applied sciences

MDPI

Article

A Lightweight Forest Pest Image Recognition Model Based on Improved YOLOv8

Tingyao Jiang and Shuo Chen *

College of Computer and Information Technology, Three Gorges University, Yichang 443002, China; jiangty@ctgu.edu.cn

* Correspondence: shawn_cs@yeah.net or 202208540021064@ctgu.edu.cn; Tel.: +86-133-0866-3753

Abstract: In response to the shortcomings of traditional pest detection methods, such as inadequate accuracy and slow detection speeds, a lightweight forestry pest image recognition model based on an improved YOLOv8 architecture is proposed. Initially, given the limited availability of real deep forest pest image data in the wild, data augmentation techniques, including random rotation, translation, and Mosaic, are employed to expand and enhance the dataset. Subsequently, the traditional Conv (convolution) layers in the neck module of YOLOv8 are replaced with lightweight GSConv, and the Slim Neck design paradigm is utilized for reconstruction to reduce computational costs while preserving model accuracy. Furthermore, the CBAM attention mechanism is introduced into the backbone network of YOLOv8 to enhance the feature extraction of crucial information, thereby improving detection accuracy. Finally, WIoU is employed as a replacement for the traditional CIOU to enhance the overall performance of the detector. The experimental results demonstrate that the improved model exhibits a significant advantage in the field of forestry pest detection, achieving precision and recall rates of 98.9% and 97.6%, respectively. This surpasses the performance of the current mainstream network models.

Keywords: forest pest; YOLOv8; object detection

Citation: Jiang, T.; Chen, S. A Lightweight Forest Pest Image Recognition Model Based on Improved YOLOv8. *Appl. Sci.* **2024**, *14*, 1941. https://doi.org/10.3390/app14051941

Academic Editors: Seokwon Yeom, Eleonora Iotti, Vincenzo Bonnici and Flavio Bertini

Received: 27 November 2023
Revised: 3 February 2024
Accepted: 14 February 2024
Published: 27 February 2024

1. Introduction

Forest pests pose a significant threat to both forest ecosystems and timber resources, thereby necessitating the timely and precise detection and management of these pests for the preservation of forest health [1]. However, conventional methods for pest detection suffer from accuracy deficiencies and slow detection speeds, thereby limiting their practical applicability in the face of ever-evolving threats.

In recent years, numerous scholars have conducted methodological research to achieve an efficient and precise detection of forestry pests, yielding promising results. Target detection based on deep learning has emerged as a prominent research focus in the field of computer vision, demonstrating robust capabilities in automatically identifying the position and categories of objects within images or videos. It has been empirically validated as an effective approach in the domain of forestry pest detection.

Target detection networks are generally categorized into two main types: one-stage target detection networks and two-stage target detection networks. One-stage target detection networks integrate object detection and classification into a single network to enhance detection speed. Representative models in this category include YOLO (You Only Look Once) [2–9] and SSD (Single Shot MultiBox Detector) [10–12]. Two-stage target detection networks, on the other hand, first generate candidate regions and then perform target classification and localization on these regions. Representative models in this category include Fast-RCNN and Faster-RCNN [13–15]. The advancement of these deep learning target detection networks has elevated detection accuracy and efficiency, rendering them powerful tools widely applied across multiple domains. In the context of pest detection,

there is a critical need for rapid and effective monitoring of forestry within a short timeframe, as well as the timely implementation of corresponding measures. On the other hand, for non-real-time applications like pest detection, the use of a one-stage object detector can achieve model lightweighting, reducing both the model's parameter count and complexity serves to diminish both storage and loading costs associated with the model. Simplifying the model structure also decreases the computational resource requirements, thereby enhancing overall efficiency. Therefore, this study opts for the one-stage object detector YOLO for pest detection.

However, despite the enormous potential of deep learning in pest recognition, its practical application still faces several challenges. For instance, deep learning models require a large amount of annotated data, which are often difficult to acquire in the field of pest recognition. Moreover, the computational cost of deep learning models is high, which may limit their application in resource-constrained regions. Therefore, future research needs to further explore how to improve the efficiency and accuracy of deep learning in pest recognition while reducing its computational cost and data requirements.

For the detection of forest pests, Sun Haiyan et al. [16] proposed a forestry pest detection method based on an attention model and lightweight YOLOv4. They achieved the detection of seven types of forest pests by improving the network structure, optimizing the loss function, and introducing an attention mechanism. However, there is significant variation in accuracy among different pest classes, and the overall average precision requires further enhancement. Hou Ruihuan et al. [17] presented a real-time forestry pest detection method based on YOLOv4-TIA. By incorporating a three-branch attention mechanism [18], they improved the backbone network of YOLOv4 and optimized the loss function, enabling the detection of seven types of forest pests. Nevertheless, this model exhibits increased complexity, slower detection speed, and an average precision of only 85.9%.

In response to the aforementioned issues, this paper introduces a lightweight forestry pest image recognition model based on an improved YOLOv8. This model not only enhances the performance of small object detection, but also ensures minimal resource consumption. The main contributions of this paper are as follows:

(1) Integrating the GSconv module [19] and employing the Slim-Neck design philosophy to refine the Neck layer of YOLOv8n, thereby achieving a lightweight architecture. This optimization reduces the network's parameter count, resulting in enhanced detection speed.
(2) Incorporating the attention module CBAM [20] into the backbone network to augment the network's focus on small objects. This enhancement significantly improves detection accuracy without introducing a substantial increase in computational complexity.
(3) Incorporating WIoU v3 [21] into the bounding box regression loss function and implementing a dynamic non-monotonic mechanism to devise a more judicious strategy for gradient gain allocation. WIoU v3 effectively mitigates gradient gain discrepancies between high-quality and low-quality samples, thereby fortifying the model's localization proficiency and generalization capabilities.

2. YOLOv8 Object Detection Algorithm

YOLOv8 stands out as the latest algorithm unveiled by Ultralytics on 10 January 2023. It represents the cutting-edge advancements in the YOLO series, showcasing outstanding detection accuracy and speed. Differentiating based on network depth and width, YOLOv8 further refines into YOLOv8n, YOLOv8s, YOLOv8m, YOLOv8l and YOLOv8x. Given the constraints of computational resources, YOLOv8n, a lightweight iteration within the YOLOv8 framework, proves instrumental in mitigating the model's storage and loading costs. In applications such as pest detection, rapid model deployment is a critical consideration, particularly in situations that require prompt responses to pest-related challenges. Pest detection often demands real-time results for swift decision-making and a timely implementation of corresponding measures. Lightweight models typically boast higher inference speeds, facilitating the fulfillment of real-time requirements. In summary, this study opts

for the lightweight version YOLOv8n within the YOLOv8 framework. The architectural components of YOLOv8n encompass the backbone feature extraction network (backbone), the feature pyramid (neck), and the prediction end (head), with specific structures outlined in Figure 1.

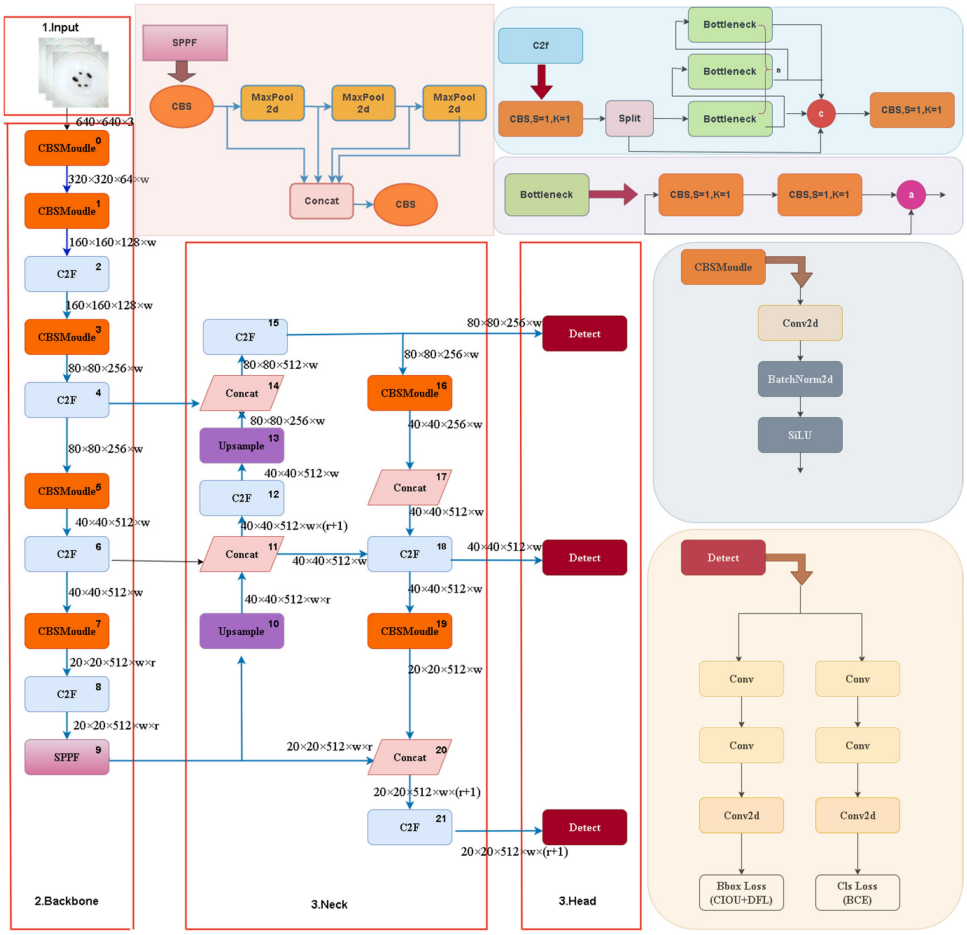


Figure 1. YOLOv8n model architecture encompassing the feature extraction network, SPPF architecture, feature fusion, and prediction network. The w (width) and r (ratio) in Figure 1 are parameters used to represent the size of the feature map.

2.1. Backbone

YOLOv8 incorporates an enhanced CSPDarknet53 [4] as its backbone network for efficient feature extraction. Upon image input, YOLOv8 employs Mosaic data augmentation at the input layer. During the final 10 epochs of training, Mosaic data augmentation is deactivated. This strategic adjustment significantly enhances the diversity of backgrounds for detected objects, thereby effectively boosting the model's performance and robustness. Within the backbone network, the CBS module conducts convolutional operations on input data, followed by batch normalization and SiLU activation, as depicted in Figure 1—the CBS module. YOLOv8 replaces the original Cross-Stage Partial (CSP) module with the C2f module in its structure, as illustrated in Figure 1—the C2f module. The C2f module, adopting gradient-concatenated connections, enriches the information flow of the feature extraction network while ensuring a lightweight design. The Spatial Pyramid Pooling Fusion (SPPF) module transforms feature maps of arbitrary sizes into adaptive-sized feature vectors. Initially, channel-balanced operations (CBSs) are applied to the input feature map to extract feature information. Subsequently, max-pooling operations with sizes of 1×1, 5×5, 5×5 and 5×5 are separately performed on the feature map. Following each pooling operation, the obtained features are sequentially output to obtain multi-scale features.

These multi-scale feature maps are then fused, followed by another round of channel-balanced operations (CBSs), resulting in the final feature map output. In comparison to the conventional SPP [22] module, the SPPF module, as illustrated in Figure 1-SPPF module, reduces computational complexity and exhibits lower latency by sequentially connecting three max-pooling operations.

2.2. Neck

Building upon the PANet [23], YOLOv8 enhances the design in the neck by introducing the PAN-FPN structure. The core design objective is to improve the effectiveness of information exchange between features, thus better addressing multi-scale object detection tasks. Firstly, a bottom-level feature extraction network acquires the low-level features of the input image. Subsequently, FPN introduces a top-down feature up-sampling path, forming a feature pyramid where each layer has a distinct resolution, providing the network with multi-scale semantic information. The innovation of the PAN structure lies in the introduction of a path aggregation mechanism. The bottom–up path generates a set of path features from the low-level feature map, while the top–down path generates corresponding path features from the high-resolution feature pyramid. Through path fusion operations, PAN integrates these two sets of path features, facilitating the exchange of multi-level, multi-resolution information from bottom to top, and vice versa. Finally, through feature fusion, the features generated by PAN are merged with the feature pyramid from the original FPN, creating the ultimate feature pyramid. This pyramid contains features from different levels and paths, providing a richer semantic understanding of the input data.

2.3. Head

YOLOv8's detection module adopts the prevalent decoupled head structure, employing two independent branches for target classification and bounding box regression predictions, each utilizing distinct loss functions. Binary cross-entropy loss (BCE loss) is employed for the classification task. For bounding box regression, we have chosen to utilize DFL [24] and CIoU [25] as the loss functions. This detection structure aims to enhance detection accuracy and accelerate the model's convergence speed. Notably, YOLOv8 departs from the anchor-based approach in its design and embraces the anchor-free [26] philosophy. This design decision is aimed at further improving detection performance, enabling the model to be more flexible in adapting to various scales and shapes of targets. Simultaneously, it enhances bounding box localization accuracy and overall detection precision in object detection tasks.

3. Methods

Currently, the YOLOv8 algorithm has achieved significant success in the field of object detection. However, challenges in its application to forestry pest detection include the computational cost increase and detection speed decrease resulting from a large number of parameters. In practical applications, rapid model deployment is crucial to meet the real-time requirements of forestry pest detection. Lightweight models often offer higher inference speeds; thus, we consider lightweighting the Neck network in YOLOv8 to reduce computational costs, accelerate detection speeds, while maintaining robust feature representation capabilities. In the context of forestry pest detection, the widespread presence of small targets (such as the acuminatus, coleoptera, armandi, and linnaeus pests in the dataset) poses challenges. YOLOv8 exhibits poor detection performance, with issues such as low detection rates and increased false positives when dealing with these small targets. To address these challenges, we introduce attention modules to focus the model more on key information in input features, enhancing the model's attention to small targets and thus improving small target detection performance. Simultaneously, we explore the design principles of Weighted Intersection over Union (WIoU). WIoU v3 employs a dynamic non-monotonic mechanism to evaluate anchor box quality, directing the model's attention toward anchor boxes of regular quality, thereby improving the model's localization capabil-

ities. In forestry pest detection, the high proportion of small targets increases the difficulty of detection. WIoU v3 further enhances the model's detection performance by dynamically optimizing the loss weights for small targets. The specific details of these optimization strategies are outlined below:

Firstly, in terms of the Neck network, we employed the Slim-Neck design approach. We replaced traditional convolution operations with lightweight GSConv convolutions and introduced the VoV-GSCSP module to replace the original C2f module, incorporating lightweight bottleneck layers (GSbottleneck). The purpose of this adjustment is to reduce computational costs and accelerate the model's detection speed without compromising feature representation capabilities. This is crucial to meet the real-time requirements of forestry pest detection. These modifications result in a more lightweight application, facilitating easier deployment while maintaining the integrity of feature representation.

Additionally, in the backbone network, we introduced the CBAM (Channel Attention and Spatial Attention) attention mechanism. The CBAM attention mechanism dynamically adjusts the weights of feature maps, directing the network's focus toward regions containing small targets. By increasing the network's perceptual range, this mechanism aids in capturing a more extensive context of information. This is particularly valuable when parts of the target may be obscured, as a larger perceptual range enables a better understanding of the contextual information surrounding the target. Overall, the incorporation of the CBAM attention mechanism holds the promise of improving small target detection performance in forestry pest detection. It enhances the network's attention to crucial target information, thereby improving adaptability to complex scenes and small targets.

Finally, we adopted WIoU v3 as a replacement for the original CIOU bounding box regression loss in YOLOv8. WIoU v3 integrates a dynamic non-monotonic mechanism and introduces a gradient gain allocation strategy to mitigate the occurrence of substantial or harmful gradients in extreme samples. This version of WIoU places a greater emphasis on samples of regular quality, enhancing the model's generalization capabilities and overall performance. Additionally, WIoU v3 dynamically adjusts the loss weights for small targets, further improving the model's detection performance. In summary, WIoU demonstrates its advantages in forestry pest detection by exhibiting flexible adaptability to targets of different sizes, shapes, and qualities. This enhances the accuracy, robustness, and generalization of the detection process, showcasing WIoU's effectiveness in addressing the challenges posed by diverse pest targets in forestry environments.

3.1. GSConv and VoV-GSCSP Modules

As the practical applications of deep learning models continue to expand, there is an urgent demand for algorithm lightweighting. This is primarily driven by the prevalent scenarios in forestry pest detection where resources are constrained, and computational capabilities are limited. Despite YOLO's outstanding performance in object detection tasks, its relatively large model size hampers its operational efficiency on lightweight devices commonly encountered in forestry pest detection. To adapt to resource-constrained environments, the imperative for refining and lightweighting YOLO becomes apparent. Optimization of the YOLO algorithm by reducing model size and enhancing computational efficiency is essential to better meet the practical requirements of forestry pest detection. Lightweighting not only facilitates real-time detection on embedded devices but also contributes to cost reduction, thereby enhancing the practical usability of the system. In the process of improving YOLO lightweighting, a holistic consideration of model accuracy, speed, and power consumption metrics is crucial to achieve a balance across diverse scenarios. Employing more efficient network architectures, streamlining parameters, and optimizing for specific hardware platforms are pivotal strategies for lightweighting. Through these concerted efforts, the adaptability of pest detection technology to the intricate and dynamic natural environment is significantly enhanced, providing a more reliable and efficient support system for forest conservation.

In the realm of lightweight models, such as Xception [27], MobileNets [28], and ShuffleNets [29], the use of Depthwise Separable Convolution (DSC) operations has significantly improved the speed of detectors. However, these models suffer from the issue of accuracy loss. To address the accuracy loss associated with Depthwise Separable Convolution, Li Hulin proposed the GSConv lightweight convolution module, the main structure of which is depicted in Figure 2. Through a shuffle operation, it achieves the fusion of information generated by traditional convolution modules (dense convolution operations) and information generated by Depthwise Separable Convolution. In this process, with an input channel number of C1 and an output channel number of C2, the following steps are taken: first, a standard convolution reduces the channel number by half to C2/2; then, the channel number remains unchanged through Depthwise Separable Convolution. Subsequently, the result of the first convolution is Concatenated and shuffled with the structure after Depthwise Separable Convolution. In the final shuffle operation, channel information is uniformly shuffled to ensure the effective retention of multi-channel information. This process aims to enhance the extraction of semantic information, strengthen the fusion of feature information, and consequently improve the expressive capabilities of image features. Through such shuffle operations, an orderly fusion of information is achieved, providing an effective mechanism for enhancing the model's performance.

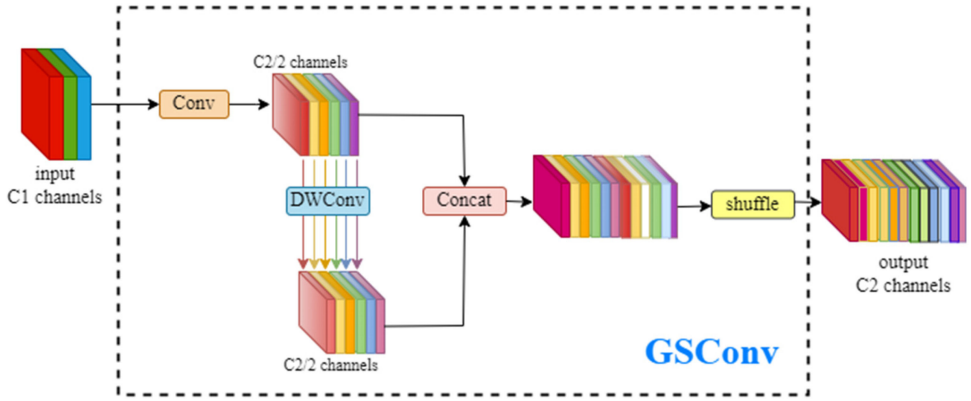


Figure 2. Structure of the GSConv Module. The "Conv" block consists of three layers: convolution layer, batch normalization layer, and activation layer. The "DWConv" marked in blue here represents Depthwise Separable Convolution (DSC) operation.

GSConv's computational cost is approximately 50% of SC, but its contribution to the model's learning capability is comparable to the latter. Building upon GSConv, a GS bottleneck module is designed in the literature, and Figure 3a illustrates the structure of the GS bottleneck module. The VoV-GSCSP module is crafted through a one-time aggregation method. Figure 3b–d depicts three design options provided for VoV-GSCSP, where (b) is straightforward with faster inference speed, and (c), (d) exhibit higher feature reuse rates. In practice, due to its hardware-friendly nature, it is more convenient to employ a structurally simpler module.

Therefore, in optimizing the Neck network layer, a design approach based on Slim-Neck is employed in this study. This involves the substitution of a standard convolution with a GSConv lightweight convolution, and the replacement of the original C2f module with the lightweight bottleneck layer (GSbottleneck) within the Vision over Visibility Skip-level Cross-Stage Partial (VoV-GSCSP) module. This implementation achieves a lightweight Neck layer, leading to a significant reduction in computational costs and, consequently, an acceleration in inference speed. The application of the Slim-Neck theory to the YOLOv8 model structure is depicted in Figure 4. This design not only ensures computational efficiency but also aligns with the standards of scientific rigor in the pursuit of model optimization.

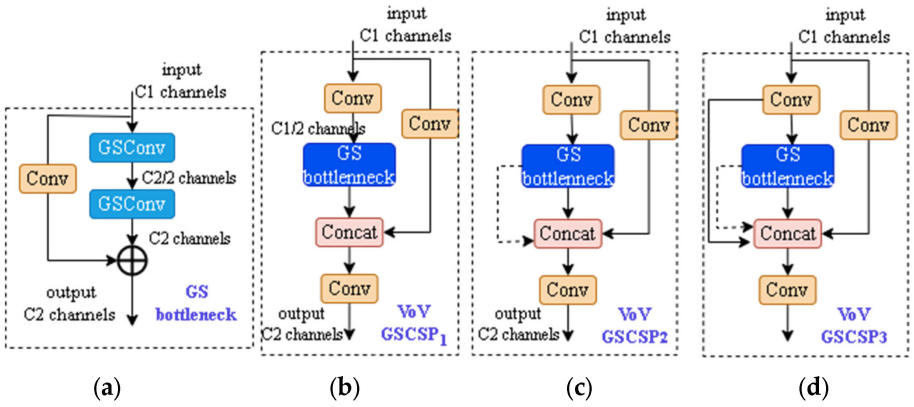


Figure 3. (**a**) Structure of the GS Bottleneck module and (**b**–**d**) Structures of VoV-GSCSP1, VoV-GSCSP2, and VoV-GSCSP3 modules.

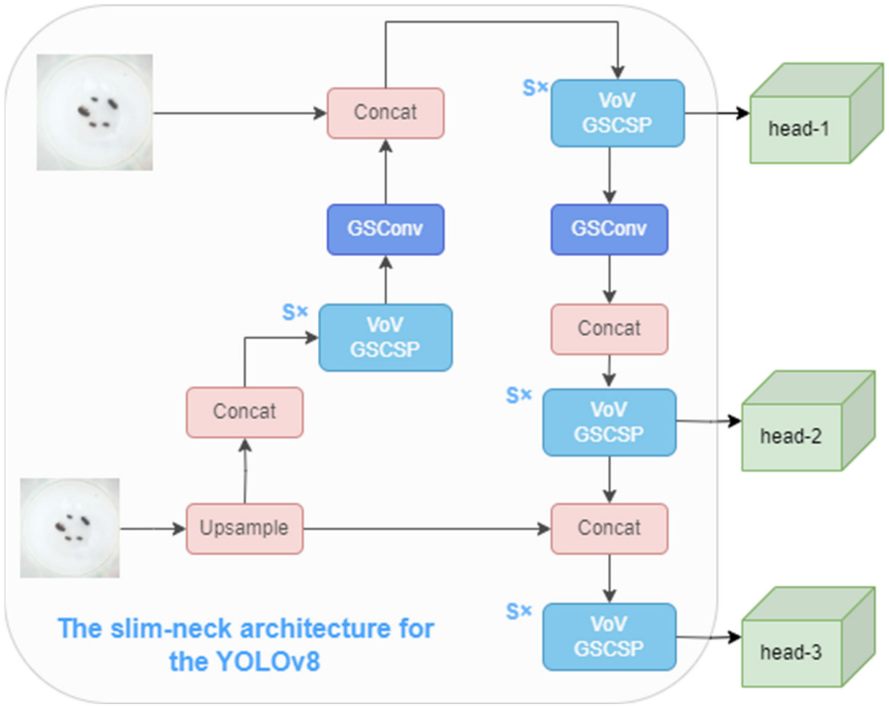


Figure 4. Slim-Neck Structure of YOLOv8.

3.2. CBAM Attention Module

Given that the background occupies a substantial portion of the images in the utilized dataset, and the predominant targets for detection are small-sized pests, the detection performance of the YOLOv8n algorithm hinges predominantly on the efficiency of the backbone network. To enhance the backbone network's capacity for extracting critical information, we introduce a Convolutional Block Attention Module (CBAM) into the YOLOv8 backbone, as depicted in Figure 5. The CBAM model comprises a Channel Attention Module (CAM) and a Spatial Attention Module (SAM), dedicated to extracting channel and spatial attention, respectively. Through adaptive feature refinement facilitated by channel and spatial attention mechanisms, the model endeavors to identify attention regions within densely populated pest scenarios. This incorporation aims to elevate the model's efficacy in discerning salient features in the presence of prevalent background and small-sized targets, thereby aligning with the rigorous standards of scientific discourse.

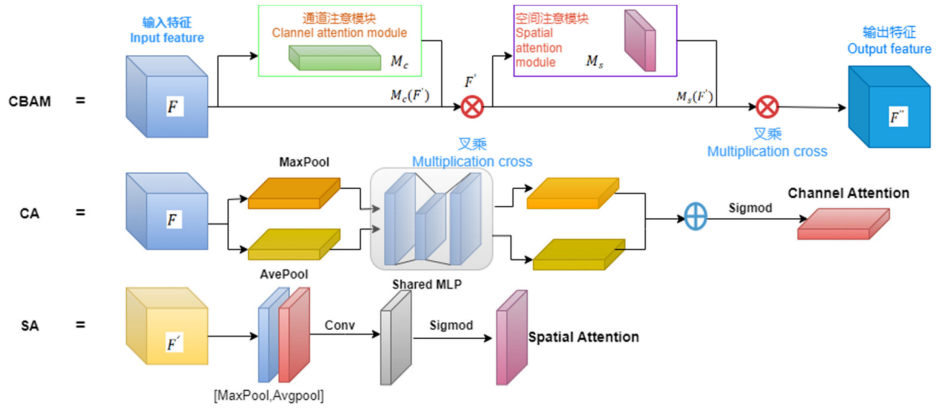


Figure 5. CBAM (Convolutional Block Attention Model).

Given the input feature $F \in \mathbb{R}^{C \times H \times W}$, where C, H, and W represent the channel number, height, and width of the feature map, respectively. After the Channel Attention Module (CAM), channel attention features denoted as M_c are obtained, as shown in Equation (1). M_c is then multiplied with the feature map F resulting in the feature F'. Following the Spatial Attention Module (SAM), spatial attention features denoted as M_s are obtained, as expressed in Equation (2). M_s is subsequently multiplied with the feature F', yielding the refined feature F''.

$$F' = M_c(F) \otimes F \tag{1}$$

$$F'' = M_s(F') \otimes F' \tag{2}$$

In this investigation, the CBAM attention module is introduced following the C2f module in the YOLOv8 backbone network, incurring a negligible computational overhead. This strategic enhancement aims to provide the deep network with more accurate feature information, thereby contributing to the reduction in loss values and ensuring precise identification and localization of small targets, especially in the context of forestry pest detection. From an intuitive perspective, during the forward propagation of gradients, crucial channels and spatial information in the feature map receive greater emphasis. This refinement is evident in the final output image, effectively accentuating regions of interest for the detection model and enhancing its ability to discern target objects accurately. This improved approach significantly elevates the model's performance in object detection tasks, demonstrating enhanced potential in effectively handling intricate image scenarios, aligning with the rigorous standards of scientific discourse.

3.3. Improved Loss Function

In the task of detecting small objects in forestry pest images, where the proportion of small objects is relatively high, the rational design of loss functions can significantly enhance the detection performance of the model. The loss function in YOLOv8 consists of multiple components, including the classification loss (VFL Loss) and the regression loss in the form of CIOU Loss + DFL. The formula for the VFL Loss function is given by Equation (3) [30].

$$\mathrm{VFL}(p, q) = \begin{cases} -q(q\log(p) + (1-q)\log(1-p)) & q > 0 \\ -\alpha p^{\gamma}\log(1-p) & q = 0 \end{cases} \tag{3}$$

In the aforementioned formula, "q" represents the Intersection over Union (IoU) between the bounding box (predicted box) and the ground truth box. IoU is calculated by dividing the intersection of the predicted box and the ground truth box by the union of the two boxes. The variable "p" represents the score or probability. If the two boxes intersect ($q > 0$), it is considered a positive sample, and if there is no intersection, then q is set to 0, indicating a negative sample.

The definition of CIoU, as given in Equation (4), incorporates an additional penalty term on top of DIoU.

$$\mathcal{L}_{\mathrm{CIoU}} = 1 - \mathrm{IoU} + \frac{\rho^2(\mathrm{b}, \mathrm{b}^{\mathrm{gt}})}{\mathrm{c}^2} + \alpha\upsilon \tag{4}$$

$$\alpha = \frac{\mathrm{v}}{(1 - \mathrm{IoU}) + \mathrm{v}} \tag{5}$$

$$\mathrm{v} = \frac{4}{\pi^2}\left(\arctan\frac{\mathrm{w}^{\mathrm{gt}}}{\mathrm{h}^{\mathrm{gt}}} - \arctan\frac{\mathrm{w}}{\mathrm{h}}\right)^2 \tag{6}$$

Among these, α serves as a weight function, defined in Equation (5). υ gauges the similarity in the aspect ratio between the predicted box and the ground truth, as outlined in Equation (6). The terms $\mathrm{w}^{\mathrm{gt}}/\mathrm{h}^{\mathrm{gt}}$ and w/h denote the aspect ratios of the ground truth box and the predicted box. b and b^{gt} represent the center points of the predicted box and the ground truth box, respectively. ρ denotes the Euclidean distance between the two rectangular boxes, and c signifies the diagonal distance of the closed region between these two rectangular boxes.

DFL loss (distribution focal loss) is a loss function designed to address the issue of class imbalance. Similar to focal loss, DFL incorporates information about class distribution, providing improved handling of imbalanced class scenarios. The formula for DFL is expressed by Equation (7).

$$\mathrm{DFL}(\mathrm{S}_i, \mathrm{S}_{i+1}) = -((y_{i+1} - y)\log(\mathrm{S}_i) + (y - y_i)\log(\mathrm{S}_{i+1})) \tag{7}$$

However, CIoU has its drawbacks. Firstly, the computation of CIoU involves the calculation of inverse trigonometric functions, which increases the computational cost of the model, particularly in large-scale object detection tasks. Secondly, CIoU does not account for the balance of hard and easy samples. Thirdly, CIoU considers aspect ratio as a penalty term. When the aspect ratio of the actual box and the predicted box is the same, but the values of width and height are different, the penalty term fails to reflect the true difference between the two bounding boxes.

Therefore, this paper introduces Wise-IoU (WIoU). In terms of computational speed, the additional computation cost of WIoU mainly lies in the calculation of the focusing coefficient and the mean statistics of IoU loss. Under the same experimental conditions, WIoU has a faster speed compared to CIoU because it does not involve aspect ratio calculations, with WIoU's computation time being 87.2% of CIoU. In terms of performance improvement, WIoU considers not only the area, centroid distance, and overlap area, but also introduces a dynamic non-monotonic focusing mechanism. When the annotation quality of the dataset is poor, WIoU performs better relative to other bounding box losses. The weight calculation of WIoU can better reflect the differences in the appearance and structure of the targets, providing better target distinctiveness, which is advantageous when dealing with targets with similar features. Specific information about WIoU is as follows:

(1) Wise-IoU v1: As it is challenging to avoid including low-quality examples in the training data, geometric metrics such as distance and aspect ratio exacerbate the penalty on low-quality examples, leading to a decrease in the model's generalization performance. A good loss function should weaken the penalty on geometric metrics when the anchor box and target box overlap well, intervening in training as little as possible to enhance the model's generalization ability. In WIoU v1, distance attention is constructed based on distance metrics. The definition of WIoU v1 is given by Formula (8).

$$\mathcal{L}_{\mathrm{WIoUv1}} = \mathcal{R}_{\mathrm{WIoU}}\mathcal{L}_{\mathrm{IoU}} \tag{8}$$

$$\mathcal{R}_{\mathrm{WIoU}} = \exp\left(\frac{\left(x - x_{gt}\right)^2 + \left(y - y_{gt}\right)^2}{\left(W_g^2 + H_g^2\right)^*}\right) \tag{9}$$

$R_{\mathrm{WIoU}} \in [1, e)$, significantly amplifying the $\mathcal{L}_{\mathrm{IoU}}$ of ordinary-quality anchor boxes, $\mathcal{L}_{\mathrm{IoU}} \in [0, 1]$, markedly reducing the $\mathcal{R}_{\mathrm{WIoU}}$ of high-quality anchor boxes, and notably decreasing their attention to the center point distance in cases where the anchor box overlaps well with the target box.

(2) Wise-IoU v2: Focal loss introduces a monotonic focusing mechanism tailored for cross-entropy, effectively reducing the contribution of easy examples to the loss value. This allows the model to focus on challenging examples, enhancing classification performance. Similarly, in v2, a monotonic focusing coefficient $\mathcal{L}_{\mathrm{IoU}}^{\gamma^*}$ is constructed for $\mathcal{L}_{\mathrm{WIoUv1}}$. The definition of Wise-IoU v2 is given by Formula (10).

$$\mathcal{L}_{\mathrm{WIoUv2}} = \mathcal{L}_{\mathrm{IoU}}^{\gamma^*}\mathcal{L}_{\mathrm{WIoUv1}}\ ,\ \gamma > 0 \tag{10}$$

In the model training process, the gradient gain Lriou decreases as Liou decreases, resulting in a slow convergence speed in the later stages of training. Therefore, the introduced mean is used as a normalization factor, as shown in Formula (11):

$$\mathcal{L}_{\mathrm{WIoUv2}} = \left(\frac{\mathcal{L}_{\mathrm{IoU}}^{\gamma^*}}{\overline{\mathcal{L}_{\mathrm{IoU}}}}\right)^{\gamma}\mathcal{L}_{\mathrm{WIoUv1}} \tag{11}$$

The term $\mathcal{L}_{\mathrm{IoU}}$ represents the moving average with momentum m, dynamically updating the normalization factor to keep the overall gradient gain $r = \left(\frac{\mathcal{L}_{\mathrm{IoU}}^{\gamma^*}}{\overline{\mathcal{L}_{\mathrm{IoU}}}}\right)^{\gamma}$ at a higher level, addressing the issue of slow convergence speed in the later stages of training.

(3) Wise-IoU v3: The concept of outlierness is introduced to characterize the quality of anchor boxes, defined as in Equation (12):

$$\beta = \frac{\mathcal{L}_{\mathrm{IoU}}^{\gamma^*}}{\overline{\mathcal{L}_{\mathrm{IoU}}}} \in [0, +\infty) \tag{12}$$

Building upon Wise-IoU v1, Wise-IoU v3 introduces a non-monotonic focusing coefficient based on β, defined as in Equation (13). A smaller outlierness implies a higher quality anchor box, resulting in a smaller gradient boost assigned to it, allowing for better bounding box regression focus on anchor boxes with common quality. For anchor boxes with larger outlierness, a smaller gradient boost is allocated, effectively preventing harmful gradients from arising in low-quality examples.

$$L_{\mathrm{WIoUv3}} = rL_{\mathrm{WIoUv1}}\ ,\ r = \frac{\beta}{\delta\alpha^{\beta-\delta}} \tag{13}$$

At that time, when $\beta = \delta$, δ makes $r = 1$. When the outlierness of the anchor box satisfies $\beta = C$ (C is a constant value), the anchor box will receive the highest gradient boost. Because $\mathcal{L}_{\mathrm{IoU}}$ is dynamic, and the quality criteria for anchor boxes are also dynamic, this enables Wise-IoU v3 to dynamically allocate gradient boosts according to the current situation at any given moment.

Through the comparative analysis mentioned above, this study achieved a significant performance improvement by replacing the traditional CIOU with Wise-IoU v3 in YOLOv8. Wise-IoU v3 utilizes a dynamic non-monotonic mechanism to evaluate anchor box quality, making the model focus more on anchor boxes of ordinary quality, and thus improving the model's object localization capability. For the task of detecting targets in forestry pests, where small targets have a high proportion, increasing the detection difficulty, Wise-IoU

v3 can dynamically optimize the loss weights for small targets, enhancing the model's detection performance.

4. Experiments

This section will delve into a series of experiments conducted, presenting the exceptional performance of the proposed model through comparisons, analyses, and result demonstrations.

4.1. Dataset and Data Preprocessing

This study is conducted using a publicly available forestry pest dataset [16,17] provided by the Beijing Forestry University, comprising a total of 2183 images. These images feature a uniform white background and cover seven distinct categories of forestry pests, namely Boerner, Leconte, Linnaeus, acuminatus, armandi, coleoptera, and linnaeus. The quantitative distribution of the seven classes of forestry pests is summarized in Table 1. The dataset is acquired using insect traps for capturing images, which compared to natural field photography scenes, results in a background that is simpler and clearer.

Table 1. Statistics of sample quantities for each pest category.

Insect Category	Number of Samples	Percentage
Boerner	1595	15.4%
Leconte	2216	21.4%
Linnaeus	818	7.9%
acuminatus	953	9.2%
armandi	1765	17.1%
colepotera	2091	20.2%
linnaeus	909	8.8%

Figure 6 illustrates the main characteristics of seven types of pests. Among them, Boerner and Leconte belong to the category of larger-sized pests. Boerner presents a body color of gray-brown, with a relatively elongated physique, while Leconte has a black body with deep brown wings. In contrast, Linnaeus, acuminatus, armandi, coleoptera, and linnaeus are considered small-sized pests. Linnaeus and linnaeus have black bodies with shorter legs; acuminatus is brown with slightly spread wings; armandi is a black beetle with longer legs; and coleoptera is the smallest among the seven types of pests. Overall, these seven types of pests exhibit a certain degree of similarity, making their detection considerably challenging. Therefore, the implementation of data preprocessing operations becomes crucial. The following outlines the specific procedures employed for data preprocessing in this study:

Figure 6. Illustration of seven types of pests.

4.1.1. Random Scaling and Cropping

Random scaling and cropping are employed to alter the size and perspective of the images. In this process, a random scaling factor is selected, and the new image dimensions are computed accordingly. Subsequently, a random window on the image is chosen to crop the final augmented image. This method contributes to increasing the diversity of the dataset, enabling the model to adapt to images with different sizes and perspectives, thereby enhancing the model's robustness.

4.1.2. Random Brightness and Contrast Adjustments

Random brightness and contrast adjustments are applied to introduce variations in the illumination and contrast of the images. This process randomly modifies the brightness and contrast levels, contributing to the augmentation of the dataset. By incorporating these random adjustments, the model becomes more resilient to variations in lighting conditions and contrast, enhancing its ability to generalize across different scenarios.

4.1.3. Mosaic Data Augmentation

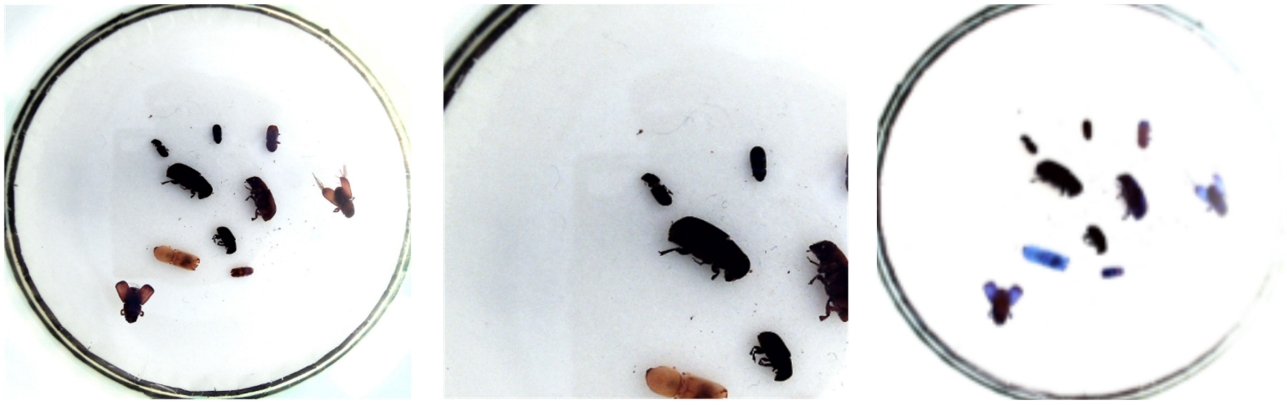
Mosaic Data Augmentation is a method that involves concatenating multiple images to create a synthetic image. This process includes randomly selecting four different images, permuting their order randomly, scaling them to the same size, and finally concatenating them to create the Mosaic image. Mosaic Data Augmentation can simulate more complex scenarios where different objects may appear in the same image, helping train the model to better adapt to the diversity and complexity of the real world, thereby improving the model's robustness.

By applying the aforementioned data augmentation methods, the enhanced results are compared in Figure 7, as illustrated.

In addition, to further increase the diversity of the dataset, random rotation, flipping, translation, and other data augmentation methods were employed in this study to ensure the comprehensiveness and balance of the dataset. After data augmentation, the sample quantities in the dataset are shown in Table 2.

Table 2. Statistics of the number of samples for each pest category after data augmentation.

Insect Category	Number of Samples	Percentage
Boerner	1595	12.2%
Leconte	2216	17.0%
Linnaeus	1636	12.6%
acuminatus	1906	14.6%
armandi	1765	13.5%
colepotera	2091	16.1%
linnaeus	1818	14.0%



(**a**) (**b**) (**c**)

Figure 7. *Cont.*

(**d**) (**e**) (**f**)

Figure 7. Augmented data results image. (**a**) Original image; (**b**) random scaling and cropping image; (**c**) random brightness and contrast adjustments image; (**d**) copying regions of less frequent classes, enlarging, rotating, translating, and pasting them back into the original image; (**e**) Mosaic data augmentation image; (**f**) Mosaic data augmentation training annotation image.

4.2. Experimental Environment and Parameter Configuration

The hardware platform and environmental parameters used in the experimental training phase are shown in Table 3.

Table 3. Training environment and hardware platform parameters.

Parameter	Configuration
CPU	Intel(R) Core(TM) i5-10400F @2.90GHz
GPU	NVIDIA GeForce RTX 3060
GPU memory size	12G
Operating systems	Windows 10
Deep learning architecture	Python 3.8.8 + Cuda 11.7 + Pytorch 2.0.0
Model	YOLOv8n

The training parameters are set as shown in Table 4.

Table 4. The training parameter settings.

Parameter	Setup	Parameter	Setup
Epochs	250	Input image size	640 × 640
Initial learning rate	0.01	Optimizer	SGD
Final learning rate	0.0001	δ (WIoU v3)	1.9
batchsize	16	α (WIoU v3)	3

4.3. Evaluation Indicators

In order to assess the detection performance of our proposed improved model, we utilize precision, recall, average precision (AP), mAP0.5, mAP0.5:0.95, number of model parameter, model size, and detection speed as evaluation metrics. The calculation formulas for each evaluation metric are as follows:

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

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

$$\text{AP} = \int_0^1 \text{Precision}(\text{Recall})\text{d}(\text{Recall}) \tag{16}$$

$$mAP = \frac{\sum_{i=1}^{N} AP_i}{N} \quad (17)$$

where TP represents the number of true positive samples correctly identified as positive, TN represents the number of true negative samples correctly identified as negative, FP represents the number of false positive samples incorrectly identified as positive, FN represents the number of false negative samples incorrectly identified as negative, and AP is the area under the precision-recall curve. N represents the number of classes (in this paper, N is 7). mAP0.5 represents the mean average precision when the detection model's IoU threshold is set to 0.5, and mAP0.5:0.95 represents the mean average precision when the detection model's IoU threshold is set to 0.5–0.95 (with a 0.05 interval).

4.4. Experiment Results

4.4.1. Comparison with YOLOv8

To validate the effectiveness of the proposed network model, we conducted objective data comparative experiments and visual performance comparative experiments. The improved model was compared with the original YOLOv8 model.

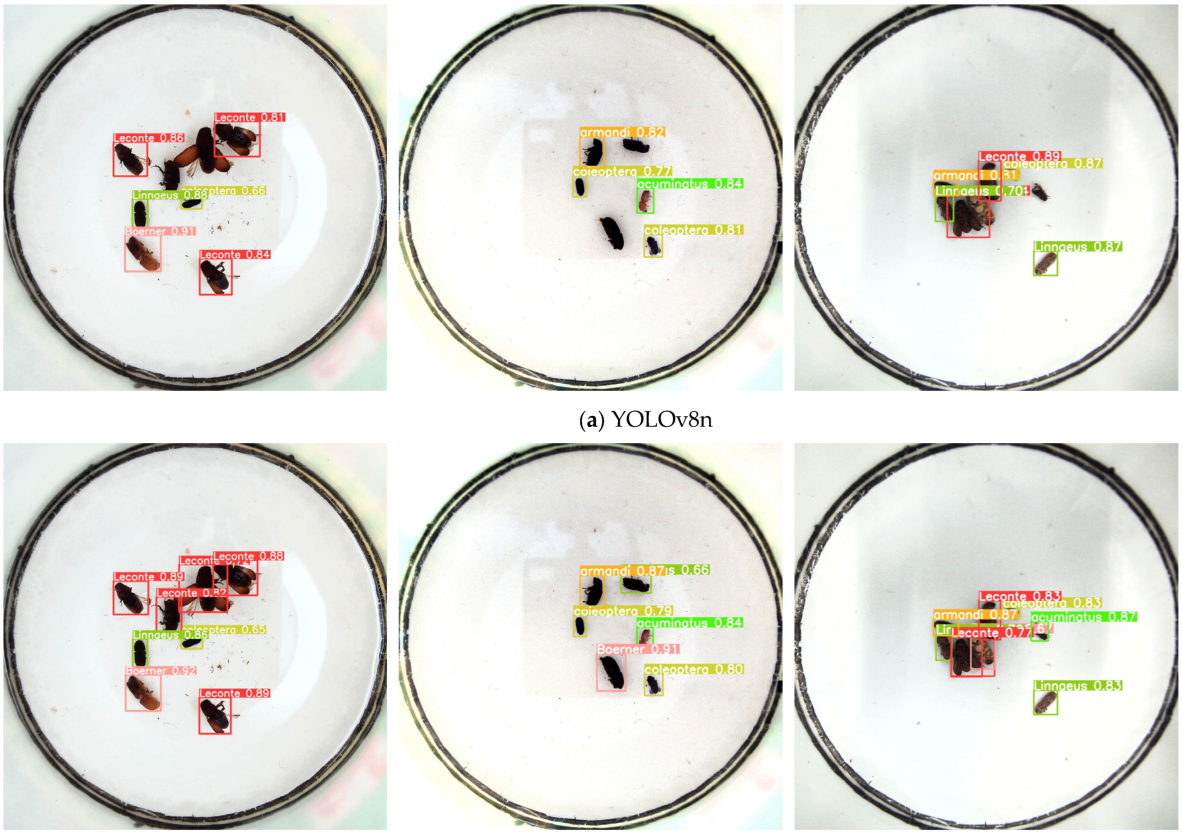
According to the results of objective data comparative experiments, as shown in Table 5, it can be observed that, compared to the original YOLOv8n model, the improved model in this study demonstrates superior detection accuracy, with a 1.5 percentage point increase in mAP, reaching a maximum AP of 99.5%. Furthermore, the improved model achieves a significant improvement in detection speed. Through the lightweight design of the neck component, the detection speed is increased by 13% compared to the original YOLOv8n model. In summary, the improved model in this study not only enhances detection accuracy but also improves detection speed, better meeting the performance requirements for real-time detection of forestry pests.

Table 5. Experimental data comparison between the proposed model and the original YOLOv8 model in this paper.

Model	AP/%							mAP/%	FPS
	Boerner	Leconte	Linnaeus	Acuminatus	Armandi	Coleoptera	Linnaeus		
YOLOv8n	99.0	98.9	97.3	97.8	97.7	97.8	97.1	97.9	81
Ours	98.2	99.3	99.2	98.7	98.7	98.4	99.5	98.9	93

On the test set, the visual comparison experiment results between the YOLOv8n model and the model proposed in this study are shown in Figure 8. From (a), it can be observed that when directly using the YOLOv8n model for detection, issues such as missed detection, false positives, and misclassification exist. In contrast, from (b), it can be seen that the model proposed in this study accurately detects each pest in the image, including occluded pest images, effectively addressing the aforementioned issues. Therefore, the detection capability of the model proposed in this study is superior.

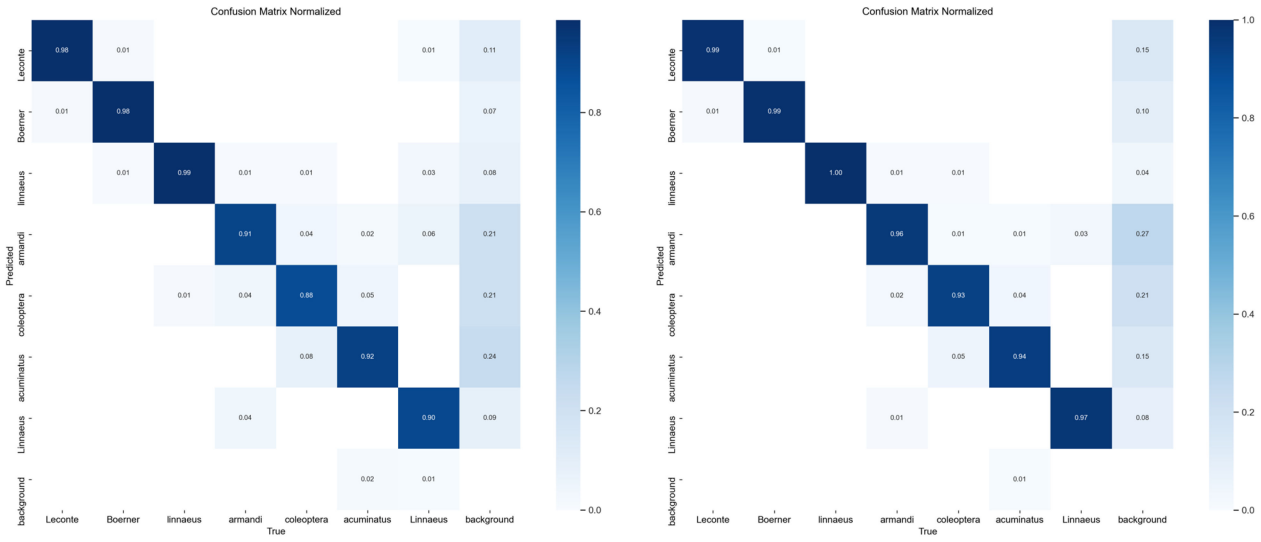
In order to visually demonstrate the performance of our method in predicting target categories, we generated a visual representation of the confusion matrix, as shown in Figure 9. The rows and columns of the confusion matrix correspond to the true and predicted categories, respectively. The values in the diagonal region represent the proportion of correctly predicted categories, while values in other regions indicate the proportion of incorrectly predicted categories. From Figure 10, it can be observed that in our algorithm, the color in the diagonal region of the confusion matrix is darker compared to YOLOv8n, indicating a significant enhancement in our model's ability to accurately predict object categories. However, for some small target pests such as armandi, coleoptera, acuminatus, there still exists a possibility of being misclassified as the background. Through improvements to the model, we have successfully reduced both the false-negative and false-positive rates for these categories, further enhancing the overall performance of the model.



(**a**) YOLOv8n

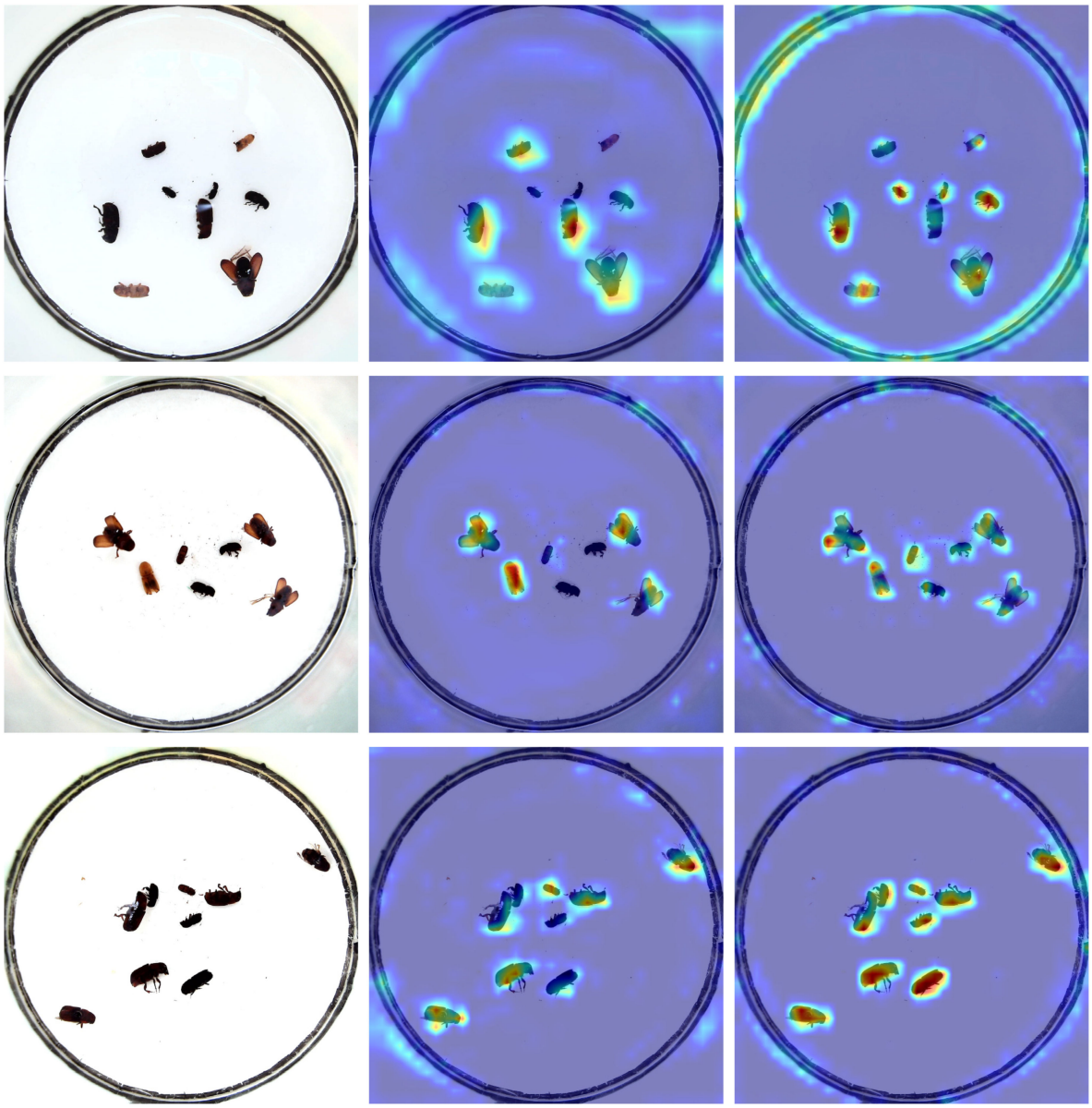
(**b**) Ours

Figure 8. Visual comparison of the original YOLOv8 Model and the model proposed in this paper. (**a**) YOLOv8n; (**b**) our model.



(**a**) (**b**)

Figure 9. (**a**) Confusion matrix plot of YOLOv8n; (**b**) confusion matrix plot of our model.



(**a**) original images (**b**) YOLOv8n (**c**) Ours

Figure 10. (**a**) Original images; (**b**) heat maps of YOLOv8n; (**c**) heat maps of our model.

4.4.2. Ablation Experiments

To comprehensively assess the effectiveness of the proposed forestry pest detection method and the individual contributions of the key enhancements, a series of meticulous ablation experiments were conducted in this study. These experiments systematically explored the influence of the Slim-Neck design, the integration of the CBAM attention mechanism, and the optimization of the WIoU loss function on model performance. The algorithm's accuracy and speed comparisons, with each improvement meticulously considered, are summarized in Table 6, where the presence of a $\surd$ signifies the incorporation of the respective enhancement point into the network.

The study leveraged Gradient-weighted Class Activation Mapping (Grad-CAM) [31] to produce heat maps for both YOLOv8n and our model. These heat maps serve as insightful visualizations, offering a clear depiction of the specific regions within the feature map that attract the model's attention. The gradients, derived through backpropagation of the model's confidence in output categories via Grad-CAM, are instrumental in crafting these heatmaps. Notably, pixels exhibiting higher gradients in the feature maps are vividly

illustrated with deeper shades of red, while those with lower gradients are elegantly rendered in deeper shades of blue. The experimental results and accompanying visual representations are depicted in Figure 10.

Table 6. Ablation experiment results.

Baseline	+Slim-Neck	+CBAM	+WIoU	Precision/%	mAP0.5/%	mAP0.5:0.95/%	Recall/%	FPS
YOLOv8n				95.6	97.9	78.6	94.5	81
	√			96.6	98.3	81.0	97.3	94
	√	√		97.2	98.6	81.4	97.4	89
	√	√	√	98.1	98.9	82.3	97.6	93

Due to the mature recognition capabilities of YOLOv8 and the relatively simple background of the pest dataset, the accuracy of the model is already high. As a result, the improvements made in this study may not show a significant increase in accuracy. However, the proposed methodology has indeed demonstrated enhanced model speed and improved recognition abilities for small targets within YOLOv8. This bears substantial research value in the field of forestry pest detection.

Through observation, it is evident that compared to YOLOv8n, our model demonstrates superior performance in detecting small targets, effectively addressing potential issues of missed detections in forestry pest detection. Our improvements are primarily manifested in the optimization of model architecture and algorithms, as robustly validated by experimental results. Specifically, the introduction of novel model structures, attention mechanisms, and loss functions has successfully enhanced the model's capability to recognize small-sized pests. This improvement not only reduces the occurrence of missed detections quantitatively, but also achieves a more precise visual effect.

Our model provides a more reliable and accurate solution for forestry pest detection. The outstanding performance is expected to yield significant benefits in practical applications, particularly in the efficient detection of small targets. These enhancements lay a solid foundation for improving the practicality and operability of the model in the field of forestry pest detection.

4.4.3. Comparison Experiment

To validate the superiority of our proposed model compared to current state-of-the-art forestry pest detection models, we conducted comparative experiments using mAP@0.5, Recall, and FPS as evaluation metrics. Our model was compared with Faster-RCNN, SSD, YOLOv5, and recent literature models under the same experimental conditions.

According to the data in Table 7, it is evident that, in terms of detection accuracy, our improved model demonstrates superior feature extraction capabilities through the introduction of attention mechanisms and enhanced loss functions. Compared to Faster-RCNN, SSD, YOLOv5, the lightweight YOLOv4 model from literature [16], and YOLOv4-TIA from literature [17], our approach exhibits stronger detection performance, better suited for the precise localization and identification of forestry pests. In terms of detection speed, our model, employing slim-neck design and Wise-IoU loss function, significantly reduces computational burden, resulting in a speed increase of 12FPS compared to the original YOLOv8 model. Overall, our model achieves a balance between accuracy and speed, outperforming other network models using the same dataset in the past two years, providing a high-precision and fast detection solution.

Table 7. Comparison between the model proposed in this paper and other models.

Model	mAP0.5/%	Recall/%	FPS/ (Frame s^{-1})
SSD	79.8	81.5	39
Faster-RCNN	86.5	84.3	17
Lightweight YOLOv4 [16]	93.7	92.9	76
YOLOv4-TIA [17]	84.5	91.2	65
YOLOv5	91.6	92.5	66
YOLOv8n	97.9	94.5	81
Ours	98.9	97.6	93

5. Conclusions

Given the current challenges in forestry pest detection methods, including large model parameters, slow detection speed, low accuracy, and issues such as missed detections, false positives, and false negatives, this study proposes a forestry pest detection model based on YOLOv8. To reduce computational complexity, the Slim-Neck approach is employed to reorganize the neck portion of the YOLOv8 network, addressing resource limitations and computational constraints for rapid deployment in forestry pest detection. Additionally, the Channel-wise and Spatial-wise Attention Mechanism (CBAM) is introduced into the backbone to enhance detection accuracy and overall performance while effectively addressing the challenges associated with missed detections and false positives for small-sized pests. The Wise-IoU improved loss function is integrated, along with a dynamic sample allocation strategy, reducing the model's focus on extreme samples and enhancing overall performance. Experimental results demonstrate that the proposed method exhibits efficient learning and high recognition accuracy, with the model achieving a maximum detection accuracy of 99.5%. Moreover, the detection speed surpasses that of other mainstream network models. Ablation experiments confirm that each improvement contributes to the enhancement of algorithmic performance. Visualization through confusion matrices and heatmaps illustrates a significant enhancement in the model's feature extraction capability, substantially improving the detection accuracy and overall performance for small targets. Comparative experiments with other mainstream network models demonstrate a balanced trade-off between accuracy and speed, providing a high-precision and efficient solution for forestry pest detection. Due to the singularity of the background of the experimental dataset, future work could further explore the model's scalability and generalization capabilities for better adaptation to diverse environments and scenarios.

Author Contributions: T.J. provided suggestions and made revisions to the manuscript. S.C. proposed and implemented the main ideas of this research and contributed to the writing of parts of the paper. T.J. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are contained within the article.

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

References

1. Zuo, Y.Z. *Pest Recognition System Based on Deep Learning*; Beijing Forestry University: Beijing, China, 2018.
2. Redmon, J.; Divvala, S.I.; Girshick, R.; Farhadi, A. You only look once:unified, realtime object detection. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 779–788.

3. Redmon, J.; Farhadi, A. YOLO9000: Better, Faster, Stronger. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 6517–6525.
4. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. Computer Vision and Pattern Recognition. *arXiv* **2018**, arXiv:1804.02767.
5. Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. YOLOv4: Optimal Speed and Accuracy of Object Detection. *arXiv* **2020**, arXiv:2004.10934.
6. Zhu, X.; Lyu, S.; Wang, X.; Zhao, Q. TPH-YOLOv5: Improved YOLOv5 Based on Transformer Prediction Head for Object Detection on Drone-captured Scenarios. In Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW), Montreal, BC, Canada, 11–17 October 2021.
7. Li, C.; Li, L.; Jiang, H.; Weng, K.; Geng, Y.; Li, L.; Ke, Z.; Li, Q.; Cheng, M.; Nie, W.; et al. YOLOv6: A Single-Stage Object Detection Framework for Industrial Applications. *arXiv* **2022**, arXiv:2209.02976.
8. Wang, C.Y.; Bochkovskiy, A.; Liao, H.Y.M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. *arXiv* **2022**, arXiv:2207.02696.
9. Reis, D.; Kupec, J.; Hong, J.; Daoudi, A. Real-Time Flying Object Detection with YOLOv8. *arXiv* **2023**, arXiv:2305.09972.
10. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.-Y.; Berg, A. SSD: Single Shot MultiBox Detector. In Proceedings of the Computer Vision—ECCV 2016, Lecture Notes in Computer Science, Amsterdam, The Netherlands, 11–14 October 2016; pp. 21–37.
11. Fu, C.Y.; Liu, W.; Ranga, A.; Tyagi, A.; Berg, A.C. Dssd: Deconvolutional single shot detector. *arXiv* **2017**, arXiv:1701.06659.
12. Li, Z.; Zhou, F. FSSD: Feature fusion single shot multibox detector. *arXiv* **2017**, arXiv:1712.00960.
13. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
14. Girshick, R. Fast R-CNN. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
15. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149. [CrossRef] [PubMed]
16. Sun, H.Y.; Chen, Y.B.; Feng, D.W.; Wang, T.; Cai, X.Q. Forest Pest Detection Method Based on Attention Model and Lightweight YOLOv4. *Comput. Appl.* **2022**, *42*, 3580–3587.
17. Hou, R.H.; Yang, X.W.; Wang, Z.C.; Gao, J. A real-time detection methods for forestry pests based on YOLOv4-TIA. *Comput. Eng.* **2022**, *48*, 255–261.
18. Song, H.; Willi, M.; Thiagarajan, J.J.; Berisha, V.; Spanias, A. Triplet network with attention for speaker diarization. *Proc. Interspeech* **2018**, *2018*, 3608–3612. [CrossRef]
19. Li, H.; Li, J.; Wei, H.; Liu, Z.; Zhan, Z.; Ren, Q. Slim-neck by GSConv: A better design paradigm of detector architectures for autonomous vehicles. *arXiv* **2022**, arXiv:2206.02424.
20. Woo, S.; Park, J.; Lee, J.Y.; Kweon, I.S. CBAM: Convolutional Block Attention Module. In Proceedings of the Computer Vision—ECCV 2018, Cham, Switzerland, 8–14 September 2018; pp. 3–19.
21. Tong, Z.; Chen, Y.; Xu, Z.; Yu, R. Wise-IoU: Bounding Box Regression Loss with Dynamic Focusing Mechanism. *arXiv* **2023**, arXiv:2301.10051.
22. He, K.; Zhang, X.; Ren, S.; Sun, J. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. *IEEE Trans. Pattern Anal. Mach. Intell* **2015**, *37*, 1904–1916. [CrossRef] [PubMed]
23. Liu, S.; Qi, L.; Qin, H.; Shi, J.; Jia, J. Path Aggregation Network for Instance Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 8759–8768.
24. Li, X.; Wang, W.; Wu, L.; Chen, S.; Hu, X.; Li, J.; Tang, J.; Yang, J. Generalized Focal Loss: Learning Qualified and Distributed Bounding Boxes for Dense Object Detection. *arXiv* **2020**, arXiv:2006.04388.
25. Zheng, Z.; Wang, P.; Liu, W.; Li, J.; Ye, R.; Ren, D. Distance-IoU loss: Faster and better learning for bounding box regression. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; pp. 12993–13000.
26. Priyantha, N.; Balakrishnan, H.; Demaine, E.D.; Teller, S. Anchor-Free Distributed Localization in Sensor Networks. In Proceedings of the International Conference on Embedded Networked Sensor Systems, Los Angeles, CA, USA, 5–7 November 2003.
27. Chollet, F. Xception: Deep learning with depthwise separable convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 1800–1807.
28. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *arXiv* **2017**, arXiv:1704.04861.
29. Zhang, X.; Zhou, X.; Lin, M.; Sun, J. ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices. *arXiv* **2017**, arXiv:1707.01083.

30. Zhang, H.; Wang, Y.; Dayoub, F.; Sünderhauf, N. VarifocalNet: An IoU-aware Dense Object Detector. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021.
31. Selvaraju, R.R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; Batra, D. Grad-CAM: Visual Explanations from Deep Networksvia Gradient-Based Localization. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 618–626.