IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 27 November 2023, accepted 11 January 2024, date of publication 19 January 2024, date of current version 26 January 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3356048

RESEARCH ARTICLE

GBCD-YOLO: A High-Precision and Real-Time Lightweight Model for Wood Defect Detection

YUNCHANG ZHENG, MENGFAN WANG, BO ZHANG, XIANGNAN SHI, AND QING CHANG
Hebei University of Architecture, Zhangjiakou, Hebei 075000, China

Corresponding author: Qing Chang (cqing220@163.com)

This work was supported in part by the Basic Scientific Research Business Fund Project of Universities in Hebei Province under Grant 2022QNJS03, and in part by the Project of Zhangjiakou Science and Technology Bureau under Grant 2121029C.

ABSTRACT With the advancement of the wood processing industry, the demand for the detection of surface defects in wood has become increasingly urgent. The application of automated production technology has enhanced the efficiency and precision of wood processing, which can significantly impact product quality and competitiveness. However, current methods for detecting surface defects in wood suffer from issues such as low detection accuracy, high computational complexity, and poor real-time performance. In response to these challenges, this paper proposes a high-precision, lightweight, real-time wood surface defect detection method based on YOLO(GBCD-YOLO) model. Firstly, the Ghost Bottleneck is introduced to improve the computational efficiency and inference speed of deep neural networks. Furthermore, the BiFormer is incorporated in the neck to enhance the performance of natural language processing tasks. Simultaneously, CARAFE is utilized as an upsampling replacement to enhance perceptual and capture abilities for details. In addition, the Dynamic Head is introduced to enhance the method's flexibility and generalization ability, and the loss function is replaced with complete intersection over union (CIoU). The proposed method was evaluated using an optimized dataset and the YOLOv5s model was chosen as the baseline. The experimental results show that compared with the original YOLOv5s, the mAP (0.5) has been improved by 13.45%, reaching 88.72%. The mAP (0.5:0.95) increased by 11.95%, and FPS increased by 6.25%. In addition, the parameter of the improved model has been reduced by 15.49%. These results indicate that the proposed GBCD-YOLO improves the real-time detection performance of wood surface defects.

INDEX TERMS Small target detection, wood defect, deep learning, transformer, YOLOv5.

I. INTRODUCTION

Surface defects in wood have a significant impact on wood processing, and the smoothness of the wood surface also affects its value. However, due to the different growth environments and processing methods of natural wood, surface defects such as cracks, dead knots, live knots, and resin are inevitable. The identification of surface defects in wood has important implications for assessing wood quality and assisting in fine wood processing. At present, the categorization of items within the wood board manufacturing process continues to depend on human expertise, resulting in challenges such as excessive workload, limited examination efficacy, and an elevated proportion of erroneous identifications. Therefore, developing an efficient and accurate method of detecting wood surface defects is an effective measure to improve the quality of wood board production.

The advancement of artificial intelligence has propelled object detection technology based on deep learning to remarkable heights in diverse sectors, including production and processing. Deep learning algorithms have become essential tools in the field of wood surface defect detection, which is currently receiving more attention because of its practical significance.

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

In the realm of computer vision algorithms applied to wood surface defect detection, two primary types have been utilized: two-stage and single-stage detection methods. In the early stages of object detection, the predominate approach involved a two-stage process. A typical representative is the R-CNN series of methods, such as R-CNN [1], Fast R-CNN [2], Faster R-CNN [3], and Mask R-CNN [4]. These techniques start with a set of candidate boxes generated by region creation algorithms, which are subsequently put through feature extraction and classification procedures. The two-stage detection methods gained prominence in the field of object detection due to their high accuracy. On the other hand, single-stage detection techniques have begun to appear with the introduction of deep learning, providing a fresh perspective on object detection problems. Representative options for single level detection algorithms include YOLO series (such as YOLOv1 [5], YOLO9000 [6], YOLOv3 [7], YOLOv4 [8], YOLOv5 [9]), SSD series (such as SSD [10], R-SSD [11], FSSD [12], and DSSD [13]), and EfficientDet [14]. The single-stage algorithm uses a single neural network to complete the entire detection process. The advantage of a single stage detector is its fast-processing speed, which is suitable for real-time applications. Both single-stage and two-stage algorithms play important roles in the field of object detection.

In the past few years, many researchers have investigated methods for detecting various surface defect objects on solid wood using deep learning. Fan et al. [15] utilized Faster R-CNN for detecting defects in solid wood, achieving a detection accuracy of 95% for woodworm holes. Nevertheless, the two-stage detection algorithm based on Faster R-CNN sacrifices real-time performance, thereby improving the accuracy of the algorithm but failing to fulfill the realistic real-time demands. With a 91.17% identification accuracy, Chacon and Alons [16] employed a fuzzy self-organizing neural network as a classifier for the detection of wood surface defects. Although accuracy meets requirements, detection time cannot yet satisfy the demands of the production line. Qi and Mu [17] obtained internal images of wood using X-ray detection and utilized artificial neural network algorithms to identify wood defects. In contrast, the modeling process of artificial neural networks is complex, with a high number of training parameters and long training time. Ye [18] employed the LBP feature extraction algorithm to detect decay, woodworm, and indentation defects on solid wood surfaces. Unfortunately, the resulting detection box regions suffer from misclassification and omission issues. Tsung-Yi et al. introduced RetinaNet in 2017 [19], which addresses the issue of imbalanced positive and negative samples in object detection by incorporating a loss function named focal loss. Furthermore, Reis and colleagues proposed YOLOv8 [20] in 2023, which builds upon the YOLO algorithm. YOLOv8 enhances the feature expression in detecting small targets, employing a multi-scale fusion approach and incorporating a data augmentation strategy.

However, as shown in FIGURE 1, the current research on wood defects, particularly complex texture and blurry clarity defects, poses a number of challenges. In particular, there is relatively little research on improving detection speed and real-time algorithms. It is crucial to address this issue, as rapid and accurate detection of wood defects is essential for the efficient operation of production lines.

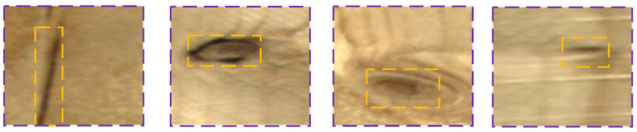

FIGURE 1. Typical wood defects.

The main contributions of this article can be summed up as follows in order to address the problems mentioned above:

1. In order to enhance computational efficiency and inference speed and achieve real-time performance, this paper introduces a lightweight Ghost Bottleneck to reduce computational burden and resource consumption while maintaining high precision.
2. BiFormer [21], which combines Transformers and feature pyramids in the neck, was incorporated to improve the performance of natural language processing tasks, extracting more spatial feature information and further enhancing detection accuracy.
3. Traditional upsampling operations have been replaced with the CARAFE [22], a superior upsampling method that focuses on capturing fine-grained details in images, preserving and refining them to enhance the model's ability to detect subtle defects.
4. To enhance model flexibility and generalizability, we introduce a Dynamic Head [23] structure, allowing for dynamic adjustments based on the shape and scale of the target. This enables the effective detection of wood defects of different sizes and shapes.
5. In order to improve the sensitivity to small objects and accurately reflect the similarities between targets, the loss function was replaced with complete intersection over union (CIoU) [24].

The rest of this paper is organized as follows: Section II introduces the original YOLOv5s model and its improvements, which is the proposed GBCD-YOLO. Section III introduces the experiment preparation and analyses the results of the experiment. Finally, the main conclusions are drawn in Section IV.

II. PRINCIPLE AND METHOD IMPROVEMENT

A. NETWORK STRUCTURE OF YOLOv5

The YOLO algorithm has become one of the most famous object detection algorithms due to its efficiency, high accuracy, and lightweight. YOLOv5 was launched by Glenn Jocher in May 2020 using the Python framework. Currently, the YOLO family has launched YOLOv8 in January 2023. However, YOLOv8 is not ideal for detecting low resolution

TABLE 1. Parameter settings for depth and width of YOLOv5 model of four different models.

Parameters	YOLOv5s	YOLOv5m	YOLOv5l	YOLOv5x
Depth	0.33	0.67	1.00	1.35
Width	0.50	0.75	1.00	1.25

objects, as it may not accurately detect small objects and requires a large number of samples and computing resources during training. Moreover, considering the outstanding practical performance of YOLOv5, we have decided to further enhance it by building on its existing framework. Currently, there are four versions of the YOLOv5 network models: YOLOv5s, YOLOv5m YOLOv5l and YOLOv5x. The depth and width parameters of four YOLOv5 versions are shown in TABLE 1. YOLO v5s is the model with the smallest network depth (the number of layers in network architecture) and width (the number of parameters and computational load of the network), while the other three models are products that have been deepened and expanded on the basis of YOLOv5s. YOLOv5s balances model size and accuracy to make it suitable for scenarios with limited computing resources or critical real-time performance.

In this paper, YOLOv5s was selected as the baseline. The image input, backbone network, neck and output are the four main parts of the YOLOv5s network model. Image input is the initial stage of passing input images into the model. The backbone network forms the core of the architecture and extracts hierarchical features from the input images. The neck component connects the backbone network and the output layer for further fusion and feature refinement. Finally, the output detection module generates predictions by processing the refined features of the neck component. To achieve a balance between accuracy and speed, YOLOv5s has been carefully designed for real-time applications with limited computing resources. It adopts advanced technologies such as anchor free detection and focus loss to improve object detection performance. In YOLOv5s, the anchor box loss function is a generalized intersection over union (GIoU) [25], which improves the accuracy of anchor box prediction. Additionally, a weighted non-maximum suppression (NMS) [26] operation is employed to filter target anchor boxes, further enhancing the precision of object detection. These modifications contribute to the overall effectiveness and reliability of the output detection head in YOLOv5s.

B. THE IMPROVED YOLO MODEL(GBCD-YOLO)

The initial YOLOv5s algorithm used a large number of Conv (convolutional layer) and Conv3 × 3(convolutional layers with 3 × 3 filters) convolutional structures in its backbone network and feature pyramid. Therefore, it leads to high parameter counts and slower detection speeds. However, in regard to the actual production of wood panels in factory workshops, achieving defect detection becomes a very challenging.

In order to meet the requirements for rapid detection of defects on the surface of wood, the proposed GBCD-YOLO method has made the following improvements to YOLOv5s: Firstly, the lightweight Ghost Bottleneck is introduced to improve the computational efficiency and inference speed of deep neural networks. Furthermore, the BiFormer is incorporated in the neck to enhance the performance of natural language processing tasks. Simultaneously, CARAFE is utilized as an upsampling replacement to enhance perceptual and capture abilities for details. In addition, the Dynamic Head is introduced to enhance the method's flexibility and generalization ability, and the loss function is replaced with CIoU.

Overall, the proposed method combines dataset construction, model architecture modification, loss function optimization, and integration of detection method to enhance the detection of wood surface defects.

1) LIGHTWEIGHT GHOST BOTTLENECK

The lightweight Ghost Bottleneck is a technique used to optimize neural network architecture. Reduce the complexity of the model by introducing ghost convolutional layers and bottleneck structures, thereby reducing computational and memory consumption while maintaining performance.

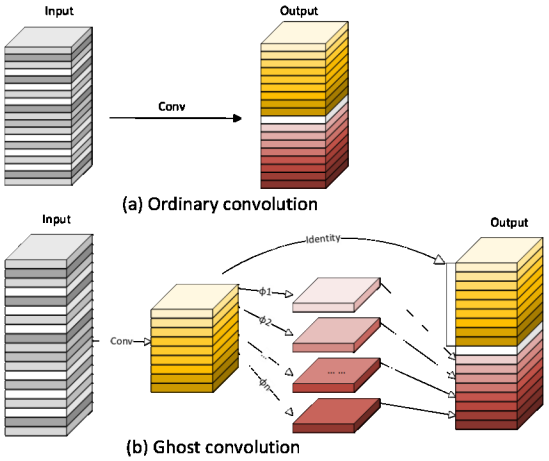

FIGURE 2. The ordinary convolution and the ghost convolution.

Compared to traditional convolution, the Ghost Net [27] follows a two-step process. As depicted in FIGURE 2, Ghost Net first utilizes regular convolution to obtain feature maps with reduced channels. Then, it employs a cost-effective operation to generate additional feature maps. A new output is created by concatenating these various feature maps. The Ghost module's main convolution allows for customized kernel sizes, unlike the widely used 1 × 1 pointwise convolution. After processing features across channels using pointwise convolution, depth convolution is used to handle spatial information. Additionally, to reduce the computational requirements, the Ghost module uses the concatenation approach. Instead of using a bottleneck structure, our work

uses a lightweight Ghost Bottleneck. The FIGURE 3 depicts the Ghost construction that was replaced.

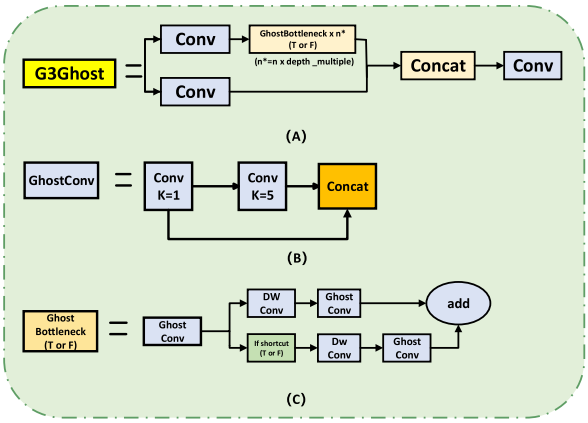

FIGURE 3. Replaced Ghost series modules: (A) C3Ghost (B) GhostConv (C) Ghost Bottleneck.

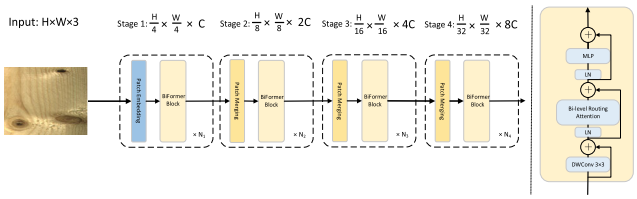

FIGURE 4. The overall architecture of BiFormer and details of a BiFormer block.

2) BIFORMER AND BI-LEVEL ROUTING ATTENTION MECHANISM

The visual Transformer model BiFormer was proposed by Zhu et al. [21]. As shown in FIGURE 4, it is a new universal visual converter BiFormer. Follow the latest visual converters [28], [29], [30] and uses a four-level pyramid structure. Specifically, in the stage i, overlapping patch embedding is used, and in the second to fourth stages, patch merging modules [31], [32] are used to reduce the input spatial resolution while increasing the number of channels. Then, Ni consecutive BiFormer blocks are used to transform features. In each BiFormer block, use 3 at the beginning × 3 deep convolutions to implicitly encode relative position information. Then, the BRA module with an extension ratio of e and the 2-layer MLP module are sequentially applied for cross positional relationship modeling and positional embedding, respectively.

In the visual converter, BiFormer implements a two-layer routing attention mechanism. By allowing information interaction between the local and global attention levels, it improves feature representation. While the local attention level is used to capture the image's specifics and local features, the global attention level is used to capture the image's overall structure and global semantic information. Better management of global and local relationships in images is made possible by the introduction of this bi-level routing attention mechanism, which successfully captures the structural and detailed aspects of the image. Dramatically increases the effectiveness of image classification tasks.

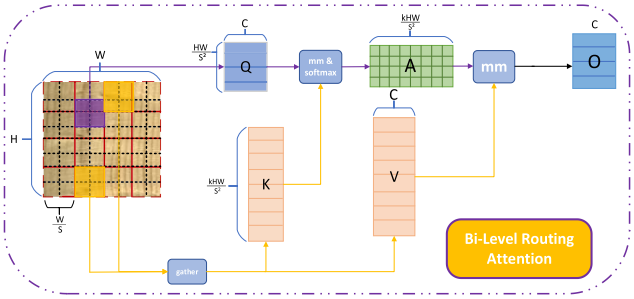

FIGURE 5. The structure of a Bi-level routing.

As shown in FIGURE 5, the query used to calculate the keywords' weighted relevance is Q. K stands for the keywords or identifiers that are used as a means of matching or information provision. V is the value linked to the keyword data and the query results. The factor C is employed to modify attention allocation and regulate attention focus. The adjacency matrix A is used to show how semantically relevant two regions are to each other. O is the output of the attention mechanism.

A convolutional neural network is used by the BiFormer model to first extract features from the input image during training, producing a feature map with dimensions of $H \times W \times C$. Subsequently, along the channel dimension C, this feature map is divided into r sub-feature maps of size $H \times W \times C/r$.

In the global attention part, the BiFormer model maps each sub-feature map to a vector of dimension d through two fully connected layers. To get a weight vector, it computes how similar this vector is to every other vector. The global relationship is then represented by the output vector, which is obtained by weighting and summing the sub-feature maps in accordance with this weight vector.

In the local attention part, the BiFormer model creates a local feature map by applying two convolutional layers to each sub-feature map. After that, another weight vector is obtained by calculating the similarity between this feature map and all other feature maps. An output vector that represents the local features is produced by weighting and adding the feature maps in accordance with this weight vector.

In brief, every sub-feature map is converted into a semantic vector that symbolizes the global relationship through the application of the global routing attention mechanism. Local features are simultaneously represented by a semantic vector created by the local routing attention mechanism. The output vectors of local attention and global attention are finally combined by the BiFormer model. These vectors are mapped to the target object's class and position data via fully

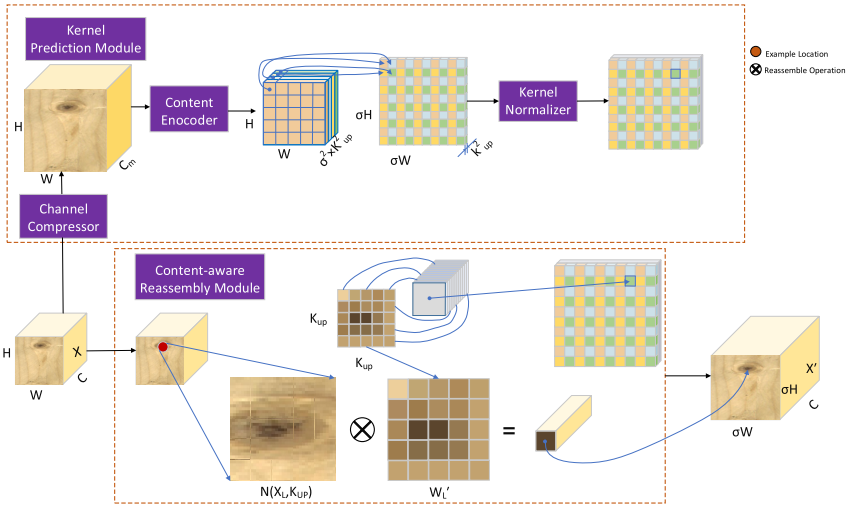

FIGURE 6. The overall framework of CARAFE.

connected layers. The target object's class and coordinates are provided by the probability output, which is produced by applying the softmax function.

The BiFormer model has been shown through a number of studies to be effective at detecting small objects [33], [34]. It is able to concurrently focus on the local characteristics and global relationships of the detection target because of the introduction of the Bi-level routing attention mechanism. This makes it possible to distinguish defective targets from the background more effectively.

In summary, the dataset utilized in this study and the BiFormer model have a high degree of compatibility, which should lead to a notable increase in accuracy.

3) CARAFE: LIGHTWEIGHT UPSAMPLING OPERATOR

Feature recombination was used to perform the upsampling. This entails taking the dot product between the upsampling kernel and the corresponding neighboring pixels in the feature map input. Given its small receptive field, the original network structure ignores some important information. Therefore, it is necessary to enlarge the receptive field. On the other hand, the reorganization process can be guided by the input features and a larger receptive field thanks to the CARAFE upsampling operation. Additionally, the compact nature of the CARAFE operator structure satisfies the need for lightweight models. In particular, the input feature map is used to predict distinct upsampling kernels for every position. These predicted kernels are then used to perform feature recombination. CARAFE adds minimal extra parameters and computational overhead while achieving notable performance gains across a range of tasks.

As seen in FIGURE 6, CARAFE is made up of two main modules: the feature recombination module and the upsampling kernel prediction module. Using the upsampling kernel prediction module, the process begins by predicting the upsampling kernel, assuming an upsampling multiplier of σ and an input feature map with dimensions $H \times W \times C$. After that, the upsampling process is carried out by the feature recombination module, producing an output feature map with dimensions of $\sigma H \times \sigma W \times C$.

The first step is to perform channel compression, which reduces the number of channels to Cm using a 1×1 operation, starting with an input feature map of size $H \times W \times C$. This compression is meant to lighten the computational burden in later stages. CARAFE then moves on to predict the upsampling kernel and encode the content. It is crucial to remember that although a larger kernel size results in a wider perceptual field, it also has higher computational costs, assuming a specific upsampling kernel size of $k_{up} \times k_{up}$. In order to apply distinct upsampling kernels to every position in the output feature map, CARAFE must forecast the upsampling kernel's shape as $\sigma H \times \sigma W \times k_{up} \times k_{up}$. Initially, a convolutional layer with $k_{encoder} \times k_{encoder}$ channels is used to predict the upsampling kernel after the input feature map has been compressed. There are Cm channels in the input and $\sigma^2 k_{up}^2$ channels in the output. Next, CARAFE produces an upsampling kernel with the shape $\sigma H \times \sigma W \times k_{up}^2$ is produced by CARAFE expanding the channel dimension across the spatial dimension.

CARAFE created a mapping from each position on the output feature map to the matching area on the input feature map. This area, which was centered at the position, measured $k_{up} \times k_{up}$. The output value was then produced by CARAFE by computing the dot product between this region and the anticipated upsampling kernel linked to that particular position. It is important to note that the same upsampling kernel was used by different channels at the same position.

4) DYHEAD: DETECTION HEAD BASED ON ATTENTION MECHANISM

FIGURE 7 illustrates the attention-based detection head, known as DyHead. DyHead introduces an innovative

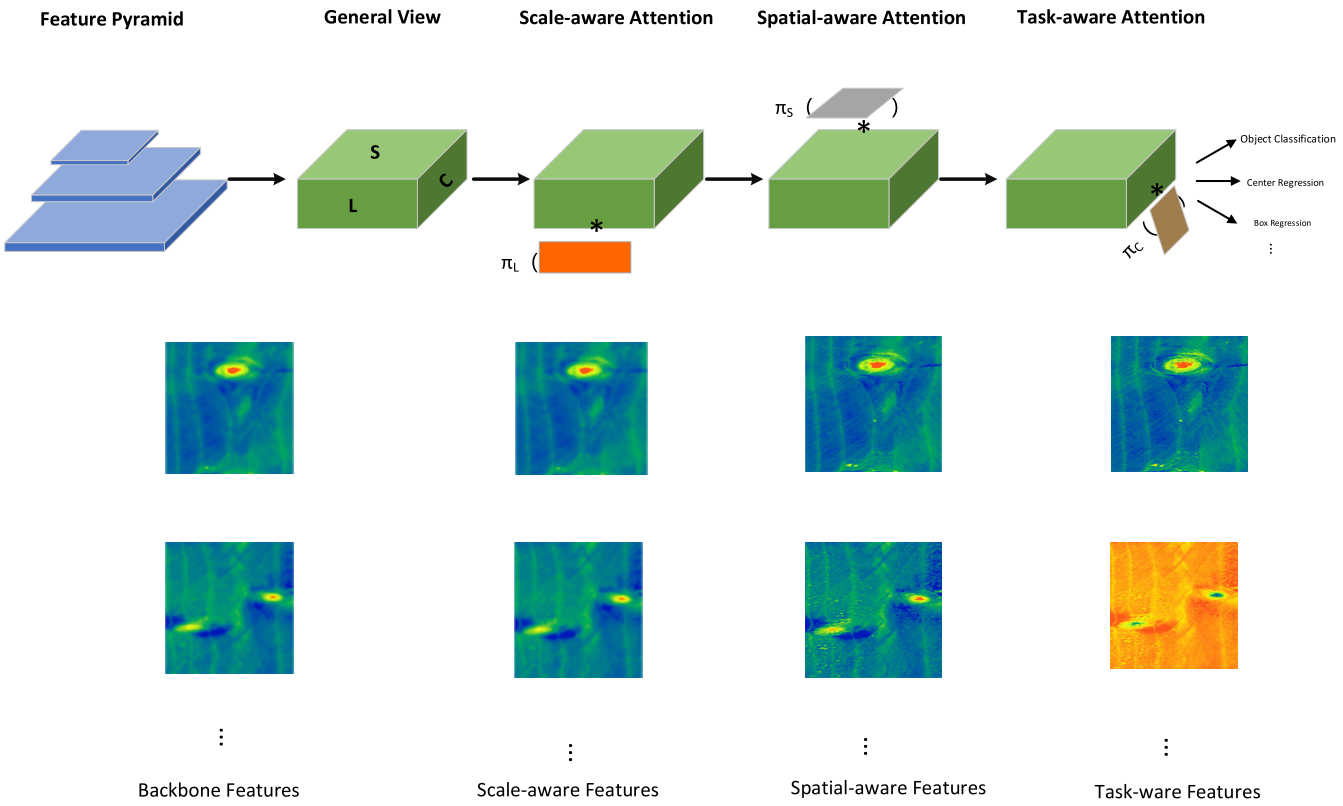

FIGURE 7. The illustration of dynamic head approach.

framework that incorporates attention mechanisms to unify different object detection heads. This method effectively increases the performance of the object detection head of the model without adding more computational overhead by integrating scale perception at the feature level, spatial perception at the spatial position, and attention between output channels for task perception.

Scale-aware attention (π_L), spatial attention (π_S), and channel attention (π_C) are the three attention mechanisms combined to form DyHead. The stacking of these attention mechanisms results in the creation of a single block. This stack of attention mechanisms is incorporated into each of the multiple blocks that make up the final detection head.

Given the feature tensor $F \in R^{L\times H\times W\times C}$, the generalized form of self-attentiveness is best described by DyHead as follows:

$$W(F) = \pi(F) \times F \tag{1}$$

where $\pi(\cdot)$ is an attention function. Using fully connected layers is one direct approach. However, directly learning attention functions on all dimensions will lead to high computational requirements and become impractical due to its high dimensionality. To solve this problem, the attention function is decomposed into three sequential attention functions, each targeting a specific dimension:

$$W(F) = \pi_C\left(\pi_S\left(\pi_L(F) \times F\right) \times F\right) \times F \tag{2}$$

Scale-aware attention π_L: In order to solve the fusion problem of different scale features based on semantic meaning, scale aware attention was first introduced:

$$\pi_L(F) \times F = \sigma\left(f\left(\frac{1}{SC}\sum_{S,C} F\right)\right) \times F \tag{3}$$

In this case, $f(\cdot)$ corresponds to using 1×1 convolutional approximation of linear functions, while $\sigma(\cdot)$ represents a hard S-shaped activation function.

Spatial Perceived Attention π_S: In order to highlight the capacity to distinguish between different spatial positions, a second module named Spatial Position Perceived Attention was added to the exploration. S is decoupled into two stages due to its large scale: first, sparse attention learning is achieved through the use of deformable convolution, and this is then completed by integrating features of different scales:

$$\pi_S(F) \times F = \frac{1}{L}\sum_{i=1}^{L}\sum_{i=1}^{K} w_{l,k} \times F\left(l; p_k + \Delta_{p_k}; c\right) \times \Delta_{m_k} \tag{4}$$

In this equation, $p_k + \Delta_{p_k}$ is a shifted location by the self-learned spatial offset Δ_{p_k} to focus on a discriminative region, K represents the count of sparsely selected positions. The remaining parameter details are similar to those in deformation convolution, where $w_{l,k}$ denotes a bias importance factor

and Δ_{m_k} stands for adaptive weighting importance factor, which is excluded here for conciseness.

Task Perceived Attention π_C: In order to promote collaborative learning and enhance the scalability of target representation ability, a task aware attention mechanism was designed. This attention mechanism helps with different tasks by dynamically adjusting feature channels as needed:

$$\pi_C(F) \times F = \max\left(\alpha^1(F) \times F_c + \beta^1(F), \alpha^2(F) \times F_c + \beta^2(F)\right) \tag{5}$$

As with DyReLU [35], hyperparameters are essential for regulating activation thresholds. α and β were used for rescaling and reorienting, respectively. Multiple instances of the previously mentioned attention mechanism can be stacked by applying it successively. FIGURE 8 shows the configuration of the dynamic head and provides one-stage detector and two-stage detectors.

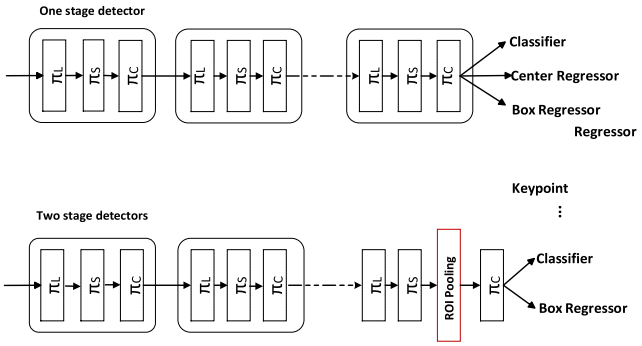

FIGURE 8. The application of dynamic head blocks to one-stage and two-stage object detector.

5) CIOU LOSS FUNCTION

In the original YOLOv5, prediction boundary boxes are handled using the GIoU, which effectively handles situations in which the predicted box does not intersect the actual box. However, the intersection position is not accurately reflected by GIoU when two boxes are contained within one another or have different aspect ratios because it is unable to accurately determine their relationship. Regression loss function known as CIoU is used in our study to address these shortcomings. The penalty function introduced by CIoU takes into account the aspect ratio, the size of the overlapping area, and the separation between the center points. Improved prediction accuracy is achieved by stabilizing the target bounding box regression through the incorporation of CIoU.

As shown in FIGURE 9, the distance between the center points of the predicted box and the ground truth is represented by ρ, and the diagonal distance of the mini-maximum enclosing area that contains both the prediction box and the ground truth is represented by the symbol c. The definition of

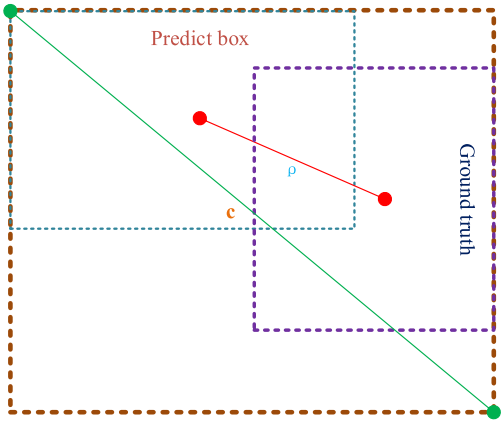

FIGURE 9. The CIoU loss function for bounding box regression.

the CIoU loss function is as follows:

$$IoU = IoU - \frac{\rho^2(b, b^{gt})}{c^2} - \alpha v \tag{6}$$

$$v = \frac{4}{\pi^2}\left(\arctan\frac{w^{gt}}{h^{gt}} - \arctan\frac{w}{h}\right)^2 \tag{7}$$

$$\alpha = \frac{v}{(1 - IoU) + v} \tag{8}$$

$$LossCIoU = 1 - IoU + \frac{\rho^2(b, b^{gt})}{c^2} + \alpha v \tag{9}$$

where, respectively, b and b^{gt} stand for the center points of ground truth box B^{gt} and prediction Box B. A parameter labeled as v is used to measure the similarity of aspect ratios, and the weight parameter is represented by α.

6) THE GBCD-YOLO MODEL

FIGURE 10 illustrates the improved GBCD-YOLO method's structure. Firstly, the lightweight Ghost Bottleneck is introduced to improve the computational efficiency and inference speed of deep neural networks. Furthermore, the BiFormer is incorporated in the neck to enhance the performance of natural language processing tasks. Simultaneously, CARAFE is utilized as an upsampling replacement to enhance perceptual and capture abilities for details. In addition, the Dynamic Head is introduced to enhance the method's flexibility and generalization ability, and the loss function is replaced with CIoU.

III. EXPERIMENTS AND DISCUSSION

As shown in FIGURE 11, the experimental procedure is divided into three parts: dataset creation, network training, and object detection. Initially, this paper used image preprocessing methods to collect images and annotate them according to experimental needs to create a comprehensive dataset. Following that, we obtained the algorithm's model weights through parameter refinement and model training. Finally, object detection was performed using carefully trained weights and the detection results of various methods were compared.

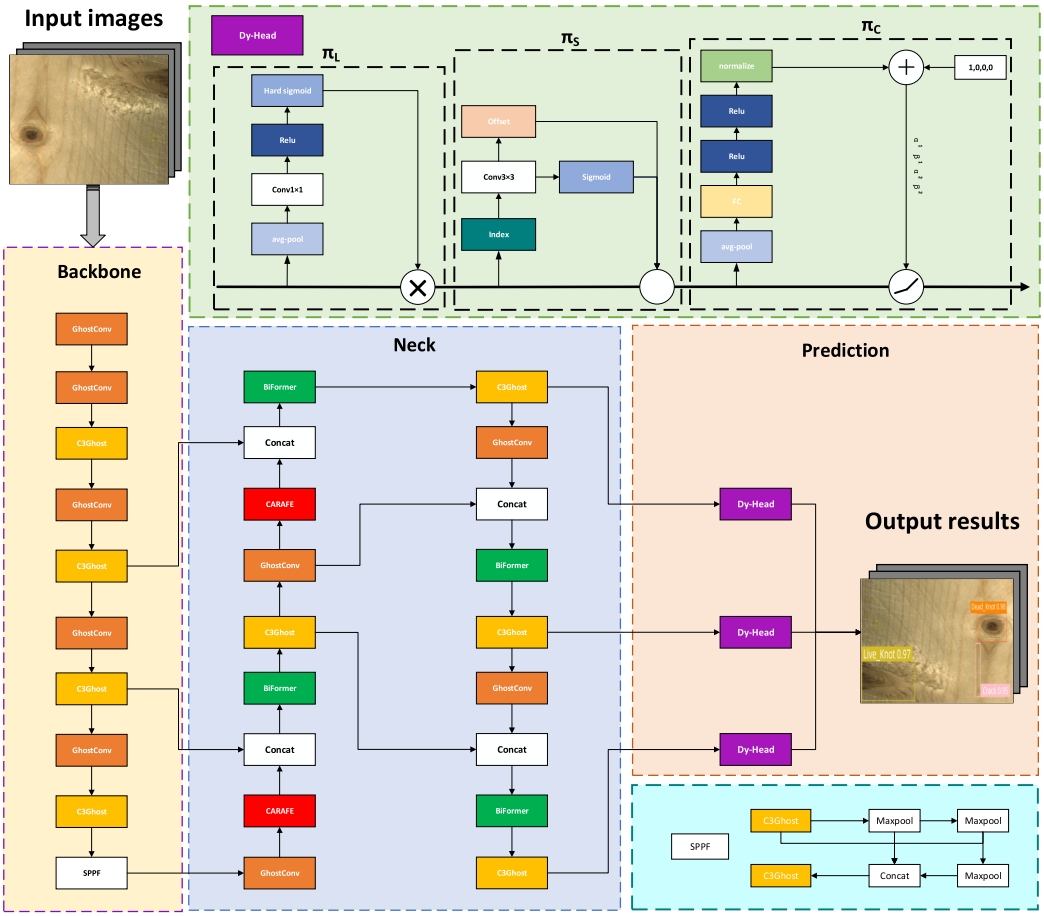

FIGURE 10. The GBCD-YOLO model.

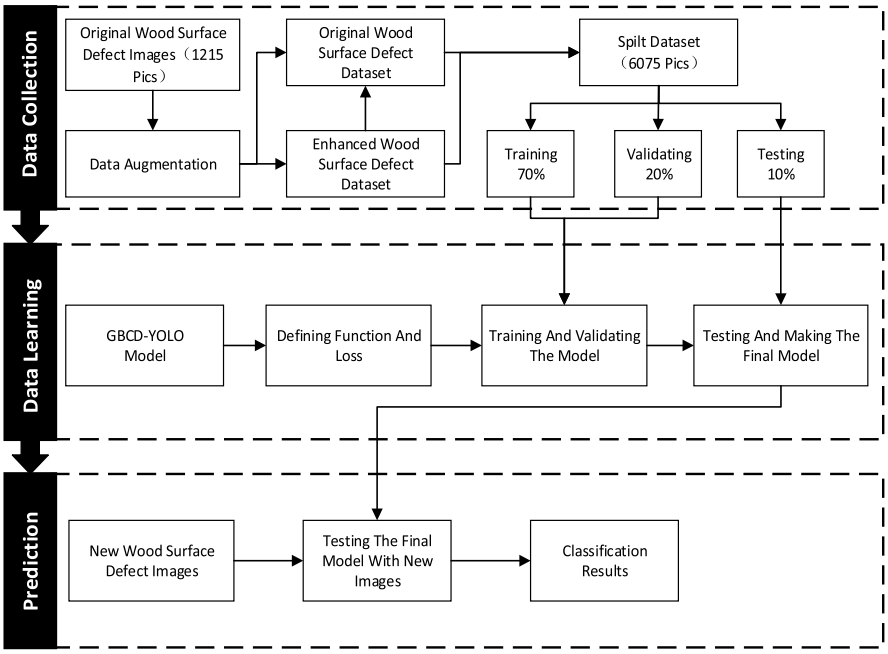

FIGURE 11. The flow chart of experimental process.

A. DATASET AND PREPROCESSING

According to our cooperation plan with the local wood processing factory, this paper has selected 756 wood surface defect images from the workshop. To ensure the diversity of the dataset, we also screened an additional 459 wood surface defect datasets from academic papers [36]. Finally, the two images were merged to create our new dataset, with a total of 1215 images, some example images are shown in FIGURE 12.

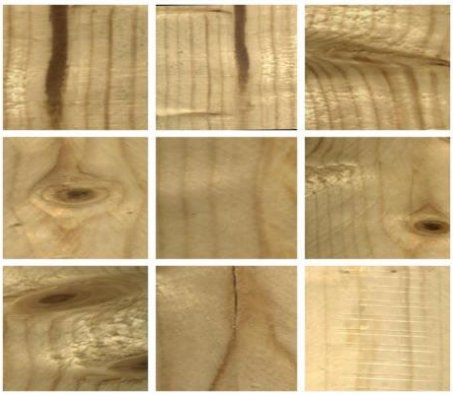

FIGURE 12. Sample images extracted from the wood defect dataset.

At the same time, to solve the problem of sample imbalance, image pre-processing methods were used for each data group. These methods include image enhancement, mirroring, filtering, and grayscale conversion. After the image pre-processing stage, we planned a custom dataset consisting of 6075 images, including both original and preprocessed images, as shown in FIGURE 13.

To better evaluate detection performance, the dataset is annotated using LabelImg in four categories: cracks, knots, dead knots, and resins. In addition, we converted the TXT files to XML files labeled as VOC format. Finally, we obtained a VOC format dataset of wood surface defect images, as shown in TABLE 2 for various types of wood defects. Then, these images were randomly assigned to training, validation, and testing in a ratio of 7:2:1.

B. NETWORK TRAINING

Before the experiments, a sophisticated experimentation and development platform is established utilizing Windows 10 (64-bit). The GPU boasts the commendable NVIDIA GeForce RTX 4090 endowed with a substantial 24GB of memory. The CUDA framework used in this endeavor operates under the prestigious 11.3 version. The deep learning framework is PyTorch.

Before network training, it is necessary to configure hyperparameters to achieve optimal model performance and avoid overfitting. The batch size is set to 32 and the learning rate is set to 0.001. We set the number of iterations to 300 and select Adam as the optimizer. As shown in FIGURE 14, the value of the loss function sharply decreases from 0 to 100 iterations and gradually decreases from 100 to 250 iterations. After 250 iterations, the loss value stabilizes, indicating that the model has reached its optimal state.

C. EVALUATION METRICS

In this study, we utilized the average accuracy (AP) and the average AP (mAP) to assess the accuracy of wood defect detection. In addition, we used frames per second (FPS) and number of parameters to evaluate the detection speed and model size.

$$Precision = \frac{TP}{TP + FP} \tag{10}$$

$$Recall = \frac{TP}{TP + FN} \tag{11}$$

$$\mathrm{AP} = \int_0^1 Precision\,(t)\,\mathrm{d}t \tag{12}$$

$$\mathrm{mAP} = \frac{1}{N}\sum_{\mathrm{i}=1}^{N} AP_{\mathrm{i}} \tag{13}$$

$$FPS = \frac{1}{AverageProcessingTime} \tag{14}$$

In equation (10) and (11), TP represents the number of correctly detected defect samples; FP represents the number of non-defect samples falsely detected as defects; and FN represents the number of defect samples that were not detected correctly. P and R, respectively, denote Precision (the ratio of TP to the sum of TP and FP) and Recall (the ratio of TP to the sum of TP and FN). In the equation (12) and (13), AP represents the average detection accuracy for each defect category. The average detection accuracy for all defect categories is represented by the mAP. It thoroughly takes into account the recall and precision of the detection results. FPS is a metric that represents the number of image frames processed per second and gauges the speed at which algorithms process data and how long it takes to draw conclusions from models. Real-time detection is made possible by algorithms that can perform computations in a comparatively shorter amount of time, as indicated by a higher FPS, which typically indicates lower computational complexity.

In addition, mAP (0.5) and mAP (0.5:0.95) are often used as evaluation metrics in experiments. mAP (0.5) is the mAP with the IoU set to 0.5 and mAP (0.5:0.95) represents the mAP computed across a range of IoU thresholds from 0.5 to 0.95.

D. ABLATION EXPERIMENTS

Using the same environmental and parameter settings, we performed five sets of ablation experiments to precisely assess the influence of each component of enhancement on wood defect detection. The purpose of these joint ablation experiments was to assess how well the changes worked. TABLE 3 presents the experiment results. For the purpose of

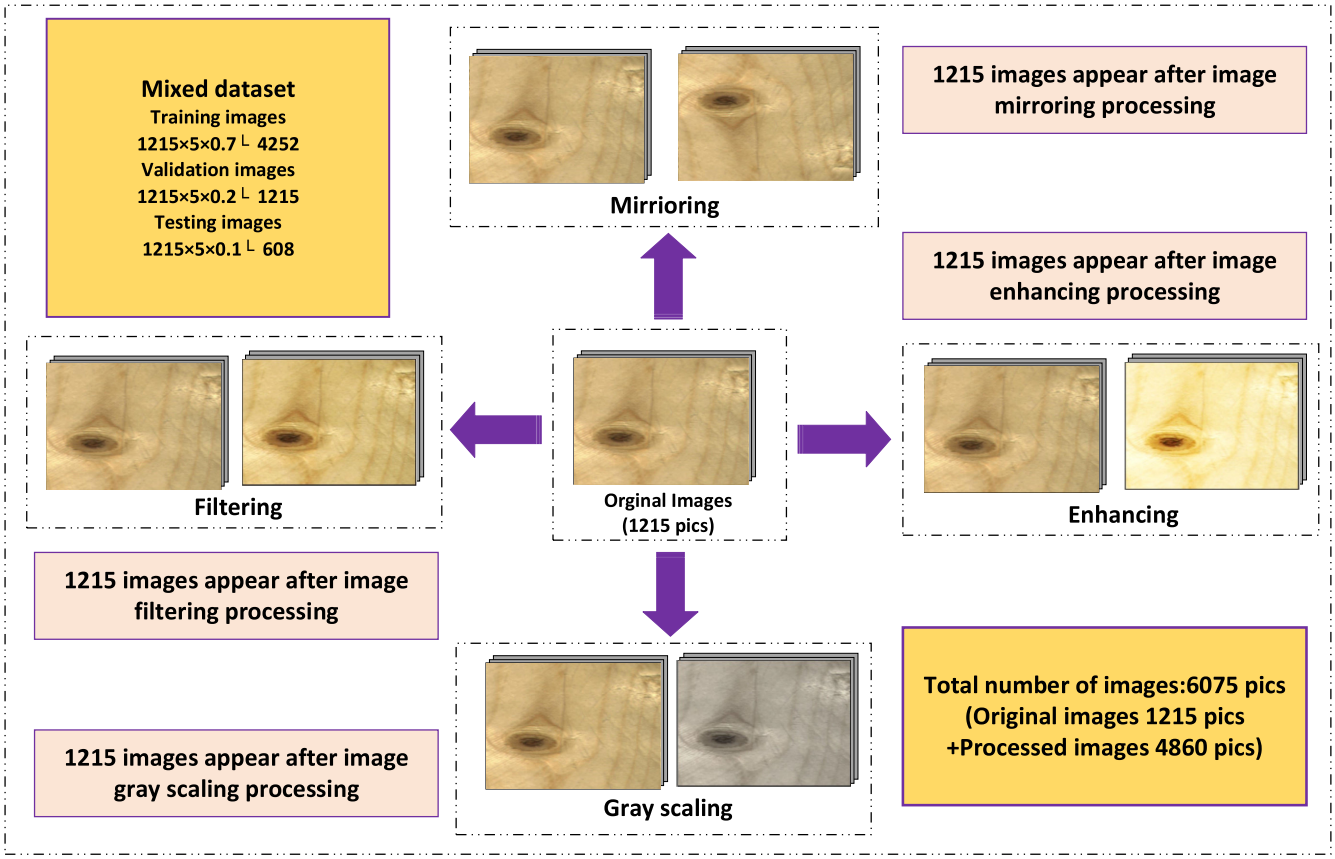

FIGURE 13. Image pre-processing.

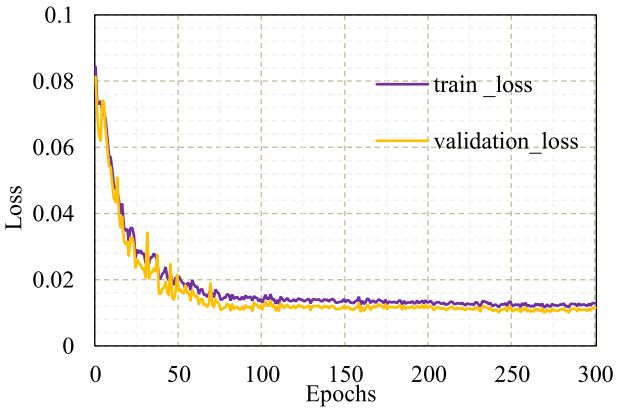

FIGURE 14. The loss of training and validation.

TABLE 2. Sample diagram of wood surface defects.

Defect Type	Defect Sample	Defect Description
Crack		Linear cracks or fissures on the surface of wood, usually caused by shrinkage and expansion of the wood.
Dead_Knot		Resin clumps in wood that have died or withered. Dead knots usually appear in circular or irregular shapes.
Live_Knot		Resin grown on the surface of wood. A live knot usually appears as a raised small or cylindrical structure.
Resin		Resin exudates on the surface of wood. Resin may scatter on the surface of wood in irregular shapes.

comparing these results with the previous five experimental sets, we used the YOLOv5s model as the baseline.

Compared to the original YOLOv5s, when only Ghost Bottleneck was introduced, the Precision increased by 1.274%, Recall increased by 2.868%, mAP (0.5) increased by 1.372%, Parameters decreased by 8.451% and FPS increased by 3.558%. These results verify that our Ghost Bottleneck has better feature extraction performance with fewer parameters.

After introducing the BiFormer model on this basis, the Precision increased by 2.397%, Recall increased by 10.457%, mAP (0.5) increased by 5.164%, Parameters slightly decreased, and FPS increased by 0.829%. These findings confirm that the overall detection accuracy has significantly increased following the addition of the BiFormer model.

TABLE 3. Ablation experiments.

Model	+Ghost	+BiFormer	+CARAFE	+DyHead	+CIoU	Precision (%)	Recall (%)	mAP (0.5) (%)	Parameters (M)	FPS (fps)
YOLOv5s						78.261	70.399	75.265	7.1	81.5
—	√					79.258	72.418	76.298	6.5	84.4
—	√	√				81.158	79.991	80.238	6.4	85.1
—	√	√	√			82.021	80.898	81.926	6.2	85.9
—	√	√	√	√		86.995	85.264	86.530	6.3	86.5
Our Method	√	√	√	√	√	**90.298**	**87.799**	**88.719**	**6.0**	**86.6**

Then, replacing the original upsampling operator with the lightweight CARAFE upsampling operator, the Precision is increased by 1.063%, Recall is increased by 0.907%, mAP (0.5) is increased by 1.134%, Parameters decreased by 3.125% and FPS is slightly increased. According to the experimental findings, CARAFE reduces the size of the model while increasing the computational accuracy. The upsampling operation model has the benefit of being lightweight and has good recognition performance.

After adding the DyHead, Precision increased by 6.064%, Recall increased by 5.397%, and mAP (0.5) increased by 5.619%. The parameters slightly increased, but compared to this, Precision, Recall, and mAP (0.5) all showed significant improvements that met the target period. According to the experimental findings, the incorporation of DyHead enables the recognition of image details.

Finally, after adding the CIoU loss function, also known as our method model, the Precision increased by 3.797%, Recall increased by 2.973%, mAP (0.5) increased by 2.530%, Parameters decreased by 4.762% FPS increased by 0.117%. The experimental results indicate that CIoU can effectively measure the degree of matching between bounding boxes. Improve the efficiency of model detection.

In conclusion, our approach has improved significantly while maintaining a high degree of accuracy in terms of minimizing parameters. These results validate our proposed model's high precision and lightweight. The ablation experiment's visual comparison results are displayed in FIGURE 15. It is evident that our approach has the highest accuracy and is most similar to the actual box.

Meanwhile, we compared our method to the original YOLOv5s to assess its detection performance and demonstrate its visual appeal. We randomly selected a subset from the dataset for evaluation, and the results are shown in FIGURE 16. In the first identical scenario, comparing the result graph (a) detected by the original YOLOv5s with our method's result graph (e), it is evident that the original YOLOv5s failed to detect the Live in the lower left corner of the Live_ Knot, there is an error detection situation, and our method accurately detected the Live in the bottom left corner of the Live_ Knot. In the other three cases, although the initial YOLOv5s was able to detect defects on the wood surface, its confidence was lower than that of our method. As shown in FIGURE 16, (b) and (f), (c) and (g), (d) and (h), it can be concluded that our method outperforms the original YOLOv5s in detecting wood surface defects.

E. COMPARATIVE EXPERIMENTS

To further validate the detection performance of the method proposed in this article, a comparative analysis was performed. Our method was compared with several one-stage object detection algorithms, including SSD and YOLOv5s, alongside well-regarded approaches such as RetinaNet and YOLOv8.

In the training process, we present a comparison of our method's mAP (0.5) over roughly 300 epochs with other methods. The results presented in FIGURE 17 demonstrate that, for each method, the mAP (0.5) increases rapidly during the first 50 epochs before beginning to trend toward stability at epoch 250. However, the mAP (0.5) of our method rises faster and finally stabilizes at maximum value of 0.88719, which is higher than other methods.

In the detection process, we also conducted comparative experiments using our method and the algorithms mentioned above. The weight file with the best training performance is saved in the same experimental environment. The evaluation metrics used for comparative experiments encompass AP, mAP (0.5), mAP (0.5:0.95), FPS, Parameters, Precision and Recall.

As shown in TABLE 3, TABLE 4 and FIGURE 18, both the speed of model inference and the accuracy of detection have been improved in our method.

Compared to SSD, the Precision and Recall have improved by 16.933% and 17.701%, the mAP (0.5) and mAP (0.5:0.95) have improved by 17.576% and 18.536%. The parameters have decreased by 82.608%, FPS has increased by 56.6% and the AP of four types of defects Crack, Dead_Knot,

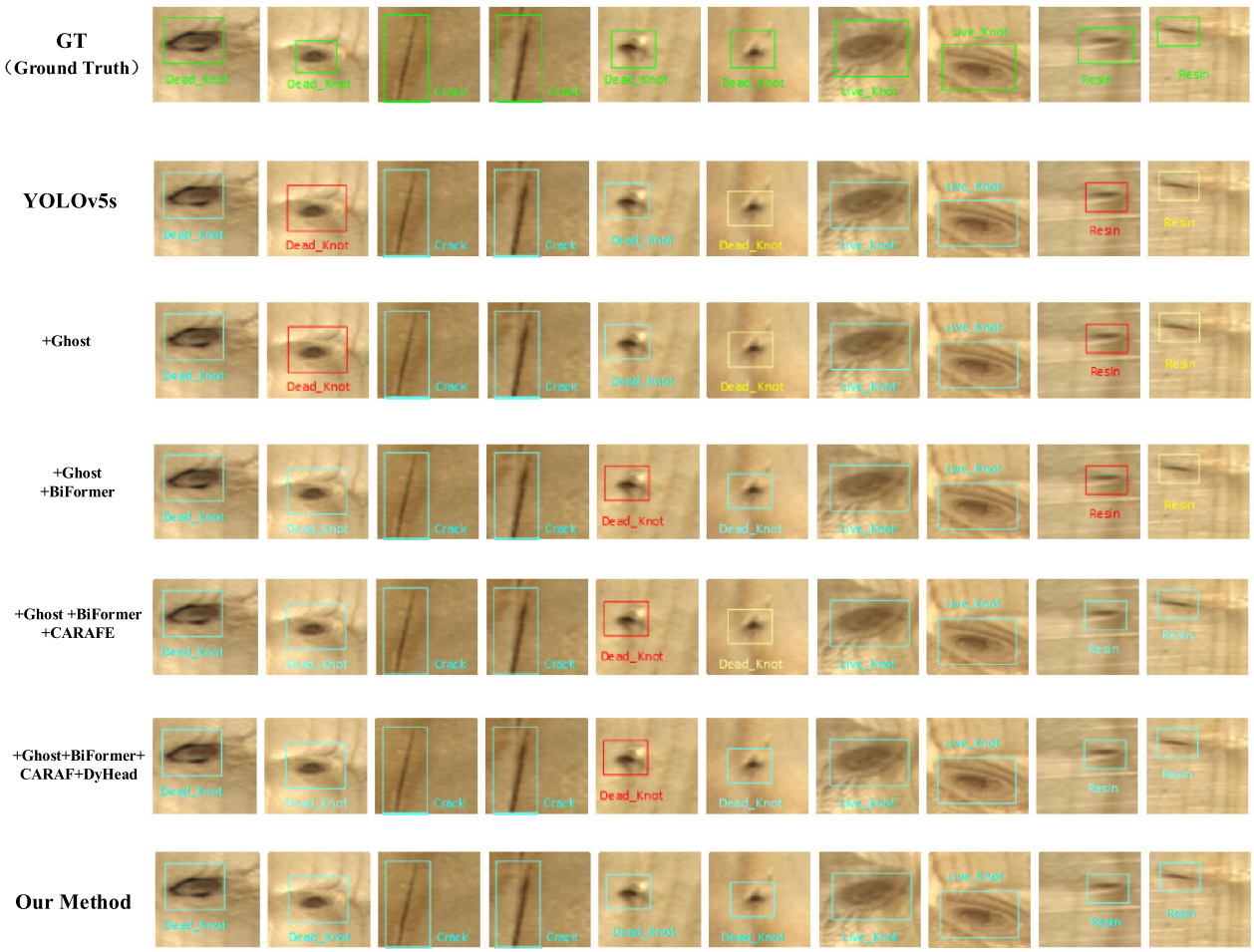

FIGURE 15. Visualization of wood surface defects through ablation experiments. The red rectangular box indicates missed recognition of the targets, the yellow rectangular box indicates misidentification of the targets, and the blue rectangular box indicates correct recognition of the targets.

TABLE 4. The Comparison results of different methods in the detection process.

Model	Parameters (M)	FPS (fps)	mAP (0.5) (%)	mAP (0.5:0.95) (%)	Precision (%)	Recall (%)
SSD	34.5	55.3	71.143	53.531	73.365	70.098
YOLOv5s	7.1	81.5	75.265	60.122	79.926	74.231
RetinaNet	36.5	47.2	81.026	62.362	82.209	80.098
YOLOv8	11.1	64.3	85.521	68.021	87.786	84.291
Our Method	**6.0**	**86.6**	**88.719**	**72.067**	**90.298**	**87.799**

Live_Knot and Resin has been increased respectively by 5.501%,11.032%,21.733% and 32.038%.

Compared with the original YOLOv5s, the Precision and Recall have improved by 10.372% and 13.568%, the mAP (0.5) and mAP (0.5:0.95) have improved by 13.454% and 11.945%. The parameters have decreased by 15.493%, FPS has increased by 6.258% and the AP of four types of defects Crack, Dead_ Knot, Live_ Knot and Resin has been increased respectively 4.7%,11.07%,11.27% and 26.127%.

Compared to RetinaNet, the Precision and Recall have improved by 8.089 % and 7.701%, the mAP (0.5) and mAP (0.5:0.95) have improved by 7.693% and 9.705%. The parameters have decreased by 83.562%, FPS has increased by 83.475% and the AP of four types of defects Crack, Dead_

TABLE 5. AP for four types of wood defects in the detection process.

Model	Crack (%)	Dead_Knot (%)	Live_Knot (%)	Resin (%)
SSD	82.598	75.259	68.325	58.390
YOLOv5s	83.399	75.021	78.339	64.301
RetinaNet	81.911	79.932	80.325	81.936
YOLOv8	86.611	83.395	86.201	85.877
Our Method	**88.099**	**86.291**	**90.058**	**90.428**

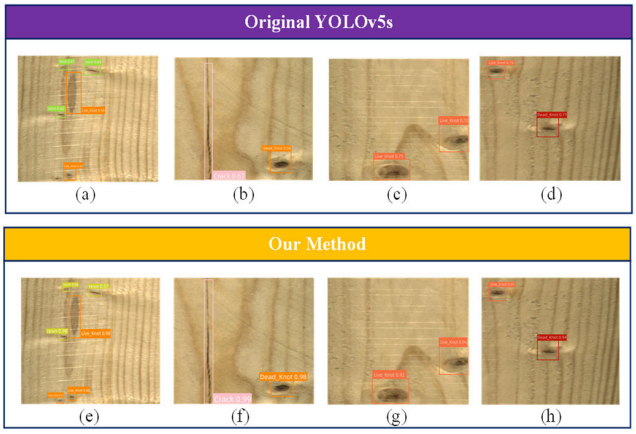

FIGURE 16. Comparison of detection results between the original YOLOv5s and our method.

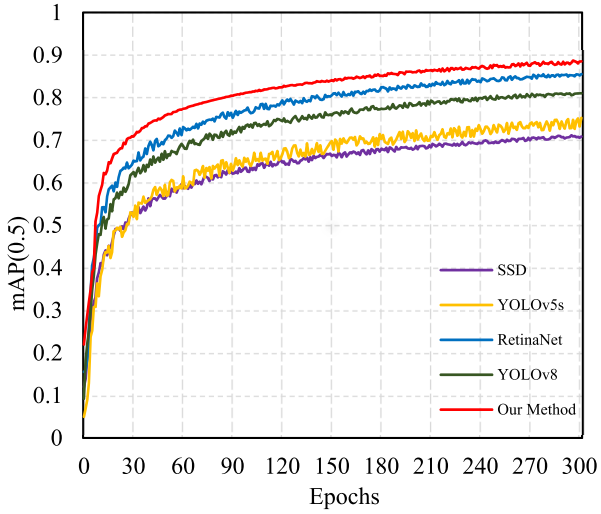

FIGURE 17. The mAP (0.5) of related algorithms in the training process.

Knot, Live_ Knot and Resin has been increased respectively 6.188%,6.359%,9.733% and 8.492%.

Compared with YOLOv8, the Precision and Recall have improved by 2.512 % and 3.508%, the mAP (0.5) and mAP (0.5:0.95) have improved by 3.198% and 4.046%. The parameters have decreased by 45.946%, FPS has increased by 34.681% and the AP of four types of defects Crack, Dead_ Knot, Live_ Knot and Resin has been increased respectively 1.488%,2.896%,3.857% and 4.551%.

Taking into account the complexity of each model and the actual detection results, it can be generally concluded that our proposed method outperforms other methods in the detection of wood defects.

Meanwhile, to facilitate a more intuitive comparison of the detection effectiveness of various methods, we conducted multiple sets of comparative experiments targeting different scenarios.

Based on scene 1 depicted in FIGURE 19, it can be inferred that during the inspection of surface defects in the wood, the presence of numerous defects and the complexity of the image pose significant challenges to the algorithm. Both SSD and YOLOv5s failed to detect the live knot at the bottom of the timber, indicating a missed detection. Although RetinaNet and YOLOv8 did not miss any detections, their confidence scores were lower compared to our algorithm. Additionally, when detecting other defects, this algorithm consistently demonstrated higher accuracy than other algorithms. Therefore, in complex scenes, our algorithm exhibits a high level of precision.

Based on the observations from scenes 2 and 3 in FIGURE 19, it can be noted that the classical SSD algorithm fails to accurately detect resin, dead knot, and live knot as small surface defects with the wood. It exhibits a phenomenon of missing detections. In contrast, YOLOv5s, RetinaNet and YOLOv8 algorithms are capable of detecting all defects on the wood surface. However, the confidence levels of these detections are lower compared to the current algorithm. In other words, our method's algorithm possesses a higher level of accuracy and precision in wood defect detection.

As shown in Scene 4 and 5 in FIGURE 19, when detecting wood surface defects, due to the small number of defects and low image complexity in Scene 4 and 5, their interference with the algorithm is relatively weak. In this case, the classic algorithms SSD, YOLOv5s, RetinaNet and YOLOv8 successfully detected defects on the wood surface without any missed detections. However, the confidence level of the

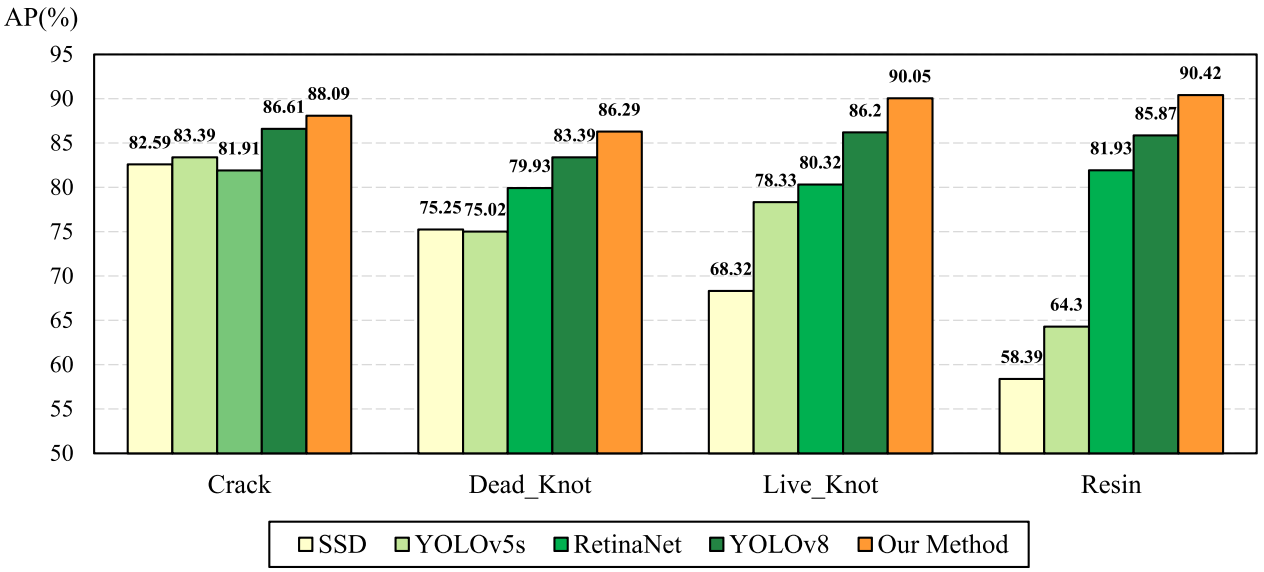

FIGURE 18. The AP of different models for four types of wood defects.

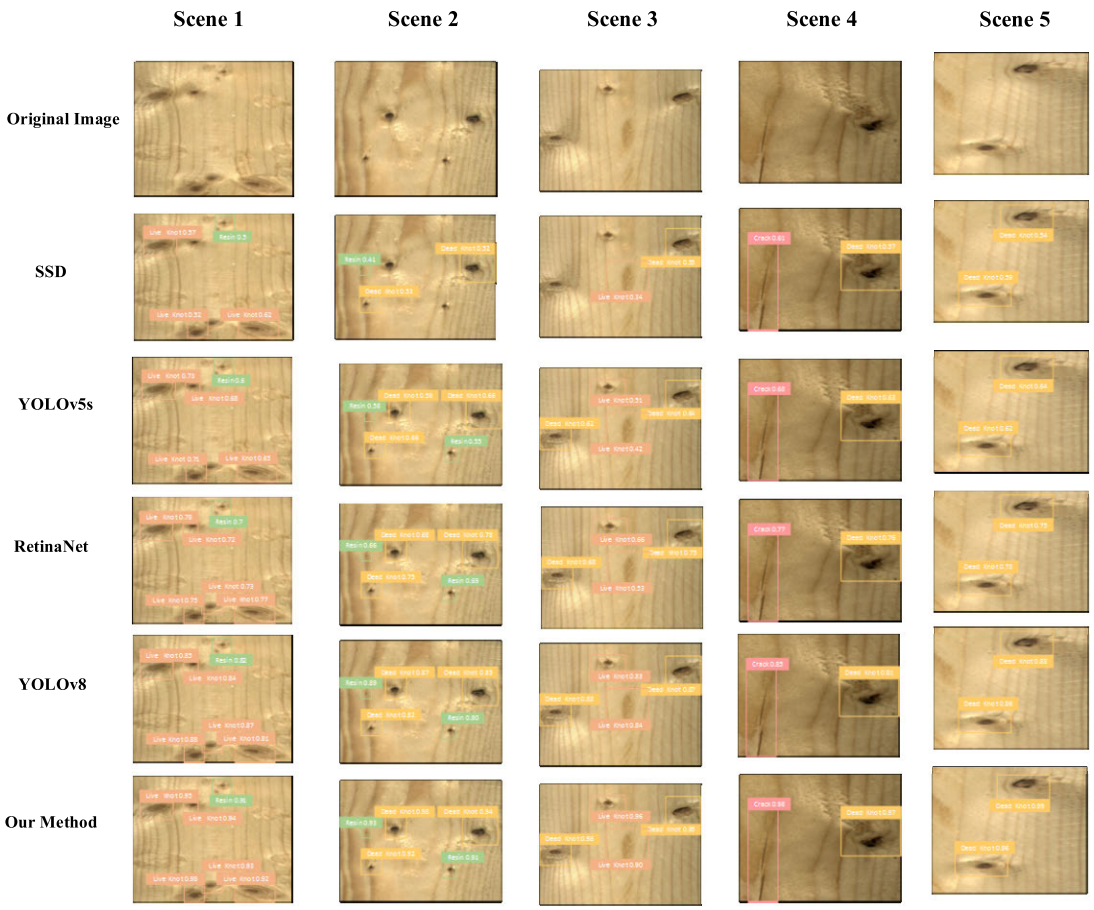

FIGURE 19. Detection comparison in multiple scenarios.

defects detected by these algorithms is lower than that of this algorithm. Therefore, our method's algorithm still exhibits high accuracy in low complexity scenarios.

In summary, our method can achieve optimal detection performance on both high and low complexity wood surfaces in different scenarios. The algorithm used in our approach

greatly minimizes errors and reduces missed detections, producing detection results that are more reliable and accurate.

IV. CONCLUSION

To achieve a more precise and rapid detection of wood surface defects and address the limitations of existing methods, this paper introduces an enhanced lightweight model, GBCD-YOLO, based on YOLOv5s. Firstly, we introduced a lightweight Ghost bottleneck to improve the computational efficiency and inference speed of deep neural networks. In addition, BiFormer was incorporated into the neck to improve the performance of natural language processing tasks. Meanwhile, CARAFE was used as an upsampling substitute to enhance the perception and capture of details. In addition, in order to enhance the flexibility and generalization ability of the method, dynamic headers were introduced, and finally CIoU was used instead of GIoU. The results of the optimized wood defect dataset experiment demonstrated a notable advancement in our approach. Specifically, compared to YOLOv5s, mAP (0.5) increased by 13.454% to 88.719%, FPS increased by 6.258% and the parameters decreased by 15.493%. In addition, our method has advantages over other models mentioned in the paper in terms of mAP and model inference speed. In summary, our method effectively solves the problems of low detection rate of small size defects on wood surfaces and incomplete recognition of complex and dense defects.

However, this paper acknowledges limitations in visualizing the internal conditions of wood with respect to size and lighting. Therefore, our future research endeavors will focus on integrating deep learning methods with techniques capable of characterizing the internal conditions of wood more effectively.

REFERENCES

[1] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Columbus, OH, USA, Jun. 2014, pp. 580–587.

[2] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1440–1448.

[3] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.

[4] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2980–2988.

[5] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," 2015, *arXiv:1506.02640*.

[6] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," 2016, *arXiv:1612.08242*.

[7] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*.

[8] A. Bochkovskiy, C.-Y. Wang, and H.-Y. Mark Liao, "YOLOv4: Optimal speed and accuracy of object detection," 2020, *arXiv:2004.10934*.

[9] Ultralytics. *YOLOv5*. Accessed: Nov. 1, 2020. [Online]. Available: https://github.com/ultralytics/YOLOv5

[10] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot MultiBox detector," 2015, *arXiv:1512.02325*.

[11] J. Jeong, H. Park, and N. Kwak, "Enhancement of SSD by concatenating feature maps for object detection," 2017, *arXiv:1705.09587*.

[12] Z. Li and F. Zhou, "FSSD: Feature fusion single shot multibox detector," 2017, *arXiv:1712.00960*.

[13] C.-Y. Fu, W. Liu, A. Ranga, A. Tyagi, and A. C. Berg, "DSSD : Deconvolutional single shot detector," 2017, *arXiv:1701.06659*.

[14] M. Tan, R. Pang, and Q. V. Le, "EfficientDet: Scalable and efficient object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 10778–10787.

[15] J. Fan, Y. Liu, Z. Hu, Q. Zhao, L. Shen, and X. Zhou, "Solid wood panel defect detection and recognition system based on faster R-CNN," *J. Forestry Eng.*, vol. 4, no. 3, pp. 112–117, 2019.

[16] M. I. Chacon and G. R. Alonso, "Wood defects classification using a SOM/FFP approach with minimum dimension feature vector," in *Proc. Int. Symp. Neural Networks*, Chengdu, China, Jun. 2006, pp. 1105–1110.

[17] D. W. Qi and H. B. Mu, "Detection of wood defects types based on Hu invariant moments and BP neural network," *J. Southeast Univ.*, vol. 43, pp. 63–66, Jul. 2013.

[18] W. Ye, "Research on rapid identification algorithm of wood defects based on LBP feature extraction," *Electron. Compon. Inf. Technol.*, vol. 3, pp. 69–72, Jan. 2019.

[19] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 2, pp. 318–327, Feb. 2020.

[20] D. Reis, J. Kupec, J. Hong, and A. Daoudi, "Real-time flying object detection with YOLOv8," 2023, *arXiv:2305.09972*.

[21] L. Zhu, X. Wang, Z. Ke, W. Zhang, and R. Lau, "BiFormer: Vision transformer with bi-level routing attention," 2023, *arXiv:2303.08810*.

[22] J. Wang, K. Chen, R. Xu, Z. Liu, C. C. Loy, and D. Lin, "CARAFE: Content-aware reassembly of features," 2019, *arXiv:1905.02188*.

[23] X. Dai, Y. Chen, B. Xiao, D. Chen, M. Liu, L. Yuan, and L. Zhang, "Dynamic head: Unifying object detection heads with attentions," 2021, *arXiv:2106.08322*.

[24] 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.

[25] 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.

[26] A. Neubeck and L. Van Gool, "Efficient non-maximum suppression," in *Proc. 18th Int. Conf. Pattern Recognit. (ICPR)*, vol. 3, Aug. 2006, pp. 850–855.

[27] K. Han, Y. Wang, C. Xu, J. Guo, C. Xu, E. Wu, and Q. Tian, "GhostNets on heterogeneous devices via cheap operations," *Int. J. Comput. Vis.*, vol. 130, no. 4, pp. 1050–1069, Apr. 2022.

[28] X. Dong, J. Bao, D. Chen, W. Zhang, N. Yu, L. Yuan, D. Chen, and B. Guo, "CSWin transformer: A general vision transformer backbone with cross-shaped windows," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 12114–12124.

[29] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 9992–10002.

[30] Z. Tu, H. Talebi, H. Zhang, F. Yang, P. Milanfar, A. Bovik, and Y. Li, "MaxViT: Multi-axis vision transformer," in *Proc. ECCV*, 2022, pp. 459–479.

[31] K. Li, Y. Wang, P. Gao, G. Song, Y. Liu, H. Li, and Y. Qiao, "UniFormer: Unified transformer for efficient spatiotemporal representation learning," 2022, *arXiv:2201.04676*.

[32] S. Ren, D. Zhou, S. He, J. Feng, and X. Wang, "Shunted self-attention via multi-scale token aggregation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 10843–10852.

[33] Z. Yang, H. Feng, Y. Ruan, and X. Weng, "Tea tree pest detection algorithm based on improved YOLOv7-tiny," *Agriculture*, vol. 13, no. 5, p. 1031, May 2023.

[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.

[35] Y. Chen, X. Dai, M. Liu, D. Chen, L. Yuan, and Z. Liu, "Dynamic ReLU," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 351–367.

[36] P. Kodytek, A. Bodzas, and P. Bilik, "A large-scale image dataset of wood surface defects for automated vision-based quality control processes," *FResearch*, vol. 10, p. 581, Jul. 2021.

YUNCHANG ZHENG received the B.S. degree in electronic science and technology and the M.S. degree in signal and information processing from the University of Electronic Science and Technology of China, in 2013 and 2016, respectively. Since 2017, he has been a Lecturer with the College of Electrical Engineering, Hebei University of Architecture, Zhangjiakou, China. His research interests include image processing, deep learning, and artificial intelligence.

MENGFAN WANG is currently pursuing the degree in electrical engineering and automation with the Hebei University of Architecture, Zhangjiakou, China. His research interests include machine learning and artificial intelligence.

BO ZHANG received the M.S. degree in electrical engineering from the Hebei University of Technology, in 2018. Since 2020, he has been a Lecturer with the College of Electrical Engineering, Hebei University of Architecture, Zhangjiakou, China. His research interests include electrical automation and artificial intelligence.

XIANGNAN SHI is currently pursuing the degree in electrical engineering and automation with the Hebei University of Architecture, Zhangjiakou, China. Her research interests include artificial intelligence and image processing.

QING CHANG received the M.S. degree in computer technology from the Hebei University of Technology, in 2011. Since 2004, she has been a Teacher with the Electrical Engineering Department, Hebei University of Architecture. From 2010 to 2018, she was with the Scientific Research Management Department. She is currently an Associate Professor with the Electrical Engineering. Her research interests include virtual instruments, signal processing, and industrial process control.