IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 8 March 2024, accepted 15 May 2024, date of publication 20 May 2024, date of current version 30 May 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3403491

RESEARCH ARTICLE

JutePest-YOLO: A Deep Learning Network for Jute Pest Identification and Detection

SHUAI ZHANG[1], **HENG WANG**[1], **CONG ZHANG**[2], **ZHENG LIU**[1], **YIMING JIANG**[1], **AND LEI YU**[1]

[1]School of Mathematics and Computer, Wuhan Polytechnic University, Wuhan 430048, China
[2]School of Electrical and Electronic Engineering, Wuhan Polytechnic University, Wuhan 430048, China

Corresponding author: Heng Wang (wh825554@163.com)

This work was supported in part by the Key Project of the Scientific Research Program of the Hubei Provincial Department of Education under Grant D20201601; in part by the Major Technical Innovation Projects of Hubei Province under Grant 2018ABA099; and in part by the Hubei Provincial Key Laboratory of Intelligent Robot under Grant HBIR202101.

ABSTRACT In recent years, jute, as an important natural fiber crop, has become more and more significant in the production process of insect pests, causing serious harm to agricultural production. Especially in the field of crop pest identification with complex backgrounds, fuzzy features, and multiple small targets, the lack of datasets specifically for jute pests has led to the large limitations of traditional pest identification models in terms of generalization. At the same time, the research on models specifically for jute pest detection is still in its infancy. To solve this problem, we constructed a large-scale image dataset containing nine types of jute pests, which was highly targeted and could effectively support model training and evaluation. In this study, we developed a deep convolutional neural network model based on YOLOv7, namely JutePest-YOLO. The model has optimized the Backbone, Head, and loss functions of the baseline model, and introduced the new ELAN-P module and P6 detection layer, which effectively improved the model's ability to identify jute pests in complex backgrounds. The experimental results showed that compared with the baseline model, the Precision, Recall, and F1 scores of the JutePest-YOLO model were improved by 3.45%, 1.76%, and 2.58%, respectively; the mAP@0.5 and mAP@0.5:0.95 was improved by 2.24% and 3.25%, and the overall model's computation (GFLOPS) was reduced by 16.05%. Compared to other advanced methods such as YOLOv8s, JutePest-YOLO has achieved superior performance in terms of detection accuracy, with a precision of 98.7% and mAP@0.5 reaching 95.68%. As a result, JutePest-YOLO not only achieved significant improvement in recognition accuracy but also optimized computational efficiency. It's a high-performance, lightweight solution for jute pest detection.

INDEX TERMS Jute pest detection, YOLOv7, PConv, wise-IoU, object detection, deep learning.

I. INTRODUCTION

This Jute is a highly versatile natural fiber, widely used in the manufacture of a variety of environmentally friendly products, such as bags, handicrafts, textiles clothing, etc [1]. It is often seen as an ideal alternative to nylon and polypropylene because jute is not only durable and reusable but also poses minimal threat to human health and the natural environment [2]. In addition, the low cost of jute is preferable to synthetic fibers, making it an affordable material choice [3].

Jute in Bangladesh is known as the ‘‘golden fiber’’, not only because of its unique golden color but also because of its important contribution to the national economy. In China, jute also has a long history, the importance of the agricultural product is self-evident [4]. One of the major challenges faced during jute production is the threat of pests. These pests not only have a serious impact on the growth of jute but also have a significant negative impact on the overall yield and quality. For example, Indigo caterpillars feed heavily on jute leaves, resulting in stunted plant growth and in severe cases, plant death. Jute semiloopers attack the tops and leaves of jute, making it difficult for the plant to flower and set seed, thus

The associate editor coordinating the review of this manuscript and approving it for publication was Alba Amato.

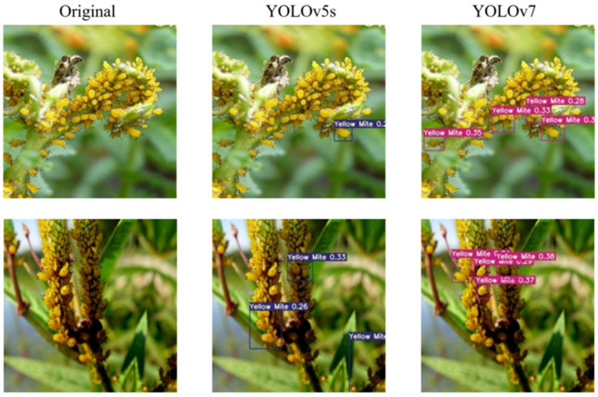


FIGURE 1. Detection results for the current stage of the network model. In the first figure, YOLOv5s uses only one detection frame, and YOLOv7 uses four detection frames, and in the second figure, YOLOv5 uses only three detection frames, and YOLOv7 uses four detection frames.

directly affecting yield. Yellow mites affect jute by causing leaf spotting, wilting, and eventual defoliation, which not only affects the quality of the jute fiber but also reduces yield [5].

Apart from the above-mentioned pests, various other pests also pose a threat to jute production such as root pests and larvae that colonize the soil and attack the root system of jute, affecting the plant's ability to absorb water and nutrients. The activities of these pests not only make jute production difficult but also increase the cost of plant protection and pest control for farmers. Therefore, effective identification and control of these pests is important to safeguard the yield and quality of jute.

In jute production, although the application of insecticides is a common and rapid method of pest control with significant cost-effectiveness, the effectiveness of most insecticides is limited to specific species of pests. Traditional visual inspection methods [6], while relying on specialized knowledge and experience, can easily lead to misuse of insecticides and affect production due to similar pest symptoms and complex detection processes. Existing inspection methods either rely on complex hardware equipment or are difficult to respond quickly in the field. Therefore, the development of an Artificial Intelligence (AI)-based real-time inspection technology, especially an efficient pest detection solution that can be adapted to mobile devices, is important for real-time monitoring and effective pest control.

In recent years, deep learning methods have made significant progress in the field of crop pest detection, especially in the application of two network models, YOLOv5s and YOLOv7. YOLOv5s, with its lightweight structure and efficient performance, performed well in small target detection and was suitable for fast detection in resource-constrained environments [7]. YOLOv7, on the other hand, has made an even greater breakthrough in pest detection accuracy and speed due to its more advanced feature extraction and target recognition capabilities. Both models demonstrate excellent recognition capabilities when dealing with complex crop backgrounds and pests of various scales, and YOLOv7, in particular, is widely regarded as the fastest and most accurate real-time object detector currently available [8].

Figure 1 demonstrates the effectiveness of YOLOv5s and YOLOv7 in detecting small target pests. However, despite the effectiveness of these two models in general-purpose object detection, they still face certain challenges when confronted with the detection of small-target pests. These challenges mainly originated from the feature ambiguity due to the complexity of the pest background and the diversity of small-target pest species. These problems triggered misdetection and omission in the detection of small target pests, thus limiting the efficiency and accuracy of the model in jute pest identification applications.

To solve the above problems, we proposed a more efficient jute pest detection model, JutePest-YOLO, which was innovatively optimized based on YOLOv7, with special consideration for the practicality of mobile devices. Our main innovations are as follows:

1) Optimize the model for mobile device compatibility: To better adapt to mobile devices, we replaced the 3 × 3 regular convolution in the ELAN module of the baseline model with PConv to form a new ELAN-P module, which not only reduced the amount of computation and the number of memory accesses of the whole model but also improved the computational efficiency of the model so that the network could extract the features of the jute insect pests more quickly.
2) Solve the feature ambiguity problem: For the complexity of the pest background, we optimized the Head part of the baseline model and added a new P6 detection layer, which extended the sensory field of the model and enhanced the feature extraction capability of the original image, so that JutePest-YOLO could recognize the ambiguous features in the complex background more clearly.
3) Enhance the detection ability of small targets: Considering the diversity of small target pests, we improved the loss function of the model by abandoning the original CIoU loss function and adopting WIoU to optimize the loss function, which effectively solved the problems such as misdetection and omission of targets of all scales.

In addition, we constructed a large-scale image dataset containing nine types of jute pests, which not only provided an effective training and tested basis for the model but also was an important contribution to the research field of jute pest recognition.

The remainder of the paper is structured as follows: Section II describes the related work, Section III details the architecture of the JutePest-YOLO model, Section IV provides the results and analyses of the comparative experiments, the ablation experiments, and the visual presentation, and finally, in Section V, we summarize the results of the research.

II. RELATED WORK

In recent years, researchers have developed an increasing number of models using different Convolutional Neural Networks (CNNs), and this section highlights some of the recent noteworthy studies. Sourav and Wang [9] proposed a target detection model based on Transfer Learning (TL) and Deep Convolutional Neural Networks (DCNN), which was capable of identifying four groups of jute pests, Field cricket, Spilosoma obliqua, Jute stem weevil, and Yellow mite with a final accuracy of 95% for the identification of the four pest categories. However, in general, the accuracy of the network may decrease as the number of categories increases. Networks such as MobileNet [10], AlexNet [11], ShuffleNet [12], or GoogLeNet [13], for example, all assert that the richness of the dataset should be increased in all cases to improve the recognition rate of the model. Therefore, the model still needs to enhance the number of categories in the dataset greatly.

Karim et al. [14] worked on the same dataset and proposed a deep CNN model called PestDetector for the classification of the jute pest population. Their model achieved an excellent 99.18% training accuracy and 99.00% validation accuracy. However, it could perform better on unseen pest test datasets. Li et al. [15] established a new large-scale image dataset of ten types of jute diseases and pests, which includes eight different diseases as well as two types of jute pests. They proposed a unique model, YOLO-JD, which integrates into its main architecture the Sand Clock feature extractor Module (SCFEM), Deep Sand Clock feature extractor Module (DSCFEM) and Spatial Pyramid Pooling Module (SPPM) three new modules to extract image features efficiently and to be able to detect multiple types of diseases and pests in the same image as well as to find multiple instances of diseases in the same image. However, YOLO-JD achieved an average mAP of 96.63% for all disease categories. It was not as effective for jute pest category recognition. To address these issues, Talukder et al. [16] prepared a jute pest dataset containing 17 categories and about 380 photographs per pest category and designed JutePestDetect from several well-known pretrained models from previous studies, which is a model based on DenseNet201 and Resilient Migration Learning (TL) jute pest detection model, which can achieve a surprising 99% accuracy, despite the excellent performance of JutePestDetect in terms of accuracy on the homemade dataset, Md. Simul Hasan Talukder et al. did not test the JutePestDetect model for metrics such as mAP and FPS and lacked comparisons with other, then newer, models for jute pest identification. The jute pest dataset prepared by them was not targeted. The dataset was not targeted and lacked a description of the jute pest species.

In addition to pest identification in the field of jute, in other areas of crop pest identification, we also learned that pest species identification has problems such as small targets being easily lost, dense distribution of pests, individual recognition rate, etc. To improve the efficiency of pest detection further, Limei et al. [17] proposed an algorithm for pest species identification based on the YOLOv4 network, DF-YOLO; they introduced the DenseNet network into the YOLOv4 backbone network CSPDarknet53 to introduce DenseNet network to enhance the feature extractor capability of the model, improve the individual recognition rate of densely distributed targets, use the focal loss function to improve the effect of sample imbalance on training and optimize the mining process of complex samples, the algorithm achieved 94.89% mAP after testing on the homemade pest dataset, which is better than the improved the previous YOLOv4 by 4.66%. Xinming and Hong [18] compared the performance of two well-known target detection and classification models, YOLOv4 and YOLOv7, in detecting different leaf diseases. The performance comparison showed that both architectures were competitive in precision, F1 score, average precision, and recall, but the composite scaling and dynamic labeling of YOLOv7 provided superior performance. In addition, several researchers have focused on defect identification in raw jute fibers, with Nageshkumar et al. [19] exploring methods to identify and classify fiber defects in this specific context.

Although researchers in various fields have utilized various deep learning algorithms and neural network models to achieve significant results in crop pest recognition and other target detection tasks, relatively few studies have been conducted for the recognition of geographically important insects, especially jute pests. Moreover, existing studies generally lack specialized image datasets for jute pest identification. Therefore, we have produced a dataset specialized for jute pests based on the report published by the Department of Agricultural Extension, Bangladesh [20], which identified a wide range of pests causing damage to large-scale jute production, using the pest species in the report as a reference.

Considering the problems of feature ambiguity due to complex pest background, misdetection and underdetection of small target pest species, and generally large arithmetic volume faced by traditional models in the pest identification task, we proposed the JutePest-YOLO detection algorithm. The algorithm aimed to effectively break through the limitations in the field of jute pest identification and provide an accurate, efficient, and convenient pest detection solution for jute growers.

III. METHODS

A. YOLOv7 DETECTION

The YOLO (You Only Look Once) family of algorithms is an efficient target detection framework that has undergone several iterations and optimizations since it was first proposed by Redmon et al. In July 2022, Wang et al. released its latest version, YOLOv7 [8]. The network architecture of YOLOv7, as shown in Figure 2, can be divided into four main components: the Input, the Backbone, the Neck, and the Head.

For the input part, the image undergoes a series of preprocessing stages, such as data enhancement, and is then fed into the backbone for the feature extractor. Next, these extracted features are partially feature-fused by the Neck to

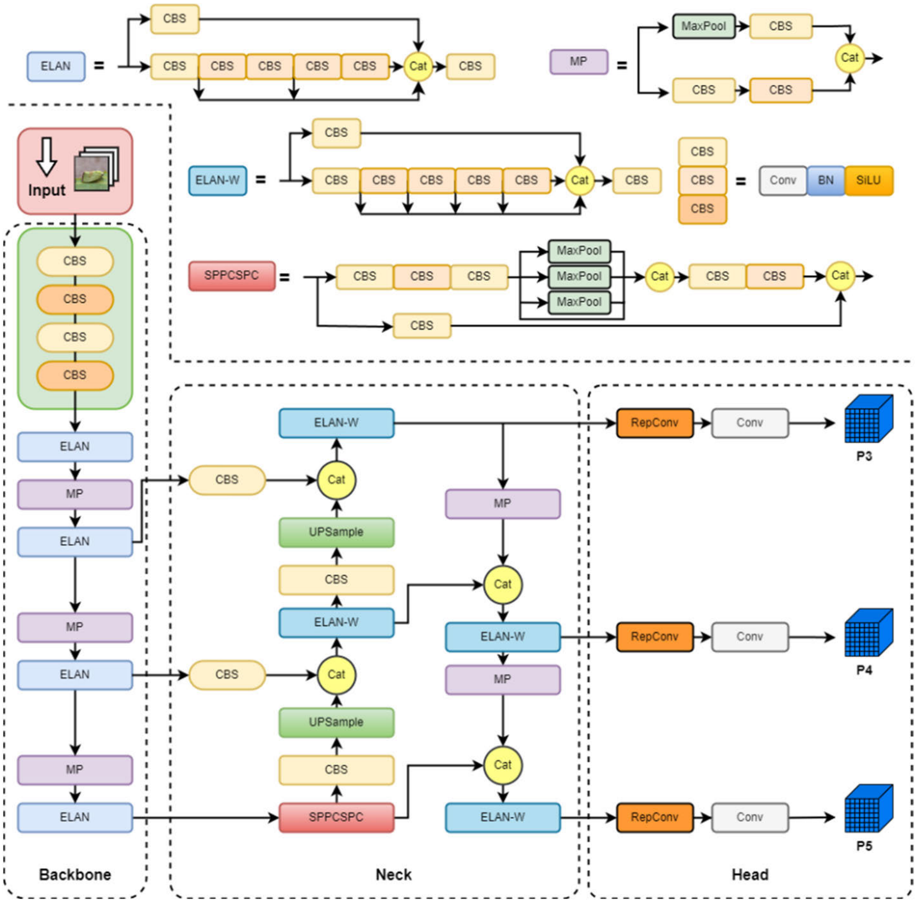


FIGURE 2. The overall structure of YOLOv7.

generate features of different sizes by fusing the three feature layers extracted by the backbone network. Finally, these fused features are fed to the Head module, which outputs the prediction results.

The input layer of YOLOv7 subjected the input images to a series of data augmentation algorithms, including color dithering, normalization, random cropping, etc., designed to improve the network's data diversity and generalization performance. Subsequently, the images after data augmentation are all subjected to a uniform scaling to scale them to the default size (640 × 640 × 3) to meet the backbone network's requirements for the input.

The main responsibility of the backbone network lies in extracting feature information from images in preparation for subsequent feature fusion and target detection tasks. The backbone network consists of three main components: the CBS, ELAN, and MPConv modules. Specifically, the CBS module consists of a convolutional layer, a batch normalization layer, and an activation function layer, whose main tasks are feature extractor and channel number transformation operations. The ELAN module is an efficient layer aggregation network that enhances the learning capability of the network without destroying the original gradient path. In addition, it guides the computation of different groups of features to induce the network to learn richer and more diverse feature information. At the same time, the MPConv module is mainly responsible for the downsampling operation, which combines the maxpool downsampling branch with the convolutional downsampling branch to merge the feature maps obtained from different downsampling methods. This fusion process preserves as much feature information as possible without increasing the computational burden.

The neck module consists of an optimized SPPCSPC module and Path Aggregation Feature Pyramid Network (PAFPN) for fusing feature maps of different sizes. Among them, the role of PAFPN is to retain the precise location information at the bottom level and fully fuse it with the abstract semantic information at the top level to achieve a complete fusion of semantic and location information at different levels. This strategy further improves the model's localization accuracy for multi-sized targets, especially for small targets in complex contexts.

In the detection head module, the number of image channels of the PAFPN output features was adjusted using the REPConv structure [21], and multi-scale target prediction

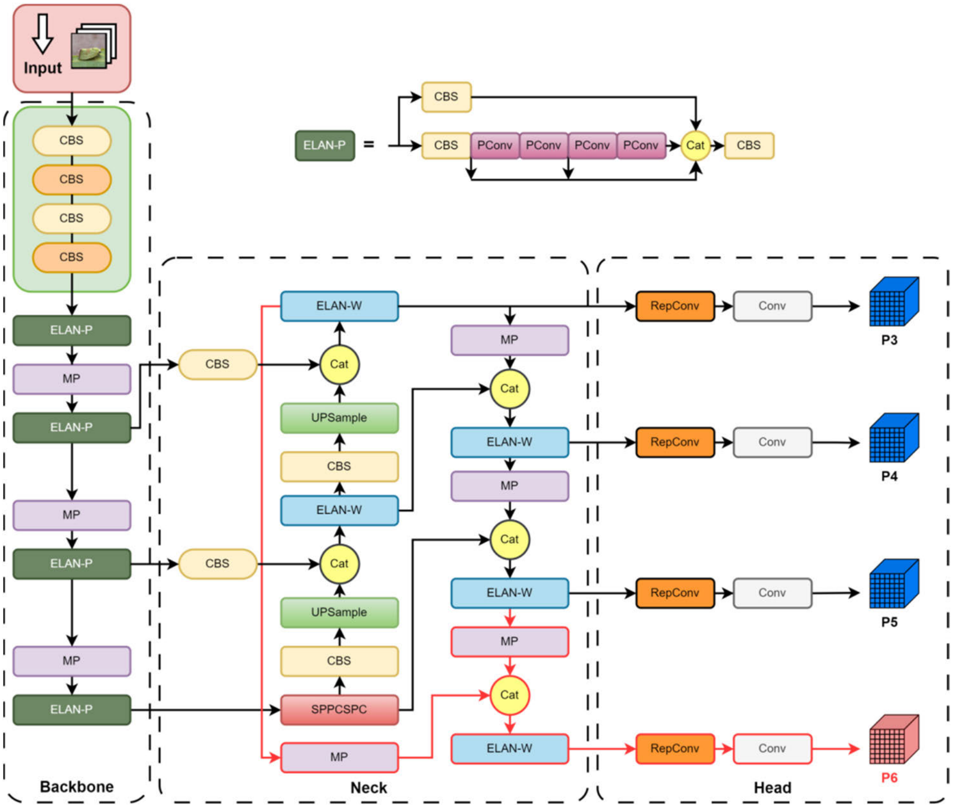


FIGURE 3. JutePest-YOLO structure diagram.

was performed by convolution on three different sizes of feature map branches output from the neck module.

B. IMPROVED JUTE PEST IDENTIFICATION ALGORITHM: JUTEPEST-YOLO

Although the traditional YOLOv7 algorithm can satisfy general image recognition tasks, its detection of jute pests still needs improvement. Most major false detections occur in scenes with small target detection and blurred pest features. In this study, we proposed an improved deep learning model for jute pest detection, JutePest-YOLO. Its structure is shown in Figure 3.

First, we replaced all the ELAN modules of the baseline model with the ELAN-P module, which was a module that replaced all the 3 × 3 regular convolutions in the ELAN module with PConv, where PConv applied regular convolutions to a single subset of the input channels as a way of extracting the spatial features, and by doing so, the sum of computational redundancy and the number of memory accesses could be reduced.

Next, we added a new P6 detection layer in the Head part of the original network, and the added P6 detection layer extended the sensory field of the model and enhanced the model's ability to extract the fuzzy features in the complex background. This is of crucial significance for the accurate localization and identification of jute pests, and can effectively solve the problems caused by the complex background of jute pests in this research field.

Finally, we improved the loss function of the model by abandoning the original CIoU loss function, because it failed to effectively distinguish the differences between targets of different sizes when dealing with aspect ratios, and was prone to cause problems such as missed detection and misdetection in small target detection. Therefore, we adopted WIoU v3 to optimize the loss function [22].WIoU v3 adopts a dynamic non-monotonic mechanism and designs a reasonable gradient gain allocation strategy, which reduces the occurrence of large gradients or harmful gradients from extreme samples.WIoU v3 can better take into account the target's size and positional information and effectively solve problems such as misdetection and omission of detection of targets at all scales.

1) ELAN-P MODULE

The conventional ELAN module enables the network to learn more features and be more robust by controlling the shortest and longest gradient paths. The structure is shown in Figure 4.

The ELAN module reaches a steady state when processing large-scale data or performing large-scale computations, regardless of the gradient path length and the number of computational modules. However, if more computational modules are stacked indefinitely, this stable state may be destroyed, reducing parameter utilization. The ELAN-P

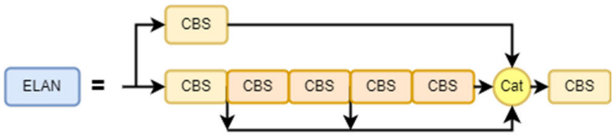


FIGURE 4. Structure diagram of the ELAN module.

proposed in this paper introduces the PConv convolution in the FasterNet network [23] to reduce the network's computation and improve the network's computational efficiency without destroying the original gradient path.

PConv is used to apply regular convolution to a single subset of input channels to extract spatial features and keep the remaining channels unchanged. This partial convolution approach reduces the sum of computational redundancy and memory accesses, thus improving detection speed. In YOLOv7, there is a certain amount of redundant computation in its neural network structure, which results in more floating point operations (FLOPs), thus increasing the latency time of the model. Equation (1) below reveals the relationship between latency time, FLOPs, and FLOPS:

$$Latency = \frac{FLOPs}{FLOPS} \tag{1}$$

Here, FLOPs represent the total number of floating point operations, and FLOPS represents the number of floating point operations per second. The ratio of FLOPs to FLOPS is a measure of computational latency. The FasterNet network increases FLOPS at the same time by effectively reducing the FLOPs, and this approach reduces the latency time and improves the computation speed, as seen from equation (1).

Depthwise Convolution (DWConv) is a commonly used method for convolutional optimization of backbone networks. Unlike conventional convolution, DWConv assigns a convolution kernel to each channel so that each channel is convolved by only one convolution kernel, effectively reducing redundant computations and FLOPs.However, DWConv cannot simply replace conventional convolution, which may lead to degradation of network accuracy.

Typically, DWConv is followed by Pointwise Convolution (PWConv) to improve accuracy. With the structure of this network combination, to compensate for the loss of accuracy caused by DWConv, the number of channels of DWConv needs to be increased from c to c', which is more than the number of channels c for regular convolution. However, this increases the number of memory accesses, which increases the latency time and decreases the overall computational speed. The memory accesses for DWConv are shown in Equation (2), where h and w represent the length and width of the image, respectively, c represents the number of channels, and k represents the convolution kernel size:

$$h \times w \times 2c' + k^2 \times c' \approx h \times w \times 2c' \tag{2}$$

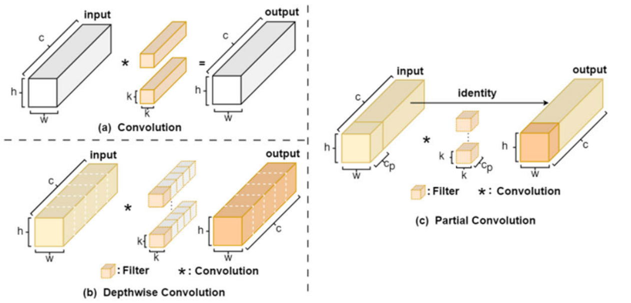


FIGURE 5. Comparison of ordinary convolution, DWConv, and PConv convolution: (a) Structural diagram of ordinary convolution, (b) Structural diagram of DWConv convolution, (c) Structural diagram of PConv convolution.

The memory access formula for regular convolution is as follows:

$$h \times w \times 2c + k^2 \times c \approx h \times w \times 2c \tag{3}$$

We found that when $c' > c$, it is evident that the memory access times of DWConv are higher than those of the regular convolution. A novel Partial Convolution (PConv) was proposed in FasterNet as a competitive alternative capable of reducing the computational redundancy and the number of memory accesses. The design of PConv is shown in Figure 5.

In contrast to regular convolution and DWConv, PConv in FasterNet requires only regular convolution to be applied to a portion of the input channels to extract spatial features, leaving the remaining channels unchanged. If the feature map is stored continuously or periodically in memory, the first or last consecutive channel represents the whole feature map. The FLOPs of Pconv are shown in Equation (4):

$$h \times w \times k^2 \times c_p^2 \tag{4}$$

In the general case of ratio $r = \frac{c_p}{c} = \frac{1}{4}$, the FLOPs of Pconv are only $\frac{1}{16}$ of the conventional convolution, achieving a significant reduction in FLOPs. In addition to this, Pconv also has a significant reduction in memory accesses compared to the regular convolution, which is shown in Equation (5):

$$h \times w \times 2c_p + k^2 \times c_p^2 \approx h \times w \times 2c_p \tag{5}$$

When $r = \frac{1}{4}$, Pconv has only $\frac{1}{4}$ of the memory access of regular convolution.

PConv enables neural network models to pursue higher FLOPS while reducing the number of parameters and increasing the FPS. Based on PConv, we constructed ELAN-P modules. Each ELAN-P module consists of three CBS modules and four Pconv modules, and the structure diagram of the whole module is shown in Figure 6.

The ELAN-P module is similar in structure to the conventional ELAN module, with two branches. The first branch passes through a 1 × 1 convolution module to change the number of channels. The second branch changes the number of channels first by a 1 × 1 convolutional module and then

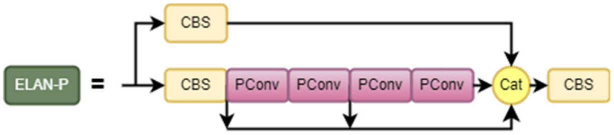


FIGURE 6. Structure diagram of the ELAN-P.

by four 3 × 3 PConv convolutional modules for the feature extractor. By replacing the original convolution module with four 3 × 3 PConv convolutions, the computation of the entire module is greatly reduced, resulting in a more efficient computation. Finally, the final feature extractor results were obtained by superimposing the four features. Using the ELAN-P module instead of the ELAN module in YOLOv7, we achieve a more efficient spatial feature extractor due to the introduction of the PConv convolution, which allows for a reduction in the amount of computation in the network and an increase in the computational efficiency of the network while keeping the original gradient paths intact. We expect that this improvement will reduce redundant computations and memory accesses, significantly reducing FLOPs while boosting FLOPS.

2) INTRODUCTION OF P6 DETECTION LAYER

In this study, we implemented a significant improvement and optimization measure for the YOLOv7 network model, i.e., a new P6 detection layer was added to the Head part of the original model, which extended the sensory field of the model, improved the network's ability to extract features from the original image, and enhanced the recognition and detection of multi-scale targets. Its structure is shown in Figure 7.

The traditional Head module predicts the objectness, class, and box components mainly by taking the three detection layers P3, P4, and P5 output from the Neck part and adjusting them by the number of RepConv channels, followed by a 1 × 1 convolution. The introduction of the P6 detection layer significantly expands the analytical scope of the network, allowing the model to more effectively capture and understand the information about large-scale blurred features in the image. It facilitates better capture and utilization of high-level semantic information by performing feature extraction and information integration at higher layers of the network.

The specific implementation of the new P6 detection layer is as follows:

Firstly, the image is processed by Backbone to output three feature maps, whose resolutions are C3, C4, and C5, from largest to smallest. Next, the network will process C5 by reducing its channel number from 1024 to 512 through the SPPCSPC module and adjusting the resolution of C5 to the size of C4 and C3 through upsampling, followed by feature fusion to get the fused D4 and D3 feature map. Among them, D3 is first adjusted for the number of channels by RepConv, and then 1 × 1 convolution is used to predict the three parts of objectness, class, and bbox, which finally forms the P3 detection layer. Subsequently, D3 was adjusted to the resolution to the size of D4 and D5 by downsampling operation and then fused with them to obtain the fused M4 and M5 feature maps.M4 and M5 were adjusted to the number of channels by RepConv and predicted using 1 × 1 convolution to form the P4 and P5 detection layers, respectively. Finally, we merge the D3 feature map with the M5 feature map following a downsampling process to form the M6 feature map. This is then subjected to channel adjustment via RepConv and 1 × 1 convolution before prediction, culminating in the formation of the P6 detection layer.

The P6 detection layer fused different levels of semantic information to enable the model to more clearly recognize ambiguous features in complex backgrounds.

3) LOSS FUNCTION IMPROVEMENT

Object Detection is one of the core problems of computer vision, and its effectiveness depends greatly on the loss function used [24]. In our proposed JutePest-YOLO model, we noticed that the accuracy of Jute mite species pest detection is low, and the targets of this species occupy fewer pixel points in the image than fewer targets are small. The traditional YOLOv7 algorithm should be more effective for jute mite pest detection, with leakage and false detection occurring mainly in the case of small target detection and background blurring. Furthermore, improving the loss function is the key to improving the accuracy of small target detection.

Many current target detection algorithms use the Intersection of Union (IoU) as the loss function because the intersection ratio can represent the error between the prediction frame and the real frame, directly affecting the prediction effect. The higher the value of the loss function, the higher the direct error between the prediction and real frames. In traditional IoU calculations, the IoU values of the predicted and actual bounding boxes are calculated by the ratio of their intersection area to the total area. However, this traditional approach sometimes leads to sub-optimal results. For example, smaller targets are given less weight in the IoU calculation due to a smaller pixel base, which may cause the model to ignore these smaller targets due to bias.

The loss function used for the original YOLOv7 network is as follows:

$$loss = loss_{ioc} + loss_{conf} + loss_{cls} \tag{6}$$

where, $loss_{ioc}$, $loss_{conf}$, and $loss_{cls}$ represent the localization loss, confidence loss, and classification loss, respectively. Among them, the confidence loss and classification loss are calculated using the cross-entropy loss function, and the localization loss is calculated using the CIoU loss function, which is shown in Equation (7):

$$\begin{aligned} \mathscr{L}_{CIoU} = 1 - IoU + \frac{\rho^2\left(b, b^{gt}\right)}{(c_w)^2 + (c_h)^2} \\ + \frac{4}{\pi^2}\left(tan^{-1}\frac{w^{gt}}{h^{gt}} - tan^{-1}\frac{w}{h}\right) \end{aligned} \tag{7}$$

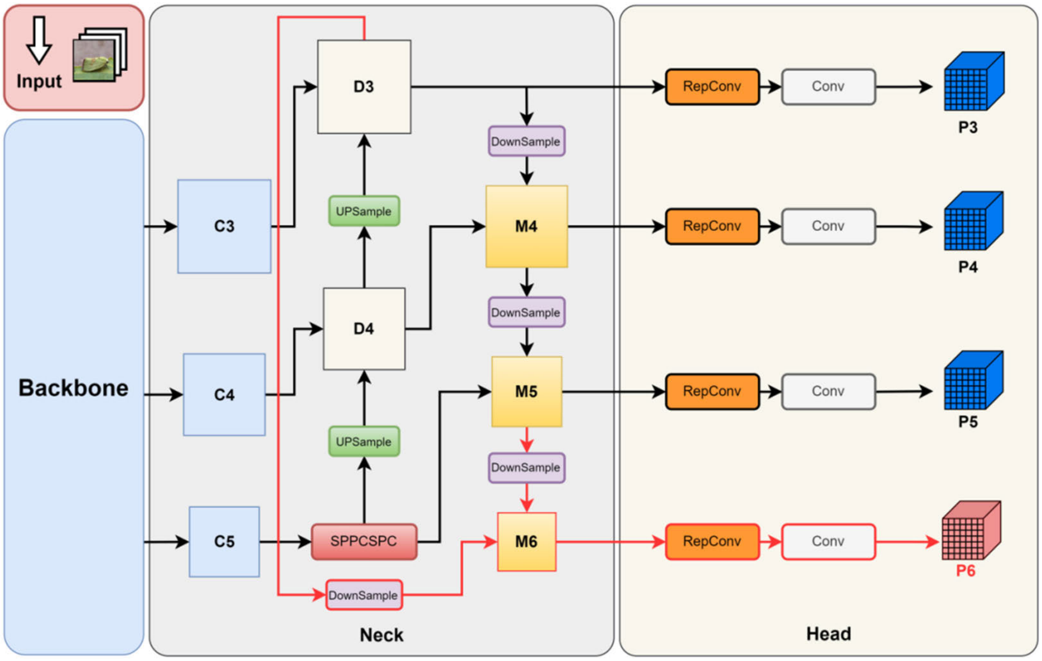


FIGURE 7. Structure of the neck and head sections with the addition of the P6 detection layer.

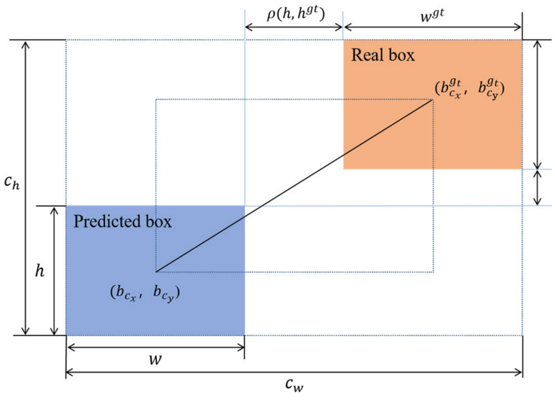


FIGURE 8. Schematic diagram of the CIoU loss function.

In Equation (7), IoU denotes the intersection ratio of the predicted and real boxes. Some of the remaining parameters involved are shown in Figure 8. ρ represents the Euclidean distance between the center of the predicted bounding box and the center of the actual bounding box, where b is the coordinate of the center of the predicted bounding box, and b^{gt} is the coordinate of the center of the actual bounding box. The terms c_w and c_h denote the width and height of the minimum enclosing rectangle (i.e., the smallest common external rectangle) of the predicted and actual bounding boxes. The w^{gt} and h^{gt} are the width and height of the actual bounding box, while w and h are the width and height of the predicted bounding box.

The CIoU loss function considers the overlap between the predicted and real frames. It introduces a penalty term for the distance between the center point of the predicted and real frames and the aspect ratio to optimize the loss function further. However, CIoU does not consider that after using the aspect ratio as a penalty factor in the loss function, if the real frame and the predicted frame have the same aspect ratio but different values of width and height, then the penalty term cannot reflect the real difference between these two frames.

Therefore, in this study, we replace the CIoU loss with the WIoU v3 loss. WIoU v3 loss places greater emphasis on the aspect ratio of bounding boxes, center distance, and overlap area. It introduces a dynamic, non-monotonic focusing mechanism and devises a rational gradient gain allocation strategy. This reduces the occurrence of large or detrimental gradients from extreme samples, enhancing the model's performance in detecting targets of varying sizes and effectively reducing false negatives and false positives. Tong et al. [22] introduced three versions of WIoU. WIoU v1 is based on attention-driven bounding box loss, while WIoU v2 and WIoU v3 incorporate a focusing coefficient through the construction of gradient gains and algorithmic methods.

WIoU v1 introduced distance as a metric of attention. Reducing the penalty of the geometric metric when the object frame and prediction frame overlap within a certain range gives the model a better generalization ability. The formulas for calculating WIoU v1 are shown in Equation (8) and Equation (9):

$$\begin{aligned}\mathcal{L}_{WIoUv1} &= \mathcal{R}_{WIoU}\mathcal{L}_{IoU}\\ &= exp\left(\frac{\left(x-x_{gt}\right)^2+\left(y-y_{gt}\right)^2}{\left(W_g^2+H_g^2\right)^*}\right)\mathcal{L}_{IoU}\end{aligned} \tag{8}$$

$$\mathcal{L}_{IoU} = 1 - IoU \tag{9}$$

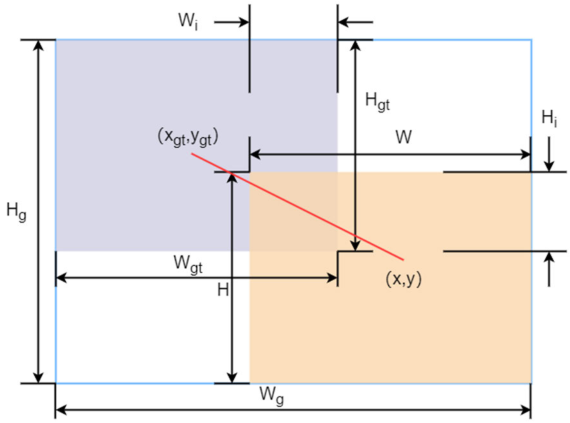


FIGURE 9. Schematic diagram of the WIoU loss function.

The weight of simple samples in the loss values is effectively reduced by constructing monotonic focusing coefficients $\mathcal{L}_{\mathrm{IoU}}^{*}$ and applying WIoU v2 to WIoU v1. Considering that $\mathcal{L}_{\mathrm{IoU}}^{*}$ decreases during the model training period as the of $\mathcal{L}_{\mathrm{IoU}}^{*}$ decreases, which leads to slower convergence, the average value of $\mathcal{L}_{\mathrm{IoU}}$ is introduced to normalize $\mathcal{L}_{\mathrm{IoU}}^{*}$. The formula for WIoU v2 is shown in Equation (10):

$$\mathscr{L}_{IoU} = 1 - IoU \tag{10}$$

where γ is a hyperparameter.

WIoU v3 defines the outlier β to measure the quality of the anchor frame, constructs the non-monotonic focusing factor γ based on β, and applies r to WIoU v1. The WIoU v3 equations are shown in Equation (11) to Equation (13):

$$\mathscr{L}_{WIoUv3} = \gamma \times \mathscr{L}_{WIoUv1} \tag{11}$$

$$\gamma = \frac{\beta}{\delta \alpha^{\beta-\delta}} \tag{12}$$

$$\beta = \frac{\mathscr{L}_{IoU}^{*}}{\mathscr{L}_{IoU}} \in [0, +\infty) \tag{13}$$

β denotes the degree of abnormality of the prediction frame, and a smaller degree implies a higher quality of the anchor frame. Therefore, using β to construct the number of non-monotonic focuses can assign smaller gradient gains to the prediction frames with larger anomalies, effectively reducing the harmful gradients of low-quality training samples; α and δ are hyperparameters. The meanings of the other parameters are shown in Figure 9. x_p and y_p denote the coordinate values of the prediction box, while x_{gt} and y_{gt} denote the coordinate values of the Ground Truth. The corresponding H and W values denote the width and height of the two boxes, respectively.

IV. EXPERIMENTAL RESULTS AND ANALYSIS

A. DATASET

1) DATASET CONSTRUCTION

In the field of target detection, model training using a single dataset containing multiple categories is usually considered to improve the recognition accuracy and training efficiency of the model, but we base on the application scenario of the jute pest recognition task, considering the diversity of pest species, the model trained using a single dataset is less generalizable and does not perform as well as the model trained with multiple category datasets in recognizing unseen or similar species. Therefore, to address the problem of missing datasets for multiple species of jute pests, the other part of the dataset we obtained from Baidu image library and Google image library. Through these sources, we collected images of nine types of pests that seriously damage jute plants, including Black hairy, Field cricket, Indigo caterpillar, Jute semilooper, Jute stem girdler, Jute stem weevil, Leaf beetle, Pod borer, and Yellow mite images. Some sample images will be shown in Figure 10.

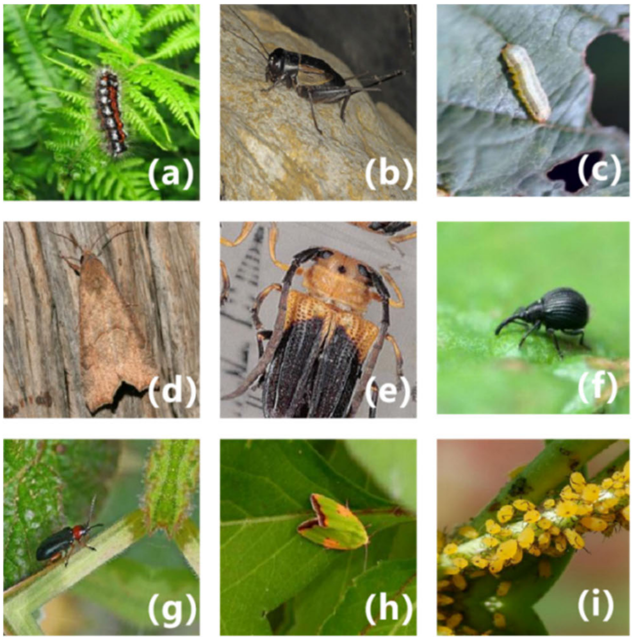


FIGURE 10. Some sample images from our jute diseases and pests data. (a) Black hairy. (b) Field cricket. (c) Indigo caterpillar. (d) Jute semilooper. (e) Jute stem girdler. (f) Jute stem weevil. (g) Leaf beetle. (h) Pod borer. (i) Yellow mite.

2) DATASET PREPROCESSING

To increase the training volume of the network model and to prevent overfitting and low model generalization ability during model training, we expanded the jute pest dataset based on the original data in this study using data augmentation methods using content and geometric transformations. Geometric transformations modify image properties without changing the image content, such as image RandomCrop, horizontal flip, and translation rotation, etc., which are designed to simulate the appearance of pests under different viewpoints and locations to meet the challenge of target localization in complex backgrounds. Content transformations include color dithering, Gaussian blurring, etc. These transformations are used to simulate pest images under different lighting and environmental conditions to increase the model's adaptability

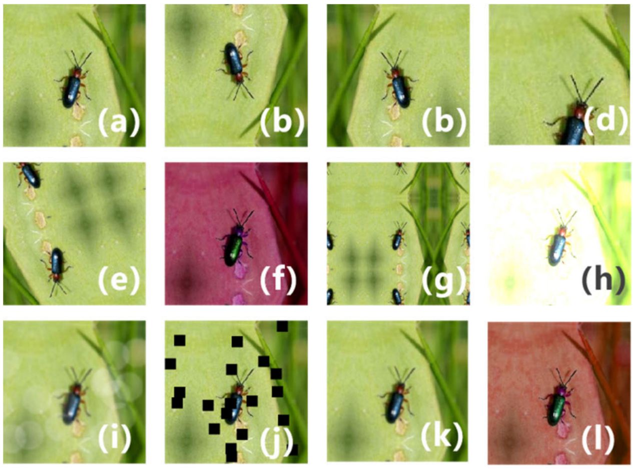


FIGURE 11. Image samples in the data augmentation. (a) Original Image. (b) VerticalFlip. (c) HorizontalFlip. (d) RandomCrop. (e) ShiftScaleRotate. (f) HueSaturationValue. (g) PadIfNeeded. (h) RandomBrightnessContrast. (i) RandomFog. (j) Cutout. (k) GaussianBlur. (l) ColorJitter.

to environmental changes, and certain transformations (e.g., Random Fogging, Gaussian Blurring) provide different textures and noise levels, which are essential to improve the model's robustness to the variations in image quality that may be encountered in real-world applications.

In summary, we included these data enhancement steps aimed at comprehensively improving the model's ability to cope with diverse environments, as detailed in Figure 11. After data enhancement, we expanded the entire dataset to 3252 jute pest images. To reduce the impact of dataset division on the experiment, this study adopts a random division method, dividing the augmented dataset into a training set and validation set according to the ratio of 8:2. Subsequently, we annotated each image in the dataset with "LabelImg" software to mark the real bounding box of the pests. All image data sizes were standardized at the initial stage of the network and fixed to a uniform resolution of 640 × 640. In Table 1, we give the details of the enhanced dataset of jute pests.

B. EXPERIMENTAL ENVIRONMENT AND PARAMETERS

In this paper, experiments were conducted on a homemade jute pest dataset with the following model experimental conditions: an Ubuntu server with a CPU of Xeon E5-2620 v4, 24 GB of RAM, and a GPU of NVIDIA GeForce RTX3090 with 24G of video memory. Python version 3.8, PyTorch version 1.13.0, and CUDA 11.7 are the programming environments. The initial learning rate for network training is set to 0.01, and the Adam optimizer is used to update the network parameters with a batch size of 8, a weight decay coefficient of 0.0005, a momentum of 0.937, and an Epoch of 500. To save time, the model is trained on the server and subsequently validated locally. The detailed environment configuration is shown in Table 2, and the training parameter settings are shown in Table 3.

TABLE 1. Jute pests dataset information.

Number	Pest Category	Number of Images in Training Set	Number of Images in Validation Set
P-1	Black hairy	263	66
P-2	Field cricket	302	76
P-3	Indigo caterpillar	370	93
P-4	Jute semilooper	312	79
P-5	Jute stem girdler	240	61
P-6	Jute stem weevil	268	67
P-7	Leaf beetle	252	64
P-8	Pod borer	324	81
P-9	Yellow mite	267	67

TABLE 2. Environment configuration.

Configuration		Local Configuration	Server Configuration
Hardware	CPU	Inter Core i7 12700h	Xeon E5-2620 v4
	GPU	GeForce RTX3060	GeForce RTX3090
	RAM	16GB	24GB
Software	System	Win11	Ubuntu 22.04
	Python	3.9	3.8
Environment	Pytorch	1.8	1.8
	CUDA	11.6	11.7
	Cudnn	8.6	8.4

TABLE 3. Parameter settings.

Parameter	Values
Base learning rate	0.01
End learning rate	0.1
Momentum	0.937
Batch size	8
Learning rate policy	Adam
Dropout	0.0005
Epoch	500

In this study, the change curve of the loss function during model training is shown in Figure 12, which shows that the improved JutePest-YOLO model in this paper is closer to the global optimum. In the early stage of model training (Epoch 1-100), the loss value decreases rapidly and shows a clear convergence trend. The model adapts quickly to the training data at this stage, and the loss value decreases significantly. In the next training process (Epoch 100-400), the decline of the loss function gradually slows down and shows a smooth trend. It indicates that the model is approaching the convergence point and learning the main features of the data. The decelerating decline at this stage indicates that the model's parameter tuning is more subtle, and more training

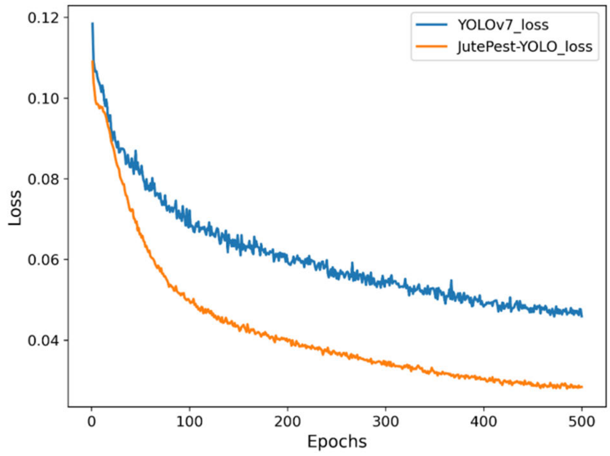


FIGURE 12. Loss function curve during model training.

iterations are needed to refine the model's performance. In the final stages of training (Epoch 400-500), the trend in the loss function indicates that the model has approached stability. At this point, the parameters of the model are nearing their optimal state, and the performance of the model has reached convergence.

C. PERFORMANCE METRICS

For our jute disease dataset, each detected bounding box can be classified into four cases, i.e., True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN), and Precision and Recall can be used to classify the results of the above four classifications for comprehensive evaluation. The F1 value represents the reconciled average of Precision and Recall, which provides a single metric when dealing with data imbalance problems, enabling the simultaneous consideration of model precision and recall. Its calculation principle is shown in Eqs. (14)-(16).

$$Precision = \frac{TP}{TP + FP} \tag{14}$$

$$Recall = \frac{TP}{TP + FN} \tag{15}$$

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{16}$$

This paper also evaluates the model's detection accuracy using mAP@0.5 and mAP@0.5:0.95 and measures the model's computational complexity using GFLOPs. Here, mAP (mean average precision) assesses the model's prediction accuracy at various recall levels through different IoU thresholds, reflecting the model's ability in localization and classification detection.mAP@0.5 and mAP@0.95 are calculated at IoU thresholds of 0.5 and 0.95, respectively. On the other hand, mAP@0.5:0.95 is calculated by averaging over the range of IoU thresholds from 0.5 to 0.95 with a step size of 0.05. This evaluation criterion is more stringent and can demonstrate the performance variation of the model at different IoU thresholds. The calculation formulas are shown in Equations (17)-(19).

$$AP = \int_0^1 P(r)\, dr \tag{17}$$

$$mAP = \frac{1}{m} \sum \left[\frac{1}{n} \sum P(r) \right] \tag{18}$$

$$mAP@0.5:0.95 = \frac{1}{10} \sum_{r=0.5}^{0.95} mAP@r \tag{19}$$

where m denotes the number of categories for classification, n denotes the number of targets predicted in a single category, and P(r) denotes the precision value when the recall is r. mAP@r denotes the mean mAP value at a specific IoU threshold r.

D. COMPARISON WITH BASELINE MODEL

To demonstrate the improvement effect of the improved model on the detection performance, we conducted a comparison experiment between the improved model and the baseline model YOLOv7. Table 4 shows the results of the detection metrics of the improved model and YOLOv7. Figure 13 shows the change curve of the detection metrics. The results indicate that compared to YOLOv7, the improved JutePest-YOLO model shows an improvement of 3.45% in Precision and 1.76% in Recall. It achieves a mAP@0.5 of 95.68% and mAP@0.5:0.95 of 67.11%, representing increases of 2.24% and 3.25%, respectively, compared to YOLOv7. The GFLOPs decreased from 105.3 to 88.4, a reduction of 16.05%. The F1 score increased from 94.19 to 96.77%, showing an overall improvement of 2.58%. The accuracy is improved in all categories, especially in the P9 category, by 12.6%, which proves the effectiveness of the redesign of the detection head in our improvement strategy, which allows the model to better capture and understand the large-scale fuzzy feature information in the image and further improves the accuracy of target detection. The results show that the improved model has better detection performance in pest target identification.

Gradient-weighted Class Activation Map (Grad-CAM) [25] is now one of the most commonly adopted techniques in computer vision, aiming to visualize the convolutional feature maps in deep neural networks and generate heat map, which in turn can identify the region of interest of the model more accurately. The heatmap visually and easily reflects which areas of the feature map the model focuses on. Figure 14 demonstrates the difference between the improved and baseline models in terms of focusing on regions of interest for specific target categories. This difference further corroborates the effectiveness of our proposed improved model JutePest-YOLO in detecting non-significant targets. We acquired the Grad-CAMs for both the YOLOv7 and JutePest-YOLO models and visualized the detection effectiveness on nine categories of jute pest damage using heat maps generated by Grad-CAM. Compared to the baseline YOLOv7 model, our JutePest-YOLO model demonstrates an enhanced focus on relevant information, particularly in augmenting the perception of non-prominent objects. This

TABLE 4. Comparison of the proposed improved model and YOLOv7 detection accuracy. (The bold data in the table indicate the best results.)

Model	Precision/%	Recall/%	F1	mAP@0.5/%	mAP@0.5:0.95%	GFLOPs
YOLOv7	95.27	93.14	94.19	93.26	63.86	105.3
JutePest-YOLO(ours)	**98.72**	**94.9**	**96.77**	**95.68**	**67.11**	**88.4**

Model	AP/%								
	P1	P2	P3	P4	P5	P6	P7	P8	P9
YOLOv7	96.4	98.4	99.2	97.6	98.4	97.8	98.5	97.5	54.9
JutePest-YOLO(ours)	**98.9**	**99.5**	**99.5**	**99.6**	**99.5**	**98.9**	**99.5**	**99.8**	**67.5**

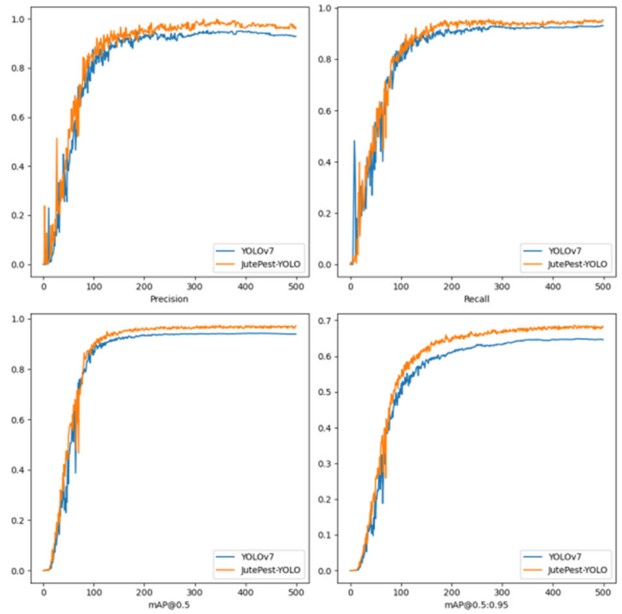


FIGURE 13. Comparison of model detection index change curves.

clearly indicates its superior performance, further confirming our model's effectiveness in addressing issues related to pest background modeling and the prevalence of small targets.

E. DIFFERENT LOSS FUNCTION COMPARISON

In the experiments of training the JutePest-YOLO network for jute pest detection, to verify the superiority of introducing WIoU v1, we conducted comparative experiments using WIoU v1 and several mainstream loss functions for JutePest-YOLO network respectively, while keeping other training conditions consistent. Table 5 demonstrates the experimental results, while Figure 15 compares the Precision, Recall, F1 score, and mAP@0.5和mAP@0.5:0.95 under different loss functions.

The experimental data show that the model achieves the best mAP performance when WIoU v3 is used as the bounding box regression loss function, which is 1.13% higher than using the WIoU v1 loss function, and 1.46% higher than the default CIoU loss function, reaching a maximum of 95.68%. the F1 score improves by 1.75%, from 95.04 to 96.79%.Moreover, as shown in Figure 15, the JutePest-YOLO model using the WIoU v1 loss function outperforms other loss functions in terms of recall rate and mAP@0.5:0.95. Therefore, we believe that introducing the WIoU v3 loss function as the bounding box loss function for the JutePest-YOLO model is an optimal choice.

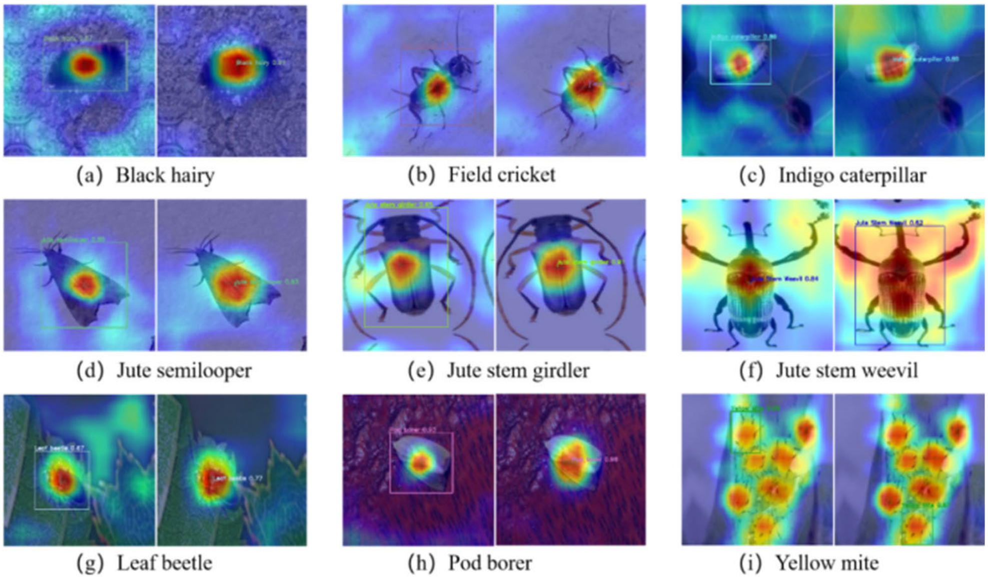
F. ABLATION STUDY

To verify the effectiveness of the various improvement strategies of the JutePest-YOLO model proposed in this paper, we designed an ablation study on the jute pest dataset in this paper. The experiments were divided into six groups, and their results are displayed in Table 6. Group 1 is the experimental results of the original model YOLOv7, and Groups 2 to 4 are the results after adding only one improvement method at a time to the original model, respectively, to verify the effectiveness of each improvement method to the original algorithm. Group 5 is the experimental results after adding two improvement methods, and Group 6 is based on the finally obtained improved algorithm JutePest-YOLO.

As shown in Table 6, the first group represents the original YOLOv7 model without the inclusion of any improvement modules, achieving accuracy and mAP@0.5 of only 95.27% and 93.26%, respectively. In comparison to the original model, all models incorporating the three improvement methods have demonstrated enhanced detection performance. The analysis of the experimental results is as follows:

In the second experimental group, the original model was augmented by introducing the WIoU v3 loss function. WIoU v3, by incorporating a dynamic, non-monotonic focusing mechanism, effectively reduces the occurrence of large or detrimental gradients from extreme samples. This enhancement resulted in an increase of 1.31% in mAP@.5 and 1.29% in mAP@.5:.95.

In the third set of experiments, the addition of the P6 detection layer enabled the model to more effectively capture large-scale, blurred feature information in complex background images. Consequently, this improvement led to a 2.22% increase in accuracy and a 1.43% increase



(a) Black hairy
(b) Field cricket
(c) Indigo caterpillar
(d) Jute semilooper
(e) Jute stem girdler
(f) Jute stem weevil
(g) Leaf beetle
(h) Pod borer
(i) Yellow mite

FIGURE 14. Heatmaps of different models on all categories. (YOLOv7 on the left, JutePest-YOLO on the right.)

TABLE 5. Comparison of detection results for different loss functions introduced by JutePest-YOLO.

IoU Loss Function	Precision/%	Recall/%	F1	mAP@0.5/%	mAP@0.5:0.95%
CIoU	97.29	92.90	95.04	94.22	66.32
GIoU[26]	95.62	92.22	93.88	93.40	64.76
DIoU[27]	95.41	91.89	93.61	93.88	65.08
EIoU[28]	95.77	91.80	93.74	93.72	64.53
SIoU[29]	95.34	90.12	92.65	94.03	65.92
NWD[30]	95.42	93.84	94.62	94.56	66.13
WIoU v1	97.87	90.18	93.86	94.55	65.57
WIoU v2	**98.73**	90.77	94.56	94.02	66.41
WIoU v3	98.72	**94.95**	**96.79**	**95.68**	**67.11**

TABLE 6. Comparison of ablation experiments of each module in JutePest-YOLO model, √ indicates that this improved strategy was used.

Group	Baseline	ELAN-P	P6 detection layer	WIou v3	Precision	Recall	F1	mAP@.5	mAP@.5:.95	GFLOPs
1	√				95.27	93.14	94.19	93.26	63.86	105.3
2	√			√	95.64	93.31	94.46	94.57	65.15	105.3
3	√		√		97.49	93.95	95.68	94.69	**67.53**	108.7
4	√	√			96.69	93.78	95.21	94.42	64.74	85.0
5	√	√	√		97.73	94.31	95.99	95.22	67.22	88.4
6	√	√	√	√	**98.72**	**94.90**	**96.77**	**95.68**	67.11	**88.4**

in mAP@.5, while mAP@.5:.95 was enhanced by 3.67%, reaching 67.53%.

Group 4 experiments improved the ELAN module of the original YOLOv7 model, and the new ELAN-P module introduced a more efficient PConv in the original module. After using the ELAN-P module, the model can effectively reduce redundant computations and memory accesses and significantly reduce the FLOPs so that the GFLOPs are reduced from 105.3 to 85.0, which is a reduction of 19.3%.

In the fifth group of experiments, the P6 detection layer was introduced on the basis of the fourth group. Compared to the original model, this resulted in a 16.05% reduction in GFLOPs, while Precision and mAP@.5 were enhanced by 2.46% and 1.96%, respectively.

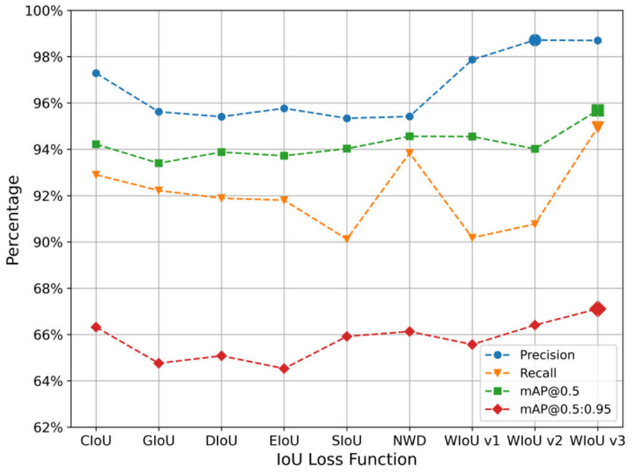


FIGURE 15. Comparison of Precision, Recall, mAP@0.5, and mAP@0.5:0.95 under different loss functions.

The sixth experimental group integrated all the improvement methods, resulting in the proposed JutePest-YOLO model. Compared to the original model, the improved JutePest-YOLO model showed enhancements in Precision, Recall, F1 score, mAP@0.5, and mAP@0.5:0.95 by 3.45%, 1.76%, 2.58%, 2.24%, and 3.25%, respectively. The overall model's GFLOPs decreased by 16.05%.

The experimental results show that the improvement strategies proposed in this paper are effective. The improved model not only enhances the accuracy but also optimizes for small target detection and blurred pest features and significantly improves the operation efficiency of the model so that the model can achieve the optimal comprehensive performance in the task of jute pest identification and detection. In addition, we demonstrated the model results of the six sets of experimental results by generating a heatmap via Grade-CAM. Figure 16 demonstrates the results of the heatmap.

The darker the color of the heatmap, the more obvious the target area is, and the more localized it is, the more important feature areas are highlighted. From the figure, we can see that the regions of interest of the heatmaps generated by the models after adding the improved methods are all enlarged, especially the JutePest-YOLO model after introducing all the methods, which can highlight the important regions in the influence more clearly, and once again proves that the overall detection performance of the improved models is better.

G. COMPARATIVE EXPERIMENTS

To demonstrate the superiority and effectiveness of the JutePest-YOLO network with better detection performance in jute pest detection, we conducted comparison experiments between the improved model, the classical model, and the recently released model in this paper dataset. The comparison results are shown in Table 7, and the comparison of metrics of different models is shown in Figure 17.

The experimental results show that the JutePest-YOLO model achieved 98.7% on the Precision metric, the highest of all the models listed. The closest of these is the YOLOv7x model, but its Precision of 96.9% is still lower than the JutePest-YOLO model. This implies that the JutePest-YOLO model performs well in reducing the number of incorrect positive examples and has a strong accuracy. Regarding the Recall metric, the JutePest-YOLO model achieves 94.9%, second only to the YOLO-JD model's 95.0%. The high Recall indicates that the JutePest-YOLO model can identify the target object well and reduce missed detection. The F1 score is the reconciled mean of Precision and Recall, and the JutePest-YOLO model got 96.7% on this metric, the highest of all models, showing that the model has a good balance between Precision and Recall. In terms of mAP@0.5, the JutePest-YOLO model outperforms all other models with a score of 95.6%, highlighting its ability to maintain high detection accuracy at higher IoU thresholds. On the mAP@0.5:0.95 metric, the JutePest-YOLO model scores 67.1%. While it may not be the highest among all models, it still surpasses the majority, such as YOLOX, YOLO-JD, and CAP-YOLOv7. This indicates that the detection performance of the JutePest-YOLO model remains relatively stable across different IoU thresholds.

In summary, The JutePest-YOLO model demonstrates strong advantages across almost all evaluation metrics, particularly in Precision, F1 score, and mAP@0.5. This highlights the effectiveness of the proposed improvement strategies, demonstrating their ability to enhance the model's recognition capabilities for complex backgrounds and targets of different scales. Additionally, its performance on the mAP@0.5:0.95 metric is commendable, showcasing stability across different IoU thresholds. This set of comparative experiments fully demonstrates the superiority of the JutePest-YOLO model, emphasizing its practical value in real-world applications.

H. GENERALIZATION STUDIES

To validate the generality and performance of our proposed JutePest-YOLO model on different datasets, we conducted a Generalisation experiment on another jute pest dataset and compared our model with other mainstream target detection models. This dataset is from the dataset used in the paper of Sourav et al. [9], which contains images of four categories of jute pests (the specific categories are Field cricket, Spilosoma Obliqu, Jute stem weevil, and Yellow mite). The names of the categories are denoted by D1, D2, D3, and D4, respectively. The experimental results are shown in Table 8 below.

From the experimental results, it is obvious that our JutePest-YOLO model achieves 97.1% in Precision, which is significantly better than other models, which indicates that our model can accurately identify jute pest targets and reduces the possibility of misdetection. Meanwhile, the JutePest-YOLO model also achieved excellent performance in Recall and F1 scores, reaching 93.4% and 95.21%, respectively, which verified the superiority and generalization ability of the model. In each category's average

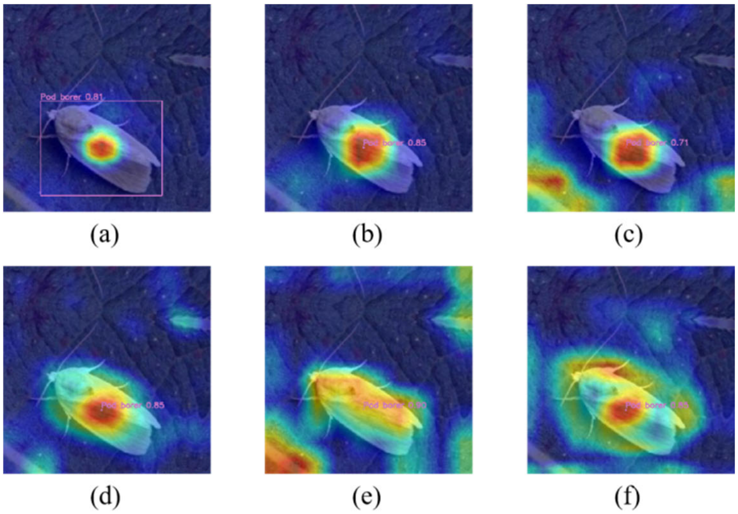


FIGURE 16. Comparison of heat maps after adding different methods. (a-f correspond to 6 groups of experiments. For example, (a) is the baseline model in the first set of experiments).

TABLE 7. Analysis of the experimental effects of the different models.

Model	Precision	Recall	F1	mAP@.5/%	mAP@.5:.95%
RetinaNet[31]	87.9	86.3	87.1	85.4	57.6
YOLOv5s	94.8	94.2	94.4	93.7	64.6
YOLOX[32]	95.2	93.1	94.1	92.4	61.7
YOLOv7x	96.9	92.2	94.4	94.8	64.2
YOLOv8s	93.0	93.3	93.1	94.6	**68.5**
YOLO-JD[15]	95.2	**95.0**	95.2	92.9	62.6
CPA-YOLOv7[33]	96.2	93.5	94.8	93.6	64.9
UVA-YOLOv8s[34]	94.5	92.9	93.6	93.1	67.2
YOLOv7	95.2	93.1	94.1	93.2	63.8
JutePest-YOLO(ours)	**98.7**	94.9	**96.7**	**95.6**	67.1

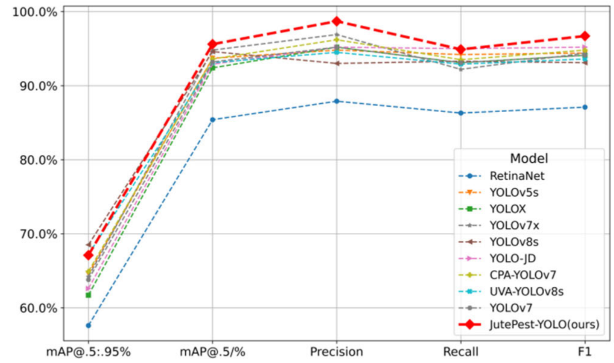


FIGURE 17. Comparison of detection performance of different models.

precision (AP) evaluation, the JutePest-YOLO model achieves excellent detection results on all four categories, especially on D1, D2, and D3, where the AP values exceed 98.5%. This indicates that the JutePest-YOLO model can handle the multi-category target detection task well with strong generalization ability.

On the mAP@0.5 metric, our model achieved a score of 88.9%, demonstrating excellent performance. This further confirms that the JutePest-YOLO model excels not only in general object detection tasks but also maintains high-precision detection even at lower IoU thresholds. On the mAP@0.5:0.95 metric, our model achieved a score of 64.4%. Compared to other models, our model demonstrates higher advantages across almost all evaluation metrics. Particularly, when compared to the RetinaNet model, our model shows a significant improvement with a 15.5 percentage point increase in mAP@0.5 and a 21.8 percentage point increase in mAP@0.5:0.95. When compared to other YOLO series models, the JutePest-YOLO model also exhibits certain advantages, indicating significant improvements in our model enhancements.

TABLE 8. Generalization experiment.

Model	Precision	Recall	F1	AP/%				mAP@.5/%	mAP@.5:.95%
				D1	D2	D3	D4		
RetinaNet[31]	84.9	81.4	83.13	89.4	86.6	85.5	32.1	73.4	42.6
YOLOv5s	95.8	91.2	93.46	96.3	95.4	97.2	51.2	85.0	59.4
YOLOX[32]	96.4	**94.2**	95.28	98.0	97.8	98.2	52.3	86.5	54.6
YOLOv7x	96.1	92.4	94.23	97.9	97.2	94.3	55.5	86.2	56.7
YOLOv8s	95.6	93.3	94.43	99.1	98.0	99.1	57.4	88.4	61.3
YOLO-JD[15]	95.2	92.9	94.04	97.2	97.3	98.5	56.4	87.3	52.6
CPA-YOLOv7[33]	93.6	91.7	92.65	98.4	98.6	98.7	57.9	88.4	53.8
UVA-YOLOv8s[34]	94.3	92.4	93.34	98.5	98.2	98.9	**58.9**	88.6	59.8
YOLOv7	95.2	93.3	94.24	97.4	98.6	98.9	56.9	87.9	58.7
JutePest-YOLO(ours)	**97.1**	93.4	**95.21**	**99.5**	**98.6**	**99.3**	58.2	**88.9**	**64.4**

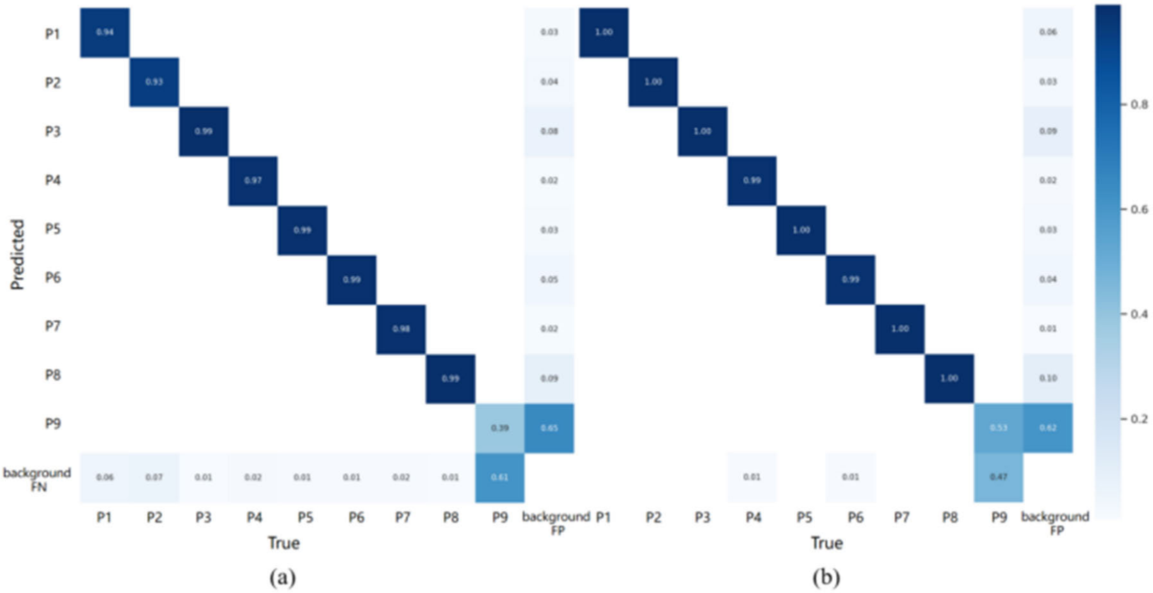


FIGURE 18. Comparison of confusion matrix results, (a) for YOLOv7, (b) for JutePest-YOLO.

In summary, as verified by the Generalisation experiment on the jute pest dataset, our JutePest-YOLO model achieves excellent performance in all evaluation metrics and has significant advantages over other mainstream target detection models, especially in terms of precision, recall, detection effect of various categories, and mAP metrics. These results fully demonstrate the generalization ability of our model and its wide applicability in practical applications.

I. VISUAL ANALYSIS

To show the detection effect of the proposed model in this study more intuitively, a confusion matrix was employed to compare the model's performance before and after improvements. In this experiment, the confusion matrix is primarily used to assess the performance of the JutePest-YOLO detection algorithm. Presented in a two-dimensional table format, the rows represent actual categories while the columns represent predicted categories. By calculating the prediction results across different categories, various metrics such as accuracy, recall rate, and false positive rate can be determined.

Darker colored blocks on the diagonal of the confusion matrix indicate high accuracy of the model's detection results; values on the off-diagonal represent misclassification, and these values should be as low as possible to show the model's high accuracy and low false alarm rate. It is evident that the YOLOv7 network has lighter color blocks on the diagonal of the confusion matrix for the category Yello mite with a Precision of 39% and shows color blocks for all categories on the FN and FP samples. This implies that

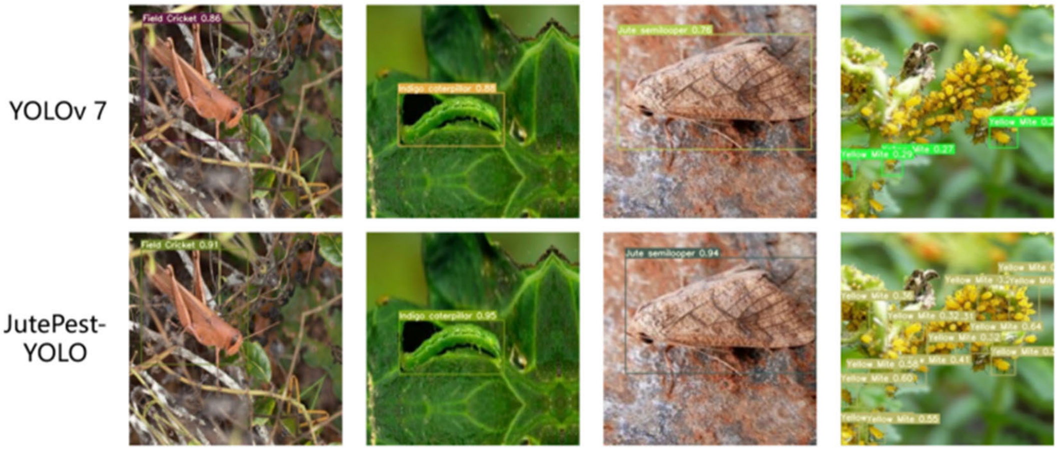


FIGURE 19. Detection results of YOLOv7.

the model has a certain error rate in detecting various categories of objects.By comparison, the confusion matrix of the JutePest-YOLO network exhibits a darker color on the diagonal for the P9 (Yello mite) category, indicating an accuracy of 53%. Meanwhile, it achieves a detection accuracy of 100% for most other categories. Additionally, only three categories show color blocks in the case of FN (False Negative) samples. Notably, P9 represents a typical example of small-target pest infestation. Therefore, based on the comparison of these confusion matrices, it can be concluded that the JutePest-YOLO model outperformed the original model in detecting objects of all categories. The results of the comparison of the confusion matrices are displayed in Figure 18.

To visually demonstrate the detection effect of our model, this study conducted inference experiments using YOLOv7 and JutePest-YOLO. We screened the images of the jute pest dataset for this experiment, and all categories tried to select images with complex image backgrounds and many small targets as the inference experiment data and compared the detection results of some categories of pests.

Figure 19 shows the comparative results of YOLOv7 and JutePest-YOLO models in detecting jute pests, respectively. It can be observed that YOLOv7 has a relatively poor detection performance, while JutePest-YOLO demonstrated the best detection performance. In the detection of (i) category Yello mite, YOLOv7 used three detection frames, and JutePest-YOLO used 14 detection frames, identifying a large number of visible Yello mite targets in the image. Overall, JutePest-YOLO was able to detect a wide range of jute pests quickly, accurately, and comprehensively, providing strong technical support for crop protection.

V. CONCLUSION

In this study, a JutePest-YOLO model for jute pest detection with high detection accuracy and good effect was proposed to solve the problems of feature ambiguity and misdetection and omission caused by the complex background and small target categories in the field of pest recognition and to satisfy the requirements of accuracy and effect of the target detection of the jute pest scene while considering the resource consumption. First, we replaced all the ELAN modules of the YOLOv7 model with the ELAN-P module, which was a module that replaced all the 3 × 3 regular convolutions in the ELAN module with PConv, where PConv applied regular convolutions to a single subset of the input channels as a way of extracting spatial features, which reduced the computational redundancy and memory accesses of the network while keeping the original gradient paths unchanged. Next, we added a new P6 detection layer, which extended the sensory field of the model and fused different levels of semantic information to enable the network to recognize fuzzy features in the background of the model more clearly. Finally, we introduced the WIoU v3 loss function, which incorporated a dynamic sample allocation strategy to effectively reduce the model's focus on extreme samples and improve the overall performance. In addition, we constructed a large-scale image dataset containing nine types of jute pests, which not only provided an effective training and testing basis for the model but also was an important contribution to the research field of jute pest recognition. The experimental results showed that the average detection accuracy of the improved model increased by 3.45%, especially in the small target P9 category with 12.6% accuracy improvement, mAP@0.5和mAP@0.5:0.95 compared to YOLOv7 with 2.24% and 3.25% respectively, and the GFLOPs were reduced by 16.05%.

The limitation of the JutePest-YOLO model is that the number of parameters and the inference speed of the model are still too high, resulting in inapplicability to target detection in other scenarios. In the following research work, we will make lightweight structural optimization of the JutePest-YOLO model so that it can be extended to target detection in other scene datasets or applied to the field of target tracking.

REFERENCES

[1] M. H. Saleem, S. Ali, M. Rehman, M. Hasanuzzaman, M. Rizwan, S. Irshad, F. Shafiq, M. Iqbal, B. M. Alharbi, T. S. Alnusaire, and S. H. Qari, ''Jute: A potential candidate for phytoremediation of metals—A review,'' *Plants*, vol. 9, no. 2, p. 258, Feb. 2020, doi: 10.3390/plants9020258.

[2] J. Ferdous, M. Hossain, M. Alim, and M. Islam, ''Effect of field duration on yield and yield attributes of tossa jute varieties at different agroecological zones,'' *Bangladesh Agronomy J.*, vol. 22, no. 2, pp. 77–82, Jun. 2020, doi: 10.3329/baj.v22i2.47622.

[3] S. Akter, M. N. Sadekin, and N. Islam, ''Jute and jute products of bangladesh: Contributions and challenges,'' *Asian Bus. Rev.*, vol. 10, no. 3, pp. 143–152, Aug. 2020, doi: 10.18034/abr.v10i3.480.

[4] S. Rahman, M. Kazal, I. Begum, and M. Alam, ''Exploring the future potential of jute in Bangladesh,'' *Agriculture*, vol. 7, no. 12, p. 96, Nov. 2017, doi: 10.3390/agriculture7120096.

[5] V. R. Babu, G. Sivakumar, and S. Satpathy, ''Characterization and field evaluation of spilosoma obliqua nucleopolyhedrosis virus (SpobNPV) CRIJAF1 strain against jute hairy caterpillar, spilosoma obliqua (Walker) infesting jute, corchorus olitorius linn,'' *Egyptian J. Biol. Pest Control*, vol. 33, no. 1, p. 8, Jan. 2023, doi: 10.1186/s41938-023-00654-7.

[6] K. Li, Q. H. Yang, H. J. Zhi, and J. Y. Gai, ''Identification and distribution of soybean mosaic virus strains in Southern China,'' *Plant Disease*, vol. 94, no. 3, pp. 351–357, Mar. 2010, doi: 10.1094/pdis-94-3-0351.

[7] F. Lei, F. Tang, and S. Li, ''Underwater target detection algorithm based on improved YOLOV5,'' *J. Mar. Sci. Eng.*, vol. 10, no. 3, p. 310, Feb. 2022, doi: 10.3390/jmse10030310.

[8] C.-Y. Wang, A. Bochkovskiy, and H.-Y.-M. Liao, ''YOLOV7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors,'' in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 7464–7475.

[9] M. S. U. Sourav and H. Wang, ''Intelligent identification of jute pests based on transfer learning and deep convolutional neural networks,'' *Neural Process. Lett.*, vol. 55, no. 3, pp. 2193–2210, Jun. 2023, doi: 10.1007/s11063-022-10978-4.

[10] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, ''MobileNetV2: Inverted residuals and linear bottlenecks,'' in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4510–4520.

[11] W. Yu, K. Yang, Y. Bai, T. Xiao, H. Yao, and Y. Rui, ''Visualizing and comparing AlexNet and VGG using deconvolutional layers,'' in *Proc. 33rd Int. Conf. Mach. Learn.*, 2016, pp. 1–18.

[12] N. Ma, X. Zhang, H.-T. Zheng, and J. Sun, ''ShuffleNet V2: Practical guidelines for efficient CNN architecture design,'' in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 116–131.

[13] P. Tang, H. Wang, and S. Kwong, ''G-MS2F: GoogleNet based multi-stage feature fusion of deep CNN for scene recognition,'' *Neurocomputing*, vol. 225, pp. 188–197, Feb. 2017, doi: 10.1016/j.neucom.2016.11.023.

[14] D. Z. Karim, T. A. Bushra, and M. M. Saif, ''PestDetector: A deep convolutional neural network to detect jute pests,'' in *Proc. 4th Int. Conf. Sustain. Technol. Ind.*, Dec. 2022, pp. 1–6.

[15] D. Li, F. Ahmed, N. Wu, and A. I. Sethi, ''YOLO-JD: A deep learning network for jute diseases and pests detection from images,'' *Plants*, vol. 11, no. 7, p. 937, Mar. 2022, doi: 10.3390/plants11070937.

[16] M. S. H. Talukder, M. R. Chowdhury, M. S. U. Sourav, A. A. Rakin, S. A. Shuvo, R. B. Sulaiman, M. S. Nipun, M. Islam, M. R. Islam, M. A. Islam, and Z. Haque, ''JutePestDetect: An intelligent approach for jute pest identification using fine-tuned transfer learning,'' *Smart Agricult. Technol.*, vol. 5, Oct. 2023, Art. no. 100279, doi: 10.1016/j.atech.2023.100279.

[17] L. Song, M. Liu, S. Liu, H. Wang, and J. Luo, ''Pest species identification algorithm based on improved YOLOV4 network,'' *Signal, Image Video Process.*, vol. 17, no. 6, pp. 3127–3134, Sep. 2023, doi: 10.1007/s11760-023-02534-x.

[18] W. Xinming and T. S. Hong, ''Comparative study on Leaf disease identification using Yolo v4 and Yolo v7 algorithm,'' *AgBioForum*, vol. 25, no. 1, pp. 58–67, Jun. 2023. [Online]. Available: https://hdl.handle.net/10355/95967

[19] T. Nageshkumar, P. Shrivastava, B. Saha, A. Subeesh, D. B. Shakyawar, G. Sardar, and J. Mandal, ''Defects identification in raw jute fibre using convolutional neural network models,'' *J. Textile Inst.*, vol. 115, no. 5, pp. 835–843, May 2024, doi: 10.1080/00405000.2023.2199489.

[20] (2019). *Agricultural Extension Manual—Dae*. [Online]. Available: https://dae.portal.gov.bd/sites/default/files/files/dae.portal.gov.bd/publications/38eaceb4_db27_48ff_83e1_8b45b01b6a79/Extension_Mannual_Chapt1.pdf

[21] X. Ding, X. Zhang, N. Ma, J. Han, G. Ding, and J. Sun, ''RepVGG: Making VGG-style ConvNets great again,'' in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 13728–13737.

[22] Z. Tong, Y. Chen, Z. Xu, and R. Yu, ''Wise-IoU: Bounding box regression loss with dynamic focusing mechanism,'' 2023, *arXiv:2301.10051*.

[23] J. Chen, S.-H. Kao, H. He, W. Zhuo, S. Wen, C.-H. Lee, and S.-H.-G. Chan, ''Run, don't walk: Chasing higher FLOPS for faster neural networks,'' in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 12021–12031.

[24] Z. Jiang, Y. Guo, K. Jiang, M. Hu, and Z. Zhu, ''Optimization of intelligent plant cultivation robot system in object detection,'' *IEEE Sensors J.*, vol. 21, no. 17, pp. 19279–19288, Sep. 2021, doi: 10.1109/JSEN.2021.3077272.

[25] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, ''Grad-CAM: Visual explanations from deep networks via gradient-based localization,'' in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 618–626.

[26] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, ''Generalized intersection over union: A metric and a loss for bounding box regression,'' in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 658–666.

[27] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, and D. Ren, ''Distance-IoU loss: Faster and better learning for bounding box regression,'' in *Proc. AAAI Conf. Artif. Intell.*, Apr. 2020, vol. 34, no. 7, pp. 12993–13000.

[28] Y.-F. Zhang, W. Ren, Z. Zhang, Z. Jia, L. Wang, and T. Tan, ''Focal and efficient IOU loss for accurate bounding box regression,'' *Neurocomputing*, vol. 506, pp. 146–157, Sep. 2022, doi: 10.1016/j.neucom.2022.07.042.

[29] Z. Gevorgyan, ''SIoU loss: More powerful learning for bounding box regression,'' 2022, *arXiv:2205.12740*.

[30] J. Wang, C. Xu, W. Yang, and L. Yu, ''A normalized Gaussian Wasserstein distance for tiny object detection,'' 2021, *arXiv:2110.13389*.

[31] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, ''Focal loss for dense object detection,'' in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2999–3007.

[32] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, ''YOLOX: Exceeding YOLO series in 2021,'' 2021, *arXiv:2107.08430*.

[33] H. Shi, W. Yang, D. Chen, and M. Wang, ''CPA-YOLOV7: Contextual and pyramid attention-based improvement of YOLOV7 for drones scene target detection,'' *J. Vis. Commun. Image Represent.*, vol. 97, Dec. 2023, Art. no. 103965, doi: 10.1016/j.jvcir.2023.103965.

[34] G. Wang, Y. Chen, P. An, H. Hong, J. Hu, and T. Huang, ''UAV-YOLOV8: A small-object-detection model based on improved YOLOV8 for UAV aerial photography scenarios,'' *Sensors*, vol. 23, no. 16, p. 7190, Aug. 2023, doi: 10.3390/s23167190.

SHUAI ZHANG received the B.E. degree from Hubei Polytechnic University, Huangshi, China, in 2021. He is currently pursuing the M.S. degree in software engineering with Wuhan Polytechnic University, Wuhan. His research interest includes artificial intelligence technology and its application.

HENG WANG received the B.E. degree from the Huazhong University of Science and Technology, in 2006, and the Ph.D. degree in engineering from Wuhan University, in 2013. He is currently a Professor with the School of Mathematics and Computer Science, Wuhan Polytechnic University. He is also a Postdoctoral Research Fellow with Alto University, Finland. His research interests include the perception characteristics of acoustic spatial parameters, artificial intelligence, and the application of 3D audio and video in virtual reality.

CONG ZHANG received the bachelor's degree in automation engineering from the Huazhong University of Science and Technology, in 1993, the master's degree in computer application technology from Wuhan University of Technology, in 1999, and the Ph.D. degree in computer application technology from Wuhan University, in 2010. He is currently a Professor with the School of Electrical and Electronic Engineering, Wuhan Polytechnic University. His research interests include multimedia signal processing, multimedia communication system theory and application, and pattern recognition.

ZHENG LIU received the B.E. degree from Wuhan Polytechnic University, Wuhan, China, in 2022, where he is currently pursuing the M.S. degree in software engineering. His research interests include music information retrieval, and artificial intelligence technology and its application.

YIMING JIANG received the B.E. degree from Wuhan Polytechnic University, Wuhan, China, in 2023, where he is currently pursuing the M.S. degree in software engineering. His research interests include music information retrieval, and artificial intelligence technology and its application.

LEI YU received the B.E. degree from Southwest Petroleum University, Chengdu, China, in 2023. She is currently pursuing the M.S. degree in software engineering with Wuhan Polytechnic University, Wuhan. Her research interest includes artificial intelligence technology and its application.