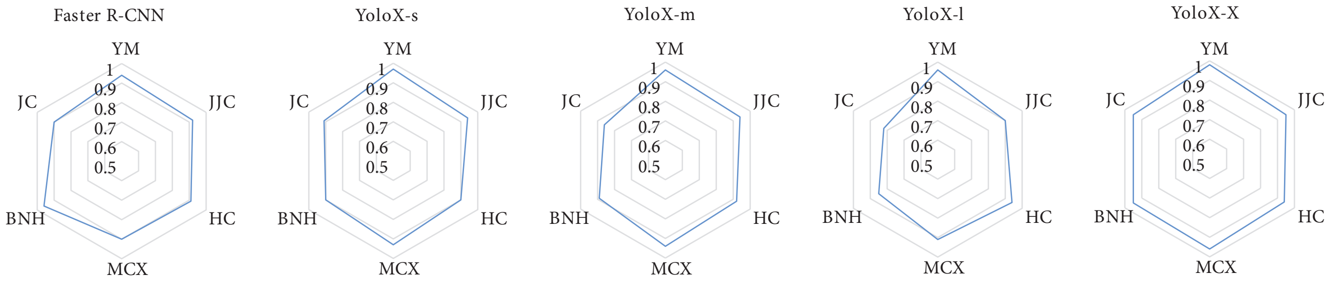


Figure 20: Radar chart of F1 scores for Faster R-CNN and YOLOX series in the TensorFlow framework.

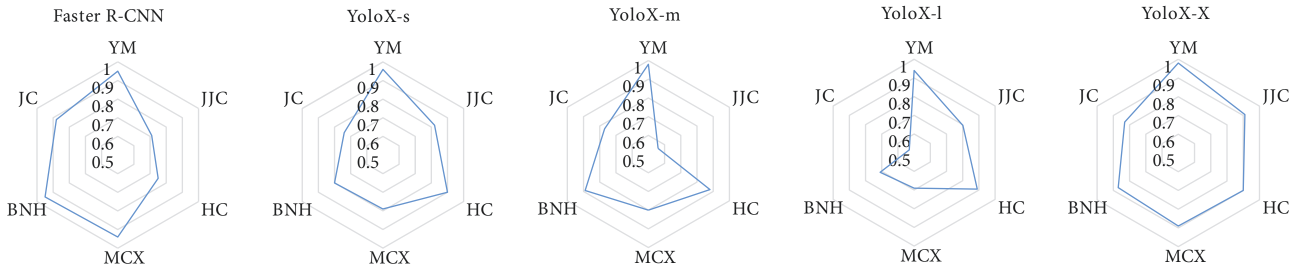
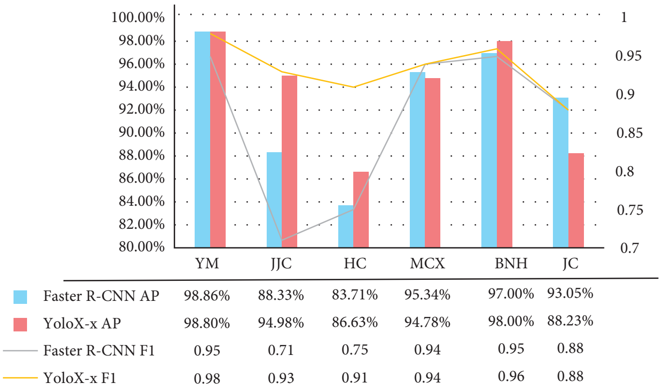
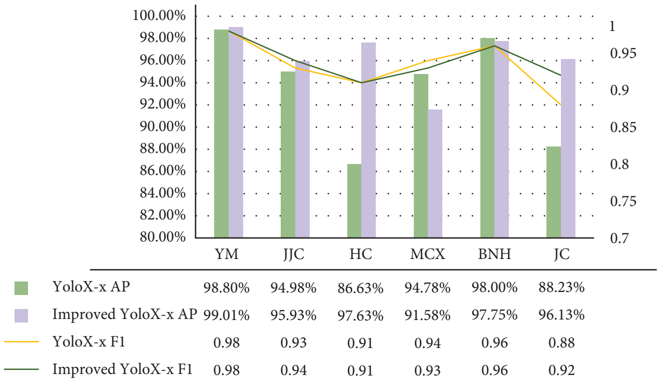


Figure 21: Radar chart of F1 scores for Faster R-CNN and YOLOX series in the PyTorch framework.



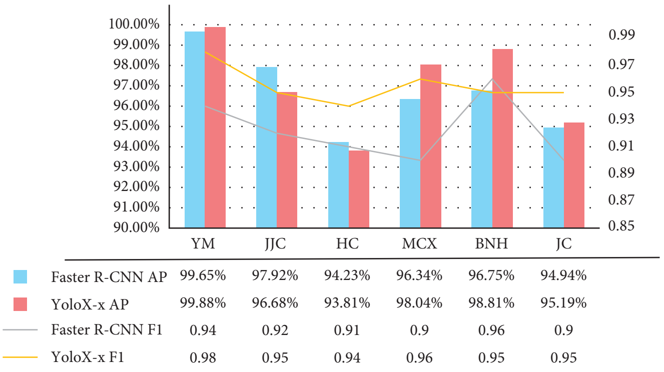
	YM	JJC	HC	MCX	BNH	JC
Faster R-CNN AP	98.86%	88.33%	83.71%	95.34%	97.00%	93.05%
YoloX-x AP	98.80%	94.98%	86.63%	94.78%	98.00%	88.23%
Faster R-CNN F1	0.95	0.71	0.75	0.94	0.95	0.88
YoloX-x F1	0.98	0.93	0.91	0.94	0.96	0.88

Figure 22: Mixed comparison of F1 score and AP value for Faster R-CNN and YOLOX-X in TensorFlow framework.



	YM	JJC	HC	MCX	BNH	JC
YoloX-x AP	98.80%	94.98%	86.63%	94.78%	98.00%	88.23%
Improved YoloX-x AP	99.01%	95.93%	97.63%	91.58%	97.75%	96.13%
YoloX-x F1	0.98	0.93	0.91	0.94	0.96	0.88
Improved YoloX-x F1	0.98	0.94	0.91	0.93	0.96	0.92

Figure 24: Mixed comparison of F1 scores and AP values for YOLOX-X and improved YOLOX-X in the TensorFlow framework.



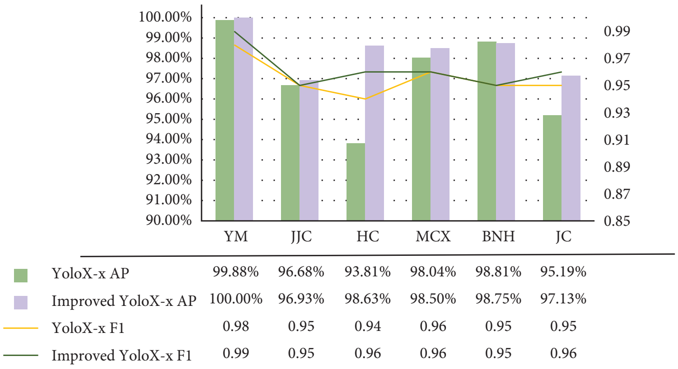
	YM	JJC	HC	MCX	BNH	JC
Faster R-CNN AP	99.65%	97.92%	94.23%	96.34%	96.75%	94.94%
YoloX-x AP	99.88%	96.68%	93.81%	98.04%	98.81%	95.19%
Faster R-CNN F1	0.94	0.92	0.91	0.9	0.96	0.9
YoloX-x F1	0.98	0.95	0.94	0.96	0.95	0.95

Figure 23: Mixed comparison of F1 score and AP value for Faster R-CNN and YOLOX-X in PyTorch framework.

(1) Critical feature selection for enhanced precision: The attention mechanism directs YOLOX's focus toward selecting more crucial features, enhancing algorithm precision and robustness.

(2) Noise ignoring and redundancy reduction: The attention mechanism allows YOLOX to ignore irrelevant background noise, filtering out redundant information. This enables the algorithm to swiftly and accurately detect target objects, enhancing efficiency and speed.

The addition of the CBAM mixed attention mechanism thus improving YOLOX-X in terms of accuracy, efficiency, and effectiveness.



	YM	JJC	HC	MCX	BNH	JC
🟩 YoloX-x AP	99.88%	96.68%	93.81%	98.04%	98.81%	95.19%
🟪 Improved YoloX-x AP	100.00%	96.93%	98.63%	98.50%	98.75%	97.13%
— YoloX-x F1	0.98	0.95	0.94	0.96	0.95	0.95
— Improved YoloX-x F1	0.99	0.95	0.96	0.96	0.95	0.96

Figure 25: Mixed comparison of F1 score and AP values for YOLOX-X and improved YOLOX-X in the PyTorch framework.

6. Discussion

Through comprehensive cross-comparison and analysis of the data, YOLOX-X has emerged as the optimal algorithm under the PyTorch framework. Further optimization through the integration of the CBAM mixed attention mechanism has led to significant improvements in relevant metrics. With the introduction of the attention mechanism, the algorithm becomes more adept at selecting critical features while disregarding background noise. This enhancement in feature selection has not only heightened detection accuracy but also improved the algorithm's robustness. Additionally, the reduction in redundant information has resulted in shortened training times and accelerated training speeds. In sum, the addition of the attention mechanism has not only refined the precision and robustness of the algorithm but also streamlined its training process and boosted its efficiency.

7. Conclusion

To address the challenges posed by traditional agricultural cultivation practices and ensure food safety, a focus has been placed on tackling weed-related issues during crop growth. In order to assess the feasibility of the YOLOX series in the realm of agricultural weed recognition, a self-constructed dataset based on weed algorithms was utilized. The performance of various object detection algorithms, including Faster R-CNN, YOLOX-S, YOLOX-M, YOLOX-L, and YOLOX-X, was tested under both the TensorFlow and PyTorch frameworks. A comprehensive evaluation encompassing AP, mAP, recognition time, and algorithmic F1 scores was conducted. Through these tests, it was concluded that the YOLOX algorithm holds strong potential for weed recognition. Specifically, YOLOX-X exhibited higher recognition accuracy, faster processing speed, and greater stability compared to the Faster R-CNN algorithm. YOLOX-X achieved an mAP of 97.07% with an average recognition time of 0.062 s. By integrating the CBAM mixed attention mechanism into the YOLO-X algorithm, the optimized YOLOX algorithm exhibited significant improvements. It achieved an mAP of 97.70% with an average recognition time of 0.029 s. These results underscore the remarkable feasibility and advancement of YOLOX algorithms in the field of agricultural weed recognition. Moreover, there remains ample room for further enhancements in the YOLOX algorithm. Ongoing research is needed to achieve a better balance between detection speed and accuracy, thereby continuing to elevate the algorithm's performance.

Data Availability

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

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The authors would like to thank their colleagues for their guidance and the help of their classmates. This paper was sponsored by the "Qing Lan Project" of Jiangsu University, the General Natural Science Research Project of Jiangsu University (19KJD510005), and 136 Engineering Project of School of Mechanical and Electrical Engineering of San Jiang University (21SJJD005).

References

[1] G. G. Peteinatos, P. Reichel, J. Karouta, D. Andújar, and R. Gerhards, "Weed identification in maize, sunflower, and potatoes with the aid of convolutional neural networks," *Remote Sensing*, vol. 12, no. 24, Article ID 4185, 2020.

[2] T. Gao and X. Zhang, "Investigation into recognition technology of helmet wearing based on HBSYOLOX-s," *Applied Sciences*, vol. 12, no. 24, Article ID 12997, 2022.

[3] A. Subeesh, S. Bhole, K. Singh et al., "Deep convolutional neural network models for weed detection in polyhouse grown bell peppers," *Artificial Intelligence in Agriculture*, vol. 6, no. h, pp. 47–54, 2022.

[4] J. Xiaojun, S. Yanxia, C. Jun, B. Muthukumar, Y. Jialin, and C. Yong, "A novel deep learning-based method for detection of weeds in vegetables," *Pest Management Science*, vol. 78, no. 5, pp. 1861–1869, 2022.

[5] Y. Wang, Y. Du, and P. Hong, "Weed fine-grained identification based on bilinear convolutional neural network," *Intelligent Computers and Applications*, vol. 11, no. 7, pp. 36–42, 2021.

[6] X. U. Yan, W. E. N. Desheng, Z. H. O. U. Jianping, F. A. N. Xiangpeng, and L. I. U. Yang, "A Faster R-CNN-based method for identifying cotton seedlings and weeds in Xinjiang," *Journal of Drainage and Irrigation Machinery Engineering*, vol. 39, no. 6, pp. 602–607, 2021.

[7] L. Quan, B. Wu, and S. Mao, "A Mask R-CNN based method for weed instance segmentation and leaf age recognition in agricultural fields," *Journal of Northeastern Agricultural University*, vol. 52, no. 4, pp. 65–76, 2021.

[8] L. Kaijing, X. Yan, Z. Jianping, F. Xiangpeng, and W. Yutong, "Faster R-CNN and data enhancement based weed identification in cotton field at seedling stage," *Journal of Xinjiang University (Natural Science Edition) (in English)*, vol. 38, no. 4, pp. 450–456, 2021.

[9] W. Qian, D. Jianwei, and Z. Qi, "Comprehensive experimental design of image denoising system," *Electronic Technology and Software Engineering*, vol. 24, pp. 89–92, 2021.

[10] W. Xin, W. Shuhong, and W. Y. Li, "Forest fire smoke detection model based on deep convolutional long and short-term memory network," *Computer Applications*, vol. 39, no. 10, pp. 2883–2887, Research on face recognition based on convolutional neural network combined with SVM, 2019.

[11] H. Chao and B. Hua, "Research on face recognition based on convolutional neural network combined with SVM," *Microcomputers and Applications*, vol. 36, no. 15, pp. 56–58 72, 2017.

[12] Z. H. Liu, Y. H. Liao, S. Z. Yuan, H. Y. Huang, and L. M. Xie, "Research on the identification of bitter melon leaf diseases based on Faster R-CNN," *Journal of Guangdong Light Industry Vocational Technology College*, vol. 20, no. 2, pp. 1–4, 2021.

[13] Z. Xi, J. Zhengmeng, and J. Yaqin, "Full-variance image coloring algorithm with fused depth image prior," *Systems Engineering and Electronics Technology*, vol. 44, no. 2, pp. 385–393, 2022.

[14] X. Wei, D. Xuewen, S. Lanchao, and L. Li, "Implementation of binocular vision ranging system based on MATLAB and OpenCV," *Journal of Tianjin Vocational and Technical Teachers' University*, vol. 14, Article ID 1116932, 2017.

[15] H. Yuping, L. Weixuan, and X. Zuhuan, "A comparative analysis of deep learning frameworks based on TensorFlow and PyTorch," *Modern Information Technology*, vol. 4, no. 4, pp. 80–87, 2020.

[16] W. Chaoyang, F. Shaosheng, L. Zheng, L. Bin, and Z. Wei, "Abnormal state detection of overhead lines in distribution networks based on improved FasterRCNN," *Journal of Electric Power*, vol. 34, no. 4, pp. 322–329, 2019.

[17] Lin Sen, Liu Meiyi, and Tao Zhiyong, "Underwater rare product detection using attention mechanism with improved YOLOv5," *Journal of Agricultural Engineering*, vol. 37, no. 18, pp. 307–314, 2021.

[18] W. Xu, L. Sun, C. Zhen, B. Liu, Z. Yang, and W. Yang, "Deep learning-based image recognition of agricultural pests," *Applied Sciences*, vol. 12, no. 24, Article ID 12896, 2022.

[19] Y. Hongpeng, C. Bo, C. Yi, and L. Zhaodong, "A review of vision-based target detection and tracking," *Journal of Automation*, vol. 42, no. 10, pp. 1466–1489, 2016.

[20] Y. Wan, Y. Han, and H. Q. Lu, "Exploration of motion target detection algorithm," *Computer Simulation*, vol. 10, pp. 221–226, 2006.