BILINGUAL PUBLISHING GROUP

Journal of Computer Science Research

https://journals.bilpubgroup.com/index.php/jcsr

ARTICLE

Improvement and Research on Object Detection Algorithm for Wind Turbine Blade Defect Scenarios Based on YOLOv8

Maoyu Zhu*

Illinois Institute of Technology, Chicago, IL 60616, USA

ABSTRACT

Wind turbine blades are vital for energy generation, where defects can cause efficiency loss and costly maintenance. This paper proposes an improved object detection algorithm based on YOLOv8 for detecting defects in wind turbine blades. Enhancements include network architecture modifications and advanced attention mechanisms, which boost detection accuracy while maintaining real-time processing. Our approach is tested on a custom dataset, showing better performance than the standard YOLOv8 model. These improvements can enhance automated defect detection in wind turbines, reducing downtime and operational costs, and contributing to more efficient renewable energy maintenance.

Keywords: Wind turbine blade defect; Object detection; YOLOv8; Algorithm improvement; Attention mechanisms; Renewable energy maintenance; Real-time processing; Defect detection automation

1. Introduction

1.1 Research background

Wind energy, as a clean and renewable energy source, has been widely applied globally. Wind turbines[1], being the core equipment for wind energy conversion, play a crucial role in the economic efficiency and operational stability of wind farms. The blades of wind turbines are exposed to harsh natural environments, making them prone to defects such as cracks, wear, and corrosion. If these defects are not detected and repaired promptly, they can lead to equipment

*CORRESPONDING AUTHOR:

Maoyu Zhu, Illinois Institute of Technology, Chicago, IL 60616, USA; Email: mzhu41@hawk.iit.edu

ARTICLE INFO

Received: 19 August 2024 | Revised: 31 August 2024 | Accepted: 3 September 2024 | Published Online: 26 September 2024
DOI: https://doi.org/10.30564/jcsr.v6i4.7072

CITATION

Zhu, M., 2024. Improvement and Research on Object Detection Algorithm for Wind Turbine Blade Defect Scenarios Based on YOLOv8. Journal of Computer Science Research. 6(4): 39–50. DOI: https://doi.org/10.30564/jcsr.v6i4.7072

failure and even severe safety accidents.

1.2 Current research status

Traditional methods for detecting blade defects mainly rely on manual inspection and regular maintenance, which are inefficient, costly, and susceptible to human error. With the advancement of deep learning technology, automated detection methods based on computer vision have gradually become a research hotspot. The YOLO (You Only Look Once) series of algorithms have been widely applied due to their efficiency and accuracy in object detection. YOLOv8 [2], as the latest version of this series, has shown significant improvements in detection performance.

1.3 Research problems

Although YOLOv8 performs excellently in many applications, directly applying it to wind turbine blade defect detection still faces challenges. These challenges include detecting small objects in complex backgrounds and distinguishing various types of defects. Therefore, it is necessary to improve and optimize the YOLOv8 algorithm to enhance its adaptability and performance in wind turbine blade defect detection.

1.4 Research objectives

This study aims to improve the YOLOv8 algorithm to enhance its effectiveness in detecting defects in wind turbine blades. The specific objectives include optimizing the algorithm structure, improving detection accuracy, increasing detection speed, and validating the effectiveness of the improved algorithm in practical applications.

2. Materials and methods

2.1 YOLOv8

YOLOv8 (You Only Look Once version 8) is an advanced object detection algorithm primarily used for identifying and locating objects within images, see **Figure 1**. It belongs to the YOLO series, characterized by its ability to simultaneously predict the positions and classes of multiple objects in a single forward pass, thereby offering efficient and real-time performance.

1) Input Layer

The input image undergoes certain preprocessing steps, such as resizing and normalization, to meet the model's input requirements. The input size is usually fixed, for example, 640x640.

2) Backbone

Purpose: Extract basic features from the image.

Structure:

- Conv Layer: Performs convolution operations with common kernel sizes of 3x3, a stride of 2, and padding of 1, to reduce the size of the feature map.
- C2f Module: A key feature extraction module in YOLOv8. It includes multiple convolution layers and Bottleneck modules, which help maintain efficient feature extraction while reducing computational load.
- SPPF (Spatial Pyramid Pooling-Fast) Module: Further extracts multi-scale features by applying max pooling at different scales, then resizes the feature maps to a fixed size before concatenating them.

3) Neck

Purpose: Further processes the multi-scale features extracted by the Backbone and fuses features from different scales to improve detection accuracy and robustness.

Structure:

- Concat Layer: Concatenates feature maps from different layers to fuse multi-scale features.
- Upsample Layer: Performs upsampling operations to enlarge low-resolution feature maps to the same size as high-resolution feature maps for easier feature fusion.
- C2f Module: Further extracts and fuses features in the Neck.

In the Neck, through upsampling and concatenation operations, high-level semantic information is combined with low-level detailed information to form richer feature representations, enhancing the model's ability to detect objects of various scales.

4) Head

Purpose: Generate the final detection results, including the locations and classes of objects.

Structure:

- Conv Layer: Performs convolution operations to further process the fused feature map.
- Detect Module: Predicts object bounding boxes and

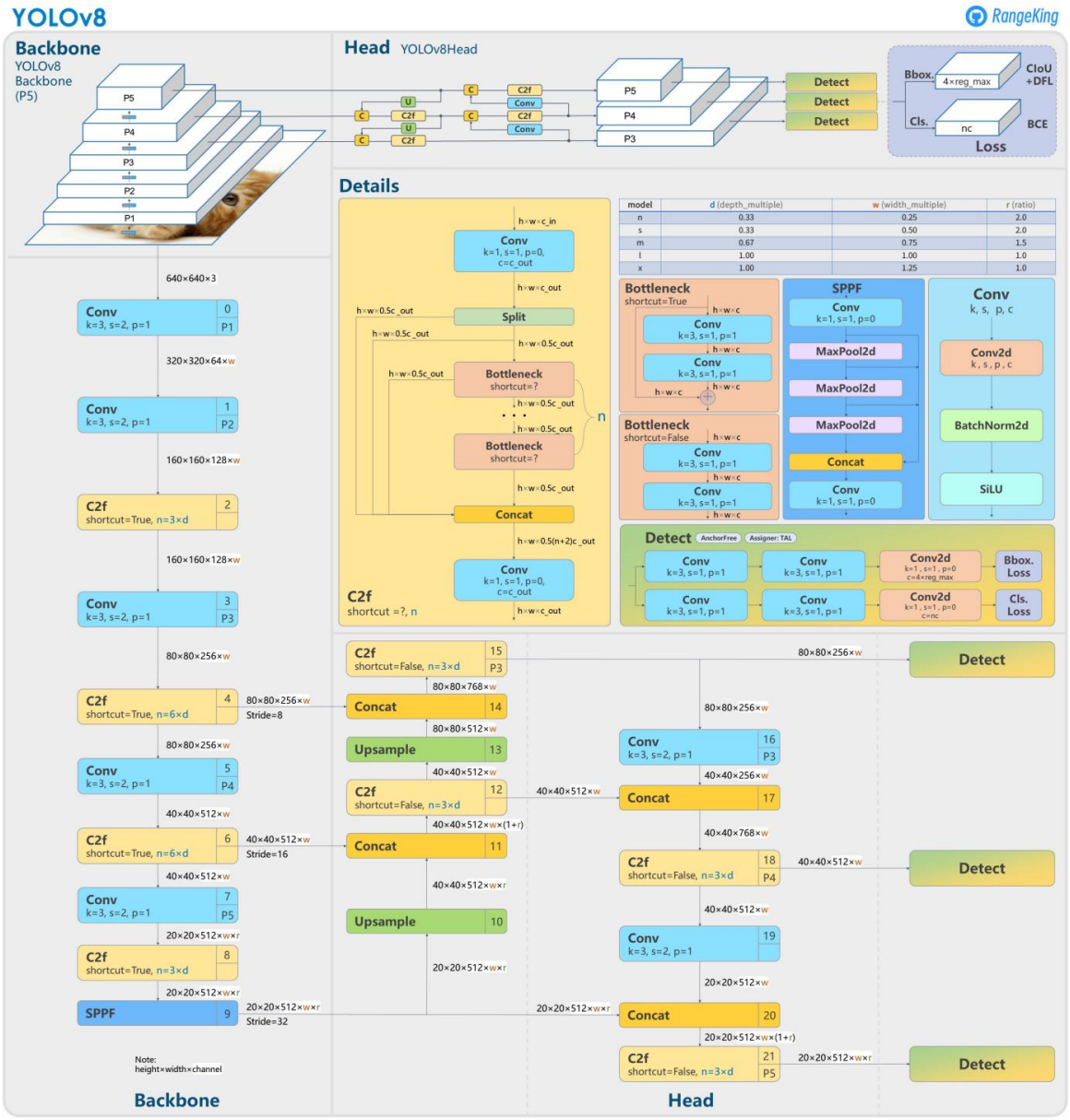

Figure 1. Yolov8 Network Architecture Diagram.

classes, including anchor box prediction and non-maximum suppression (NMS) operations.

- Bbox, Cls, Obj Outputs: Represent bounding box regression, class prediction, and object confidence prediction, respectively.

5) Technical Details:

a) Anchor-Free:

The Anchor-Free[3] mechanism in object detection algorithms does not use predefined anchor boxes to locate and classify objects but directly predicts the center point and size of the object. This method simplifies the detection process and can often improve detection accuracy and efficiency. Specifically, Anchor-Free directly predicts whether each pixel in the image is the center point of an object and, if so, further predicts the width and height of the object without relying on predefined anchor boxes for adjustment. Compared to traditional anchor-based methods, Anchor-Free reduces the setup and tuning work of anchor box sizes and ratios, offering greater flexibility. It can better adapt to objects of different scales and shapes, reducing the problem of anchor box mismatches and improving detection accuracy. In YOLOv8, this mechanism is mainly reflected in the output layer, where feature maps are processed through multiple convolution operations to directly predict the object

confidence, category, width, and height for each pixel point.

b) CIoU:

CIoU[4] (Complete Intersection over Union) is an improved bounding box regression loss function that not only considers the overlapping area between two bounding boxes but also includes the distance between their center points, the aspect ratio of the bounding boxes, and their relative scale. This allows for better optimization of the bounding box's position and shape, enabling the predicted box to more accurately fit the target object. CIoU is particularly effective in cases where the aspect ratio and position of the objects vary significantly.

$$CIoU = 1 - IoU + \frac{p^2(b,b^{gt})}{c^2} + av$$

1) IoU is Intersection over Union.
2) $p^2\left(b, b^{gt}\right)$ is the Euclidean distance between the center points of the predicted box(b) and the ground truth box(b^{gt}).
3) c is the diagonal length of the enclosing box.
4) a is the adjustment parameter, defined as $a = \frac{v}{(1-IoU)+v}$.
5) v is a parameter that measures the consistency of the aspect ratio, defined as $v = \frac{4}{\Pi^2}\left(arctan\frac{w^{gt}}{^{gt}} - arctan\frac{w}{h}\right)^2$

c) DFL:

DFL (Distribution Focal Loss) is a loss function used in classification tasks for object detection. By re-weighting the class probability distribution, it assigns higher weights to samples that are difficult to classify, thereby enhancing the model's ability to recognize hard-to-distinguish categories. DFL effectively addresses the imbalance between positive and negative samples, improving the model's classification accuracy across various scenarios.

d) BCE:

BCE (Binary Cross-Entropy) is a commonly used loss function for binary classification problems. It measures the difference between the model's predicted results and the true labels. The core idea is to evaluate the model's prediction performance by calculating the cross-entropy between the predicted probabilities and the true labels.

2.2 Method

YOLOv8 algorithm improvement

1) Loss Function: Improving powerfuIou using the concept of InnerIoU

a) InnerIoU:

InnerIoU[5] (Inner Intersection over Union) is an improved IoU (Intersection over Union) metric designed to better evaluate the performance of object detection models. Traditional IoU only considers the overlapping area between the predicted box and the ground truth box. In contrast, InnerIoU introduces more geometric information to enhance this evaluation method.

The core idea of InnerIoU is to incorporate internal geometric information to more accurately assess the degree of overlap between the predicted box and the ground truth box. This method not only considers the overlapping area but also takes into account the relative position and shape of the predicted box and the ground truth box. Specifically, InnerIoU performs a fine-grained calculation within the overlapping area to reflect more geometric characteristics between the two.

Advantages of InnerIoU:

- Robustness: InnerIoU better reflects the degree of overlap between the predicted box and the ground truth box in cases of partial occlusion or significant positional deviations.
- More reasonable measurement: Especially when detecting small targets, InnerIoU can more accurately evaluate the detection results.

Calculation Method for InnerIoU:

- Determine the Inner Box: The Inner Box is formed by the minimum bounding rectangle of the overlapping region between the predicted box and the ground truth box. This involves taking the maximum coordinates of the top-left corner and the minimum coordinates of the bottom-right corner of the overlapping part of the two boxes.
- Calculate the Area of the Inner Box: The calculation steps are similar to those of traditional IoU, but here the overlapping part is limited to the Inner Box.
- Compute InnerIoU: Calculate the InnerIoU based on the area of the Inner Box

$$InnerIoU = \frac{InnerBox(Area)}{PredictedBoxArea + GroundTrutBoxArea - InnerBox(Area)}$$

b) PIoU:

PIoU[6] (Precise Intersection over Union) is an improvement over the traditional IoU, aiming to more accurately evaluate the overlap between the predicted box and the

ground truth box. Traditional IoU calculates the overlapping area of two boxes divided by their union area, which, while simple and effective, may not be precise enough in certain situations, especially when there are significant differences in the position and shape of the predicted box and the ground truth box.

PIoU improves the accuracy of IoU calculation by introducing the concepts of boundary offset and area offset. It not only considers the overlapping area but also takes into account the shape differences and boundary alignment of the predicted box and the ground truth box.

PIoU Calculation Method:

- Boundary Offset: Calculate the boundary differences between the predicted box and the ground truth box. The boundary offset considers the offsets in four directions (top, bottom, left, right). These offsets provide a more accurate reflection of the boundary alignment between the two boxes.
- Area Offset: Besides the overlapping area, PIoU also considers the area differences between the two boxes. The area offset measures the area similarity between the predicted box and the ground truth box.
- PIoU Calculation: After integrating the boundary offset and area offset, the PIoU formula is as follows:

$$PIoU = IoU - \alpha \cdot BoundaryOffset - \beta \cdot AreaOffset$$

Among them, α and β are parameters used to adjust the weights of the boundary offset and area offset.

Advantages:

- Higher Accuracy: PIoU introduces boundary and area offsets, allowing for a more precise evaluation of the matching degree between the predicted box and the ground truth box.
- Improved Model Performance: When used as part of the loss function in training object detection models, PIoU can significantly enhance the detection accuracy of the model, especially when handling targets with complex shapes and significant positional differences.
- Robustness: PIoU demonstrates greater robustness for targets of different scales and shapes, making it better suited to adapt to the complex scenarios found in practical applications.

2) Modifying the C2f Module Using the SPD Method [7]

By transforming the spatial information of the input feature map into depth information, the number of channels is increased while retaining fine-grained spatial information. Traditional strided convolution and pooling layers are replaced to avoid the loss of fine-grained information. Non-uniform strided convolution is used to perform convolution operations pixel by pixel, preserving more detailed information. This approach is particularly suitable for low-resolution images and small object detection.

The design is simple and easy to integrate into various CNN architectures, such as YOLOv5 and ResNet. It significantly enhances the performance of low-resolution image and small object detection tasks while maintaining relatively low computational complexity.

3) Enhancing Feature Extraction Using BiFPN (Bidirectional Feature Pyramid Network)

BiFPN[8] adopts a bidirectional fusion path, combining both top-down and bottom-up feature fusion approaches. This design allows for better integration of features from different levels, capturing high-level semantic information while retaining low-level detail information. To optimize feature fusion, BiFPN uses a weighted mechanism on each fusion path, adjusting weights through training to make the feature fusion more efficient.

In terms of computational efficiency, BiFPN employs repeated fusion layers and depthwise separable convolutions, significantly reducing the computation and the number of parameters. As a result, BiFPN not only improves performance but also maintains low computational overhead, making it suitable for real-time object detection tasks.

The structure of BiFPN is highly flexible, capable of handling any number and scale of feature maps. It can be combined with various backbone networks, such as ResNet and EfficientNet, adapting to different object detection tasks. This flexibility ensures that BiFPN can perform well in a variety of application scenarios.

By introducing BiFPN, the detection accuracy and efficiency of YOLOv8 can be significantly improved, especially when dealing with targets that have large scale variations and complex backgrounds. The lightweight design of BiFPN also ensures its applicability in real-time detection tasks.

4) Using GAM [9] Global Attention Mechanism

By leveraging global context information to compute attention weights, the feature extraction capabilities of convolutional neural networks are enhanced. Specifically, GAM first performs global pooling on the input feature map to

generate a global feature vector. Then, a series of fully connected layers and activation functions are used to calculate the attention weights for each channel. Finally, these weights are multiplied element-wise with the original feature map, enhancing important features and suppressing less important ones.

- Global Pooling: Perform global average pooling on the input feature map X, resulting in a global feature vector.
$$Z_c = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} X_{c,i,j}$$
- Attention Weight Calculation:
Using fully connected layers and an activation function (usually sigmoid), generate the attention weights.
$$s = sigmoid\left(W_z + b\right)$$
- Weighted Feature Map:
Multiply the attention weights s element-wise with the input feature map X to obtain the output feature map Y.
$$Y_{c,i,j} = s_c \cdot X_{c,i,j}$$

GAM's Design Balances Efficiency and Effectiveness

GAM is designed to balance computational efficiency and effectiveness. By utilizing global pooling and lightweight fully connected layers, GAM introduces global information calculation while maintaining low computational overhead. Compared to other complex attention mechanisms, GAM has a lower computational complexity yet can still significantly enhance model performance.

GAM can be seamlessly integrated into various convolutional neural network architectures, providing effective feature enhancement for both classification and detection tasks, thereby improving model performance. Its simple structure and good compatibility make it widely applicable across different scenarios.

The GAM attention mechanism enhances feature representation by calculating attention weights using global feature information. It is computationally efficient and easy to integrate, significantly boosting the performance of convolutional neural networks. Introducing GAM can improve the detection accuracy and robustness of YOLOv8, particularly excelling in tasks involving complex backgrounds and multi-object detection.

2.3 Experimental Setup and Model Training

Dataset

a) Using the Wind Turbine Blade Defect Dataset

The defect categories are:

- Burning
- Crack
- Deformity
- Dirt
- Oil
- Peeling
- Rusty

Number of training images: 3,422 images (resolution 640x640), as shown in **Figure 2** and **Figure 3**.

Number of validation images: 856 images (resolution 640x640), as shown in **Figure 4** and **Figure 5**.

b) Data Preprocessing

YOLOv8 provides built-in image preprocessing features, including image scaling, normalization, and data augmentation. The specific preprocessing steps are as follows:

- **Image Scaling**: Scale all images to the required input size of the model.
- **Normalization**: Normalize the pixel values of the images to the [0, 1] range.
- **Data Augmentation**: Use data augmentation techniques such as random cropping, flipping, rotation, and color jitter to increase data diversity and prevent model overfitting.

By utilizing YOLOv8's preprocessing features, the data preprocessing steps can be simplified, ensuring consistency in the dataset and the effectiveness of model training.

Experimental environment

- **Operating System**: Linux
- **Hardware Configuration**: NVIDIA RTX 4090D
- **Software Configuration**: Python 3, PyTorch 2.0, CUDA 11.8, YOLOv8

Model configuration

- **Input Size**: 640x640 pixels
- **Batch Size**: 64
- **Learning Rate**: 0.001
- **Optimizer**: Adam optimizer
- **Loss Function**: A combination of the original YOLOv8 loss and the improved PIoU regression loss based on the innerIoU concept.

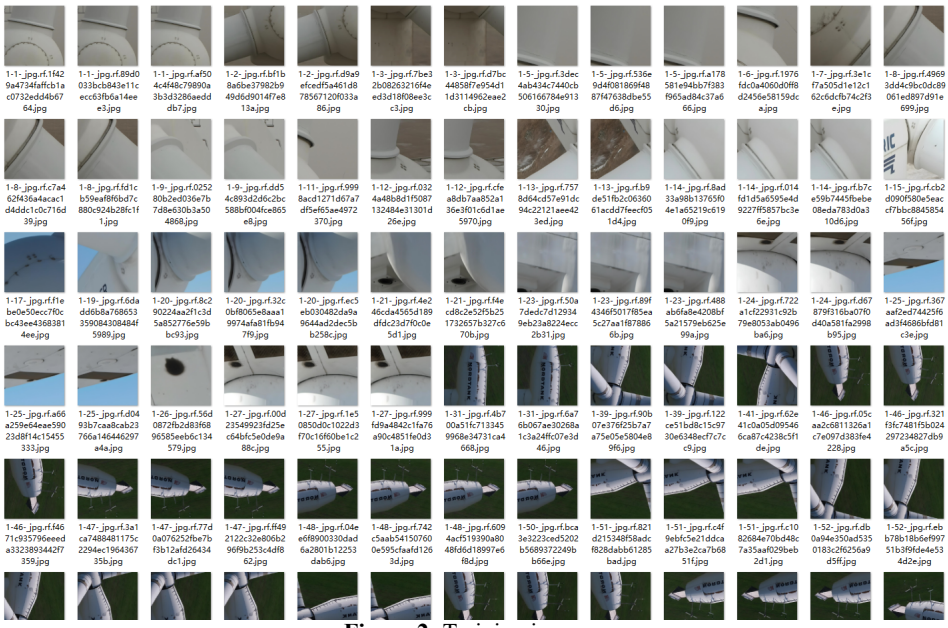

Figure 2. Training images.

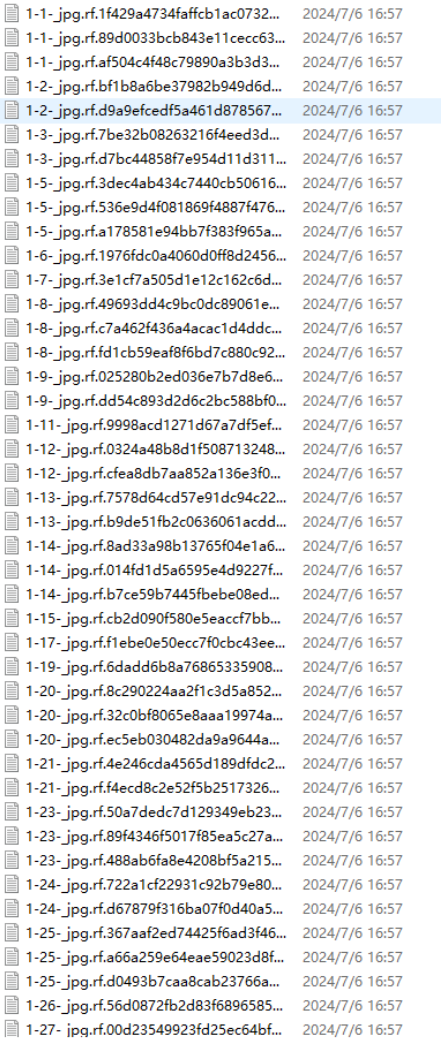

Figure 3. Training labels.

Training process

- **Data Loading**: Use YOLOv8's DataLoader to load training and validation data.
- **Model Initialization**: Initialize the model with pre-trained weights.
- **Training Process:**
    1. **Number of Iterations**: Train for 100 epochs, with each epoch representing one complete iteration over the entire training set.
    2. **Learning Rate Adjustment**: Use a cosine annealing learning rate scheduler to dynamically adjust the learning rate during training.
    3. **Data Augmentation**: During training, use YOLOv8's built-in data augmentation features such as random cropping, flipping, and color jitter.

  Model Validation: At the end of each epoch, evaluate the model performance using the validation set, recording loss values and accuracy metrics.

Performance evaluation

- **Precision**: The proportion of true positive samples among all detected positive samples.
- **Recall**: The proportion of actual positive samples that are correctly detected.
- **F1 Score**: The harmonic mean of precision and recall.
- **Mean Average Precision (mAP)**: The average precision across all classes.

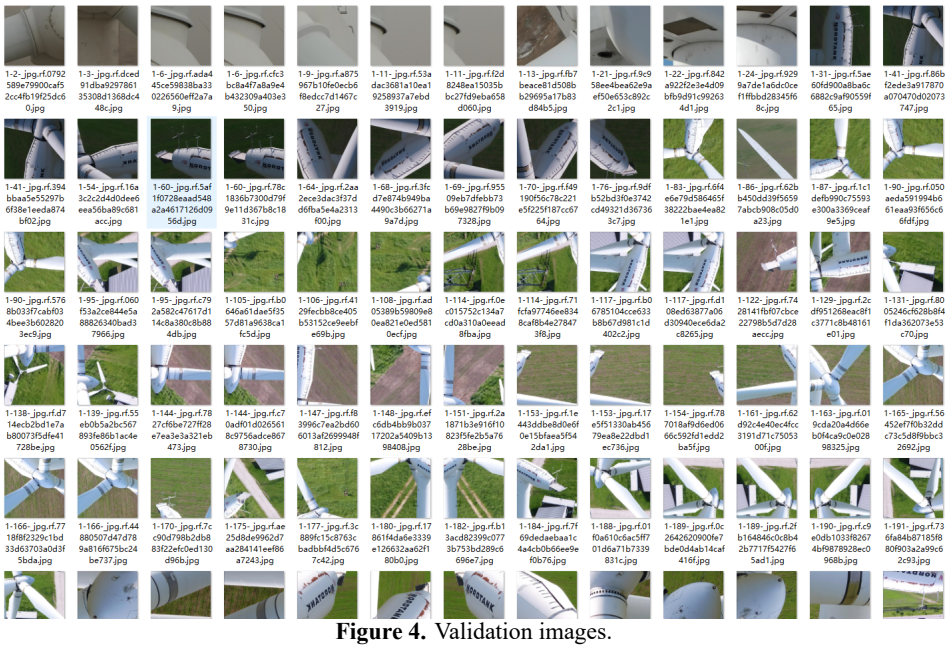

Figure 4. Validation images.

Figure 5. Validation labels.

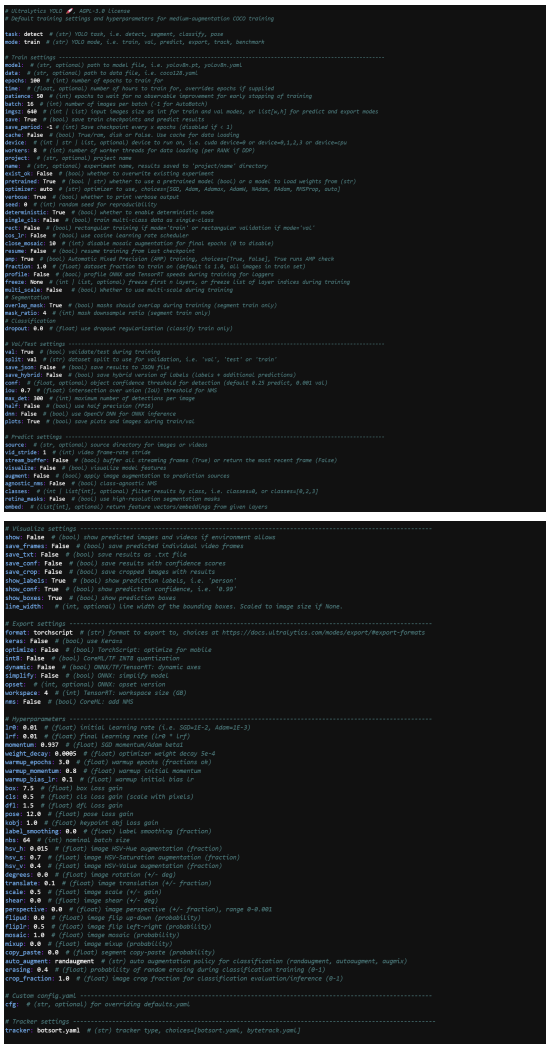

Figure 6. Yolov8 hyperparameters.

3. Experiment Results and Analysis

3.1 Ablation study

In this experiment, we conducted ablation studies with several different algorithms. The results on the validation set are shown in the **Table 1** below:

a) Analysis of Experimental Results

In this experiment, we conducted multiple sets of comparative experiments on YOLOv8 and its improved methods. The results are shown in the table:

1. **YOLOv8**: a. Precision and Recall are both 1 and 0.95, respectively, with an F1 score of 0.83 and a mean Average Precision (mAP) of 0.876. b. This is the baseline model, which shows high precision but relatively low F1 score, indicating a trade-off between precision and recall.
2. **YOLOv8 + SPD**: a. After adding the SPD (Spatial Pyramid Dilated) module, precision remains the same, but the F1 score slightly decreases to 0.82, and the mAP also slightly drops to 0.874. b. This may indicate that the SPD module has limited effectiveness in improving precision under the current settings.
3. **YOLOv8 + PloU (Inner)**: a. By replacing the standard IoU loss function with PloU (Position Loss under Uncertainty), both precision and recall improve to 0.96, with an F1 score of 0.83, and the mAP increases to 0.888. b. This indicates that the PloU loss function has a significant effect on improving the overall performance of the model.
4. **YOLOv8 + SPD + PloU (Inner)**: a. Combining SPD and PloU (Inner) methods, precision remains at 1, recall increases to 0.96, the F1 score rises to 0.84, and the mAP significantly improves to 0.894. b. This suggests that the combination of these two methods further enhances model performance.
5. **YOLOv8 + SPD + PloU (Inner) + GAM**: a. After incorporating GAM (Global Attention Mechanism), precision remains at 1, while recall and the F1 score both increase to 0.96 and 0.87, respectively, and the mAP rises to 0.914. b. This improvement significantly enhances the recall rate and F1 score, indicating the effectiveness of GAM in focusing on important features.
6. **YOLOv8 + SPD + PloU (Inner) + BiFPN + GAM**: a. The final model, which combines BiFPN (Bidirectional Feature Pyramid Network) and GAM, maintains precision at 1, further increases recall to 0.98, achieves an F1 score of 0.87, and reaches the highest mAP of 0.916. b. This combination demonstrates the best overall performance, significantly improving recall and mAP, indicating the effective synergistic effect of various improvements in feature extraction and attention mechanisms.

b) Conclusion

Through the above experiments, we found that:

- Adding the SPD module alone does not significantly improve the model's performance and may even result in a decrease in the F1 score and mean Average Precision (mAP).
- Adding the PloU (Inner) module alone can significantly improve the model's recall rate and mAP.
- Combining the SPD and PloU (Inner) modules can comprehensively improve the model's recall rate, F1 score, and mAP to a certain extent.

These experimental results provide valuable references for model improvement, indicating that adding appropriate improvement modules to the YOLOv8 model can effectively enhance its detection performance. This offers theoretical support and practical guidance for further optimization and application.

3.2 Testing results

All testing results are in **Figure 7** below:

(A)

Figure 7. *Cont.*

Table 1. Comparison of evaluation metrics for yolov8 ablation experiments.

Algorithm	Precision	Recall	F1 Score	mAP
YOLOv8	1	0.95	0.83	0.876
YOLOv8+SPD	1	0.95	0.82	0.874
YOLOv8+PIoU(Inner)	1	0.96	0.83	0.888
YOLOv8+SPD+PIoU(Inner)	1	0.96	0.84	0.89
YOLOv8+SPD+PIoU(Inner)+ GAM	1	0.96	0.87	0.914
YOLOv8+SPD+PIoU(Inner)+BiFPN+GAM	1	0.95	0.87	0.916

(**B**)

(**C**)

Figure 7. Inference images from the validation set during training.

4. Discussion

The main objective of this experiment is to improve the YOLOv8 model to enhance the accuracy and efficiency of wind turbine blade defect detection. Through a series of comparative experiments, we have derived some important conclusions.

4.1 Performance of the baseline model YOLOv8

The baseline model YOLOv8 performs excellently in wind turbine blade defect detection, with a precision of 1, a recall of 0.95, an F1 score of 0.83, and a mean Average Precision (mAP) of 0.876.

This indicates that YOLOv8 effectively balances precision and recall when handling wind turbine blade defect detection. However, the F1 score and mAP suggest that there is still room for improvement, particularly in further enhancing the recall rate to improve overall detection effectiveness.

4.2 Introduction of the spatial pyramid dilated (SPD) module

After introducing the SPD module, although precision remains unchanged, the F1 score slightly decreases to 0.82, and the mAP also slightly drops to 0.874.

This indicates that the SPD module does not significantly improve model performance under the current configuration. One possible reason for the limited performance enhancement is that the SPD module may not capture the details of blade defects adequately during feature extraction. Future research can further optimize SPD's parameter settings or combine it with other feature extraction methods to improve its performance in defect detection.

4.3 Application of the PloU (Inner) loss function

By replacing the standard IoU loss function with the PloU (Inner) loss function, both precision and recall improve to 0.96, the F1 score increases to 0.83, and the mAP rises to

0.888.

This demonstrates that PloU (Inner) has an advantage in handling uncertainty and subtle defects. PloU (Inner) better optimizes the model's detection capabilities, particularly in detecting defects in large, complex structures like wind turbine blades, where it is especially effective.

4.4 Effect of combining SPD and PloU (inner)

Combining the SPD and PloU (Inner) methods significantly enhances model performance, with precision remainingat 1, recall improving to 0.96,the F1 score rising to 0.84, and the mAP significantly increasing to 0.894.

This indicates that the combination of these two methods can complement each other in feature extraction and loss optimization, significantly improving the model's detection performance.

4.5 Introduction of global attention mechanism (GAM) and bidirectional feature pyramid network (BiFPN)

Further introducing GAM and BiFPN leads to a significant improvement in recall and F1 scores, with the mAP reaching its highest value.

This suggests that combining multiple improvement methods enhances model performance at different levels. The synergy between modules significantly improves feature extraction and attention mechanisms, thereby enhancing detection accuracy. In particular, the addition of GAM greatly enhances the model's adaptability to complex backgrounds.

4.6 Comprehensive performance of the optimal model

Ultimately, the improved model combining BiFPN and GAM demonstrates the best overall performance, with precision remaining at 1, recall increasing to 0.98,the F1 score reaching 0.87, and the mAP achieving the highest value of 0.916.

This multi-module combination method not only captures the details of wind turbine blade defects but also enhances the model's adaptability to complex backgrounds, thereby improving overall detection effectiveness.

5. Conclusions

This study significantly improved the accuracy and efficiency of wind turbine blade defect detection by enhancing the YOLOv8 model. The experimental results show that the baseline YOLOv8 model already has high detection performance, but the introduction of the PloU (Inner) loss function, Global Attention Mechanism (GAM),and Bidirectional Feature Pyramid Network (BiFPN) further significantly improved the model's recall rate and mean Average Precision (mAP). These improvement methods excel in handling complex backgrounds and subtle defects, demonstrating their synergistic effects in feature extraction and attention mechanisms.

These technological innovations not only enhance detection performance but also show great potential for practical industrial applications. By detecting and identifying minor defects in real-time, potential failures can be effectively prevented, reducing downtime and maintenance costs, thus improving the overall efficiency and safety of wind power generation systems.

6. Future work

Future research will focus on further optimizing the parameter settings of each module and exploring new deep learning techniques such as self-supervised learning [10] and transfer learning to enhance the model's generalization ability. Additionally, we will conduct validations in more practical wind power scenarios to ensure the model's efficient and stable operation in different environments. We will also develop efficient industrial deployment solutions to promote the application and popularization of the improved model in actual production environments. Through these efforts, we aim to further improve the accuracy and efficiency of wind turbine blade defect detection, providing a solid guarantee for the stable operation of wind power generation systems.

Data Availability Statement

To support the main conclusions of this study, key data screenshots and summaries have been provided in this paper. Due to the special nature or restrictions of the dataset, the full dataset has not been included in the publicly available materials. Readers who require further information are welcome

to contact the corresponding author for discussion.

Conflict of Interest

The author declares that there is no conflict of interest regarding the publication of this paper.

Funding

This research received no external funding.

Acknowledgments

The author would like to acknowledge that there were no additional contributors or support that need to be mentioned for this research.

References

[1] Schubel, P.J., Crossley, R.J., 2012. Wind turbine blade design. Energies. 5(9), 3425–3449.

[2] Sohan, M., Sai Ram, T., Reddy, R., et al., 2024. A review on yolov8 and its advancements. In International Conference on Data Intelligence and Cognitive Informatics (pp. 529–545). Singapore: Springer.

[3] Duan, K., Xie, L., Qi, H., et al., 2020. Corner proposal network for anchor-free, two-stage object detection. In European Conference on Computer Vision (pp. 399–416). Cham: Springer International Publishing.

[4] Gao,J., Chen,Y., Wei,Y., et al., 2021. Detection of specific building in remote sensing images using a novel YOLO-S-CIOU model. Case: gas station identification. Sensors. 21(4), 1375.

[5] Zhang, H., Xu, C., Zhang, S., 2023. Inner-IoU: more effective intersection over union loss with auxiliary bounding box. arXiv preprint arXiv:2311.02877.

[6] Chen, Z., Chen, K., Lin, W., et al., 2020. Piou loss: Towards accurate oriented object detection in complex environments. In Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part V 16 (pp. 195–211). Springer International Publishing.

[7] Sunkara, R., Luo, T., 2022. No more strided convolutions or pooling: A new CNN building block for low-resolution images and small objects. In Joint European conference on machine learning and knowledge discovery in databases (pp. 443-459). Cham: Springer Nature Switzerland.

[8] Tan, M., Pang, R., Le, Q.V., 2020. Efficientdet: Scalable and efficient object detection. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 10781–10790).

[9] Liu,Y., Shao, Z., Hoffmann,N., 2021. Global attention mechanism: Retain information to enhance channel-spatial interactions. arXiv preprint arXiv:2112.05561.

[10] Hendrycks, D., Mazeika, M., Kadavath, S., et al., 2019. Using self-supervised learning can improve model robustness and uncertainty. Advances in neural information processing systems, 32.