Journal of Electronic Research and Application, 2026, Volume 10, Issue 4
http://ojs.bbwpublisher.com/index.php/JERA
Online ISSN: 2208-3510
Print ISSN: 2208-3502

An Improved YOLOv8-Based Algorithm for Industrial Metal Surface Defect Detection

Yinghao Zhu, Shijie Jia*

Dalian Jiaotong University, Dalian 116028, Liaoning, China

Corresponding author: Shijie Jia, jsj@djtu.edu.cn

Abstract: To address challenges in industrial metal surface defect detection, including tiny defects, significant scale variation, and complex backgrounds, this study proposes an enhanced YOLOv8s-based model, termed MEAF-YOLOv8s. Based on the original YOLOv8s architecture, the model introduces several improvements to enhance feature extraction and multi-scale representation. First, a CSP-MSEE module is incorporated in the feature extraction stage to strengthen the capture of edge and detail information of tiny defects, thereby effectively alleviating the problem of insufficient feature representation for small targets. Second, an AFRBN module is introduced to establish long-range spatial dependencies. By leveraging global contextual information, the module suppresses texture background interference, while a re-parameterization strategy is adopted to maintain the lightweight nature of the model and ensure that inference efficiency is not compromised. In addition, a CA-HFPN feature fusion structure is employed, which incorporates a direction-aware coordinate attention mechanism and a hierarchical pyramid architecture to promote precise cross-scale feature alignment and adaptive fusion, thereby improving the model's adaptability and localization accuracy for defects of different sizes. To evaluate the proposed method, experiments are carried out on seven common defect types collected from real industrial environments. The results indicate that MEAF-YOLOv8s improves mAP50 by 4.72% and mAP50-95 by 1.28%, while decreasing the number of parameters by approximately 5M. These findings confirm that the proposed model can effectively enhance defect detection performance under complex background conditions.

Keywords: Deep learning; Defect detection; YOLOv8; Feature extraction; Multi-scale feature fusion

Online publication: May 22, 2026

1. Introduction

In industrial production and manufacturing, the quality standards for manufactured components have become increasingly rigorous, which makes defect inspection an essential step in ensuring product reliability and consistency. Surface defect detection is mainly concerned with recognizing flaws on industrial component surfaces, including scratches, contaminant residues, and dents, while also determining critical defect attributes

such as size, position, and type. For common industrial parts, frequently encountered surface defects include scratches, cracks, pits, stains, and holes. Such defects not only degrade the visual quality of components, but can also shorten their operational lifespan; in serious cases, they may even impair the functionality and dependability of products built from these components [1].

Traditional methods for metal surface defect inspection primarily depended on manual observation and non-destructive evaluation techniques, among which ultrasonic inspection and infrared-based detection were widely used [2–4]. Nevertheless, these approaches are associated with considerable labor demands, strong reliance on specialized equipment, and limited robustness under complicated working conditions, which restricts their effectiveness in practical industrial applications. Along with the continuous advancement of computer vision and deep learning, as well as the extensive adoption of industrial imaging devices, intelligent defect inspection approaches driven by deep learning have developed rapidly.

Currently, deep learning-based object detection methods are mainly grouped into two major types. The first type is the two-stage detection paradigm represented by the R-CNN series, including Faster R-CNN and Mask R-CNN [5–7]. The second type is the one-stage detection paradigm represented by the YOLO family (such as YOLOv5, YOLOv8, and YOLOv11) and the SSD series [8,9]. In industrial applications, both detection precision and inference speed are crucial considerations. Two-stage detectors usually obtain strong accuracy by generating candidate regions in advance and then conducting category prediction together with bounding-box refinement. However, their relatively high computational burden and limited inference efficiency make them less suitable for real-time industrial inspection tasks. Yang *et al*. proposed an improved Faster R-CNN approach that enhanced both precision and efficiency through the optimization of region proposal generation and feature extraction [10]. Huang *et al.* presented a parts surface defect detection method based on Faster R-CNN, where clustering theory was introduced to determine anchor configurations and a multi-level ROI pooling strategy was adopted to alleviate localization errors [11]. However, the detection of very small targets still remained challenging.

In contrast, one-stage detection algorithms formulate object localization and classification as a unified regression problem, enabling the direct prediction of object positions and categories through a single forward propagation process. Owing to their faster inference speed and simpler architecture, these methods are more suitable for real-time industrial applications. Nevertheless, most one-stage detectors adopt non-maximum suppression (NMS) to remove overlapping prediction boxes, which may lower processing efficiency and increase the risk of missed and incorrect detections, particularly when small targets are involved. Cheng *et al.* proposed an improved metal defect detection method based on YOLOv3 to address missed detections caused by small object sizes and indistinct features in metal surface defects [12]. Cao *et al.* improved the YOLOv5 algorithm for steel surface defect detection, enhancing the detection accuracy of small and complex defects by optimizing the feature extraction module and training strategy [13]. Zhang *et al.* further improved YOLOv8 by introducing a DRB module to strengthen the capture of small-scale defect patterns, achieving better performance than the original YOLOv8n on the NEU-DET dataset, with the detection accuracy increased by 3.3% [14]. In recent years, Transformer models, which first achieved breakthrough success in natural language processing, have gradually been introduced into computer vision tasks [15]. DETR (Detection Transformer), as the first end-to-end object detection model based on the Transformer architecture, discards the traditional anchor-based mechanism and NMS through set prediction, thereby significantly simplifying the detection pipeline. However, it still exhibits limitations in handling small targets, large-scale variations, and background disturbances in complex scenes.

To cope with these practical issues, this paper develops an enhanced YOLOv8-based model, termed MEAF-YOLOv8s. Experimental results show that the proposed model improves detection performance while also reducing parameter size and computational cost, thus offering a more effective approach for industrial defect detection.

2. Methodology

2.1. Overview of methodology

YOLOv8 belongs to the family of one-stage object detectors and is available in several variants with different model scales [16]. Among these versions, YOLOv8s provides a relatively suitable trade-off between computational burden and detection performance, making it appropriate for the present task. As illustrated in **Figure 1**, the YOLOv8s network consists of three main components, namely the backbone, the neck, and the detection head. The backbone is mainly used to extract image features. It is built upon the CBS basic unit, where convolution, batch normalization, and the SiLU activation function are combined to process the input features. In addition, the C2f module is employed to strengthen feature representation by preserving shallow details while further exploring high-level semantic information through branch-based transmission and feature concatenation, thereby balancing information completeness and computational cost. The neck is designed for multi-scale feature aggregation and incorporates the SPPF module together with the PAN-FPN structure. Specifically, the SPPF module enlarges the receptive field through successive pooling operations and integrates contextual information, whereas the PAN-FPN structure promotes bidirectional interaction among features at different levels. In this way, high-level semantic cues and low-level spatial details can be effectively combined, which improves the model's capability in recognizing and localizing targets of different scales. The detection head is used to generate the final predictions. By adopting a decoupled head and an anchor-free mechanism, the model handles classification and localization separately and outputs the target class, confidence score, and bounding box coordinates, which is beneficial to both detection performance and convergence speed.

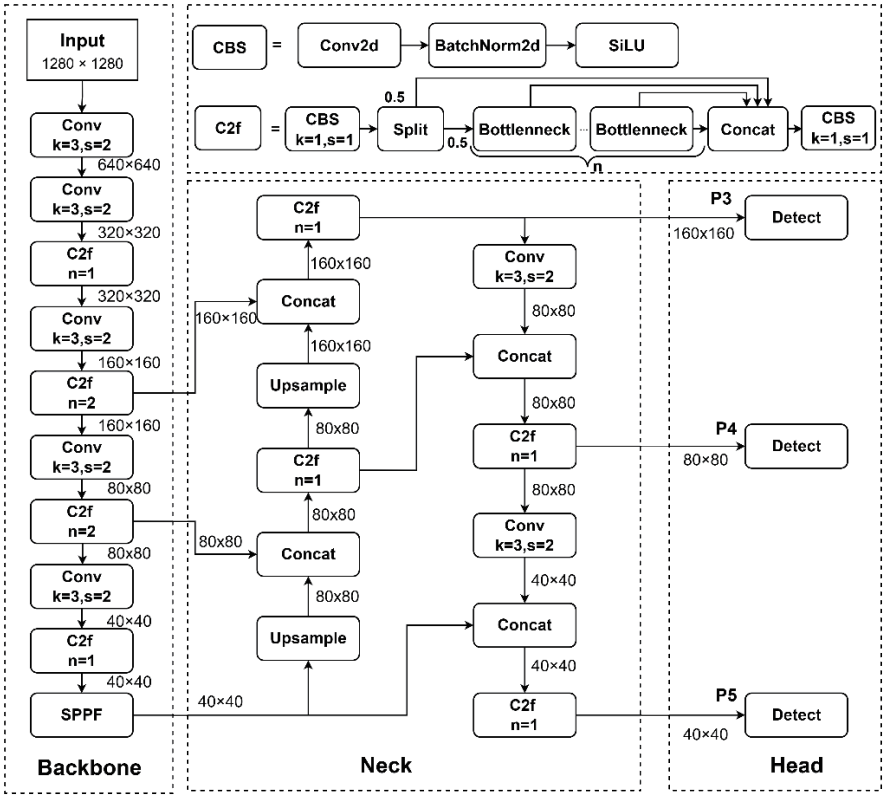

Figure 1. Overall architecture of YOLOv8s.

In industrial applications, metal surface defect detection is frequently affected by weak small-target characteristics and interference caused by complicated texture backgrounds. Although YOLOv8 performs well in balancing accuracy and efficiency, it still shows certain shortcomings in feature extraction, attention modeling, and multi-scale feature fusion when applied to industrial defect detection. These limitations become more evident in cases involving blurred defect boundaries, insufficient texture information, illumination changes, complex backgrounds, and obvious scale variations among defects. In view of these problems, this study develops an improved model, named MEAF-YOLOv8s, on the basis of the original YOLOv8s architecture. In the backbone stage, the initial C2f module is replaced by the CSP-MSEE module, which obtains semantic information at multiple levels through multi-scale adaptive pooling and further strengthens contour representation by introducing an edge enhancement unit, thereby alleviating the recognition difficulty caused by indistinct boundaries and missing detail features. In addition, the original SPPF module is replaced with the AFRBN module. By incorporating the Transformer self-attention mechanism, this module enhances the modeling of long-range spatial relationships in feature maps. Combined with the RepBN re-parameterization strategy, it preserves rich feature representation during training while maintaining relatively high inference efficiency, thus improving the model's robustness to illumination variation and complex background disturbance. During feature fusion, the conventional PAN-FPN is replaced with CA-HFPN. This structure applies coordinate attention to capture horizontal and vertical spatial dependencies separately, which improves the precision of defect localization. At the same time, its hierarchical pyramid design and adaptive weighted fusion strategy strengthen the interaction among features at different scales. As a result, the improved MEAF-YOLOv8s enhances the detection performance for multi-scale metal surface defects while keeping the parameter quantity at a reasonable level. The structure of the improved model is presented in **Figure 2**.

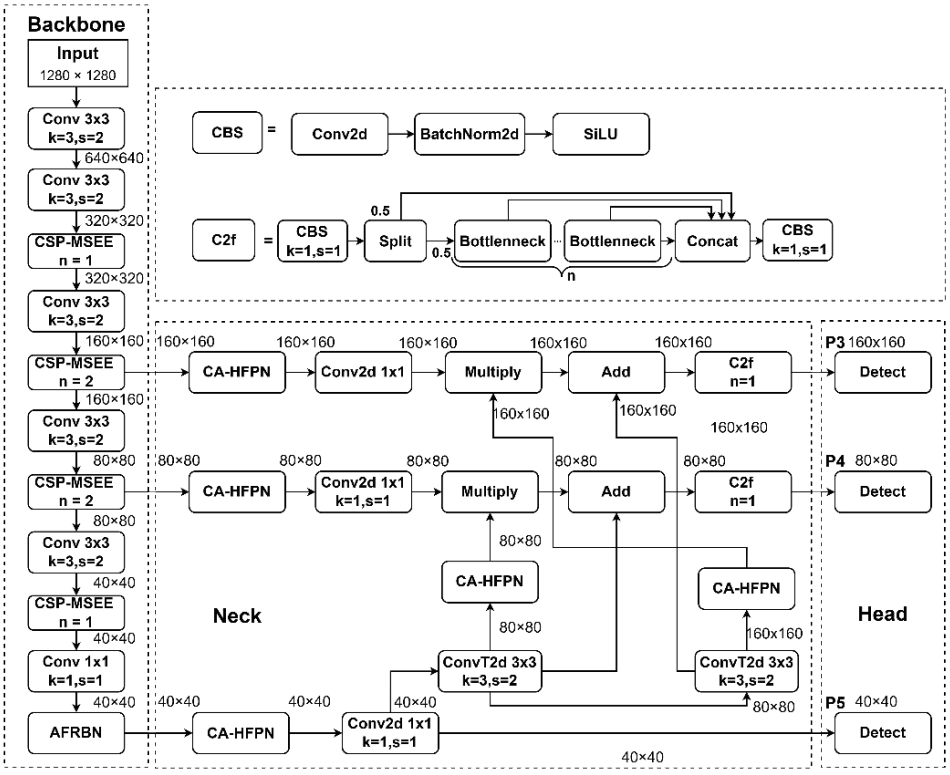

Figure 2. Overall architecture of MEAF-YOLOv8s.

2.2. CSP-MSEE module based on multi-scale edge information enhancement

In industrial defect detection tasks, target defects often exhibit significant scale diversity. For example, the pixel area of tiny scratches may account for only 0.1% of the entire image, whereas large-area abrasions or dents may cover more than 30% of the inspection region. Although the conventional C2f module enables feature reuse by means of cross-stage partial connections, its Bottleneck-based stacking design under a relatively fixed receptive field limits its ability to preserve local details while capturing global semantic information. In addition, for low-contrast defects in industrial scenarios, such as faint oxidation traces on metal surfaces, edge information is easily suppressed during the downsampling process, which consequently leads to a decline in detection recall.

To overcome the above limitations, a CSP-MSEE (Multi-Scale Edge Information Enhancement) module is designed in this study. On the basis of the original C2f framework, this module integrates an MSEE unit to extract defect-related information across different scales in an adaptive manner while reinforcing edge-related features. Through stronger multi-scale representation and improved retention of weak boundary cues, the proposed module enhances the model's ability to perceive subtle defects and increases its adaptability under complicated industrial conditions. The overall design of CSP-MSEE is presented in **Figure 3**.

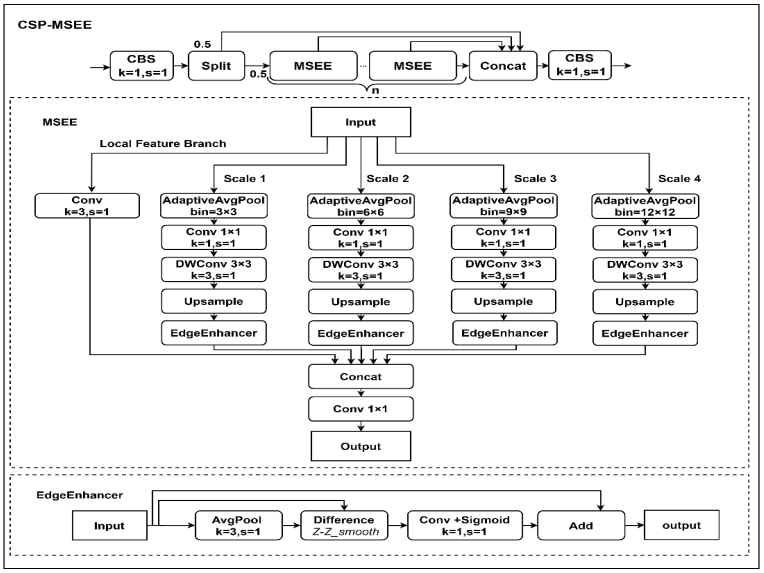

Figure 3. Architecture of the CSP-MSEE module.

The MSEE module is mainly designed to generate feature representations at different granularities by performing discrete sampling over multiple spatial scales. Given the input feature map X, the module first generates one local-fidelity branch and four scale-reduction branches, as formulated in **Equation (1)**, where θ_{local} denotes the parameters of the standard convolution:

$$F_{\text{local}} = \text{Conv}_{3\times3}\left(X;\theta_{\text{local}}\right) \in R^{C/4\times H\times W} \tag{1}$$

For the i-th scale branch, where $i \in 1,2,3,4$, a sequential processing chain consisting of adaptive pooling, convolution, and edge enhancement is adopted, as expressed in **Equation (2)**. Here, $\text{bin}_i \in 3,6,9,12$ corresponds to four grid partitions ranging from coarse to fine granularity, and $U(\cdot)$ represents a bilinear upsampling operation that resizes the feature map to(H,W). By introducing depthwise convolution (DWConv), the parameter count is reduced from $C^2\times k^2$ in standard convolution to $C^2\times k^2$ times, where k=3

denotes the kernel size.

$$F_{\text{scale}_i} = U\left(\text{EE}\left(\text{DWConv}\left(\text{Conv}_{1\times1}\left(\text{AdaptivePool}_{\text{bin}_i}\left(X\right)\right)\right)\right)\right) \tag{2}$$

The structure of the edge enhancement unit, denoted as EdgeEnhancer, is shown in Figure 3. It explicitly extracts high-frequency edge information by adopting local differencing and residual enhancement. For an intermediate feature $Z \in R^{C/4\times h\times w}$, a smoothed version is first generated through average pooling, as shown in Equation (3). Then, the Difference operation is used to compare the original input feature Z with the smoothed feature map Z_{smooth}. The subtraction between them produces the edge information E with blurred components suppressed, as calculated in Equation (4). This residual map represents the detailed and edge-related features in the image. Subsequently, the residual edge map E is processed by a 1×1 convolutional layer followed by a Sigmoid activation function to obtain the edge response feature A, as also shown in Equation (4). In this equation, θ_{edge} denotes the learnable parameters of the edge-mapping convolutional layer. Its initial values are automatically generated by the model parameter initialization strategy and are optimized through backpropagation during training. Finally, the edge response feature is added back to the input feature through a residual connection to obtain the enhanced output feature, as expressed in Equation (5). This design enables the pooled features to capture richer contextual information while still preserving relatively clear edge structures, thereby alleviating the detail blurring problem caused by conventional pooling operations.

$$Z_{\text{smooth}} = \text{AvgPool}_{3\times3}\left(Z\right) \tag{3}$$

$$E = Z - Z_{\text{smooth}}, \quad A = \sigma\left(\text{Conv}_{1\times1}\left(E;\theta_{\text{gate}}\right)\right) \tag{4}$$

$$\text{EE}\left(Z\right) = Z + A \tag{5}$$

Finally, the five feature branches are fused by concatenating them along the channel dimension and then applying a 1×1 convolution, as described in **Equation (6)**. The fusion convolution parameter implicitly learns the θ_{fusion} importance weights of F_{scale} different scales, enabling the network to adaptively strengthen the response of $F\text{scale}_4$ (*bin=3*) when detecting large-area defects, while increasing the contribution of (*bin=12*) when capturing tiny targets.

$$Y_{\text{MSEE}} = \text{Conv}_{1\times1}\left(\text{Concat}\left[F_{\text{local}}, F_{\text{scale}_1}, F_{\text{scale}_2}, F_{\text{scale}_3}, F_{\text{scale}_4}\right]; \theta_{\text{fusion}}\right) \tag{6}$$

As illustrated in **Figure 3**, the CSP-MSEE module is built following the Cross Stage Partial design principle. The input feature map is first passed through a 1×1 convolution for channel reduction and is then divided into two separate paths along the channel dimension. One path is retained directly as a shortcut connection, whereas the other is processed by several Bottleneck blocks before being fed into the MSEE module for multi-scale feature enhancement. Afterward, the resulting three feature paths are merged along the channel dimension and mapped to the desired output dimension through a final 1×1 convolution layer.

Compared with the original C2f module, the proposed design introduces an additional multi-scale edge enhancement pathway, allowing the final feature representation to simultaneously contain shallow gradient flow information, deep semantic abstraction, and multi-scale edge-enhanced features. The interaction of these components enhances the network's ability to represent complex defect patterns.

2.3. AFRBN module based on attention and re-parameterized batch normalization

Surface defect detection on mechanical parts often suffers from strong specular reflection interference and

complex background textures. In the original YOLOv8 backbone, the SPPF module at the final stage enlarges the receptive field only through local spatial pyramid pooling, which presents clear limitations in complex industrial scenarios. To overcome this issue, this paper proposes the AFRBN module. Inspired by the attention-based feature interaction mechanism in RT-DETR, the proposed module replaces the original SPPF with AIFI (Attention-based Intrascale Feature Interaction) and further introduces RepBN for optimization [17]. In this way, direct associations can be established among all spatial positions in the feature map, enabling long-range feature interaction beyond the restriction of local receptive fields. The structural design of the AFRBN module is presented in **Figure 4**.

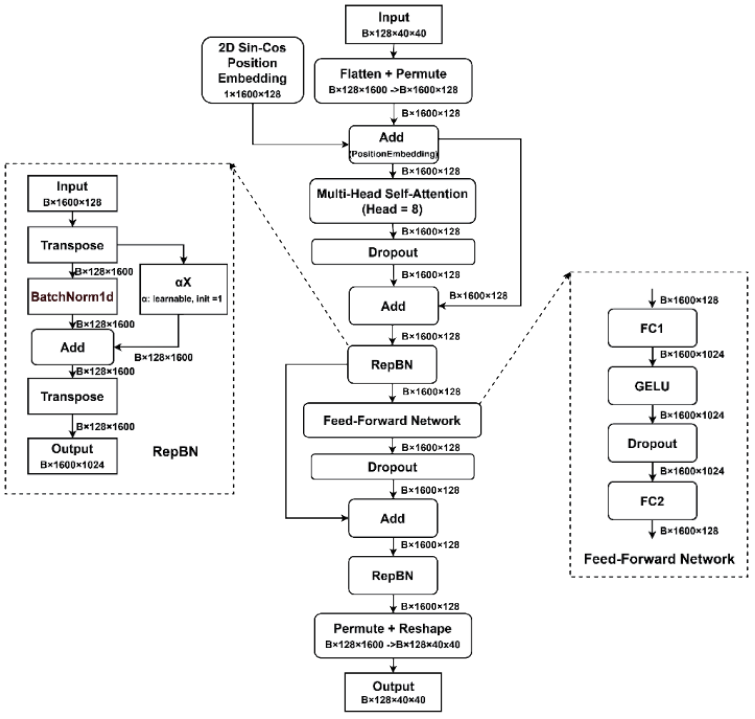

Figure 4. Structural diagram of the AFRBN module.

The AFRBN module first generates a two-dimensional sine-cosine positional encoding matrix (2D sin-cos position embedding) for the input feature map. The low-frequency components capture coarse-grained spatial layouts, while the high-frequency components encode fine positional differences. Together, they provide rich geometric prior knowledge, ensuring that the geometric location and topological structure of defects can be preserved during network propagation. The expression is presented in **Equation (7)**, $i \in [0, W-1]$ and $j \in [0, H-1]$ represent the coordinate indices in the width and height directions, respectively; d denotes the channel index, and C refers to the encoding dimension.

$$PosEmbed_{ij}(d) = sin\left(\frac{i}{10000^{2d/C}}\right) + cos\left(\frac{j}{10000^{2d/C}}\right) \tag{7}$$

The generated positional encoding is subsequently combined with the original feature map in an element-wise manner to explicitly incorporate spatial geometric information, as expressed in **Equation (8)**, X' where denotes the enhanced feature representation. Through this operation, the subsequent self-attention computation can rely not only on content similarity, but also on the spatial positional relationship between

feature points, which is particularly important for understanding different morphological structures of the same defect.

$$X' = X + PosEmbed\left(X\right) \tag{8}$$

The position-enhanced feature X' is then fed into the Transformer encoder layer, where global contextual modeling is achieved through the multi-head self-attention mechanism, as formulated in **Equation (9)**. In this equation, the query Q, key K, and value V matrices are all obtained from different linear projections of X'. The term d_K denotes the dimensionality of the key vector and is used to scale the dot-product results for gradient stabilization. The multi-head mechanism projects features into multiple representation subspaces and computes attention in parallel, allowing different heads to focus on different feature patterns, such as edges, textures, and illumination variations. After concatenation, these outputs form a rich multi-view feature representation. This capability of global information interaction enables the model to establish contrastive relationships between defect regions and surrounding backgrounds across long spatial distances, thereby improving sensitivity to anomalous patterns.

$$Attention\left(Q,K,V\right) = softmax\left(\frac{QK^T}{\sqrt{d_K}}\right)V \tag{9}$$

In terms of structural optimization, which is also an important aspect of the proposed module, RepBN is adopted to replace the standard Layer Normalization after the AIFI module is used in place of SPPF. Although Layer Normalization is widely used in Transformers, it cannot fully exploit hardware parallelism during batch inference. Based on Batch Normalization, RepBN introduces an additional learnable identity mapping branch to form a dual-path structure, as described in **Equation (10)**. In this formulation, $BN(X)$ performs standard batch normalization by reducing inter-batch distribution differences through mean subtraction and variance normalization, while is a trainable scaling factor used to dynamically adjust the contribution of the residual path. This design preserves the regularization effect and training acceleration advantages of Batch Normalization, while simultaneously enhancing gradient flow through the identity shortcut and alleviating information loss caused by excessive normalization.

$$RepBN\left(X\right) = BN\left(X\right) + \alpha \cdot X \tag{10}$$

Finally, the features generated by the self-attention mechanism are delivered to a Feed-Forward Network (FFN) for subsequent nonlinear projection. In comparison with ReLU, GELU (Gaussian Error Linear Unit) yields a smoother activation pattern and preserves gradient information more effectively for negative-valued inputs, which is advantageous for modeling subtle contrast changes in metal surface defects. Structurally, the FFN contains two fully connected layers separated by a GELU activation unit. Owing to its continuously differentiable property, GELU enables more effective gradient propagation than ReLU, which helps strengthen defect-related feature responses while suppressing background interference. The FFN is formulated in **Equation (11)**, where $W_1 \in R^{C \times C_{\text{hidden}}}$ and $W_2 \in R^{C_{\text{hidden}} \times C}$ are the weight matrices of the first and second layers, respectively, and b_1 and b_2 are the corresponding bias vectors.

$$\text{FFN}\left(X\right) = \text{FC}_2\left(\text{GELU}\left(\text{FC}_1\left(X\right)\right)\right) = W_2 \cdot \text{GELU}\left(W_1 \cdot X + b_1\right) + b_2 \tag{11}$$

Through the above steps, the AFRBN module effectively integrates spatial position awareness, global contextual modeling, and efficient feature transformation. Finally, after reshaping and transposition, the output feature map is obtained, thereby improving the model's performance in metal surface defect detection

under complex background conditions.

2.4. CA-HFPN: Hierarchical feature pyramid network with coordinate attention

In the original detection head of YOLOv8, multi-scale features are mainly fused through upsampling and direct concatenation. However, features from different levels lack an explicit selection or guidance strategy before fusion, which may cause mutual interference between deep semantic cues and shallow detail information. For metal surface defect detection, the model must accurately distinguish defect regions from non-defect regions under complex background conditions. Although low-level features contain abundant edge and texture information, they also introduce a large amount of background redundancy. If these features are directly fused, the response capability of the network to tiny defects may be weakened. Therefore, it is necessary to introduce a more effective feature selection and fusion mechanism into the multi-scale feature interaction process.

According to the above analysis, CA-HFPN is introduced into the feature fusion stage of YOLOv8 to strengthen the distinction and stability of multi-scale features during the fusion process. The structure of CA-HFPN is illustrated in **Figure 5**. This module mainly contains two key parts, namely feature selection and adaptive feature fusion, and is constructed with a hierarchical modeling strategy. By taking into account the scale diversity, morphological variation, and contrast differences of mechanical surface defects, CA-HFPN provides an effective solution for attention-guided feature interaction.

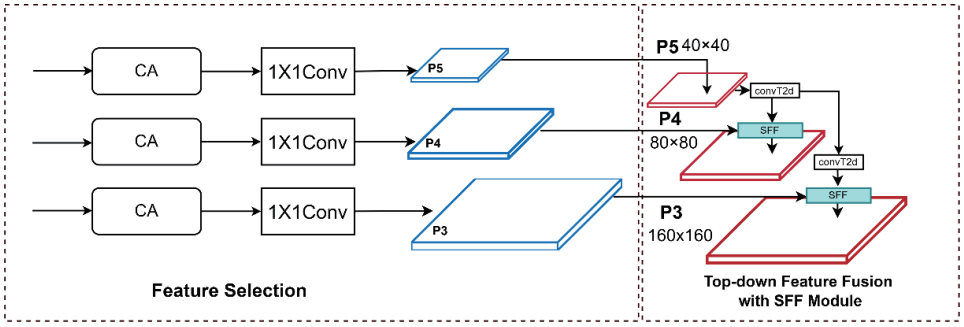

Figure 5. Architecture of the CA-HFPN.

In the feature selection stage, CA-HFPN adopts an improved Coordinate Attention mechanism (Coordinate Attention_HFPN) to assign adaptive weights to the input feature maps [18]. The corresponding structure is presented in **Figure 6**. Specifically, one-dimensional global average pooling is performed independently in the height and width directions to obtain direction-aware channel statistics. The resulting features are then combined in the spatial dimension and passed through a 1×1 convolution, batch normalization, and a nonlinear activation layer, so that channel compression and feature integration can be achieved. After that, the fused representation is divided into two directional branches associated with height and width, from which the corresponding attention coefficients are produced. Finally, the coordinate attention module applies these weights to the input feature map by element-wise multiplication.

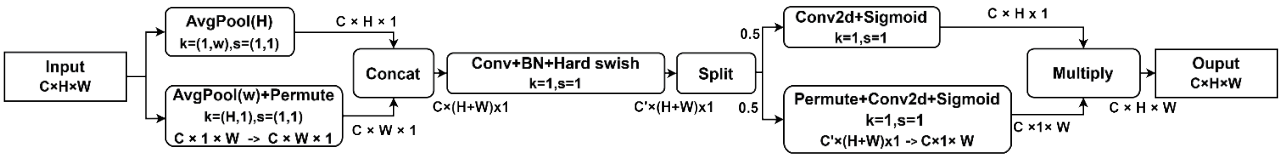

Figure 6. Structural diagram of the coordinate attention module.

Through the above mechanism, the coordinate attention module is capable of emphasizing critical channel responses while preserving spatial positional information. As a result, it effectively suppresses background redundancy in low-level features and provides more salient feature representations for tiny defects.

During feature fusion, CA-HFPN adopts the Selective Feature Fusion (SFF) module as illustrated in **Figure 7**[19]. The core idea of this module is to dynamically screen low-level features using attention weights generated from high-level semantic features, rather than concatenating features from different scales. In implementation, SFF first upsamples the high-level features to the same resolution as the low-level features, ensuring spatial alignment between them and laying the foundation for subsequent feature fusion. Then, the attention weights generated by the CA module are multiplied element-wise with the low-level features to achieve selective filtering. Finally, the filtered low-level features are added to the upsampled high-level features through residual addition, and the fused features are further processed by the C2f module to generate the final detection feature map. This hierarchical fusion strategy effectively reduces information redundancy caused by cross-level feature interaction, enabling the network to focus more on the channels and spatial regions that contribute most to the detection task.

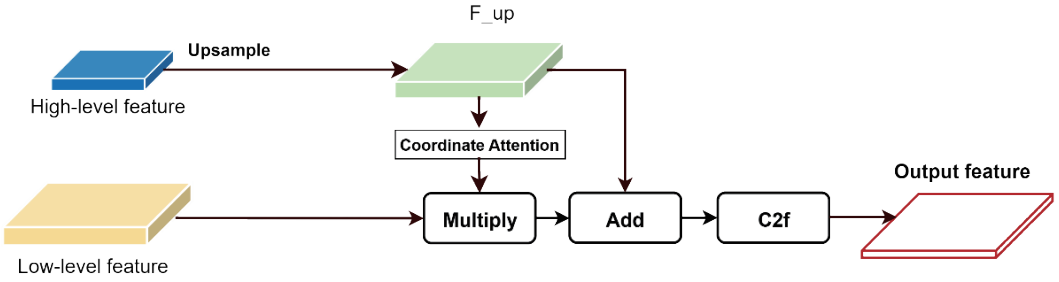

Figure 7. Structure of the selective feature fusion module.

3. Experiments

3.1. Experimental settings

The experimental environment adopted in this study is shown in **Table 1**, while the corresponding hyperparameter settings are provided in **Table 2**. Notably, among the models included in the comparative experiments, only YOLO26 is optimized using the MuSGD algorithm, whereas all the remaining models are trained with SGD.

Table 1. Experimental platform and environment

Parameter	Configuration
Operating System	Windows 10
CPU	Intel® Xeon® W-2245
GPU	NVIDIA GeForce RTX 3090
CUDA Version	CUDA 11.8
Deep Learning Framework	PyTorch 2.1.0
Programming Language	Python 3.8

Table 2. Experimental hyperparameter settings

Parameter	Configuration
Epochs	300
Momentum	0.9
Initial Learning Rate	0.0001
Min Learning Rate	0.00001
Weight Decay	0.0001
Batch Size	8
Workers	8
Image Size	1280X1280
Optimizer	SGD

3.2. Dataset construction

According to existing studies, most commonly used defect detection datasets are primarily designed for small-target defect detection, while only a few directly employ high-resolution images for global defect inspection. Moreover, real metal defects are highly diverse, and even common defect categories can present in different forms, yet few datasets provide a reasonable and detailed classification of such defects. In this study, all images were collected from real high-resolution on-site images provided by a cooperating enterprise, and the defect categories were annotated and classified by the company's quality assurance personnel. Taking metal defects on aluminum parts as the research object, this paper constructs a dataset named Dataset-AMP (Defect Dataset of Aluminum Metal Parts). At present, the dataset contains seven defect categories: white scratches, severe black scratches, white abrasions, severe black abrasions, indentations, deformation, and material loss.

During the image acquisition process, a PA8KGV-45KM industrial line-scan camera was used, together with a constant-current line-light controller for auxiliary illumination. As shown in **Figure 8**, the acquisition system consists of two cameras mounted above and below the production line to capture the front and back surfaces of the same product, respectively. To obtain clear and complete images, the image width and height were set to 4480 and 3360, respectively. This resolution represents an appropriate image resolution for presenting the overall defects of the parts as completely as possible without introducing distortion. The self-constructed dataset contains a total of 1,821 high-resolution images with a resolution of 4480 × 3360, all collected under real industrial conditions. Each image contains different categories of defects, covering the seven categories mentioned above. Among them, 1,459 images were used for training, 184 images for validation, and 178 images for testing. Representative samples of the dataset are shown in **Figure 9**.

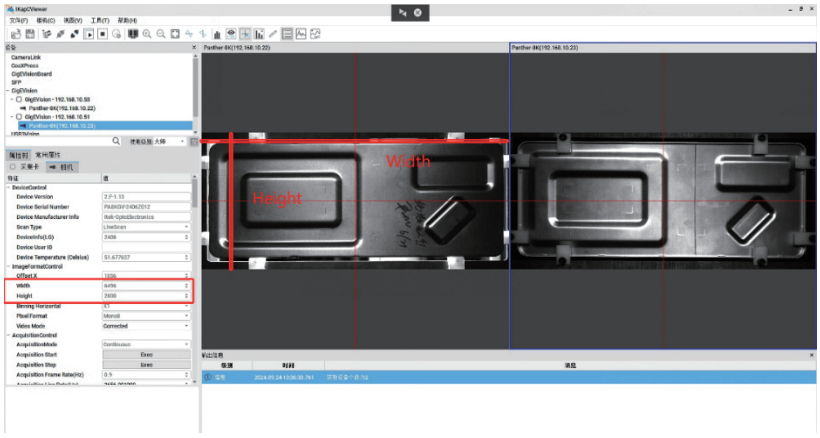

Figure 8. Camera parameter settings.

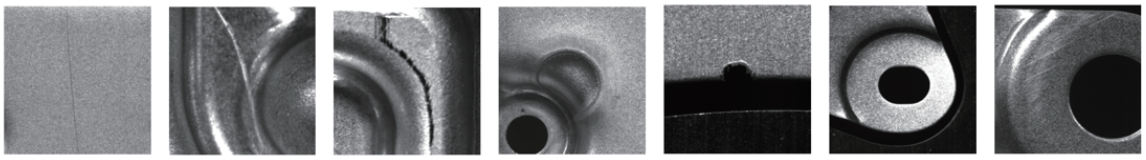

Figure 9. Metal defect samples from the dataset-AMP.

3.3. Evaluation metrics

To provide a comprehensive assessment of the proposed model, a series of evaluation metrics are employed in this study, including Precision, Recall, Average Precision (AP), mean Average Precision (mAP), Parameters, and GFLOPs. Among them, Precision is used to measure the proportion of correctly predicted positive samples among all positive predictions, thereby reflecting the reliability of the detection results. Recall describes the proportion of true positive samples that are successfully identified among all actual positive samples, which indicates the model's ability to detect targets. The corresponding definitions are given in **Equation (12)** and **Equation (13)**, respectively:

$$Precision = \frac{TP}{TP + FP} \tag{12}$$

$$Recall = \frac{TP}{TP + FN} \tag{13}$$

AP is defined as the area under the Precision-Recall (PR) curve and is commonly adopted to evaluate the detection performance of a single category. On this basis, mAP represents the mean value of AP over all categories. In particular, under a specified IoU threshold such as mAP50, it can provide an overall measure of detection performance across different classes. In general, a larger mAP value indicates better detection capability of the model. The calculation of AP and mAP is shown in **Equation (14)** and **Equation (15)**, respectively:

$$AP = \int_0^1 P(R) \tag{14}$$

$$mAP = \frac{1}{N}\sum_{i=1}^{n} AP_i \tag{15}$$

In addition, Parameters indicate the model size and complexity, where fewer parameters generally imply a more compact model. GFLOPs represent the computational cost of a single forward pass, where lower values indicate higher computational efficiency.

3.4. Comparative experiments

In this paper, we compare the improved YOLOv8s model with the baseline YOLOv8s and several mainstream object detection algorithms to verify the effectiveness of the proposed improvements. The comparison methods include YOLOv5, YOLOv8, YOLOv10, YOLO11, YOLO12, YOLO26, as well as lightweight and larger-scale variants. The experiments were conducted on the self-built Dataset-AMP metal surface defect dataset, and the models were evaluated in terms of detection accuracy and model complexity. The results are shown in **Table 3**.

YOLO models of different scales show clear differences in terms of detection performance and computational burden. As a lightweight model, YOLOv8n contains only 3.0M parameters and requires 8.1 GFLOPs, which makes it advantageous for deployment. Nevertheless, its mAP50 and recall are only 55.72% and 57.73%, respectively, indicating that small defects are more likely to be missed and that its performance is insufficient for industrial inspection tasks with high reliability requirements. In comparison, medium-scale compact models provide a more suitable compromise between accuracy and efficiency. Among them, YOLOv8s achieves 74.09% precision and 62.43% mAP50, reflecting relatively strong detection performance. Although the latest YOLO12s reaches 63.1% in mAP50 and 32.24% in mAP50-95, slightly surpassing YOLOv8s in overall accuracy, its precision is only 67.82%, which is 6.27 percentage points lower than that of YOLOv8s. This result suggests that YOLO12s may produce more false positives in practical applications, thereby increasing the cost of subsequent manual inspection.

Table 3. Comparison of results among different object detection algorithms 窗体顶端窗体底端

Model	P(%)	R(%)	mAP50(%)	mAP50-95(%)	Params(M)	GFLOPs
YOLOv8n	64.7	57.73	55.72	24.69	3.0	8.1
YOLOv5s	65.72	60.47	62.2	31.26	9.1	23.8
YOLOv8s	74.09	58.26	62.43	30.88	11.1	28.5
YOLOv10s	63.39	50.94	54.27	23.88	7	21.4
YOLO11s	64.89	61.41	59.73	30.13	9.4	21.3
YOLO12s	67.82	62.57	63.1	32.24	9.2	21.2
YOLO26s	70.18	60.4	58.8	29.54	9	20.5
YOLOv5m	74.34	62.04	61.4	31.34	25.0	64.0
YOLOv8m	71.02	56.73	62.79	31.95	25.8	78.7
YOLOv10m	60.23	60.31	60.34	29.66	15	58.9
YOLO11m	58.6	64.16	60.84	31.89	20.0	67.7
YOLO26m	72.98	53.17	56.77	30.49	20	67.9
Ours	65.64	65.27	67.15	32.16	6.1	21.5

When extending to medium-scale models, performance shows a more complex trend. YOLOv5m

achieves a precision of 74.34% and recall of 62.04%, but its parameters increase significantly to 25.0M, and GFLOPs reach 64.0G, approximately 2.25 times those of YOLOv8s. Such expansion imposes higher demands on edge devices, while its mAP50 drops to 61.4%, indicating that increasing model capacity does not necessarily improve detection performance.

From the perspective of YOLO version evolution, different generations show varying adaptability to metal defect detection tasks. The YOLOv8 series, benefiting from the C2f module and decoupled detection head, achieves significant accuracy improvements. Compared to YOLOv5s, YOLOv8s increases precision from 65.72% to 74.09%, an improvement of 8.37 percentage points, which has direct economic value in reducing false detections in industrial scenarios.

However, most newer models perform less effectively in this task. YOLOv10s achieves mAP50 of only 54.27% and recall of 50.94%, making it the weakest among small models. YOLO11s achieves mAP50 of 59.73%, which is 2.7 percentage points lower than YOLOv8s. Although YOLO26s achieves a precision of 70.18%, its mAP50 is only 58.8%, still lower than YOLOv8s and YOLO12s. This trend is more evident in medium-scale models. YOLO11m achieves a recall of 64.16% but has a precision of only 58.6%, far below YOLOv5m. YOLOv10m and YOLO26m achieve precisions of 60.23% and 72.98%, respectively, but neither surpasses YOLOv8m in mAP50. This indicates that newer models are optimized for general object detection tasks and are less suitable for industrial defect detection scenarios involving small objects, low-contrast textures, and complex backgrounds.

Considering detection accuracy, model complexity, and deployment requirements, YOLOv8s is selected as the baseline model for further improvement. The YOLOv8 framework provides a mature development ecosystem and strong community support, facilitating subsequent optimization and TensorRT deployment. Experimental results further validate this choice: the improved model based on YOLOv8s reduces parameters to 6.1M while achieving an mAP50 of 67.15%, significantly outperforming other YOLO models of similar scale.

From the perspective of training process stability, **Figure 10** illustrates the training loss comparison curve between the improved model and the original YOLOv8s model. As seen from the image, the improved model demonstrates a faster convergence rate in the early stages of training, with a significantly better loss function descent gradient compared to the baseline model. Furthermore, the loss curve becomes more stable in the later stages of training, with a notable reduction in fluctuation amplitude. This indicates that the proposed improvement strategy not only enhances the detection performance of the model but also strengthens the stability and robustness of the training process, facilitating the acquisition of more reliable model weights in practical applications.

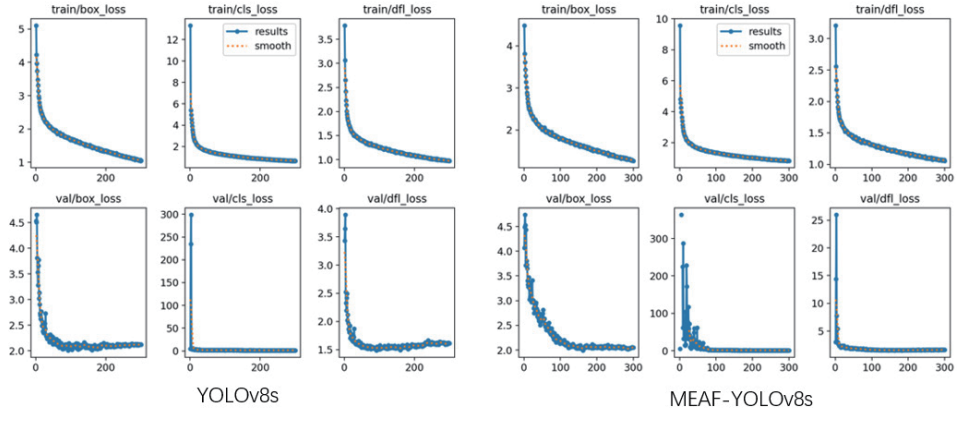

Figure 10. Training loss comparison between the improved and original models.

3.5. Ablation experiments

To verify the effectiveness of each module introduced in MEAF-YOLOv8s, ablation studies are carried out in this work. The original YOLOv8s serves as the baseline model, and the CSP-MSEE, AFRBN, and CA-HFPN modules are incorporated step by step. To guarantee a fair comparison, all experiments are performed under identical hardware conditions and training configurations. The corresponding results are presented in **Table 4**.

Table 4. Results of the ablation experiments

Model	Baseline	CA-HFPN	AFRBN	CSP-MSEE	mAP50(%)	Params(M)	GFLOPs
A	√				62.43	11.13	28.5
B	√	√			65.21	7.09	23.9
C	√	√	√		65.32	6.69	23.6
D	√	√		√	66.18	6.48	21.8
E	√	√	√	√	67.15	6.08	21.5

Scheme A represents the baseline YOLOv8s model. Scheme B introduces the CA-HFPN feature fusion module into the neck. Scheme C further adds the AFRBN module based on B. Scheme D adds CSP-MSEE based on B. Scheme E integrates all modules.

Compared with Scheme A, Scheme B improves mAP50 from 62.43% to 65.21% (+2.78%), while reducing parameters from 11.13M to 7.09M (−36.3%) and GFLOPs from 28.5G to 23.9G (−16.1%). This demonstrates that CA-HFPN significantly improves feature fusion efficiency while reducing model complexity.

In Scheme C, the AFRBN module further improves mAP50 to 65.32% (+2.89% over baseline), while reducing parameters to 6.69M (−39.9%). Compared to Scheme B, mAP50 increases by 0.11%, and parameters decrease by 5.6%. AFRBN utilizes Transformer-based self-attention and re-parameterization to achieve both accuracy improvement and model simplification.

Scheme D integrates CSP-MSEE, which enhances fine-grained feature extraction. Compared to Scheme A, mAP50 increases by 3.75% to 66.18%, while parameters and FLOPs decrease by 41.8% and 23.5%, respectively. Compared to Scheme B, mAP50 improves by 0.97%. Compared to Scheme C, Scheme D improves mAP50 by 0.86%, demonstrating superior capability in capturing defect boundaries and weak textures.

Scheme E integrates all modules, achieving mAP50 of 67.15% (+4.72%), while reducing parameters by 45.4% to 6.08M and FLOPs by 24.6% to 21.5G. This confirms the synergistic effect of all modules and indicates that the proposed model achieves an optimal balance among accuracy, complexity, and efficiency.

3.6. Visualization analysis

As shown in **Figure 11**, the first and third images contain defects labeled as baicashang, which are incorrectly detected by the baseline model but correctly recognized by the improved model. The improved model also yields higher confidence scores for the detected defects.

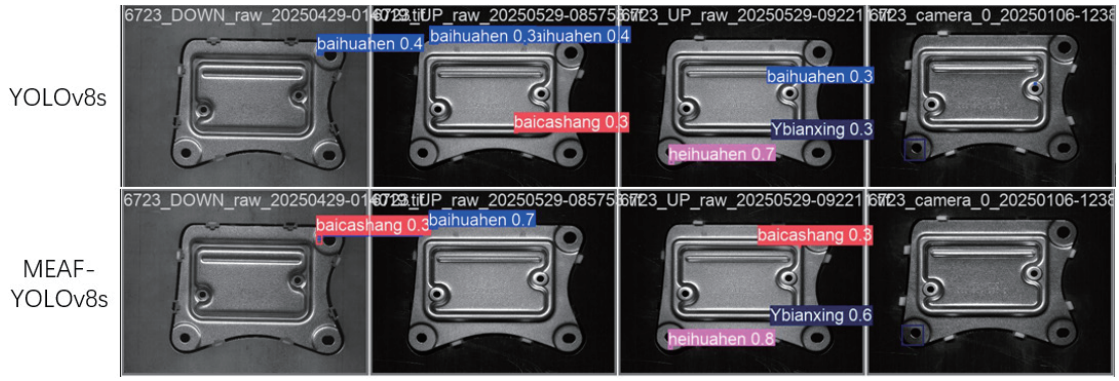

Figure 11. Visualization 1.

In **Figure 12**, the baseline model misclassifies Yyahen as baicashang, while the improved model correctly identifies it with higher confidence.

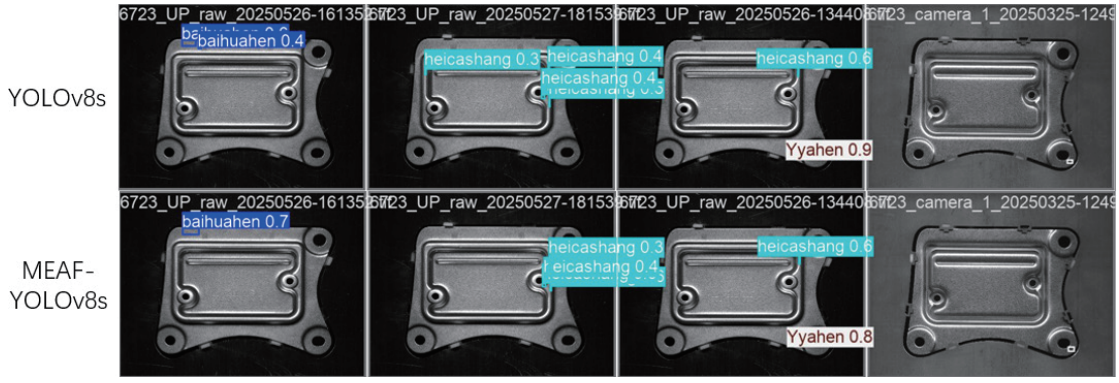

Figure 12. Visualization 2.

In **Figure 13**, the baseline model produces multiple bounding boxes for a single defect and shows false detections. In contrast, the MEAF-YOLOv8s model achieves accurate detection without false positives. In practical applications, although both models can detect high-confidence defects under predefined class-specific confidence thresholds, the improved model shows better reliability and fewer false detections.

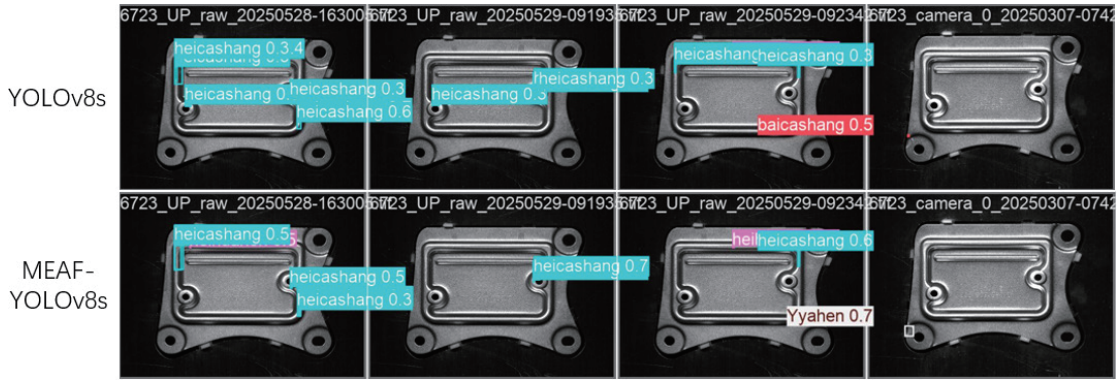

Figure 13. Visualization 3.

In **Figure 14**, the defect heicashang in the lower-left corner is correctly detected by the improved model.

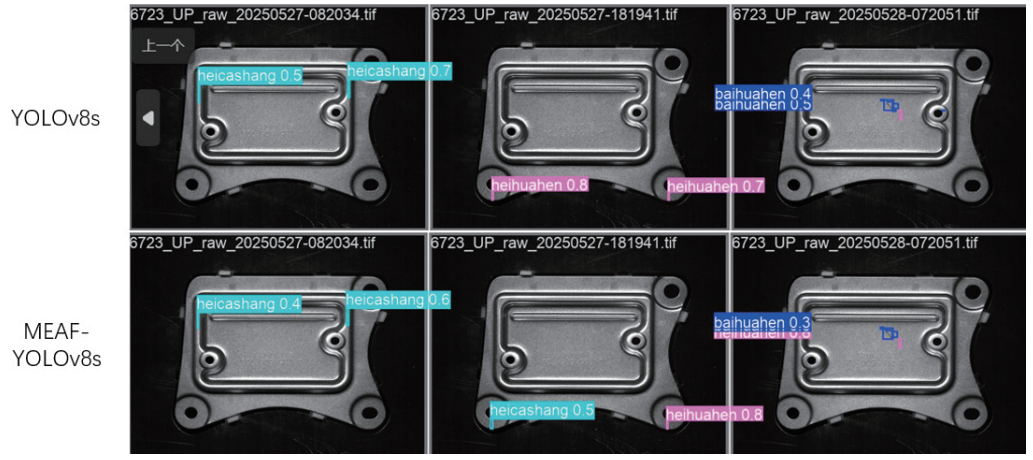

Figure 14. Visualization 4.

In **Figure 15**, the improved model successfully detects heicashang with high confidence, demonstrating its strong defect recognition capability.

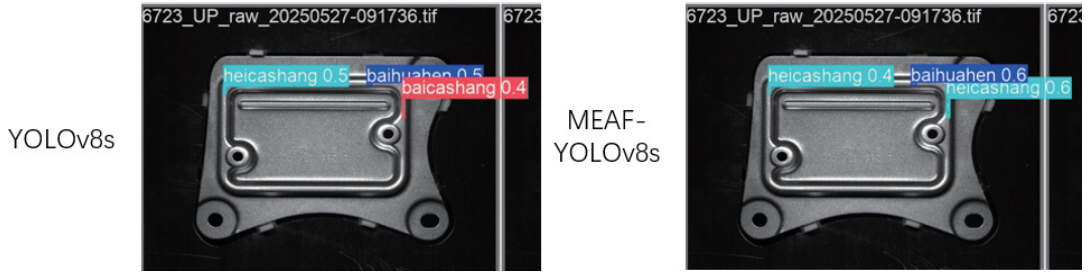

Figure 15. Visualization 5.

4. Conclusion

In this study, an enhanced YOLOv8s-based model, termed MEAF-YOLOv8s, is proposed for metal surface defect detection. The proposed model is designed to address several practical challenges encountered in industrial applications, including tiny defect targets, substantial scale variations, and interference from complex background textures. In the feature extraction stage, the CSP-MSEE module is introduced to optimize the backbone network, thereby improving the preservation of edge details and local information for small defects through multi-scale receptive field modeling and edge information enhancement. Moreover, the AFRBN module is incorporated to strengthen the modeling of long-range spatial dependencies. By integrating two-dimensional sine-cosine positional encoding with a multi-head self-attention mechanism, the proposed model effectively suppresses background texture interference while maintaining a relatively lightweight architecture. Furthermore, the CA-HFPN structure is employed in the feature fusion stage to enhance the interaction and integration of multi-scale features. Through the combination of coordinate-aware attention and a hierarchical pyramid design, the proposed model further improves its adaptability to defects with varying scales. The experimental results on the Dataset-AMP dataset show that, compared with the original model, the proposed method increases mAP50 by 4.72% and mAP50-95 by 1.28%, while

decreasing the parameter count by approximately 5 million. These findings suggest that MEAF-YOLOv8s can enhance detection performance and, at the same time, lower the overall model complexity. As a result, the proposed method achieves an effective compromise between lightweight architecture and detection capability, indicating its practical value for deployment in industrial scenarios. Although the proposed model has achieved promising results in the task of surface defect detection for aluminum parts, it still faces several challenges, such as the diversity of defect morphologies, robustness under extreme illumination conditions, and insufficient cross-material generalization capability. Future research should focus on further improving inference efficiency and exploring techniques such as knowledge distillation and model quantization. Meanwhile, introducing industrial defect datasets containing a wider variety of material types would help enhance the cross-domain generalization ability of the algorithm. In addition, with the development of edge computing devices, validating the deployment performance of the model on embedded platforms and combining it with active learning strategies for online incremental updates will provide stronger technical support for defect detection in intelligent manufacturing.

Disclosure statement

The author declares no conflict of interest.

References

[1] Li Z, Gao C, Lv X, et al., 2023, A Review of Metal Material Surface Defect Detection based on Deep Learning. Manufacturing Technology & Machine Tool, 2023(6): 61–67.

[2] Wang H, Yang C, Lv Q, 2022, A Review of Metal Surface Defect Detection Methods based on Machine Vision. Micro/Nano Electronics and Intelligent Manufacturing, 4(4): 71–81.

[3] Sun M, Cheng X, Wang Y, et al., 2016, Surface Defect Detection Method for High-Speed Railway Rails based on Photoacoustic Signals. Acta Physica Sinica, 65(3): 351–360.

[4] Chen Y, Xin Y, 2016, An Efficient Infrared Small Target Detection Method based on Visual Contrast Mechanism. IEEE Geoscience and Remote Sensing Letters, 13(7): 962–966.

[5] Girshick R, Donahue J, Darrell T, et al., 2014, Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation, 2014 IEEE Conference on Computer Vision and Pattern Recognition, 580–587.

[6] Ren S, He K, Girshick R, et al., 2017, Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. IEEE Transactions on Pattern Analysis and Machine Intelligence, 39(6): 1137–1149.

[7] He K, Gkioxari G, Dollár P, et al., 2017, Mask R-CNN, 2017 IEEE International Conference on Computer Vision (ICCV), 2980–2988.

[8] Redmon J, Divvala S, Girshick R, et al., 2016, You Only Look Once: Unified, Real-Time Object Detection, 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 779–788.

[9] Liu W, Anguelov D, Erhan D, et al., 2016, SSD: Single Shot MultiBox Detector, Lecture Notes in Computer Science, 21–37.

[10] Yang L, Zhang Y, Wang T, et al., 2021, Steel Surface Defect Detection Method based on Improved Faster R-CNN. Journal of Jilin University (Information Science Edition), 39(4): 409–415.

[11] Huang F, Li Y, Guo L, et al., 2020, Surface Defect Detection Algorithm for Parts based on Faster R-CNN. Journal of Computer-Aided Design & Computer Graphics, 2020(6): 883–893.

[12] Cheng J, Duan X, Zhu W, 2021, Metal Surface Defect Detection based on Improved YOLOv3. Computer Engineering and Applications, 57(19): 252–258.

[13] Cao Y, Wu M, Xu L, 2023, Steel Surface Defect Detection based on Improved YOLOv5 Algorithm. Journal of Graphics, 44(2): 335–345.

[14] Zhang T, Pan P, Zhang J, et al., 2024, Steel Surface Defect Detection Algorithm based on Improved YOLOv8n. Applied Sciences, 14(12): 5325.

[15] Vaswani A, Shazeer N, Parmar N, et al., 2017, Attention is all you Need, 31st International Conference on Neural Information Processing Systems (NIPS'17), 6000–6010.

[16] Jocher G, Chaurasia A, Qiu J, 2023, Ultralytics YOLO (Version 8.0.0).

[17] Guo J, Chen X, Tang Y, et al., 2024, SLAB: Efficient Transformers with Simplified Linear Attention and Progressive Re-Parameterized Batch Normalization, International Conference on Machine Learning.

[18] Hou Q, Zhou D, Feng J, 2021, Coordinate Attention for Efficient Mobile Network Design, 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 13708–13717.

[19] Chen Y, Zhang C, Chen B, et al., 2021, Accurate Leukocyte Detection based on Deformable-DETR and Multi-Level Feature Fusion for Aiding Diagnosis of Blood Diseases. Computers in Biology and Medicine, 2024(170): 107917.