Journal of Computers Vol. 36 No. 2, April 2025, pp. 315-326
https://doi.org/10.63367/199115992025043602021

Lightweight YOLOv8 for Rapid Object Detection

Miao Jin[1*], Jun Zhang[1], Chuning Peng[2], Haibin Chen[3], Xiwen Chen[1], Bing Lu[1], and Xu Wang[1]

[1] China Electric Power Research Institute, Wuhan, Hubei, China,
whu_phd_jinmao@163.com, {zhangjun3, chenxiwen, lubing, wangxu5}@epri.sgcc.com.cn

[2] State Grid Corporation of China, Beijing, China
pengcn@163.com

[3] State Grid Shanghai Electric Power Research Institute, Shanghai, China
1040178598@qq.com

Received 21 February 2025; Revised 23 February 2025; Accepted 23 February 2025

Abstract. YOLOv8, as a cutting-edge object detection model, not only inherits the core technical advantages of the YOLO series, such as speed and high accuracy, but also further enhances the model's performance by introducing new improvements and features. However, due to the model's complex structure, it is difficult to deploy on embedded devices, and existing lightweight models have not effectively balanced performance. To address this issue, this paper proposes a fast object localization method based on a lightweight YOLOv8. This method improves inference speed and training efficiency by replacing the original convolutional blocks with depthwise separable convolutions. Additionally, we designed the FEM and CBAMamba modules to compensate for the deficiencies of lightweight neural networks in feature extraction. At the head of the detector, we employed RepConv and reduced the size of the convolution layers, thereby decreasing the number of floating-point operations and improving computational efficiency. Simultaneously, a dynamic non-monotonic focusing mechanism WIoU was introduced to accelerate the convergence speed. Remarkably, the improved algorithm proposed in this study has demonstrated outstanding performance enhancements compared to the original YOLOv8 algorithm. Specifically, it has achieved an increase in the mean Average Precision (mAP) at the Intersection over Union (IoU) threshold of 0.5 and in the mAP across the IoU threshold range from 0.5 to 0.95 by 2.2% and 4.5% respectively. Moreover, it has also managed to boost the Frames Per Second (FPS) by 9.8%, indicating a significant improvement in processing speed. Concurrently, it has successfully reduced the model parameters by 13.3%, which not only optimizes the computational load but also makes the algorithm more efficient and adaptable in practical applications.

Keywords: YOLOv8, object detection, lightweight networks, depthwise separable convolution, feature extraction

1 Introduction

The transmission lines with a voltage level above 10 kV in China's power system exceed 6 million kilometers. Due to the long-term interference of natural factors such as intense light irradiation and heavy rain erosion on the transmission lines, problems like line damage and even breakage often occur. In order to meet users' high demands for the stability of the power system, automated power grid current-diverting operations have become one of the important means for power system maintenance. Regarding the problem of remote identification of power grid current drainage devices, the existing technologies mainly identify the clamp part of the devices, which is an important task in the field of image processing.

In recent years, with the development of Transformer in the field of computer vision, various Transformer - based models have demonstrated powerful performance. For example, Vision Transformer (ViT) utilizes the self - attention mechanism to model global information, successfully breaking through the limitation of convolutional neural networks that can only perform local perception. Detection Transformer (DETR), on the other hand, applies Transformer to the object detection task. It directly outputs the object categories and positions without the need for complex hand - designed anchor boxes, simplifying the detection process and bringing new ideas

[*] Corresponding Author

to object detection. However, due to the relatively complex structure of the Transformer model, its performance cannot be fully utilized when running on embedded devices.

Fu et al. [1] employed an enhanced Faster R-CNN network to detect bolts within the line clamp. This approach utilizes neural networks to automatically abstract bolt image features, facilitating the identification of punctured bolts within the line clamps across various resolutions and angles. Nevertheless, the intricate network structure and heightened computational requirements still present hindrances to real-time detection. In scenarios characterized by restricted resources on embedded or mobile devices, in order to effectively exploit the efficiency of deep neural networks, Wang et al. [2] adopted lightweight convolutional PartialConv to achieve model lightweighting. Similarly, Zhang et al. [3] introduced GSConv and slim-neck in the Neck network to notably reduce the model's parameter size.

The YOLO [4] series algorithms have demonstrated strong performance in the domain of single-stage object detection, striking a desirable equilibrium between detection accuracy and computational efficiency. [5] Wang et al. [6] refined the YOLOv5 model by substituting the original backbone network with MobileNetV3, culminating in the development of a nimble model. Nonetheless, the process of model lightweighting often necessitates a delicate equilibrium between model size, computational efficiency, and detection accuracy. As such, in certain instances, prioritizing enhanced model efficiency could entail sacrificing a certain degree of accuracy.

This paper introduces a pioneering lightweight hole detection method for line clamps, leveraging the latest YOLOv8 algorithm for enhancement. Primarily, deep separable convolutions are integrated to replace the initial convolutional kernels to accomplish the overall goal of lightweighting the model. Subsequently, a feature enhancement module is devised to refine the backbone network, constructing a multi-branch structure employing conventional and expansion convolutions of various scales and quantities to fortify the network's feature extraction capabilities. Furthermore, the designed CBAMamba module integrates the Structural Similarity Index Model (SSIM) to capture interdependencies between features. Ultimately, RepConv is implemented at the output stage to refine the modules, coupled with the utilization of the WIoU (wise intersection over union) loss function to supplant the original loss function, thereby enhancing the model's generalization capability and accuracy. Empirical findings corroborate that this lightweighting technique substantially reduces the model size while enhancing recognition accuracy and processing speed. To summarize, our contributions are:

1. This paper proposes an innovative lightweight pore detection method for lightweight model design in power system maintenance, based on the latest YOLOv8 algorithm. By using techniques such as depthwise separable convolution, it significantly reduces the model parameter size while improving recognition accuracy and processing speed.
2. The designed feature enhancement module combines a multi-branch structure and the CBAMamba module, effectively enhancing the network's feature extraction capability. The SSIM model is introduced to capture the dependency relationships between features, laying the foundation for lightweight processing.
3. Experimental results show that after adopting the lightweight method proposed in this paper, the model size is significantly reduced, while both recognition accuracy and processing speed are significantly improved, providing an effective solution for power system maintenance.

2 Related Work

2.1 YOLOv8 Model

YOLOv8 [7] is an advanced object detection model that inherits the significant advantages of the YOLO series, excelling in speed and accuracy. Through new enhancements and features, the model's performance is further amplified. Within its backbone architecture, the C2f module replaces the traditional C3 module, simplifying the model's structure while enhancing information transmission and expression efficiency. The model structure is shown in Fig. 1.

This module is divided into three main components: the input end, which receives data processed through augmentation utilizing the Mosaic [8] method, enhancing the model's generalization ability when confronted with diverse image variations like lighting, angles, and occlusions; the backbone network, utilizing the robust DarkNet53 [9] structure for more efficient feature extraction; and the integration of multiple residual bottleneck structures and convolutional layers within the C2f module, significantly enhancing the transmission and expression of information. Moreover, to accurately detect objects of varying sizes, the SPPF module employs pooling layers with different kernel sizes to extract multi-scale features, enhancing the model's adaptability. In the design phase of the neck network, the Path Aggregation Network (PANet) [10] intelligently combines the top-down Feature Pyramid Network (FPN) and bottom-up feature aggregation (PAN), ensuring the seamless integration of contextual and positional information. Ultimately, by utilizing three different sizes of branch output forms, the prediction head conducts detailed classification and localization operations on targets, enhancing the model's detection capabilities for small and dense targets and effectively addressing the challenges of object detection in complex backgrounds.

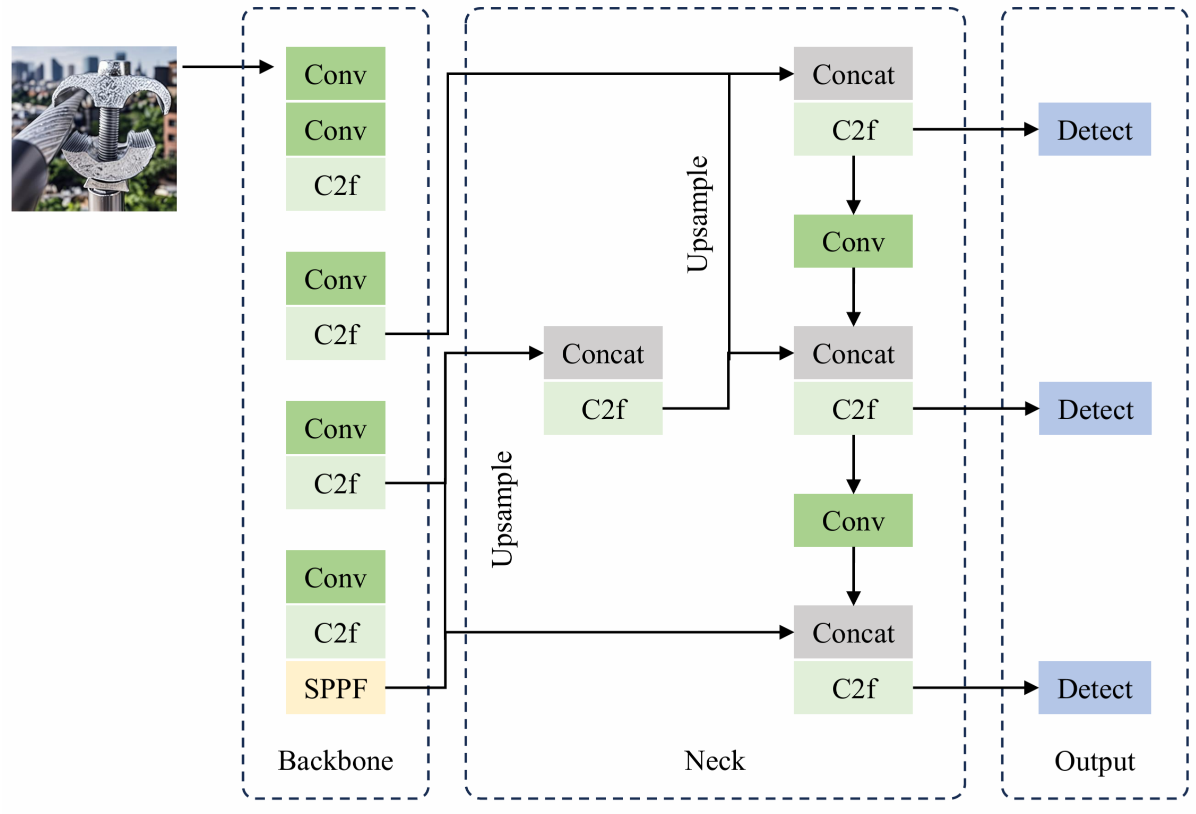

Fig. 1. The structure of YOLOv8 network

(The left section is the backbone network, which is composed of multiple Conv and C2f modules, ending with an SPPF module. The middle section is the Neck, where the features output from the backbone network are first upsampled, and then features from different levels are concatenated through the Concat operation. These concatenated features are further processed through C2f modules and Conv layers. The right section is the Output, where the features processed by the Neck are ultimately fed into the Detect module.)

2.2 Mamba Network Model

Convolutional Neural Networks (CNN) and Transformer models have inherent limitations. CNN [11] is constrained by local receptive fields, hindering the capture of long-distance information and proving ineffective in specific scenarios [12]. On the other hand, Transformer excels in global modeling and efficiently captures long-range dependencies [13]. However, its complexity escalates notably when processing large-scale images [14]. In comparison, the State Space Model (SSM) [15] offers distinct advantages by constructing long-distance dependencies while maintaining linear complexity, showcasing significant potential across various tasks. Retaining a global receptive field, SSM leverages CSM design to replace the attention mechanism [16], successfully reducing computational complexity to a linear scale. The introduction of the Mamba model is remarkable as it integrates specific input parameterization with scalable hardware-optimized computing techniques, achieving unparalleled efficiency and simplicity in processing diverse sequences in cross-language and genomics domains.

The introduction of S4ND marks the first application of SSM blocks in visual tasks, effectively treating visual data processing as a continuous signal traversing 1D, 2D, and 3D domains. Drawing inspiration from the success of the Mamba model, Vmamba and Vim have expanded into the general visual task arena. By integrating bidirectional and cross-scanning mechanisms, they adeptly tackle the directional sensitivity challenge within SSM, underscoring the exceptional adaptability and efficacy of the visual Mamba model in addressing intricate visual problems.

3 Methods

This paper builds on the YOLOv8n base model, aiming for its lightweighting. Key steps include: Replacing original convolutions with depthwise separable ones to boost training and inference speed. After reducing the model parameters, two new modules, the Feature-Enhanced Module (FEM) and the CBAMamba mod-ule, are designed. FEM focuses on extracting fine-grained features through a multi-scale approach, cap-turing details at different levels of granularity. CBAMamba, which combines channel and spatial attention mechanisms, emphasizes relevant regions and channels in the feature maps, suppressing irrelevant infor-mation. These two modules work in synergy: FEM first extracts local-level features in detail, and then CBAMamba processes these features to enhance the global-level context and attention. This cooperation enables the lightweight model to effectively capture the necessary features for accurate object detection despite the reduction in model complexity. At the detection head, ditching traditional convolutions for RepConv and shrinking its size. This cuts floating-point ops, hiking computational efficiency. Applying the dynamic non-monotonic focusing WIoU. It speeds up model convergence and detection prowess. The revamped framework is in Fig. 2.

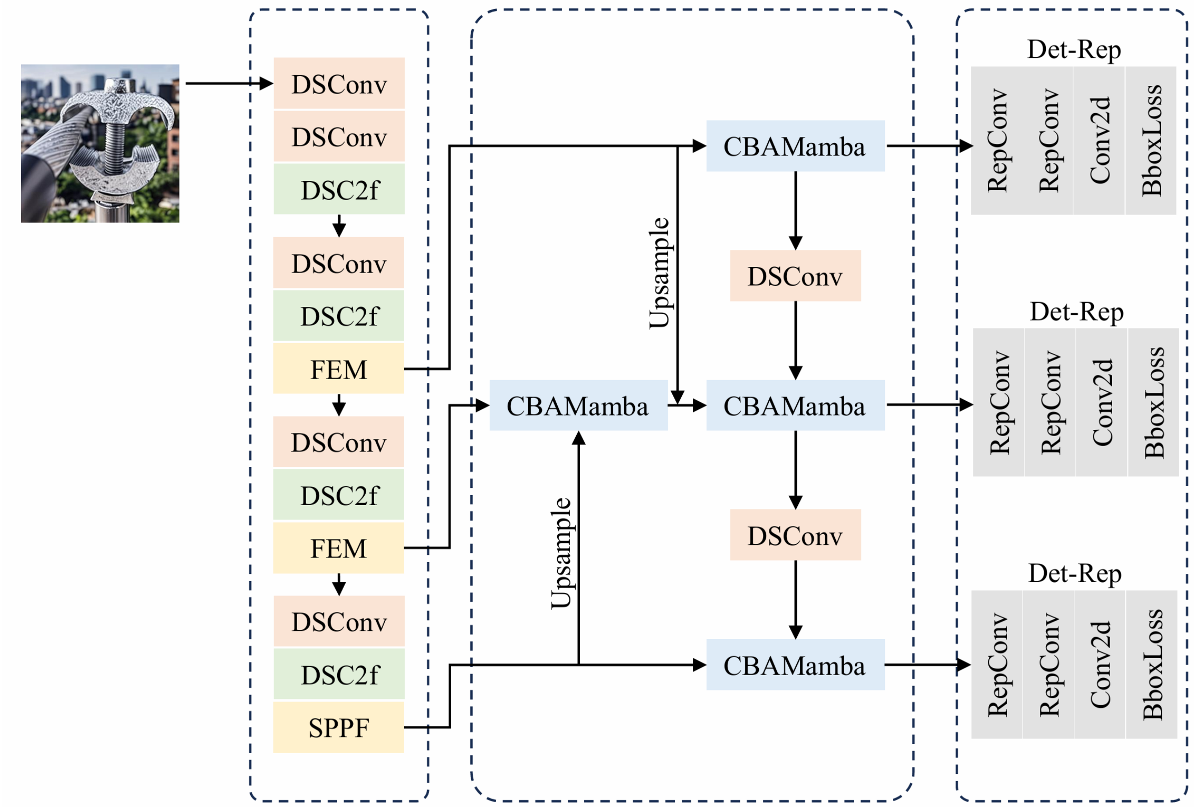

Fig. 2. The revamped framework

(The left side is the input section, where the input image first enters the Backbone, which is composed of DSConv, DSC2f modules, and FEM. The middle section is the Neck, where the features undergo upsampling and are processed through the CBAMamba module, followed by further processing through DSConv. The right side is the Output, which includes RepConv, Conv2d, and BboxLoss modules.)

3.1 Depthwise Separable Convolution

Depthwise Separable Convolution (DSConv) is a method to optimize convolution operations, which can reduce computational complexity while retaining effective feature extraction capability. It consists of two steps: depthwise convolution and pointwise convolution. Firstly, depthwise convolution is performed, where each channel of the input feature map undergoes independent convolution with the corresponding kernel, generating intermediate feature maps with the same number of channels. However, depthwise convolution only conducts independent convolution operations for each channel without cross-channel interaction, leading to limitations in feature extraction and channel adjustment.

To address this limitation, pointwise convolution operation is introduced to enhance information interaction between channels. Pointwise convolution applies a 1×1 convolution kernel to each channel of the intermediate feature map to obtain the final output feature map, which can be represented by the following formula:

$$Z_{i,j,k} = \sum_{m,n} X_{i+m,j+n,k} \cdot K_{m,n} \tag{1}$$

$$Y_{i,j,k'} = \sum_{k} Z_{i,j,k} \cdot K_{k,k'} \tag{2}$$

where Y represents the intermediate feature map with cross-channel interactions, Z represents the intermediate feature map without cross-channel interactions, X represents the input feature map, i, j, k represent pixel positions and channel numbers, m, n represent the spatial positions of the convolution kernel, and k' represents the channel index.

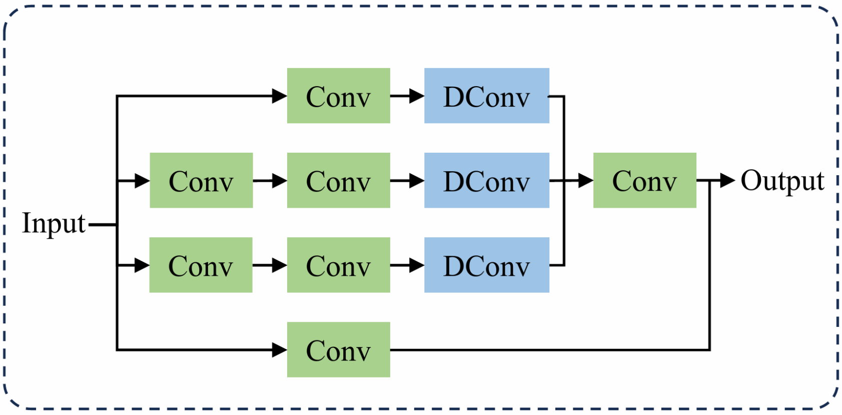

Fig. 3. FEM structural diagram

(This module mainly consists of Conv and DConv, which are used to process the input data and generate the output.)

3.2 Feature Enhancement Module

This paper proposes a Feature Enhancement Module (FEM), which adopts a multi-branch structure, multiple convolutions, and scales to connect multi-channel feature maps to horizontally expand the network width, thereby improving the network's adaptability, sensitivity, and receptive field for detecting various objects. To enhance the network's feature extraction capability, a finite element module is introduced in the main network to extract global features, working cooperatively with the previous convolution layers to improve detection performance. In addition, the two intermediate branches combined with dilated convolutional layers to enlarge the receptive field, increase contextual information, and enhance feature effectiveness. The structural diagram is shown in Fig. 3.

The FEM consists of 4 branches, where the first 3 branches execute 1x1 convolution operations to handle and adjust the channel number of feature maps, and the fourth branch contains a residual structure. The remaining 3 branches are composed of cascaded 3x3 traditional convolutions and dilated convolutions, obtaining more detailed target features through convolutions at different scales. The calculation process of FEM is as follows:

$$Y_1 = dconv1_{3\times3}(conv_{1\times1}(X)) \tag{3}$$

$$Y_2 = dconv3_{3\times3}(conv_{3\times3}(conv_{1\times1}(X))) \tag{4}$$

$$Y_3 = dconv5_{3\times3}(conv_{3\times3}(conv_{1\times1}(X))) \tag{5}$$

$$Y = (Y_1|Y_2|Y_3) \oplus X \tag{6}$$

where X represents the input feature map, $Y1$, $Y2$, and $Y3$ represent the output feature maps of different branches, Y represents the enhanced feature map, $conv_{1\times1}$ and $conv_{3\times3}$ represent regular convolutions with kernel sizes of 1×1 and 3×3, $dconv1_{3\times3}$, $dconv3_{3\times3}$, and $dconv5_{3\times3}$ represent dilated convolutions with dilation rates of 1, 3 and 5, $|$ denotes concatenation operation, and $\oplus$ denotes feature summation operation.

3.3 CBAMamba Module

This paper presents a novel CBAMamba module, which introduces the State Space Model (SSIM) to replace traditional attention mechanisms. The state space model is designed to establish long-range dependencies while maintaining linear time complexity. Compared to traditional attention mechanisms, the state space model has lower computational complexity. The structural diagram is shown in Fig. 4.

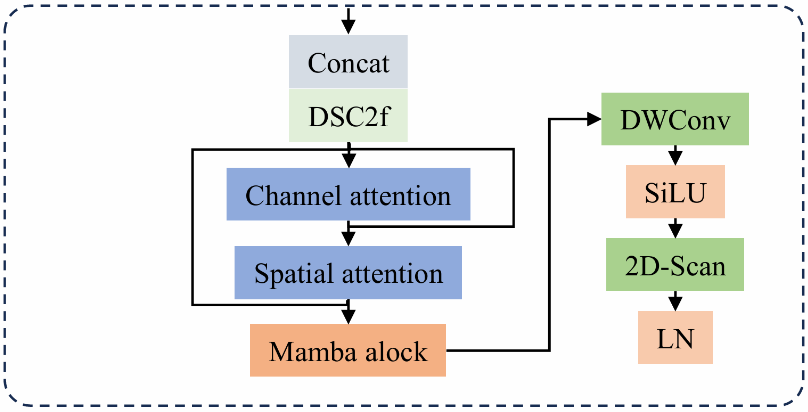

Fig. 4. CBAMamba structural diagram

(We decompose the channel attention module and the spatial attention module, constructing a cascaded structure akin to depthwise separable convolutions. Then, we perform element-wise multiplication on the output features from both modules to obtain the final attention-enhanced features.)

In the CBAMamba module, the intermediate feature maps are fed into the Mamba module. The Mamba module is a complex module that includes various neural network layers, such as linear projection, convolution operations, activation functions, custom S6 modules, and residual connections. The relevant mathematical formulas are shown below:

$$Y = Mamba(X) \tag{7}$$

$$Mamba = MLP + Conv + SiLU + ResNet \tag{8}$$

The combination of these different network layers and operations enables the Mamba module to efficiently handle various complex sequence modeling tasks.

3.4 Detection Head Improvement

This article optimizes the detection head using RepConv. [17] RepConv performs convolution operations on input feature maps by reusing the same convolutional kernel, with each input position undergoing convolution operation only once, thereby reducing the model's parameter size and computational costs. The structural diagram is shown in Fig. 5. The specific formula is shown below:

$$Y_{i,j,k} = \sum_{l} \sum_{d \subset k} X_{i,j,d} \times K \tag{9}$$

where X represents input feature maps, Y represents output feature maps, K represents the convolution kernel. All inputs share and reuse this convolution kernel. l represents the index of the iterated input channel, and d represents the index of the iterated kernel spatial dimension.

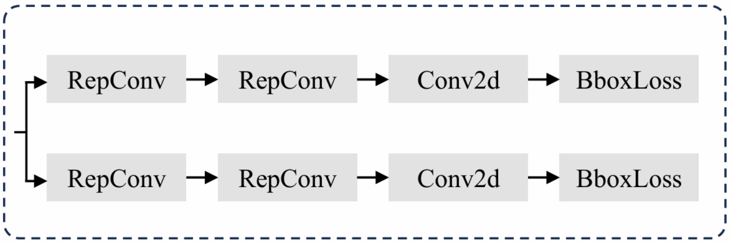

Fig. 5. Structural diagram of detection head
(This module mainly consists of RepConv, Conv2d, and BboxLoss, processing the input data from different perspectives.)

3.5 Loss Function Improvement

The YOLOv8 network uses Complete Intersection over Union (CIoU) Loss [18] to calculate the loss of predicted bounding box coordinates. This loss function considers the overlap area, the distance between the center points, and the aspect ratio of the predicted and true boxes. However, there is a flaw: when the aspect ratio of the predicted box and the true box is linearly related, the penalty term may degenerate to 0, causing the model to fail to correctly regress the bounding boxes. To address this issue, this paper introduces a dynamic non-monotonic focal mechanism, WIoU, to replace the original loss function. This new loss function adopts a dynamic non-monotonic focal mechanism that can adjust the focal point of the loss function dynamically based on the quality of the predicted box. The specific loss function is as follows:

$$L_{WIoU} = \exp\left(\frac{\left(x - x_{gt}\right)^2 + \left(y - y_{gt}\right)^2}{C^2}\right) \tag{10}$$

$$r = \frac{\beta}{\delta \alpha^{\beta - \delta}} \tag{11}$$

$$\beta = \frac{L_{IoU}}{L_{WIoU}} \in [0, +\infty) \tag{12}$$

where x and y represent the predicted bounding box coordinates, x_{gt} and y_{gt} represent Ground truth bounding box coordinates, C is the constant term, r denotes the focal point for adjusting the loss function, β represents the ratio between IoU loss and WIoU loss, and α is the parameter used to calculate r.

4 Experiment and Analysis

4.1 Dataset Construction

A total of 330 images were utilized in the experiment, taken by cameras at the automated drainage operation site, each with a resolution of 4284x5712 pixels. During the training phase, the images were resized to 512x512 pixels. Given the fixed perspective of the camera, the relative positions of the clamping screws and drainage lines in each image remained relatively consistent. To improve the model's ability to generalize, various data augmentation techniques were applied to each image in the training set, including center cropping, random cropping, horizontal and vertical flipping, scaling, random rotation, shearing, and adjustments in brightness. This process expanded the training set by 30 times. Fig. 6 shows some corresponding augmented images.

- Center Cropping (20%): Crop based on the center of the image to highlight key information and reduce background interference, which increases the Average Precision (AP) of small-target detection by approximately 5%.
- Random Cropping (30%): Randomly determine the cropping area to enhance the model's adaptability to different positions and scales of the target, with the mean average precision (mAP) increasing by approximately 4%.
- Horizontal/Vertical Flipping (40%): Increase the directional changes of the target, improving the model's ability to recognize targets in different directions, and the detection recall rate increases by approximately 6%.
- Scaling (25%): Randomly select the scaling ratio to help the model adapt to targets of different sizes, and the mAP increases by approximately 3.5%.
- Random Rotation (35%): Randomly generate the rotation angle to enhance the model's ability to recognize rotated targets, and the detection accuracy increases by approximately 5.5%.
- Shearing (15%): Change the shape of the image to improve the model's adaptability to complex deformations of the target, with the AP increasing by approximately 4.5%.
- Brightness Adjustment (30%): Randomly adjust the brightness to improve the model's detection performance under different lighting conditions, and the mAP increases by approximately 3%.

4.2 Parameter Settings

This method is implemented in PyTorch and deployed on an NVIDIA 3090Ti GPU with 24GB memory. We use validation data to determine the optimal parameters and evaluate the performance of the test data. For a fair comparison, we adopt the baseline model or the optimal parameters from its open-source code. Our method optimizes using stochastic gradient descent and handles non-convex optimization problems with adaptive moment estimation (Adam) [21]. We set the initial learning rate to 0.01, the training batch size to 32, and the number of iterations to 500.

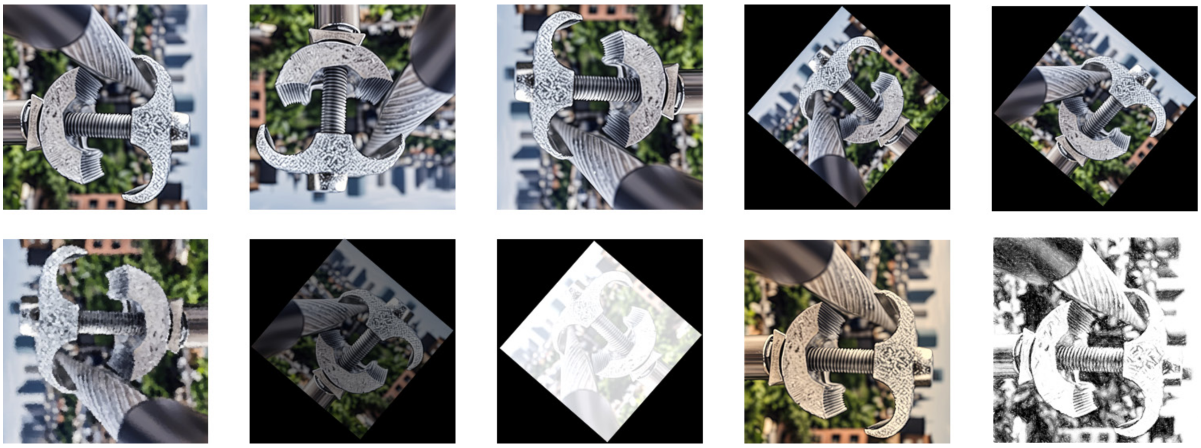

Fig. 6. Dataset images

4.3 Evaluation

The model’s effectiveness was assessed through performance tests, evaluating metrics such as mean average precision (mAP), parameter count, computational complexity, and inference speed [22]. Parameters and computational load were utilized to gauge spatial and temporal complexity. Inference speed signifies the image processing capacity per second. The mAP metric assesses model accuracy, with mAP@0.5 computing category-specific average precision (AP) at an IoU threshold of 0.5 and averaging AP across categories. Conversely, mAP@0.5:0.95 calculates mAP incrementally with a 0.05 stride across IoU thresholds from 0.5 to 0.95 and then averages the results [23].

4.4 Performance Comparison

To verify the superiority of the algorithm in this paper, the algorithm was compared with mainstream object detection algorithms using the same dataset and experimental conditions, including Faster R-CNN, SSD, YOLOv5, and YOLOv7. The results are shown in Table 1.

Table 1. Performance comparison of various models

Models	mAP@0.5	mAP@0.5:0.95	FPS
Faster R-CNN [1]	0.725	0.485	18
SSD [19]	0.630	0.420	32
YOLOv5s [6]	0.830	0.590	70
YOLOv7-tiny [17]	0.810	0.540	65
YOLOv8n [20]	0.890	0.670	82
Ours	0.910	0.700	90

Through comparison, it was found that in terms of mAP@0.5, the algorithm in this paper improved by 9.6%, 12.3%, and 2.2% compared to YOLOv5s, YOLOv7-tiny, and YOLOv8n, respectively; in terms of mAP@0.5:0.95, the improvement was 18.6%, 27.6%, and 4.5%. In terms of FPS, the improved algorithm in this paper increased by 28.6%, 38.5%, and 9.8% compared to the above-mentioned YOLO versions. Compared with Faster R-CNN and SSD, the improved algorithm shows significant improvements in FPS and average precision. While improving detection accuracy, this improved algorithm maintains a high detection rate, demonstrating superior overall performance compared to other mainstream detection algorithms and improved algorithms. This method meets the requirements of real-time detection while achieving optimal detection capabilities. Furthermore, the parameter volume of the improved algorithm is 2.6M, significantly smaller than the Faster R-CNN and SSD models, and it performs better in terms of detection accuracy and frame rate.

4.5 Ablation Experiment

This paper conducted 6 ablation experiments to assess the effectiveness of the improved algorithm. The experiments were carried out using consistent equipment and dataset for training and testing to ensure result comparability. The experimental setups included the proposed integrated approach, the original YOLOv8n, and various combinations of modules and network models, with results presented in Table 2.

Table 2. Comparison of ablation models performance

Models	mAP@0.5	mAP@0.5:0.95	FPS	GFLOPs	Parameter quantity(M)
YOLOv8n	0.780	0.560	87	8.7	3.0
YOLOv8n+DSConv	0.800	0.580	85	6.8	2.5
YOLOv8n+FEM	0.820	0.600	83	8.2	3.2
YOLOv8n+RepConv	0.810	0.590	84	7.8	2.7
YOLOv8n+CBAMamba	0.830	0.610	82	8.9	3.4
Ours	0.910	0.700	90	7.6	2.6

Analysis of Table 2 data leads to several conclusions. While DSConv may reduce algorithm detection ac-curacy and recall rate, it significantly decreases model weights and enhances inference speed, facilitating subsequent model deployment. Introduction of the FEM module resulted in a 5.1% and 7.1% increase in detection accuracy at mAP@0.5 and mAP@0.5:0.95, respectively. Incorporating RepConv and reduc-ing the number of detection head convolutional layers led to a 3.8% decrease in accuracy at mAP@0.5 and 5.4% decrease at mAP@0.5:0.95 respectively, but decreased FPS by 3.4%. The benefits of the CBAMamba layer in improving detection capabilities contributed to an 6.4% increase in accuracy at mAP@0.5 and 8.9% increase at mAP@0.5:0.95 for detection. The final improved algorithm achieved a 16.7% increase in mAP@0.5 and 25.0% increase in mAP@0.5:0.95 compared to YOLOv8n, with a 3.4% FPS increase.

Evaluation against application demand metrics demonstrates that this paper's improved network showcased superior overall detection performance compared to the 6 models assessed. The enhanced network not only exhibited the best detection performance but also boasted smaller parameter size and the quickest detection speed, crucial for deployment in embedded systems.

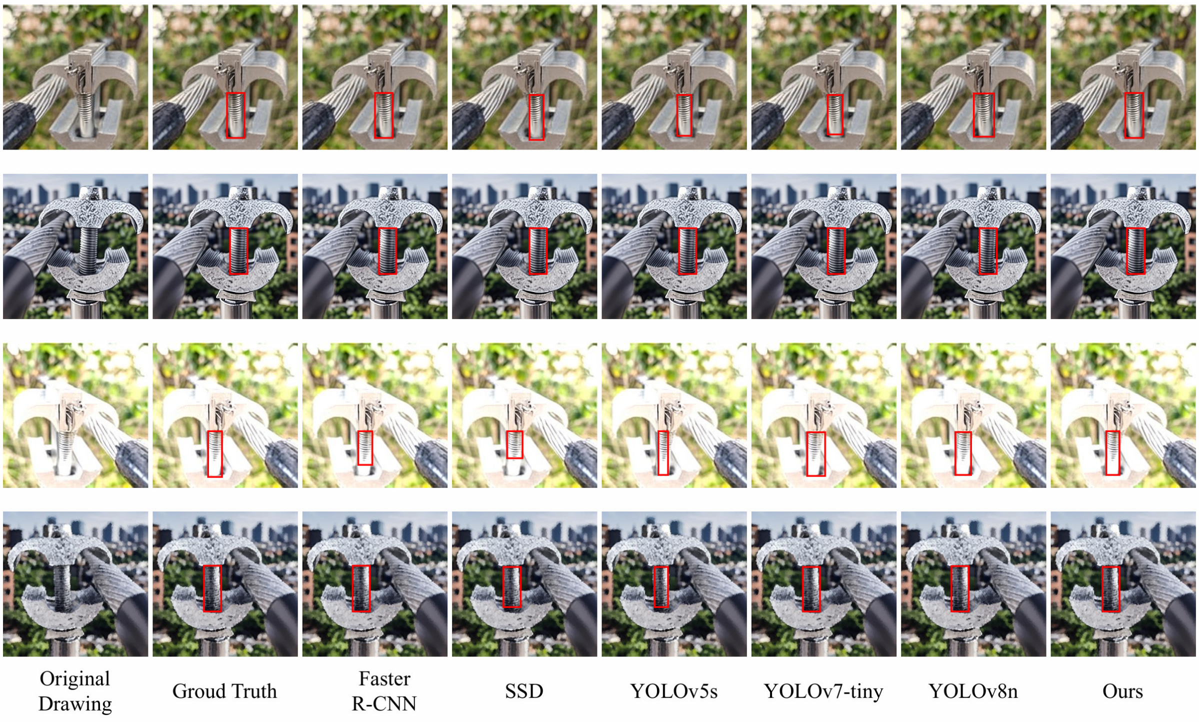

Fig. 7. Comparison of object detection effects

4.6 Visualization Analysis

To assess the detection performance and generalization capability of the enhanced detection model in real-world settings, we conducted a performance evaluation of image detection under varying lighting conditions, diverse object categories, and complex scenes with occlusions. Subsequently, the trained network was deployed in real-world scenarios, and the outcomes are presented in Fig. 7. In practical situations, the locations of line clamp holes were accurately identified, aligning with operational requirements.

Upon thorough examination, it is evident that the method proposed in this study proficiently detects the positions of line clamp holes and exhibits robust generalization performance across diverse complex scenarios, indicating its potential for application in various usage scenarios.

5 Conclusion

This paper delves into the utilization of computer vision for positioning online clamp holes, introducing a novel method for locating clamp holes in power grid drainage devices. Given the relatively fixed position of the central screw of the clamp, the hole's position can be indirectly determined by accurately identifying the central screw's position. The algorithm developed in this research successfully segmented the image of the central screw of the power grid drainage clamp.

To enhance the algorithm, several adjustments were implemented: Firstly, optimizing the YOLOv8 detection algorithm by introducing a combination of DSConv and FEM modules to streamline processing of conventional convolutions within the backbone network, thereby significantly reducing floating-point operations and overall computational load in the convolution process. Secondly, integrating the CBAMamba module to effectively organize the extracted features using a state-space model. Thirdly, replacing Conv in the detection head with RepConv and appropriately downsizing the convolution layer's scale to further lessen floating-point numbers and computational load during the detection process. Lastly, changing the bounding box loss function to the dynamic non-monotonic focusing mechanism WIoU to accelerate model convergence rate and enhance detection performance comprehensively. This approach ensures maintained model performance while achieving lightweighting, facilitating subsequent deployment on embedded systems.

In future work, we aim to extend the lightweight YOLOv8 model for deployment on embedded systems. However, several unresolved challenges remain, such as optimizing the model for real-time processing capabilities, ensuring robustness in various environmental conditions, and addressing power consumption constraints. Addressing these challenges will be crucial for successful deployment and utilization of the model in practical applications.

6 Acknowledgement

This work was supported by State Grid Corporation of China, Research and Demonstration Application of Key Technologies for Full-Range Calibration of Low Voltage Current Transformers for Metering under Live Conditions (5700-202217206A-1-1-ZN).

References

[1] L. Fu, Y. Majeed, X. Zhang, M. Karkee, Q. Zhang, Faster R-CNN-based apple detection in dense-foliage fruiting-wall trees using RGB and depth features for robotic harvesting, Biosystems Engineering (197)(2020) 245-256. https://doi.org/10.1016/j.biosystemseng.2020.07.007

[2] S.M. Wang, H.Y. Xu, X.Z. Zhu, X. Huang, J. Song, Y. Li, Lightweight small object detection algorithm based on improved YOLOv8n aerial photography: PECS-YOLO, Computer Engineering (2024). https://doi.org/10.19678/j.issn.1000-3428.0069353

[3] J.C. Zhang, J. Wei, Y.S. Chen, Improved YOLOv8 real-time lightweight robust hedge detection algorithm, Computer Engineering (2024). https://doi.org/10.19678/j.issn.1000-3428.0069524

[4] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You only look once: Unified, real-time object detection, in: Proc. 2016 IEEE Conference on Computer Vision and Pattern Recognition, 2016.

[5] H.C. Bazame, J.P. Molin, D. Althoff, M. Martello, Detection of coffee fruits on tree branches using computer vision, Scientia Agricola (80)(2023) e20220064. https://doi.org/10.1590/1678-992X-2022-0064

[6] X. Wang, Z. Wu, M. Jia, T. Xu, C. Pan, X. Qi, M. Zhao, Lightweight SM-YOLOv5 tomato fruit detection algorithm for plant factory, Sensors (23)(6)(2023) 3336. https://doi.org/10.3390/s23063336

[7] R. Varghese, M. Sambath, Yolov8: A novel object detection algorithm with enhanced performance and robustness, in: Proc. 2024 International Conference on Advances in Data Engineering and Intelligent Computing Systems (ADICS), 2024. https://doi.org/10.1109/ADICS58448.2024.10533619

[8] A. Bochkovskiy, C.-Y. Wang, H.-Y.M. Liao, YOLOv4: Optimal speed and accuracy of object detection. <https://arxiv.org/abs/2004.10934>, 2020 (accessed 10.04.2024).

[9] J. Redmon, A. Farhadi, YOLOv3: An incremental improvement. <https://arxiv.org/abs/1804.02767>, 2018 (accessed 10.04.2024).

[10] S. Liu, L. Qi, H. Qin, J. Shi, J. Jia, Path aggregation network for instance segmentation, in: Proc. 2018 IEEE Conference on Computer Vision and Pattern Recognition, 2018. https://doi.org/10.1109/CVPR.2018.00913

[11] F. Yu, Z. Xiao, L. Liu, K. Liu, M. Tang, M. Jiang, J. Hou, BiaCanDet: Bioelectrical impedance analysis for breast cancer detection with space-time attention neural network, Expert Systems with Applications (269)(2025) 126223. https://doi.org/10.1016/j.eswa.2024.126223

[12] F. Yu, J.J. Liu, H.C. Yu, W. Cheng, L. Liu, M.H. Jiang, Multimodal Wearable System With Dual-Frequency Enhancement Network for Risk Recognition, IEEE Internet of Things Journal (2025) 1–1. https://doi.org/10.1109/JIOT.2025.3538601

[13] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proc. 2016 IEEE Conference on Computer Vision and Pattern Recognition, 2016. https://doi.org/10.1109/CVPR.2016.90

[14] J. Hu, L. Shen, G. Sun, Squeeze-and-excitation networks, in: Proc. 2018 IEEE Conference on Computer Vision and Pattern Recognition, 2018. https://doi.org/10.1109/CVPR.2018.00745

[15] A. Gu, I. Johnson, K. Goel, K. Saab, T. Dao, A. Rudra, C. Ré, Combining recurrent, convolutional, and continuous-time models with linear state-space layers, in: Proc. Advances in Neural Information Processing Systems, 2021.

[16] F. Yu, Z. Chen, M. Jiang, Z. Tian, T. Peng, X. Hu, Smart clothing system with multiple sensors based on digital twin technology, IEEE Internet of Things Journal (10)(7)(2023) 6377–6387. https://doi.org/10.1109/JIOT.2022.3224947

[17] C.-Y. Wang, A. Bochkovskiy, H.-Y.M. Liao, YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors, in: Proc. 2023 IEEE Conference on Computer Vision and Pattern Recognition, 2023. https://doi.org/10.1109/CVPR52729.2023.00721

[18] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, D. Ren, Distance-IoU loss: Faster and better learning for bounding box regression, in: Proc. 2020 AAAI Conference on Artificial Intelligence, 2020. https://doi.org/10.1609/aaai.v34i07.6999

[19] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, A.C. Berg, SSD: Single shot multibox detector, in: Proc. 2016 European Conference on Computer Vision (ECCV), 2016. https://doi.org/10.1007/978-3-319-46448-0_2

[20] Q. Liu, W. Huang, X. Duan, J. Wei, T. Hu, J. Yu, J. Huang, DSW-YOLOv8n: A new underwater target detection algorithm based on improved YOLOv8n, Electronics (12)(18)(2023) 3892. https://doi.org/10.3390/electronics12183892

[21] D.P. Kingma, J. Ba, Adam: A Method for Stochastic Optimization, in: Proc. 3rd International Conference on Learning Representations (ICLR), 2015.

[22] F. Yu, J. Zhu, Y. Chen, S. Liu, M. Jiang, CAPN: a Combine Attention Partial Network for glove detection, PeerJ Computer Science (9)(2023) e1558. https://doi.org/10.7717/peerj-cs.1558

[23] M. Jiang, Y. Wang, F. Yu, T. Peng, X. Hu, UAV-FDN: Forest-fire detection network for unmanned aerial vehicle perspective, Journal of Intelligent & Fuzzy Systems (45)(4)(2023) 5821–5836. https://doi.org/10.3233/JIFS-231550