

# Steel Surface Defect Detection Method Based on Improved YOLOv8

Yang Long<sup>1, \*</sup>, Yingrui Jia<sup>1</sup>

<sup>1</sup> School of Computer Science, Yangtze University, Jingzhou, 434023, China

\* Corresponding author: Yang Long (Email: 2022710657@yangtzeu.edu.cn)

**Abstract:** This research aims to improve the accuracy and efficiency of steel surface defect detection and address the current insufficient algorithm performance in steel defect detection, especially the challenges in multi-scale feature extraction and parameter efficiency. Based on the improved YOLOv8 network structure, three key improvement points are introduced: the DBB module optimizes multi-scale feature extraction, the MSCA attention mechanism enhances the model's defect detection accuracy, and the optimized GDetect module reduces the amount of parameters and calculations. Extensive experiments and comparative analysis using the NEU-DET data set verify the effectiveness of the improved method. Ablation experiments show that using DBB or MSCA alone has limited performance improvement, but when MSCA and GDetect are combined, the model has significant improvements in Precision, Recall, and mAP50 indicators. Comparative experiments show that the performance of this algorithm is close to or even surpasses some more complex models while remaining relatively lightweight. The average accuracy mean analysis for six defect categories also shows the superior performance of this algorithm for various types of defects. The steel surface defect detection method proposed in this study based on the improved YOLOv8 network has made significant progress in the field of steel defect detection. The improved method improves detection accuracy, solves common problems, and achieves high performance while maintaining lightweight, providing an efficient and feasible solution for steel surface defect detection.

**Keywords:** Steel surface defect detection; YOLOv8; DBB; MSCA; GDetect; Lightweight.

## 1. Introduction

With the continuous development of industrial production, steel, as an important material widely used in construction, transportation, energy and other fields, its quality is directly related to the safety and reliability of the project. However, in the process of steel production and processing, the generation of surface defects inevitably affects its quality, which may lead to problems such as reduced structural strength and corrosion resistance. Therefore, realizing accurate and efficient detection of steel surface defects has become an important part of guaranteeing the quality of steel.

With the rapid development of computer vision and deep learning technology, defect detection methods based on image processing have gradually become a key means to solve this problem. In recent years, target detection algorithms have made significant progress in the field of image processing, and the more popular two-stage detection models are Faster R-CNN<sup>[1]</sup>, FPN<sup>[2]</sup>, Fast R-CNN<sup>[3]</sup> and so on. One-stage classical algorithms are FCOS<sup>[4]</sup>, SSD<sup>[5]</sup>, Retina-Net<sup>[6]</sup>, YOLO<sup>[7,11]</sup> series of algorithms, etc. Li<sup>[12]</sup> et al. proposed an improved YOLOv4 steel surface defect detection method to address the problem of low accuracy of steel defects detection by embedding the attention mechanism into the backbone network structure and modify the path aggregation network into a customized feeling field structure, which enhances the feature extraction capability of the network model and achieves an average accuracy of 85.41% for steel defect detection in four classes. Chen<sup>[13]</sup> et al. proposed a strip steel surface defect detection network based on deformation convolution and attention mechanism, which first enhances the dataset, and according to the proposed novel enhanced deformation feature extraction block and attention mechanism, the The average accuracy of the experimental

results is 82.6%. Zhang<sup>[14]</sup> et al. proposed an improved PP-YOLOE-m's network for detecting surface defects on strip steel, firstly data enhancement is performed to improve the generalization ability of the model, secondly, the coordinate attention is embedded to improve the feature extraction ability of the backbone network, and finally the sensory field is broadened by using a multi-scale network, and the results show that the average accuracy is 80.3%. Zhao<sup>[15]</sup> et al. proposed a steel surface defect detection method with RDD-YOLO based on the YOLOv5 model, the backbone network used Res2Net to widen the receptive field, and BFPN was designed to enhance the neck network, and finally the separation of decoupled headers, and the experimental results showed an average accuracy of 81.1%. Wang<sup>[16]</sup> et al. proposed an improved YOLOv7 steel surface defect detection method, which first uses BiFPN structure to strengthen the feature fusion, and combines the ECA attention mechanism to strengthen the feature channel, and finally uses the SIOU loss function to improve the detection accuracy, and the experimental results show that the average accuracy is 81.9%. Although the above improved method has achieved better results in steel surface defect detection, the performance of the network model is still to be improved, and the number of parameters and computational volume of the model are relatively large, which is not easy to be deployed in embedded and other devices.

In summary, the main contributions of the steel surface defect detection method based on improved YOLOv8 proposed in this paper are as follows:

- (1) Using the DBB<sup>[17]</sup> module to replace the Bottleneck module in C2f to enhance the network's ability to extract multi-scale defect features and improve the performance of the model.
- (2) Add the MSCA<sup>[18]</sup> attention mechanism to the backbone

network to improve the detection accuracy of steel surface defects and reduce the false detection rate and leakage rate.

(3) Improve the Detect part of the original YOLOv8 network by proposing the GDetect module, which reduces the number of parameters and computation of the network, and improves the detection accuracy and efficiency.

## 2. Improve YOLOv8 Network Structure

### 2.1. Improved YOLOv8 network

The YOLOv8 model is composed of Backbone, Neck and Head. YOLOv8's Backbone still uses the design idea of CSP, and designed the C2F module. In the C2F module, only one-half of the number of channels in each layer is used for calculation, and the other half of the channel features are connected across layers, making the C2F module Richer gradient flow and less computation. The dual-stream FPN

structure is used in the Neck part, which can efficiently extract features and speed up module operation. The Head part uses classification and regression decoupling heads instead of coupling heads. The classification loss uses BCE\_Loss, and the regression loss uses Distribution Focal Loss+CIoU Loss. DFL Loss can solve the problem of sample imbalance and replaces the Anchor-based method of the traditional YOLO algorithm with Anchor-free. Use the dynamic sample allocation strategy Task-AlignedAssigner to make the network pay more attention to high-quality anchor boxes.

As shown in Figure 1, this article first replaces all C2f in the original YOLOv8 with C2f\_DBB integrated with the DBB module to improve the feature extraction capability of the network. Then the MSCA attention mechanism was added to the Backbone part to improve the detection accuracy of the network. Finally, the improved GDetect part replaces the original Detect part to improve the performance of the network and reduce the amount of parameters and calculations.

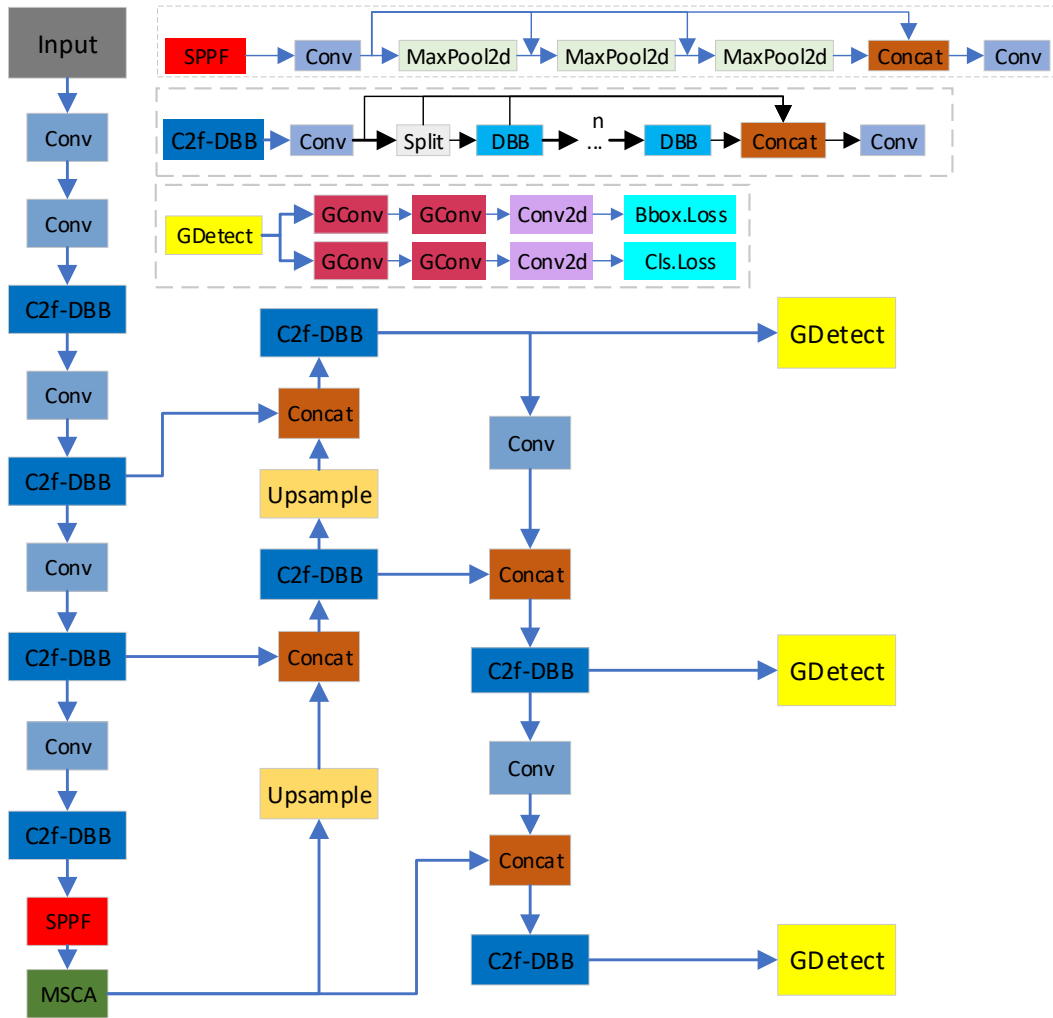


Figure. 1 Improved YOLOv8 network structure diagram

### 2.2. DBB module

The bottleneck module of C2f in the original YOLOv8 network is usually used to reduce the feature dimension. However, it may limit the model's ability to express features because the convolution operation inside may lose some spatial information. In the steel surface defect data set, due to the diversification of defect scales, the original bottleneck module does not perform well in extracting multi-scale

features of defects, resulting in a decrease in accuracy and efficiency. The DBB module draws on the re-parameterization ideas of RepVGG<sup>[19]</sup>, ExpandNet<sup>[20]</sup>, DOConv<sup>[21]</sup> and ACNet<sup>[22]</sup> to enrich the features of a single convolution by combining multiple branches with different scales and complexity. space, thereby improving the performance of the model. As shown in Figure 2 below, the branches of the DBB module include a series of convolution, multi-scale convolution and average pooling operations.

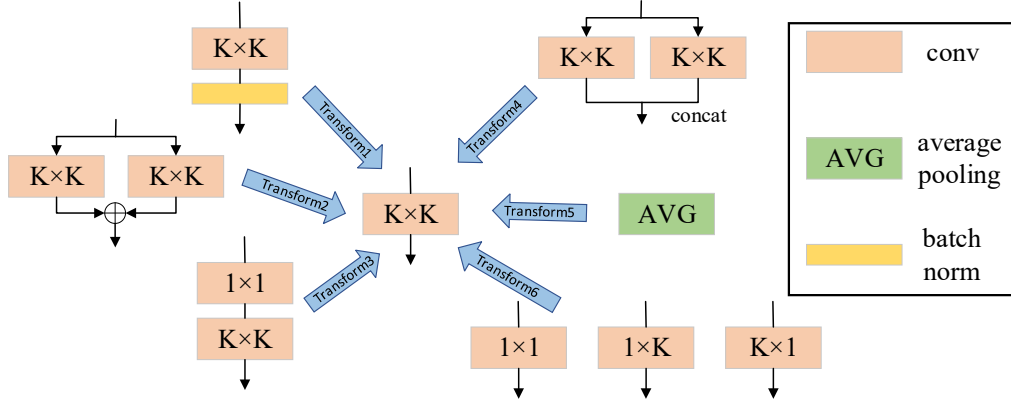


Figure 2. DBB structure

Branch 1 is a convolutional and BN layer to perform channel normalisation and prior scaling; branch 2 is the transform of addition, where additivity ensures that if the outputs of two or more convolutional layers with the same configuration are summed, they can be merged into a single convolution; branch 3 is a sequential convolutional operation, which performs a linear combination of channels through a  $1 \times 1$  convolution, and then performs spatial aggregation through a  $K \times K$  convolution; Branch 4 uses a depth connection to merge  $K \times K$  convolutions, and each  $K \times K$

convolution can in turn be operated by the convolution of branch 3; branch 5 is an average pooling operation, which performs downsampling when  $s > 1$ , but actually performs smoothing when  $s = 1$ ; and branch 6 is a multi-scale convolutional operation, which can cope with multi-scale feature extraction.

As shown in Figure. 3 below, the DBB module is used in this paper to replace the bottleneck module in C2f to cope with multi-scale steel surface defect features and improve the feature extraction capability of the model.

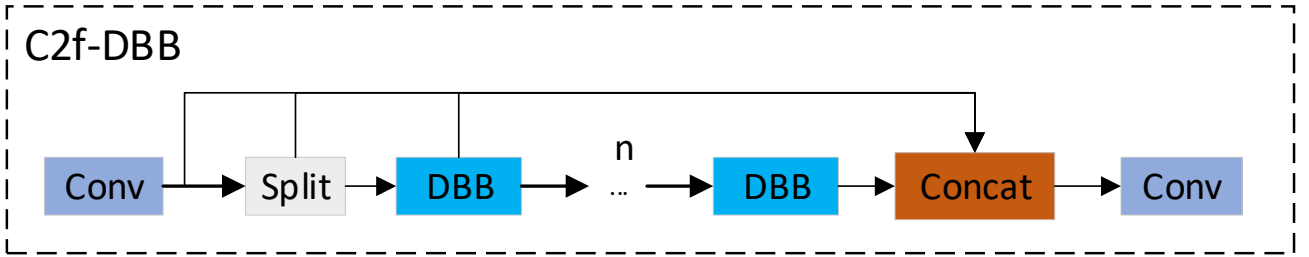


Figure 3. Structure of C2f-DBB

### 2.3. MSCA Attention Mechanism

In order to improve the feature extraction capability of YOLOv8 network and the detection accuracy of multi-scale steel surface defects, MSCA attention mechanism is added to the Backbone part of YOLOv8 network. MSCA is a multi-scale convolutional attention mechanism as shown in Figure. 4 below, which consists of three parts: depth convolution to gather local information, multi-branch depth bar convolution to capture multi-scale context, and  $1 \times 1$  convolution to construct the relationship between different channels. Because strip convolution is lightweight, in each branch, two depth strip convolutions are used as the standard depth convolution of the larger kernel. Mathematically, the formula for MSCA is:

$$Att = Conv_{1 \times 1} \left( \sum_{i=0}^3 Scale_i (DW - Conv(F)) \right) \quad (1)$$

$$Out = Att \otimes F \quad (2)$$

Where  $F$  denotes the input features.  $Att$  and  $Out$  are the attention graph and the output, respectively,  $\otimes$  is an element-level multiplication operation.  $DW - Conv$  denotes the depth-level convolution,  $Scale_i$ ,  $i \in \{0, 1, 2, 3\}$ , which denotes

that the  $i$ th branch in the graph scales to a value of 0, and is the identity connection.

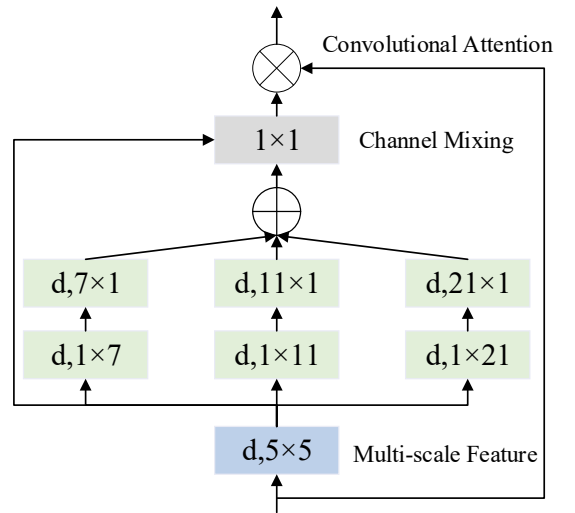


Figure 4. MSCA structure

### 2.4. Improved GDetect module

In order to reduce the number of parameters and computation of the YOLOv8 network, and at the same time to improve the accuracy and efficiency of detection, this paper improves the Detect module of the original YOLOv8

network.GConv<sup>[23]</sup> module, as shown in Figure. 5, in ordinary convolutional operation, the connection between the input and the output channels is fully connected, i.e., each input channel is connected to the output channel. Whereas in grouped convolution, the input channels are uniformly divided into groups, and the channels in each group are subjected to convolution operation, and then the outputs of the groups are combined together to form the final output. Grouped convolution can reduce the number of parameters of the model, thus reducing the computational cost and improving the computational efficiency of the model to some extent.

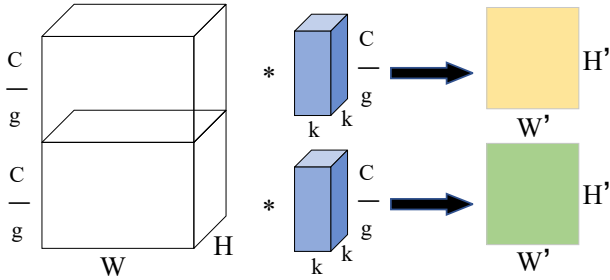


Figure 5. GConv structure

The number of parameters and FLOPs for ordinary convolutional layers are calculated as:

$$\text{params} = C_1 \times h_1 \times w_1 \times C_2 \quad (3)$$

$$FLOPs = C_1 \times C_2 \times W_2 \times H_2 \times w_1 \times h_1 \quad (4)$$

The number of parameters and GLOPs for grouped convolutions are calculated as:

$$\text{params} = \frac{1}{g} \times C_1 \times h_1 \times w_1 \times C_2 \quad (5)$$

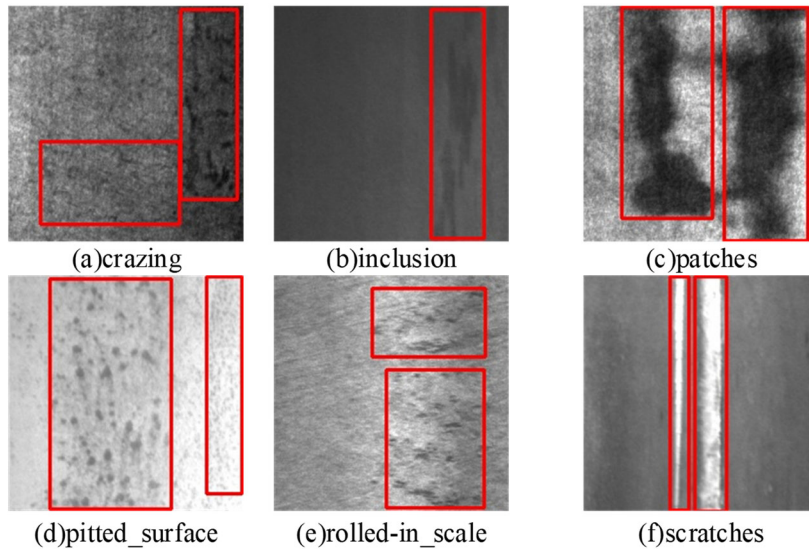


Figure 7. Example of steel surface defects

### 3.2. Experimental environment configuration

In this paper, the experimental hardware environment is configured as NVIDIA GeForce RTX 3080 with 12G of video memory, and the compilation environment is

$$FLOPs = \frac{1}{g} \times C_1 \times C_2 \times W_2 \times H_2 \times w_1 \times h_1 \quad (6)$$

It can be seen that this grouping structure reduces the number of parameters per convolutional kernel and thus reduces the computational complexity. GDetect is shown in Figure. 6. In this paper, GConv is replaced with the standard convolutional module in the Detect section of the original YOLOv8.

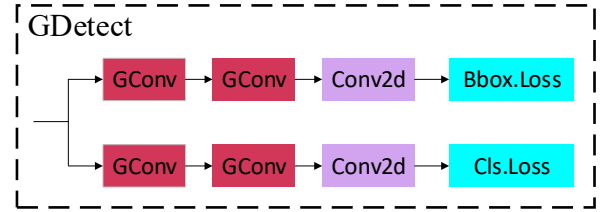


Figure 6 GDetect structure

## 3. Experimental Results and Analysis

### 3.1. Dataset Introduction

This paper uses the NEU-DET steel surface defects dataset published by Northeastern University, which has six types of defects, namely crazing, inclusion, patches, pitted\_surface, rolled-in\_scale, and scratches. The images are all greyscale images of 200×200 size. 300 images for each type of defect, 1800 images in total. The defective images are annotated and preprocessed using the Labeling tool to get the label file in xml format, and then converted to a txt format file for YOLO training, and the training set and validation set are divided according to the ratio of 8:2, and some sample dataset graphs are shown in Figure. 7.

Python3.8.16+torch2.0.0+CUDA11.8. Training is carried out in the PyTorch network framework, and the optimiser is SGD, with hyper-parameters set to a batch size of 32, a training period of 150, and an initial learning rate of 0.01, the size dimension of the input image is 640×640, and none of the pre-training weights are added during the training process.

### 3.3. Evaluation indicators

The evaluation metrics used in this paper are Precision P, Recall R, F1 Factor, Mean Precision mAP, Number of Parameters Params, Giga Floating Point Budgets per Second GFLOPs. the relevant formulas are as follows:

$$P = \frac{TP}{(TP + FP)} \quad (7)$$

$$R = \frac{TP}{TP + FN} \quad (8)$$

$$F1 = \frac{2 \times (P \times R)}{(P + R)} \quad (9)$$

$$mAP = \frac{1}{n} \sum_{i=1}^n \int_0^1 P(R) d(R) \quad (10)$$

In the formula: TP represents the number of true examples, FP represents the number of false positive examples, and FN represents the number of false negative examples.

Among them, the precision rate measures the accuracy of the model in positive category prediction; the recall rate measures the model's coverage rate in the positive category samples; the F1 factor is used to comprehensively measure the performance of the model; the average accuracy rate calculates the accuracy of the model in different categories. The average accuracy value is then averaged; the number of parameters refers to the number of trainable parameters in the model. Generally, the more parameters, the more complex the model; GFLOPs represents the billions of floating point

operations performed by the model per second, and Model speed is related to performance;

### 3.4. Ablation experiment

In order to verify the effectiveness of this article's improvement of the YOLOv8 network structure, ablation experiments were conducted on different improvement points under the same conditions in the experimental environment. The results are shown in Table 1. The first set of experiments did not make changes to specific improvement points and served as a baseline experiment. The second set of experiments added the DBB module, and the precision, recall and mAP were improved, while the number of parameters and calculations of the model remained unchanged. The third set of experiments added the MSCA attention mechanism. Precision, recall and mAP were significantly improved, reaching 70.4%, 73% and 77.3% respectively, and the number of parameters increased slightly to 8.2M. The fourth group improved the Detect module, with Precision reaching 74.9%, but Recall dropped slightly to 69.8%, and mAP50 was 77.2%. The number of parameters is reduced to 5.6M, and GLOPs are also significantly reduced to 2.416. Group 5 improved both DBB and MSCA, with a mAP50 of 76.1%. The parameter size increases to 8.2M. The sixth set of experiments improves DBB and GDetect. Precision dropped to 65.5%, but Recall increased to 72.4%, and mAP50 was 76.0%. The number of parameters and GLOPs is significantly reduced to 5.6M and 2.417. Group 7 experiments improve MSCA and GDetect. Precision reached 74.5%, Recall was 72.6%, and mAP50 was significantly improved to 78.6%. The parameter size increased slightly to 5.7M. The 8th set of experiments improved all three aspects. Precision and Recall reached 75.7% and 71.8% respectively, and mAP50 was 78.9%. The number of parameters and GLOPs remains at 5.7M.

Table 1. Ablation experiment results

No.	DBB	MSCA	GDetect	Precision(%)	Recall(%)	mAP50(%)	Params(M)	GLOPs
1				68.6	70.4	75.5	3.007	8.1
2	✓			68.8	72.2	75.7	3.007	8.1
3		✓		70.4	73.0	77.3	3.101	8.2
4			✓	74.9	69.8	77.2	2.416	5.6
5	✓	✓		69.6	71.2	76.1	3.101	8.2
6	✓		✓	65.5	72.4	76.0	2.417	5.6
7		✓	✓	74.5	72.6	78.6	2.510	5.7
8	✓	✓	✓	75.7	71.8	78.9	2.510	5.7

In general, using DBB or MSCA alone has limited performance improvement. However, the application of GDetect improves Precision while reducing parameters and calculation amount, while the combination of MSCA and GDetect performs well in all indicators, especially when all three improvement points are applied, the model reaches the best performance.

### 3.5. Comparative Experiment

In order to verify the effectiveness of the algorithm in this paper, a comparative experiment as shown in Table 2 was established. The experimental results show that after comparing multiple YOLO series models and the improved algorithm proposed in this article. In these models, the algorithm in this paper achieves satisfactory performance

with relatively low parameters and computational complexity. YOLOv8s has reached the highest level in performance, but the model is larger and requires more computing resources. In contrast, the algorithm in this article achieves performance close to YOLOv8s while keeping the model lightweight, especially in mAP50 and Precision indicators. YOLOv6n shows a relatively balanced state, with both performance and model complexity at a medium level. However, the performance of YOLOv7-tiny is slightly insufficient in lightweight models, while YOLOv5s has relatively high performance but larger models and more complex calculations. Taken together, the improvement of the YOLOv8 network structure by this algorithm proves its effectiveness in model simplification and performance improvement.

**Table 2.** Comparative test results

Model	Precision(%)	Recall(%)	F1(%)	Map50(%)	Params(M)	GLOPs
YOLOv5n	67.2	67.9	65	71.2	1.767	4.2
YOLOv5s	64.5	70.2	66	72.3	7.026	15.8
YOLOv6n	70.5	72.3	71	76.3	4.234	11.8
YOLOv7-tiny	57.3	68.6	59	67.3	6.028	13.2
YOLOv8n	68.6	70.4	70	75.5	3.007	8.1
YOLOv8s	75.9	73.1	74	77.0	11.127	28.4
<b>Our</b>	75.7	71.8	73	78.9	2.510	5.7

Table 3 shows the average accuracy of each category under different YOLO models. It can be found that for each defect category, the algorithm in this paper has achieved the highest mAP value in most cases. Especially in the categories of "crazing", "patches" and "rolled-in\_scale", this algorithm shows obvious advantages, reaching 44.9%, 93.2% and 69.6% mAP respectively. In comparison, there are differences in the performance of other models in different categories, but

overall, YOLOv8s and YOLOv8n also have better performance in mAP of each category, following the algorithm of this article. However, the YOLOv5 series and YOLOv7-tiny have significantly lower mAP than other models in most categories. Taken together, the algorithm in this paper has achieved a high average accuracy in each defect category, showing good recognition capabilities and overall performance improvement for different defect types.

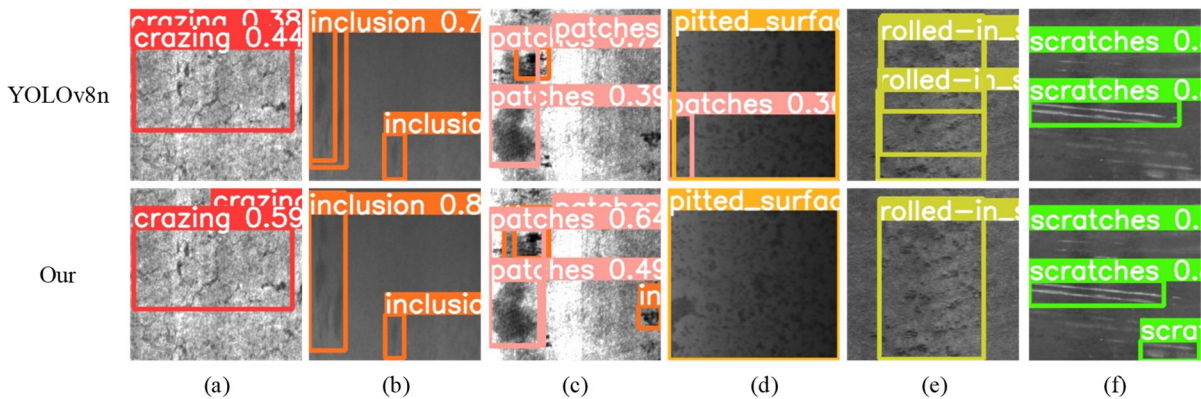
**Table 3.** Average accuracy results of six types of defects

Model	crazing	patches	inclusion	pitted_surface	rolled-in_scale	scratches
YOLOv5n	36.8	90.4	79.6	80.2	54.4	85.9
YOLOv5s	33.3	89.3	81.1	79.5	63.0	87.8
YOLOv6n	41.5	92.7	80.3	87.7	64.8	91.0
YOLOv7-tiny	28.4	87.8	71.9	81.4	52.4	82.0
YOLOv8n	39.2	89.1	86.0	84.7	63.9	89.9
YOLOv8s	42.2	92.0	84.4	85.6	66.8	91.0
<b>Our</b>	44.9	93.2	86.5	86.5	69.6	92.7

### 3.6. Visual result analysis

In order to verify the superiority of this algorithm in actual detection, 6 sets of experimental comparisons are designed as shown in Figure 8. First, group a results show that the algorithm in this paper surpasses the original YOLOv8n in terms of detection accuracy, which means that the algorithm in this paper identifies defects more accurately. Secondly, the results of groups c and f show that compared with the original YOLOv8n, the algorithm in this paper has a lower miss detection rate, which means it is easier to capture potential defects, thus improving the comprehensiveness of detection.

The results of groups b and e show that the original YOLOv8n has the problem of bounding box overlap. The algorithm in this paper can improve the comprehensive extraction of defect features, solve the problem of bounding box overlap, and improve detection accuracy. Finally, the results of group d show that the algorithm in this paper has reduced the false detection rate, which means that normal objects are less likely to be misjudged as defects, thereby reducing the risk of false positives. Taken together, the algorithm in this paper has achieved significant improvements in accuracy, missed detection rate, and false detection rate, making it a more reliable and efficient defect detection tool.

**Figure 8** Visualization results chart

## 4. Conclusion

This study proposes a steel surface defect detection method based on the improved YOLOv8 model, aiming to cope with the limitations of existing models in terms of performance and

computing resource requirements. By introducing the DBB module, MSCA attention mechanism and improvements to the GDetect module, the algorithm in this paper has achieved significant performance improvements in the steel surface defect detection task.

First of all, experimental results show that using DBB or

MSCA alone has limited performance improvement, while the application of the GDetect module improves detection accuracy while reducing parameters and calculation volume. Especially when combined with MSCA and GDetect improvements, the model performs excellently in precision, recall and average precision. The ablation experiment verified the effectiveness of the improvement method proposed in this article, and pointed out the contribution of each improvement point, providing guidance for further improvements.

Comparative experimental results show that compared with other YOLO series models, the algorithm in this paper achieves satisfactory performance while keeping the model lightweight. Especially in terms of mAP50 and Precision indicators, the algorithm in this paper achieves performance levels close to the optimal model with lower parameters and computational complexity. In addition, the average accuracy of different defect categories shows that the algorithm in this paper has good recognition ability and performance in each category.

Through visual result analysis, the algorithm in this paper has achieved significant improvements in detection accuracy, missed detection rate, and false detection rate. Compared with the original YOLOv8 model, the algorithm in this paper can identify defects more accurately, improve detection comprehensiveness, and reduce the missed detection rate and false detection rate, making it a more reliable and efficient steel surface defect detection tool.

In summary, the improved method proposed in this paper has achieved significant performance improvement in the task of detecting steel surface defects. Future research can continue to explore more effective model fusion strategies, data enhancement techniques, and more refined model parameter tuning to further improve the accuracy and efficiency of steel surface defect detection.

## References

- [1] REN S, HE K, GIRSHICK R, et al. Faster r-cnn: Towards real-time object detection with region proposal networks [J]. *Advances in neural information processing systems*, 2015, 28.
- [2] Lin T Y, Dollár P, Girshick R, et al. Feature pyramid networks for object detection[C]//*Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017: 2117-2125.
- [3] Girshick R. Fast r-cnn[C]//*Proceedings of the IEEE international conference on computer vision*. 2015: 1440-1448.
- [4] Tian Z, Shen C, Chen H, et al. Fcos: Fully convolutional one-stage object detection[C]//*Proceedings of the IEEE/CVF international conference on computer vision*. 2019: 9627-9636.
- [5] Liu W, Anguelov D, Erhan D, et al. Ssd: Single shot multibox detector[C]//*Computer Vision—ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*. Springer International Publishing, 2016: 21-37.
- [6] Lin T Y, Goyal P, Girshick R, et al. Focal loss for dense object detection[C]//*Proceedings of the IEEE international conference on computer vision*. 2017: 2980-2988.
- [7] Redmon J, Farhadi A. Yolov3: An incremental improvement[J]. *arXiv preprint arXiv:1804.02767*, 2018.
- [8] Bochkovskiy A, Wang C Y, Liao H Y M. Yolov4: Optimal speed and accuracy of object detection[J]. *arXiv preprint arXiv:2004.10934*, 2020.
- [9] Ge Z, Liu S, Wang F, et al. Yolox: Exceeding yolo series in 2021[J]. *arXiv preprint arXiv:2107.08430*, 2021.
- [10] Li C, Li L, Jiang H, et al. YOLOv6: A single-stage object detection framework for industrial applications[J]. *arXiv preprint arXiv:2209.02976*, 2022.
- [11] Wang C Y, Bochkovskiy A, Liao H Y M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors[C]//*Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023: 7464-7475.
- [12] Li M, Wang H, Wan Z. Surface defect detection of steel strips based on improved YOLOv4[J]. *Computers and Electrical Engineering*, 2022, 102: 108208.
- [13] Chen H, Du Y, Fu Y, et al. DCAM-Net: A Rapid Detection Network for Strip Steel Surface Defects Based on Deformable Convolution and Attention Mechanism[J]. *IEEE Transactions on Instrumentation and Measurement*, 2023, 72: 1-12.
- [14] Zhang Y, Liu X, Guo J, et al. Surface Defect Detection of Strip-Steel Based on an Improved PP-YOLOE-m Detection Network[J]. *Electronics*, 2022, 11(16): 2603.
- [15] Zhao C, Shu X, Yan X, et al. RDD-YOLO: A modified YOLO for detection of steel surface defects[J]. *Measurement*, 2023, 214: 112776.
- [16] Wang Y, Wang H, \*\*n Z. Efficient detection model of steel strip surface defects based on YOLO-V7[J]. *IEEE Access*, 2022, 10: 133936-133944.
- [17] Ding X, Zhang X, Han J, et al. Diverse branch block: Building a convolution as an inception-like unit[C]//*Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021: 10886-10895.
- [18] Guo M H, Lu C Z, Hou Q, et al. Segnext: Rethinking convolutional attention design for semantic segmentation[J]. *Advances in Neural Information Processing Systems*, 2022, 35: 1140-1156.
- [19] Ding X, Zhang X, Ma N, et al. Repvgg: Making vgg-style convnets great again[C]//*Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2021: 13733-13742.
- [20] Marnerides D, Bashford-Rogers T, Hatchett J, et al. Expandnet: A deep convolutional neural network for high dynamic range expansion from low dynamic range content[C]//*Computer Graphics Forum*. 2018, 37(2): 37-49.
- [21] Cao J, Li Y, Sun M, et al. Do-conv: Depthwise over-parameterized convolutional layer[J]. *IEEE Transactions on Image Processing*, 2022, 31: 3726-3736.
- [22] Tian C, Xu Y, Zuo W, et al. Asymmetric CNN for image superresolution[J]. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2021, 52(6): 3718-3730.
- [23] \*\*e S, Girshick R, Dollár P, et al. Aggregated residual transformations for deep neural networks[C]//*Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017: 1492-1500.