Research on the Improved YOLOv8 Algorithm for Steel Surface Defect Detection Method

Kaidi Liu, and Ji Zhao*

Abstract—**The detection of surface defects in the steel production process is crucial for ensuring product quality. To address the issues of large variations in defect sizes, high similarity between abnormal and normal regions, and low detection accuracy in traditional steel surface defect detection, this paper proposes an improved YOLOv8-based method for steel surface defect detection, named CEC-YOLO. Firstly, a Cross Channel Group Interaction (CCGI) module is designed to enhance the model's feature extraction and fusion capabilities. Secondly, a Sparse Spatial Reduction Attention (SSRA) module is introduced, enabling the model to focus on the most critical regions for defect detection. Finally, a Dynamic Ratio IoU (DRI) regression strategy is implemented, which adaptively adjusts the size of detection boxes based on the size of the target and is suitable for different datasets, making it more appropriate for industrial inspection tasks. Experimental results show that the proposed method achieves a mAP of 85.5% on the GC10-DET dataset and a mAP of 78.8% on the NEU-DET dataset. The proposed model demonstrates superior detection accuracy and computational efficiency compared to existing approaches, addressing critical industry requirements for high-speed and reliable steel surface defect identification.**

Index Terms—**Steel surface fault detection, Attention mechanism, Group convolution, Dynamic regression, Object detection**

I. Introduction

STEEL, as a crucial material widely used in construction, machinery, automobiles, aerospace, and other fields, has surface quality that directly impacts the performance, useful life and safety of final products. To ensure the quality and safety of steel and prevent defective products from flowing into downstream industries and causing economic losses or safety accidents, detecting steel surface defects has become an urgent task [1]. Traditional steel defect detection is based on manual visual inspection and the stroboscopic method. Although these traditional methods can identify steel surface defects, they are limited by the experience and skills of inspectors, resulting in inconsistent detection speed and quality. Moreover, training qualified inspectors requires significant time and financial investment [2]. With technological advancements, machine vision technology has been introduced. Although traditional machine learning algorithms cover multiple critical steps such as image pre-processing, feature extraction, and classification, they face challenges such as slow processing speed and insufficient accuracy [3]. In recent years, the integration of deep learning and machine vision technology has demonstrated strong capabilities in automated detection, enabling accurate localization and prediction of defect types. Methods based on Convolutional Neural Networks (CNN) exhibit greater robustness and generalization capabilities, making them a key detection method in the industrial sector [4].

Current deep learning-based object detection algorithms are primarily categorized into two types: two-stage detectors and single-stage detectors.Two-stage algorithms, such as R-CNN [5], Fast R-CNN [6], and Faster R-CNN [7], achieve higher detection accuracy by first generating region proposals in the initial stage and then classifying and refining their locations in the second stage. In contrast, single-stage detectors, such as SSD [8] and the YOLO series [9–11], perform classification and localization simultaneously on feature maps, offering faster inference speeds at the cost of slightly lower accuracy. Given the industrial demand for real-time processing, single-stage detectors are more suitable for steel surface defect detection tasks, where both speed and efficiency are critical.

Chen et al. [12] proposed substituting the Spatial Pyramid Pooling (SPP) module in YOLOX's backbone network with a Coordinate Attention (CA) mechanism. Their approach also implemented the CLAHE technique to enhance the low-contrast characteristics typical of steel surface defect imagery. Cai et al. [12] proposed a multi-stage object detection framework, Cascade R-CNN (Region-CNN), which progressively improves defect detection quality through resampling and effectively minimizes overfitting issues.Dou et al. [13] proposed a lightweight YOLOv8-based algorithm for few-shot steel plate defect detection, addressing the issue of network model training failure due to insufficient training samples. Zhou et al. [14] proposed an efficient detection network, ETDNet, based on transformers. This network extracts global features using a lightweight vision transformer, fuses multi-level features through a channel-modulated feature pyramid network, and resolves representation inconsistencies between classification and regression tasks in steel surface defect detection using a task-oriented decoupled head. However, these methods exhibit relatively low detection accuracy when addressing the problem of small target defect detection.

To overcome the limitations of excessive parameterization and computationally intensive operations in existing steel defect detection algorithms, an improved algorithm based on YOLOv8 for steel strip surface defect detection. Cao et al. [15] introduced the E-YOLOX algorithm for steel strip surface detection, which achieves real-time detection by employing a novel feature extraction network, ECMNet, and a new data augmentation method called Edge Cutout. Yang et al. [16] designed an efficient aggregation network

Manuscript received Mar 20, 2025; revised Jun 8, 2025.

The research work was supported by a scientific research project fund from the Liaoning Provincial Department of Education,and key project of Liaoning Provincial Department of Education (LJKZZ2022043).

Kaidi Liu is a Postgraduate of University of Science and Technology Liaoning, Anshan, Liaoning, China. (e-mail: liukaidi_1008@163.com).

Ji Zhao* is a Professor of University of Science and Technology Liaoning, Anshan, Liaoning, China. (corresponding author to provide phone: +086-139-9808-6167; e-mail: zhaoji_1974@126.com).

(ELAN-G), making the steel strip surface detection algorithm more deployable on terminal devices. However, these methods perform poorly when dealing with the detection of similar defects, indicating a need for targeted improvements.

In conclusion, while existing defect detection algorithms have achieved notable advancements, several critical challenges persist in this domain. Specifically, the substantial variation in target sizes leads to suboptimal detection performance for defects with diverse shapes. Additionally, the high similarity between abnormal and normal regions results in issues such as false positives and missed detections.

To address the aforementioned issues, this study proposes a CEC-YOLO method for steel surface defect detection. The main contributions of this study are as follows:

1) A Dynamic Ratio IoU (DRI) regression strategy is proposed, which adaptively adjusts the size of detection boxes based on the size of the target and accommodates different datasets, thereby achieving an IoU loss function more suitable for industrial detection tasks.

2) A Cross-Channel Group Interaction (CCGI) module is introduced to enhance the model's feature extraction and fusion capabilities, addressing the issue of low detection accuracy for defects with diverse shapes.

3) A Sparse Spatial Reduction Self-Attention (SSRA) module is proposed, enabling the model to focus on the most critical regions for defect detection, thereby reducing the occurrence of false positives and missed detections. Experimental results on the NEU-DET and GC10-DET datasets demonstrate that the improved model is well-suited for real-time and high-quality steel surface defect detection tasks.

II. Related Work

A. YOLOv8

YOLOv8's architectural design incorporates three principal components: a feature extraction backbone utilizing C2f modules, a multi-scale feature aggregation neck based on PAN-FAN, and a task-specific detection head. The C2f module serves as the core structural unit in the backbone, achieving parameter efficiency while maintaining robust feature representation capabilities through integrated residual connections and bottleneck operations. The detection head of YOLOv8 utilizes the Decoupled-Head concept, which separates the regression and classification branches. This design improves the efficiency of both training and inference. Furthermore, YOLOv8 abandons the traditional anchor-based approach and adopts an anchor-free strategy, simplifying the model architecture and accelerating the post-processing steps of non-maximum suppression (NMS).

B. Attention Mechanism

In defect detection tasks, relying solely on local features may sometimes be insufficient to capture critical defect regions. The self-attention mechanism addresses this by computing attention weights between different regions, enabling the model to focus on the most crucial areas for defect detection. This mechanism enhances the model's localization precision and accuracy in scenarios with complex backgrounds and multiple defects. In industrial defect detection, defects may exist across large areas of an image. Traditional convolutional neural networks are often limited by their local receptive fields, which can hinder their ability to fully capture global image information and long-range dependencies. The self-attention mechanism effectively models relationships between different regions of an image without introducing significant computational overhead or additional parameters. This capability allows the model to better understand the overall context, thereby improving the accuracy of defect detection.

Spatial reduction attention (SRA) [17] has been widely applied in previous work, effectively extracting global information by leveraging sparse token-region relationships. However, non-overlapping spatial reduction, used to reduce token counts, can disrupt the spatial structure near patch boundaries and degrade token quality. To address this issue, Overlapping Spatial-Reduction Attention (OSRA) [18] introduces overlapping spatial reduction (OSR) into SRA, which better represents the spatial structure near patch boundaries by using larger and overlapping patches. In practice, OSR is implemented as depthwise separable convolution. This paper further improves it into a sparse spatial-reduction attention module, enabling more focused attention on defect regions in detection tasks.

C. Intersection over Union (IoU)

This study conducted a statistical analysis of the USC pedestrian dataset, where the aspect ratios of detection boxes range from 0.26 to 0.48. In contrast, the aspect ratios of detection boxes in the NEU-DET and GC10-DET datasets range from 0.0027 to 0.952. Compared to datasets for pedestrian detection and face recognition in daily life, industrial detection datasets exhibit more complex distributions of detection box sizes, making the detection of small targets and similar defects more challenging. Traditional Intersection over Union (IoU) calculation methods may not effectively adapt to these variations, as they simply treat two bounding boxes as fixed sizes for overlap computation. InnerIoU [19] addresses this by dynamically adjusting the scale factor based on the dimensions of each pair of bounding boxes, enabling a more accurate evaluation of their overlap. However, the current implementation of InnerIoU uses a fixed scale factor for a given dataset, which limits its adaptability to different tasks and datasets. Therefore, we propose a more flexible Dynamic Ratio IoU regression strategy to better meet the demands of industrial detection tasks.

III. Method

A. CEC-YOLO Model Structure

Due to the influence of lighting, the grayscale values of defect images within the same category can vary significantly. Additionally, defects from different categories may exhibit similarities. These factors can prevent the model from focusing on critical defect regions, making it difficult for the network to extract meaningful features. The original YOLOv8 network does not fully utilize these features, and the distribution of detection box sizes in industrial detection datasets is more complex. Traditional IoU calculation methods may not effectively adapt to these

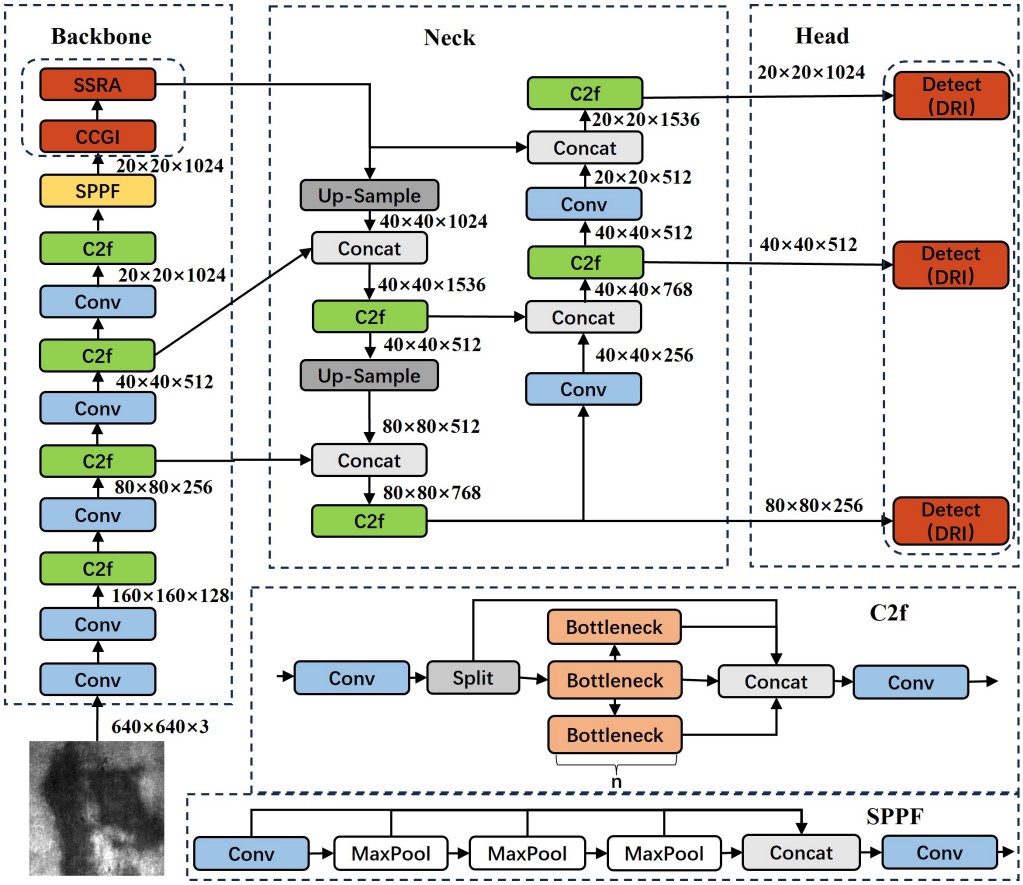


Fig. 1. Overall Structure of CEC-YOLO

variations, necessitating the introduction of a more flexible and dynamic IoU algorithm. To address these limitations of the original YOLOv8 network, we propose an improved CEC-YOLO network architecture, as shown in Fig.1.First, a CCGI module is added after the SPPF module to enhance the model's feature extraction capabilities. Second, an SSRA module is incorporated to enable the model to focus on critical defect regions. Finally, a DRI method is integrated into the detection head to better detect defects with varying characteristics, aiming to achieve a balanced improvement in both real-time performance and detection accuracy.

B. Cross Channel Group Interaction(CCGI)

Recent studies [20] have shown that introducing 3 × 3 depthwise convolutions [21] into vision transformers can effectively capture positional information from zero-padding. The cross-channel grouped interaction module enables the model to dynamically adjust its activation levels and outputs based on the input content. Specifically, it filters out unimportant information through a branch structure while enhancing information interaction across different groups and channels via grouped convolutions and channel shuffle operations, thereby improving the model's feature extraction capabilities. This mechanism endows the model with greater adaptability and flexibility, enabling it to handle complex input data more effectively.

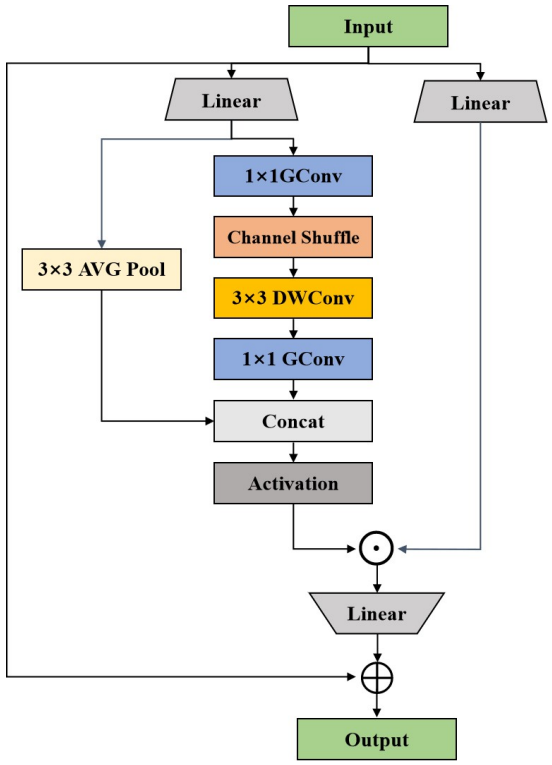


Fig. 2. CCGI moudule

The CCGI module is illustrated in Fig.2. The input of the network first passes through two parallel linear transformation layers to extract preliminary features. Subsequently, one branch of the data flow passes through a 3×3 average pooling layer (AVG Pool) to reduce the spatial dimensions of the feature maps while preserving critical discriminative characteristics. The other branch of the data flow passes through a 1×1 group convolution layer (GConv), which groups the input feature maps and performs convolution operations on each group separately, thereby

reducing the number of parameters and computational cost. Specifically, after dividing into G groups, the number of parameters in this layer is reduced to 1/G of the original. By adjusting the number of groups, an optimal performance balance can be achieved for different tasks and datasets. The two branches of data flow merge after passing through a channel shuffle layer [22]. The channel shuffle operation rearranges the feature matrix, enabling better information exchange between different groups. The network captures positional information through a 3×3 depthwise separable convolution, followed by a 1×1 group convolution for deeper feature extraction and integration. This design retains the advantage of reduced computational cost from depthwise separable convolution while enhancing the model's feature extraction capability through channel shuffling. The output is concatenated with a 3×3 average pooling layer to prevent overfitting.

C. Sparse Spatial Reduction Attention(SSRA)

To address the issues of false detection and missed detection induced by the high similarity between abnormal and normal regions, we propose a sparse spatial reduction self-attention module that enables the model to focus on critical defect regions. The principle involves exchanging the output channel features of intermediate layers with retained features, fully utilizing the information from intermediate layers to avoid information loss. Additionally, the module incorporates the idea of TripletAttention [23] to interact across all dimensions of the input tensor with negligible computational overhead.This helps the model better capture both global and local features, thereby improving the accuracy of classification and detection. Furthermore, a probabilistic sparse self-attention mechanism and a multi-head self-attention spatial reduction mechanism are introduced to enhance the computational efficiency and performance of the model.

1) Probabilistic Sparse Self-Attention Mechanism

The probabilistic sparse self-attention mechanism [24] selects the most important subset of attention weights for computation using a probabilistic approach, while disregarding weights that have minimal impact on the results. Specifically, it measures the relative importance using Kullback-Leibler (KL) divergence, and the metric is defined by the following equation:

$$M(q_i, K) = \ln \sum_{j=1}^{L_K} e^{\frac{q_i k_j^T}{\sqrt{d}}} - \frac{1}{L_K} \sum_{j=1}^{L_K} \frac{q_i k_j^T}{\sqrt{d}} \tag{1}$$

The first part is the Log-Sum-Exp (LSE) operation, which approximates the maximum value while maintaining smoothness. The second part is their arithmetic mean, used to measure the overall relevance between the query and the keys. Here, q_i represents the i-th value of the query vector, and k_j represents the j-th value of the key vector.

By incorporating the Kullback-Leibler (KL) divergence constraint into the Softmax, the attention weight distribution is guided to approximate the target sparse distribution.

$$P_{\text{sparse}} = \text{Softmax}\left(\frac{QK^T}{\sqrt{d}} - \lambda M(q_j, K)\right) \tag{2}$$

Based on the above formula, the formula for the probabilistic sparse self attention mechanism can be obtained:

$$\overline{A}(Q, K, V) = \text{Softmax}\left(\frac{\overline{Q}K^T}{c}\right) V \tag{3}$$

Here, $A(Q, K, V)$ represents the probabilistic sparse self-attention score matrix, where Q is the set of query elements corresponding to the top u largest $M(q_i, K)$ values. Here, $c = \ln L_Q$, where c is the sampling factor, and L is the length of the input sequence. Notably, to avoid the issue of excessively small gradients, a spatial reduction operation is applied to the key-value pairs using a scaling factor of $1/\sqrt{d}$. The computational complexity of the attention mechanism is optimized from the original $O(L^2)$ to $O(\ln L)$.

2) Multi-head Self-attention Spatial Reduction Mechanism

Through parallelized processing of multiple attention heads, the self-attention mechanism effectively learns heterogeneous feature interactions across different input subspaces. When combined with the probabilistic sparse self-attention mechanism, each attention head can generate distinct sparse key-value pairs, empowering the model to adaptively attend to spatially distributed information. This enhances the model's representational capacity and generalization ability. Assuming there are h attention heads, each head has independent linear transformations for queries, keys, and values. For the i-th head, its output h_i can be expressed as:

$$h_i = f\left(W_i^{(q)} q, W_i^{(k)} k, W_i^{(v)} v\right) \tag{4}$$

Here, $W_i^{(q)}$, $W_i^{(k)}$, and $W_i^{(v)}$ are the trainable weight matrices for the query, key, and value of the i-th head.

The outputs of all heads are concatenated and then mapped to the output dimension through an additional trainable weight matrix:

$$\text{Head}(Q, K, V) = \text{Concat}(h_1, h_2, \ldots, h_i) W_O \tag{5}$$

Here, $h_1, h_2, \cdots, h_i$ are the outputs of each head, and W_O is a trainable weight matrix.

As shown in Fig.3, the sparse spatial reduction self-attention module begins by passing the input through a series of 3x3 depthwise separable convolution (DWConv) layers. These layers effectively extract spatial features from the input data while reducing computational costs. Subsequently, a 1x1 pointwise convolution layer is used to further refine the feature extraction process, enhancing the model's capability to capture local features. The extracted features are then concatenated and subjected to a channel shuffle operation to increase feature diversity and expressive power. Next, the network integrates features through a triplet structure, which combines features and generates higher-level abstract representations. A linear layer is then applied to transform and adjust the feature dimensions, preparing them for subsequent processing steps. In another branch of feature processing, the model employs a spatial reduction operation, which downsamples and pools the feature maps to reduce data dimensionality and extract more abstract features. After this, a sparse operation is applied to further filter important features and remove redundant information. Finally, the

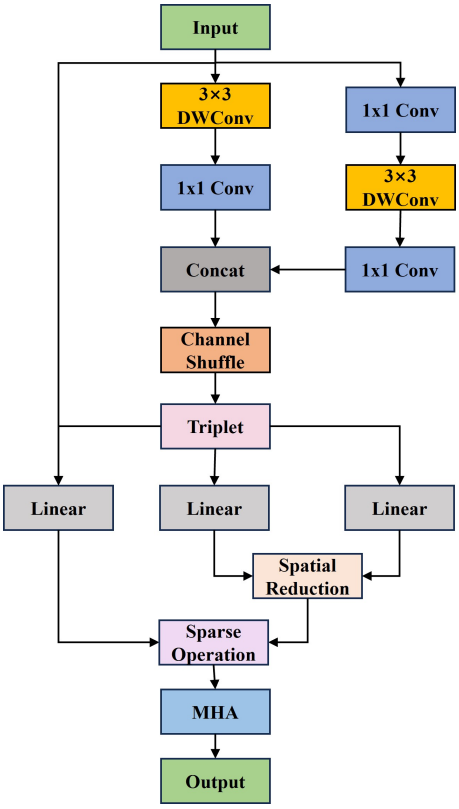


Fig. 3. SSRA moudule

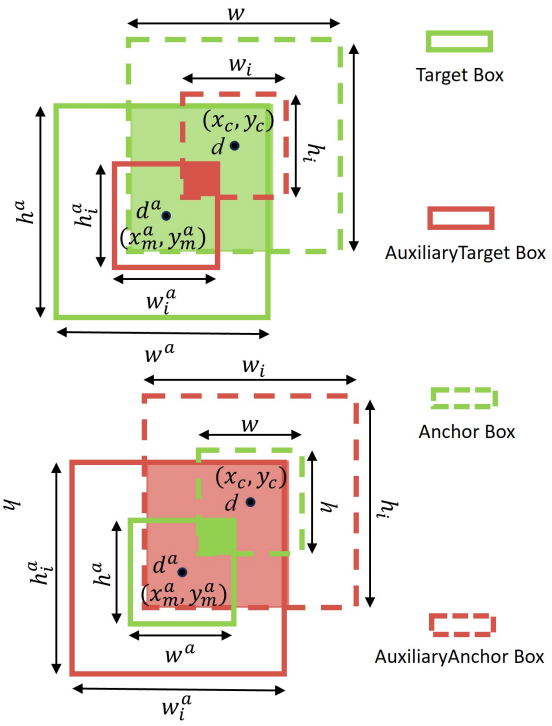


Fig. 4. Dynamic Ratio IoU

model utilizes a multi-head self-attention mechanism (MHA) to establish connections between different feature subspaces, capturing long-range dependencies and enhancing the model's ability to understand complex patterns. Through this series of feature fusion and information processing steps, the model can efficiently focus on critical defect regions.

D. Dynamic Ratio IoU(DRI)

In object detection, Intersection over Union (IoU) is used to evaluate the intersection between two bounding boxes. Samples with high IoU are easier for the model to learn, while samples with low IoU typically contain more challenging bounding boxes, such as those with partial overlap or imperfect matches. Therefore, by dynamically adjusting weights, the model focus more on these challenging instances. As shown in Fig.4, the Dynamic Ratio IoU regression strategy calculates IoU loss using auxiliary bounding boxes. High-IoU samples utilize compact auxiliary regions for loss computation to expedite convergence, whereas expanded auxiliary regions are allocated to low-IoU samples to enhance learning efficacy. For different datasets and detectors, we introduce a dynamic weight adjustment strategy. The core logic is to dynamically compute the weight of each sample based on the difference between the historical IoU average and the current IoU, thereby adapting to changes in bounding box matching in object detection tasks. This approach aims to balance the differences between samples, emphasize the contribution of challenging samples, and prevent underfitting caused by over-learning of simple samples during training.

The calculation of dynamic weights primarily relies on the difference between the current IoU and the historical average IoU, as detailed below:

1) Auxiliary Prediction Box Calculation

$$d_l^a = x_m^a - \frac{w^a \cdot p}{2}, \quad d_r^a = x_m^{st} + \frac{w^{st} \cdot p}{2} \tag{6}$$

$$d_t^a = y_m^a - \frac{h^a \cdot p}{2}, \quad d_b^a = y_m^a + \frac{h^a \cdot p}{2} \tag{7}$$

$$d_l = x_m - \frac{w \cdot p}{2}, \quad d_r = x_m + \frac{w \cdot p}{2} \tag{8}$$

$$d_t = y_m - \frac{h \cdot p}{2}, \quad d_b = y_m + \frac{h \cdot p}{2} \tag{9}$$

Here, x_m^a and y_m^a are the center of the annotated box, x_m and y_m are the center of the predicted box, d_l^a, d_r^a, d_t^a, and d_b^a are the left, right, top, and bottom of the auxiliary annotated box in the red region, d_l, d_r, d_t, and d_b are the left, right, top, and bottom of the auxiliary predicted box in the green region, and p is an adjustable ratio factor ranging from 0.5 to 1.5.

2) Dynamic IoU value calculation

$$\begin{aligned}\text{inter} = &\left(\min\left(d_i^a, d_r\right) - \max\left(d_i^a, d_l\right)\right) \\ &* \left(\min\left(d_b^a, d_b\right) - \max\left(d_l^a, d_l\right)\right)\end{aligned} \tag{10}$$

$$\text{union} = \left(w^a \cdot h^a\right) \cdot p^2 + (w \cdot h) \cdot p^2 - \text{inter} \tag{11}$$

$$d_{\text{center}} = \sqrt{\left(x_m^a - x_m\right)^2 + \left(y_m^a - y_m\right)^2} \tag{12}$$

$$\text{IoU} = \frac{\text{inter} + \epsilon}{\text{union} + \epsilon} e^{-\lambda d_{\text{center}}} \tag{13}$$

Here, ϵ is a small constant used to prevent the IoU from being zero. λ is a hyperparameter used to adjust the weight of the center point distance. d_{center} represents the Euclidean distance between the centroids of the predicted bounding box and its corresponding ground truth annotation.

3) Updating the Historical IoU Mean

The historical IoU mean is calculated using exponential moving average:

$$\mathrm{IoU}_{\mathrm{mean}} = (1-m)\cdot \mathrm{IoU}_{\mathrm{mean}} + m \cdot \mathrm{mean}\left(\mathrm{IoU}_{\mathrm{current}}\right) \quad (14)$$

Here, $\mathrm{IoU}_{\mathrm{mean}}$ is the historical mean of IoU, m is the momentum coefficient that determines the update speed of the historical mean, with $0 < m < 1$, and $\mathrm{mean}\left(\mathrm{IoU}_{\mathrm{current}}\right)$ is the mean IoU of the current batch of samples.

4) Weight Calculation

The weights are adjusted based on the difference between the current IoU and the historical mean IoU:

$$w_i = \begin{cases} 1, & \text{if } \mathrm{IoU}_i > \mathrm{IoU}_{\mathrm{mean}} \\ \exp\left(-\left(\mathrm{IoU}_{\mathrm{mean}} - \mathrm{IoU}_i\right)\right), & \text{if } \mathrm{IoU}_i \leq \mathrm{IoU}_{\mathrm{mean}} \end{cases} \quad (15)$$

Here, IoU_i is the IoU value of sample i, and $\exp\left(-\left(\mathrm{IoU}_{\mathrm{mean}} - \mathrm{IoU}_i\right)\right)$ is the dynamic weight used to slow down the weight decay for low IoU samples, allowing challenging samples to receive higher weights. The IoU obtained using Dynamic Ratio IoU results in a loss function more suitable for industrial detection tasks. Additionally, the improved Dynamic Ratio IoU can automatically adjust the detection box size based on different datasets, demonstrating the generalization capability of the enhanced model.

IV. Experimental Setting

A. Datasets and evaluation indicators

1) GC10-DET Dataset

The GC10-DET dataset includes 2,257 high-definition (2048×1000) steel surface images representing ten industrial defect categories, organized as: Welding Line (WI), Water Spot (Ws), Crease (Cr), Crescent Gap (Cg), Oil Spot (Os), Inclusion (In), Waist Crease (WI), Punching (Pu), Silk Spot (Ss) and Rolled Pit (Rp).

2) NEU-DET Dataset

The NEU-DET dataset includes 1,800 steel surface images(200×200). It contains six types of defects: patches, pitted surface, inclusion, crazing, scratches and rolled-in scale.

The datasets follows an 8:2 training-testing split protocol for machine learning applications. This paper uses AP, mAP, FPS, Precision and Recall to evaluate the strip defect detection model. The formulas are as follows:

$$AP = \int_0^1 P(R)\, dR \quad (16)$$

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

FPS represents the frame rate, reflecting the model's inference speed. The formula is as follows:

$$FPS = \frac{\mathrm{Framenum}}{\mathrm{ElapsedTime}} \quad (18)$$

Here, Framenum represent the cardinality of processed image samples, and ElapsedTime corresponds to the temporal duration required for complete inference execution.

$$\mathrm{Precision} = \frac{TP}{TP+FP} \quad (19)$$

$$\mathrm{Recall} = \frac{TP}{TP+FN} \quad (20)$$

Here, TP represent correctly identified defect instances, FP indicate misclassified non-defective regions, and FN correspond to undetected actual defects in the evaluation set.

B. Experimental Environment and Setup

The proposed model was implemented and evaluated using the PyTorch deep learning framework. The experimental configuration comprised a CPU: Intel Xeon E5, GPUs: NVIDIA GTX TITAN XP * 2, and an SGD optimizer for model optimization. The model was trained with a batch size of 8 using 640 × 640 pixel images over 300 epochs in this investigation. The training pipeline incorporates multiple augmentation operations including random spatial transformations and pixel-level intensity modifications, supplemented by multi-scale processing.

C. Experimental Verification and Analysis

To systematically evaluate the enhanced model's performance, comprehensive experiments were performed using the NEU-DET dataset. The experimental results are presented in Table I, while a detailed comparison with algorithms from the YOLO series is shown in Table II. CEC-YOLO demonstrates significant superiority over other methods in terms of both detection accuracy and efficiency. Specifically, CEC-YOLO achieves a mAP of 78.8%, which is notably higher than YOLOv8n (77.0%). In defect category detection, CEC-YOLO exhibits particularly outstanding performance for cracks (Cr) and scratches (Sc), achieving detection rates of 45.8% and 96.7% respectively, significantly surpassing other algorithms. Furthermore, CEC-YOLO has only 3.36 million parameters, substantially fewer than the 12.1M of YOLOv3-tiny. It achieves 72 frames per second (FPS), second only to YOLOv8n's 75 FPS, demonstrating its advantages in both lightweight design and real-time performance.

TABLE I
Performance Metrics on NEU-DET Dataset

Metrics	all	Cr	In	Ps	Pa	Rs	Sc
Precision/%	79.4	64.5	81.6	86.3	84.1	69.4	90.2
Recall/%	71.2	32.9	79.8	84.9	76.2	61.4	92.1
mAP@0.5/%	78.8	45.8	85.2	92.1	87.1	65.9	96.7

To assess cross-dataset generalization performance, additional evaluations were performed on the GC10-DET dataset. The experimental results are presented in Table III, while a detailed comparison with algorithms from the YOLO series is provided in Table IV. CEC-YOLO achieves a mAP of 85.5%, representing a 3.5% improvement over the baseline model YOLOv8n and a 15.4% enhancement compared to YOLOv3-tiny.Notably, these performance improvements are achieved while maintaining excellent computational efficiency. Compared to YOLOv3-tiny, CEC-YOLO reduces the number of parameters by 72.2%, and achieves a 63.1% reduction compared to YOLOv5s, while maintaining

TABLE II
COMPARISON OF DETECTION RESULTS OF NEU-DET DATASET

Model Name	Params	FPS	mAP/%	Cr	In	Ps	Pa	Rs	Sc
YOLOv3-tiny	12.1M	50	69.7	35.6	76.2	81.9	78.2	60.2	86.1
YOLOv5n	2.39M	36	74.0	37.1	84.2	89.0	82.6	61.8	89.4
YOLOv5s	9.1M	48	76.1	39.2	86.3	91.1	84.7	63.9	91.5
YOLOv6n	4.04M	45	74.9	38.5	81.7	90.3	83.4	62.9	92.5
YOLOX-Tiny	5.04M	35	74.5	38.9	82.4	89.1	85.6	60.2	90.9
YOLOv7-tiny	6.2M	62	74.7	39.3	83.4	88.6	83.3	62.3	91.2
YOLOv8n	2.87M	75	77.0	39.8	84.1	90.3	86.3	66.6	94.9
CEC-YOLO	3.36M	72	78.8	45.8	85.2	92.1	87.1	65.9	96.7

TABLE III
PERFORMANCE METRICS ON GC10-DET DATASET

Metrics	all	Pu	WI	Cg	Ws	Os	Ss	In	Rp.	Cr	Wf
Precision/%	0.886	0.957	0.859	0.842	0.958	0.808	0.862	0.904	0.868	0.852	0.956
Recall/%	0.760	0.929	0.968	0.922	0.832	0.770	0.740	0.444	0.809	0.564	0.781
mAP@0.5/%	0.855	0.959	0.923	0.943	0.885	0.832	0.848	0.618	0.773	0.842	0.930

TABLE IV
COMPARISON OF DETECTION RESULTS OF GC10-DET DATASET

Model Name	Params	FPS	mAP/%	mAP/%									
				Pu	WI	Cg	Ws	Os	Ss	In	Rp	Cr	Wf
YOLOv3-tiny	12.1M	51	70.1	86.0	79.2	78.8	73.2	66.0	67.2	49.0	61.3	66.8	73.7
YOLOv5n	2.39M	33	78.1	93.7	88.6	92.7	85.1	77.7	82.7	43.6	65.6	68.1	82.9
YOLOv5s	9.1M	49	80.1	92.6	91.3	91.6	86.8	81.1	83.2	46.0	68.7	71.2	88.4
YOLOv6n	4.04M	46	79.2	91.6	91.3	90.6	87.8	80.1	81.9	45.3	67.4	69.1	86.5
YOLOX-Tiny	5.04M	34	79.6	92.4	90.1	91.4	85.6	80.9	82.2	50.0	67.7	70.2	85.4
YOLOv7-tiny	6.2M	60	80.2	95.2	90.9	93.1	85.6	76.9	83.8	49.1	67.8	73.6	85.8
YOLOv8n	2.87M	71	82.0	95.6	92.0	92.8	89.2	82.5	83.8	54.9	70.2	68.7	90.3
CEC-YOLO	3.36M	69	85.5	95.9	92.3	94.3	88.5	83.2	84.8	61.8	77.3	84.2	93.0

comparable inference speed to baseline models with improved accuracy.

To validate the superiority of the proposed model, we conducted comparative experiments with several advanced models from recent years, with the results presented in Table V. Experimental results demonstrate that CEC-YOLO achieves outstanding performance on both NEU-DET and GC10-DET benchmark datasets for industrial defect detection. In terms of model complexity, CEC-YOLO contains only 36.1% of the parameters of GDM-YOLO while achieving superior detection accuracy (0.8% improvement on GC10-DET). Compared with the lightweight HA-YOLO model, CEC-YOLO improves mAP by 3.7% (NEU-DET) and 4.0% (GC10-DET) while introducing only an additional 0.85M parameters. Notably, CEC-YOLO achieves an 11.1% improvement in inference speed compared with Literature[28], while simultaneously increasing detection accuracy by 1.8% (GC10-DET), demonstrating balanced enhancements in both speed and precision. Furthermore, when compared with the two-stage object detection algorithm Faster R-CNN, CEC-YOLO achieves an 8.7% higher mAP (NEU-DET) while reducing the parameter count to one-tenth of the Faster R-CNN. Overall, CEC-YOLO demonstrates comprehensively superior performance in detection accuracy, category-specific performance, and computational efficiency.

TABLE V
COMPARISON WITH DIFFERENT METHODS

Model	Params	FPS	mAP%	
			NEU-DET	GC10-DET
Faster RCNN	41.25M	11	70.8	76.9
ETDNet[14]	6.99M	50	77.5	82.4
Literature[25]	9.1M	24	76.6	81.8
Literature[26]	25.9M	63	77.6	83.7
Literature[27]	5.06M	38	76.1	81.0
HA-YOLO[28]	2.51M	65	75.1	81.5
YOLOv5-CGC[29]	7M	38	77.8	83.6
GDM-YOLO[30]	9.3M	52	78.3	84.7
CEC-YOLO	3.36M	70	78.8	85.5

Fig.5 and Fig.6 comparatively present the precision-recall characteristics of both baseline and enhanced models across the NEU-DET and GC10-DET datasets, respectively. It can be observed that the accuracy for all defects except Rs improved on the NEU-DET dataset. On the GC10-DET

dataset, the accuracy for all defects except Ws improved. It performs particularly well in challenging defect categories such as In, Rp and Cr, which achieving mAP increases of 6.9%, 7.1%, and 15.5% compared with the baseline model YOLOv8. It demonstrates that the improved model has significantly improved its ability to detect defects of different shapes and similar defects.

D. Comparison of different IoU loss functions

To validate the effectiveness of Dynamic Ratio IoU, we conducted comparative experiments with other IoU loss functions. GIoU is insensitive to changes in the aspect ratio of bounding boxes. When objects are highly overlapping, GIoU degenerates into IoU, resulting in reduced optimization efficiency. Although CIoU introduces penalties for center point distance and aspect ratio, thereby improving performance, it exhibits weak adaptability to extreme scale variations (small objects). EIoU decouples the width and height penalties based on WIoU, but fails to dynamically adjust the optimization target ratio. MPDIoU optimizes based on vertex distance, but provides weak constraints on the overall shape of bounding boxes. DRI can dynamically adjust the scale factor according to the size of each bounding box pair, thereby more accurately evaluating their overlap degree. Moreover, DRI can adaptively adjust the ratio based on the selected dataset. As shown in Table VI, the Dynamic Ratio IoU achieved optimal mAP and F1-score performance on both NEU-DET and GC10-DET datasets. DRI demonstrated improvements of 1.5% in mAP and 1.7% in F1-score compared to GIoU on the NEU-DET dataset. For the GC10-DET dataset, the maximum improvements reached 1.3% in mAP and 2.2% in F1-score.

TABLE VI
COMPARISON OF IOU LOSS FUNCTIONS

Loss Function	NEU-DET		GC10-DET	
	mAP%	F1%	mAP%	F1%
GIoU	76.3	72.8	81.9	79.1
CIoU	76.8	73.0	82.2	79.6
WIoU[31]	77.1	73.5	82.7	80.0
EIoU[32]	76.7	73.3	82.5	80.2
MPDIoU[33]	77.2	73.8	82.8	80.5
Dynamic Ratio IoU	77.8	74.5	83.2	81.3

E. Analysis of the Effectiveness of Attention Mechanism

To verify that the sparse spatial reduction self-attention module can focus on key defect regions, a comparative experiment was conducted under controlled conditions with different attention mechanisms, and the results were visualized for defects of various sizes. The visualization results are shown in Fig.7. It clearly indicate that SE attention missed small-sized defects, while PSA and ECA attention did not sufficiently focus on the defects. CBAM attention focused on non-defect regions. For large-sized defects, other attention mechanisms failed to fully focus on the defect regions, whereas the sparse spatial reduction self-attention performed well across defects of all sizes. The detection results of different attention mechanisms are shown in Table VII. The sparse spatial reduction self-attention mechanism outperformed the SE attention by 1.5% in mAP and 1% in F1-score on the NEU-DET dataset. On the GC10-DET dataset, SSRA achieved greater improvements of 3.1% in mAP and 1.7% in F1-score.

TABLE VII
EVALUATION OF DIFFERENT ATTENTION STRATEGIES

Loss Function	NEU-DET		GC10-DET	
	mAP%	F1%	mAP%	F1%
CBAM[34]	77.0	73.1	81.4	79.8
SE	76.5	72.9	81.2	79.5
PSA[35]	77.1	73.4	82.8	80.1
ECA[36]	77.2	73.6	82.3	80.7
Ours	78.0	74.6	84.3	81.2

F. Ablation Experiment

Under controlled conditions, the impact of each enhancement and the combination of various improvement strategies on the enhanced model were tested. The ablation experiment results are shown in Table VIII.

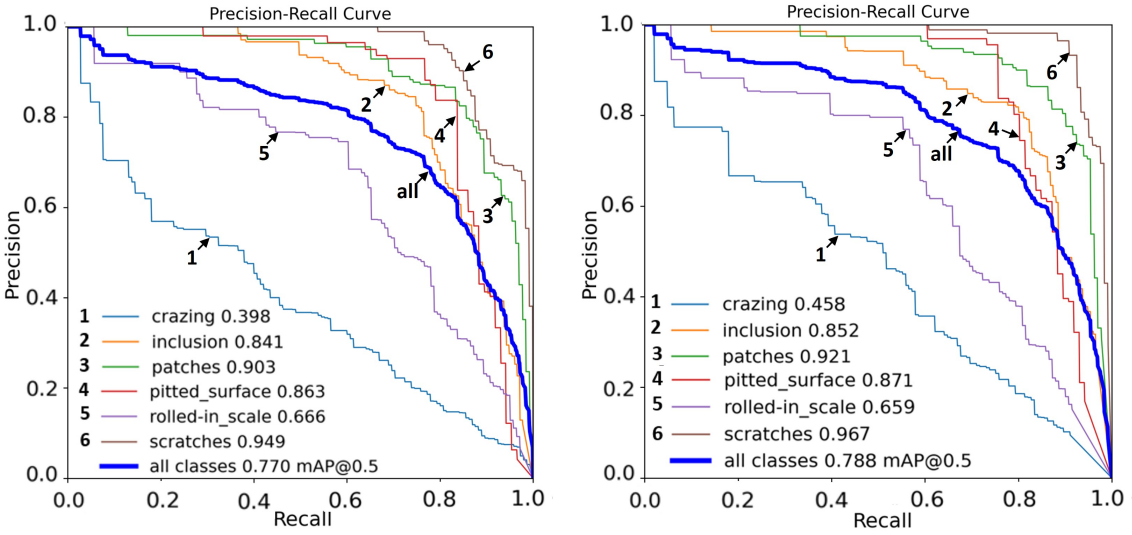
TABLE VIII
ABLATION RESULTS

Experiment	DRI	CCGI	SSRA	mAP%	
				NEU-DET	GC10-DET
YOLOv8				77.0	82.0
Experiment 1	✓			77.8	83.2
Experiment 2		✓		78.1	83.0
Experiment 3			✓	78.0	84.3
Experiment 4	✓	✓		78.1	83.8
Experiment 5	✓		✓	78.1	85.0
Experiment 6		✓	✓	78.3	84.8
Experiment 7	✓	✓	✓	78.8	85.5

By comparing the YOLOv8 baseline model with Experiment 7, quantitative analysis indicates that the enhanced model achieved a significant enhancement in detection accuracy, with improvements of 1.8% and 3.5% on the NEU-DET and GC10-DET datasets, respectively. From Experiment 1, it can be seen that DRI enables flexible and dynamic adjustment of detection boxes across different datasets, partially addressing the issue of excessive variation in defect sizes. Experiments 2 and 3 demonstrate that the improved model has enhanced feature extraction capabilities and is more likely to focus on defect regions, thereby mitigating issues of missed and false detections to some extent. Experiments 4, 5, and 6 show that different modules can be effectively combined, all contributing to the performance improvement of the model.

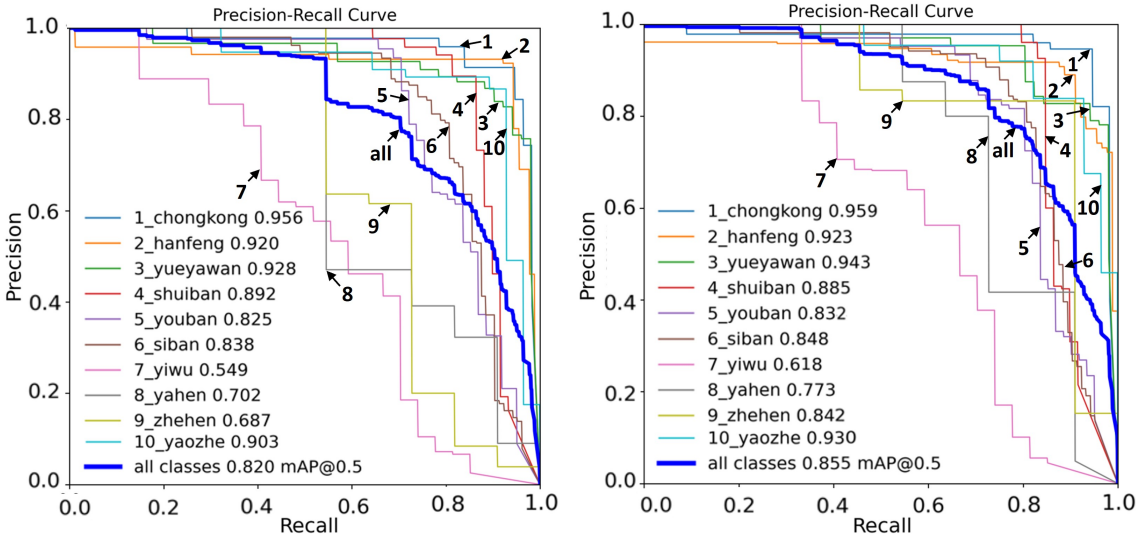
G. Visualization of Test Results

To make the empirical findings more intuitive, we visualized the detection results under different methods. The visualization of the detection effects is shown in Fig.8. (a)BBox displays the original images with bounding boxes, where the first three rows present visualization results



(a) P-R curves of YOLOv8 (b) P-R curves of CEC-YOLO

Fig. 5. PR curves: Baseline vs. Enhanced on NEU-DET dataset



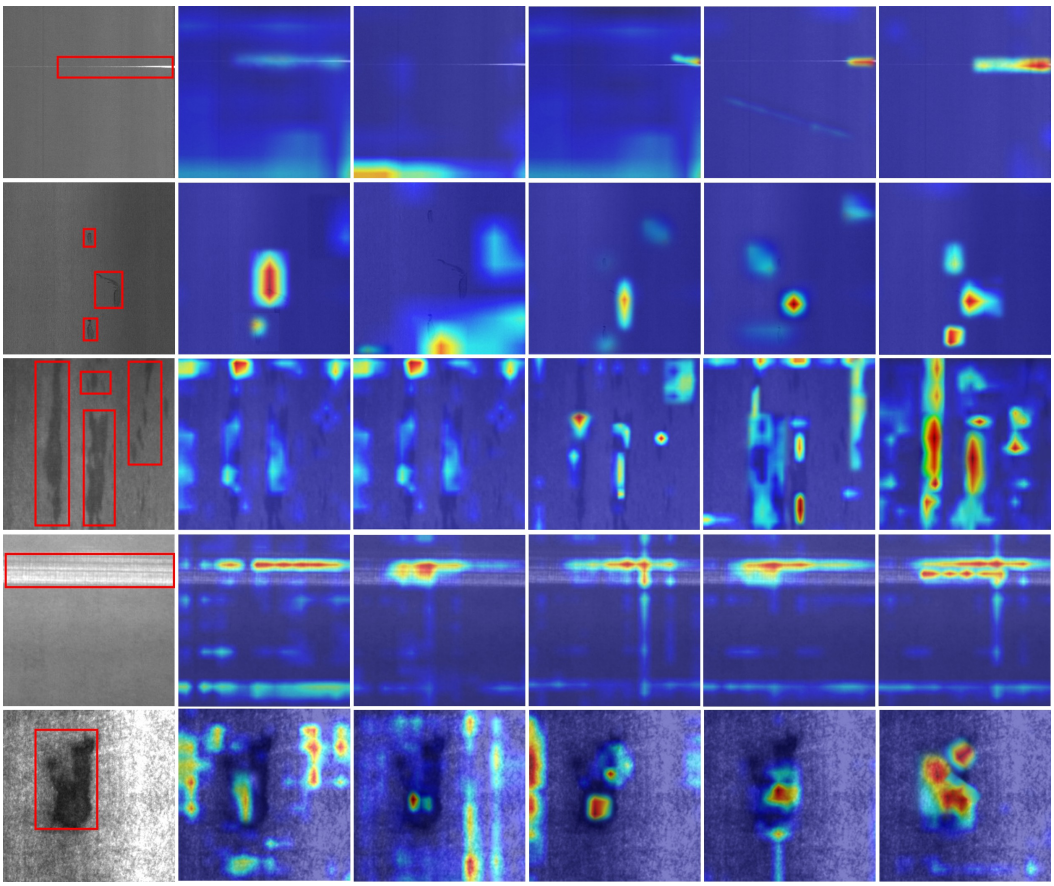
(a) P-R curves of YOLOv8 (b) P-R curves of CEC-YOLO

Fig. 6. PR curves: Baseline vs. Enhanced on GC10-DET dataset

from the GC10-DET dataset and the last two rows show detection outcomes from the NEU-DET dataset.As shown, YOLOv5, YOLOv6, and YOLOv7 failed to effectively detect defects, while the original YOLOv8 misclassified Os defects as Rp defects and exhibited some false negatives. Our improved model accurately detected both defect locations and categories, demonstrating enhanced feature extraction capability. Its superior performance across different datasets further indicates improved generalization ability and robustness. Additionally, we visualized the attention regions of the model's channels during the detection process. As shown in Fig.9, the improved model can better focus on the regions where defects exist, thereby enhancing the accuracy and efficiency of the defect detection task.

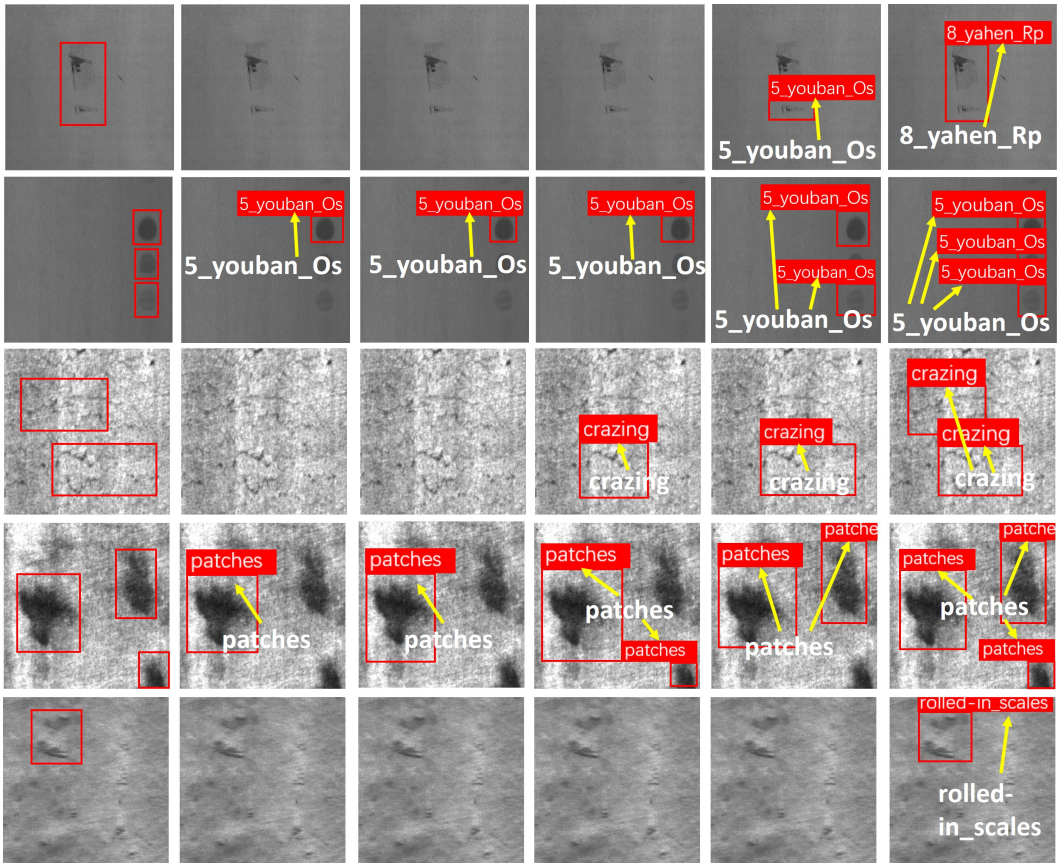
V. Conclusions

In this paper, we propose an improved steel surface defect detection model, CEC-YOLO, based on the YOLOv8 model, with enhancements made to the backbone network and the IoU component. Firstly, to enhance feature extraction capability and expand the receptive field, we introduce an improved cross-channel grouped interaction module. Secondly, we propose an improved sparse spatial reduction self-attention module, enabling the model to focus on the most critical regions for defect detection. Finally, we introduce an improved Dynamic Ratio IoU regression strategy, which adaptively adjusts the size of detection boxes based on the target size and adapts to different datasets, thereby achieving an IoU loss function more suitable for industrial detection tasks. Comparative experiments were



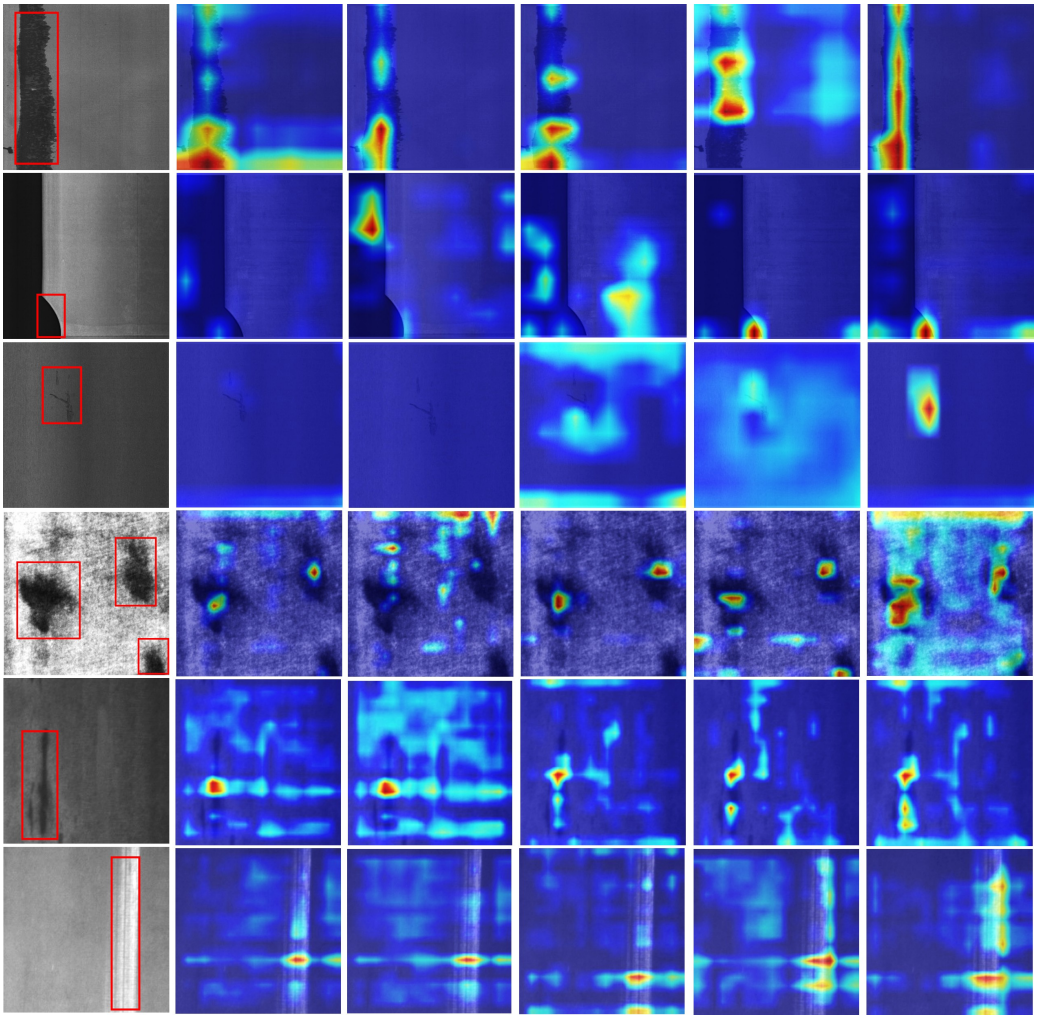
(a) BBox (b) CBAM (c) SE (d) PSA (e) ECA (f) DRI

Fig. 7. Heat maps of different sizes



(a) BBox (b) YOLOv5n (c) YOLOv6n (d) YOLOv7-tiny (e) YOLOv8n (f) CEC-YOLO

Fig. 8. Visualization of detection results



(a) BBox (b) YOLOv5n (c) YOLOv6n (d) YOLOv7-tiny(e) YOLOv8n (f) CEC-YOLO

Fig. 9. Visualization of channel focus

conducted on different attention mechanisms and IoU loss functions on the different datasets to verify the effectiveness and robustness of the added module. The enhanced model demonstrates superior performance with mAP scores of 85.5% (GC10-DET) and 78.8% (NEU-DET), achieving absolute gains of 3.5% and 1.8% respectively compared to baseline methods. The experimental results substantiate the robust performance of the proposed model across diverse steel surface defect detection scenarios, demonstrating its competitive advantage over existing approaches. Finally, the detection outputs and corresponding channel attention maps were visualized to facilitate model interpretation. In summary, the proposed model meets the requirements of high detection accuracy and real-time performance. However, the performance of the model for some subtle small defects still requires further improvement. In future research, we plan to refine the network architecture, fine-tune multiple pre-trained models, and explore multi-model fusion.

REFERENCES

[1] S. Kim, W. Kim, Y.-K. Noh, and F. C. Park, "Transfer learning for automated optical inspection," 2017 International Joint Conference on Neural Networks (IJCNN). IEEE, 2017, pp. 2517–2524.

[2] C. C. Yeung and K. M. Lam, "Efficient fused-attention model for steel surface defect detection," IEEE Transactions on Instrumentation and Measurement, vol. 71, pp. 1–11, 2022.

[3] W. Zhao, F. Chen, H. Huang *et al.*, "A new steel defect detection algorithm based on deep learning," Computational Intelligence and Neuroscience, vol. 2021, no. 1, p. 5592878, 2021.

[4] S. A. Singh and K. A. Desai, "Automated surface defect detection framework using machine vision and convolutional neural networks," Journal of Intelligent Manufacturing, vol. 34, no. 4, pp. 1995–2011, 2023.

[5] R. Girshick, J. Donahue, T. Darrell *et al.*, "Rich feature hierarchies for accurate object detection and semantic segmentation," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. IEEE, 2014, pp. 580–587.

[6] R. Girshick, "Fast r-cnn," arXiv:1504.08083, 2015.

[7] S. Ren, "Faster r-cnn: Towards real-time object detection with region proposal networks," arXiv:1506.01497, 2015.

[8] W. Liu, D. Anguelov, D. Erhan *et al.*, "Ssd: Single shot multibox detector," Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14. Springer International Publishing, 2016, pp. 21–37.

[9] J. Redmon, "You only look once: Unified, real-time object detection," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. IEEE, 2016.

[10] C. Y. Wang, A. Bochkovskiy, and H. Y. M. Liao, "Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. IEEE, 2023, pp. 7464–7475.

[11] D. Reis, J. Kupec, J. Hong, and A. Daoudi, "Real-time flying object detection with yolov8," arXiv:2305.09972, 2023.

[12] Z. Cai and N. Vasconcelos, "Cascade r-cnn: High quality object detection and instance segmentation," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 43, no. 5, pp. 1483–1498, 2019.

[13] Z. Dou, H. Gao, G. Liu, and B. Chang, "Small sample steel plate defect detection algorithm of lightweight yolov8," Computer Engineering and Applications, vol. 60, no. 9, pp. 90–100, 2024.

[14] H. Zhou, R. Yang, R. Hu *et al.*, "Etdnet: efficient transformer-based

detection network for surface defect detection," IEEE Transactions on Instrumentation and Measurement, 2023.
[15] Y. Cao, Y. Zhou, and L. Xu, "A real-time metallic surface defect detection algorithm based on e-yolox," Journal of Graphics, vol. 44, no. 4, pp. 677–690, 2023.
[16] L. Yang, M. Li, J. Hu, and D. Wang, "Strip steel surface defect detection algorithm based on improved yolov7-tiny," Computer Engineering, vol. 51, no. 1, pp. 208–215, 2025.
[17] W. Wang, E. Xie, X. Li *et al.*, "Pyramid vision transformer: A versatile backbone for dense prediction without convolutions," Proceedings of the IEEE/CVF International Conference on Computer Vision. IEEE, 2021, pp. 568–578.
[18] M. Lou, H. Y. Zhou, S. Yang *et al.*, "Transxnet: learning both global and local dynamics with a dual dynamic token mixer for visual recognition," arXiv:2310.19380, 2023.
[19] D. Misra, T. Nalamada, A. U. Arasanipalai *et al.*, "Rotate to attend: Convolutional triplet attention module," Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. IEEE, 2021, pp. 3139–3148.
[20] M. A. Islam, S. Jia, and N. D. B. Bruce, "How much position information do convolutional neural networks encode?" arXiv:2001.08248, 2020.
[21] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. IEEE, 2017, pp. 1251–1258.
[22] X. Zhang, X. Zhou, M. Lin *et al.*, "Shufflenet: An extremely efficient convolutional neural network for mobile devices," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. IEEE, 2018, pp. 6848–6856.
[23] H. Zhang, C. Xu, and S. Zhang, "Inner-iou: more effective intersection over union loss with auxiliary bounding box," arXiv:2311.02877, 2023.
[24] H. Y. Zhou, S. H. Zhang, J. Q. Peng *et al.*, "Informer: beyond efficient transformer for long sequence time-series forecasting," Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, no. 12, pp. 11 106–11 115, 2021.
[25] Y. Su, Y. Deng, N. Zhou *et al.*, "Steel surface defect detection algorithm based on improved yolov5s," 2024 IEEE 6th Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC), vol. 6. IEEE, 2024, pp. 865–869.
[26] F. Miao, Y. Huang, W. Qian *et al.*, "Intelligent surface defect detection of hot rolled strip steel using yolov8 framework," 2023 IEEE 23rd International Conference on Communication Technology (ICCT). IEEE, 2023, pp. 678–682.
[27] C. Li, A. Xu, Y. Cai *et al.*, "Improved yolox-based method for steel surface defect detection," 2023 5th International Academic Exchange Conference on Science and Technology Innovation (IAECST). IEEE, 2023, pp. 929–934.
[28] Y. Wang, J. Duan, J. Huang, and K. Fu, "Ha-yolo: A real-time method for detecting surface defects in steel," 2024 5th International Conference on Artificial Intelligence and Computer Engineering (ICAICE), 2024, pp. 1038–1041.
[29] L. Sha, Z. Zhou, L. Wang, and Y. Zheng, "The steel defect detection algorithm based on yolov5-cgc," 2024 6th International Conference on Intelligent Control, Measurement and Signal Processing (ICMSP), 2024, pp. 248–252.
[30] T. Zhang, H. Pang, and C. Jiang, "Gdm-yolo: A model for steel surface defect detection based on yolov8s," IEEE Access, vol. 12, pp. 148 817–148 825, 2024.
[31] T. Zhu, Y. Liu, X. Li, Y. Li, H. Liu, and J. Yang, "Wise-iou: Bounding box regression loss with dynamic focusing mechanism," arXiv:2301.10051, 2023.
[32] Z. Ge, S. Liu, Z. Li, and J. Sun, "Eiou: Efficient iou for object detection," arXiv:2101.08158, 2021.
[33] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, "Mpdiou: A loss for efficient and accurate bounding box regression," arXiv:2307.07662, 2023.
[34] S. Woo, J. Park, J. Y. Lee *et al.*, "Cbam: Convolutional block attention module," Proceedings of the European Conference on Computer Vision (ECCV). Springer, 2018, pp. 3–19.
[35] H. Liu, F. Liu, X. Fan *et al.*, "Polarized self-attention: Towards high-quality pixel-wise regression," arXiv:2107.00782, 2021.
[36] Q. Wang, B. Wu, P. Zhu *et al.*, "Eca-net: Efficient channel attention for deep convolutional neural networks," Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. IEEE, 2020, pp. 11 534–11 542.