MAM-YOLO: A Real-Time Algorithm for Detecting Defects in Steel Surfaces Using Multi-Scale Feature Enhancement

Yu Wang[1], Chao Mao[1], Fengling Jiang[2], Le Zou[3,*] and Jian Liu[4,*]

[1] School of Energy Materials and Chemical Engineering, Hefei University, Hefei, China

[2] School of Computer and Artificial Intelligence, Hefei Normal University, Hefei, China

[3] School of Artificial Intelligence and Big Data, Hefei University, Hefei, China

[4] School of Chemical and Pharmaceutical Engineering, Hefei Normal University, Hefei, China

INFORMATION

Keywords:
Defect detection
multi-scale convolution
attention mechanism
MPD-IoU loss function

DOI: 10.23967/j.rimni.2026.10.76720

SCIPEDIA Revista Internacional de Métodos Numéricos para Cálculo y Diseño en Ingeniería

Revista Internacional Métodos numéricos para cálculo y diseño en ingeniería RIMNI

In cooperation with CIMNE

MAM-YOLO: A Real-Time Algorithm for Detecting Defects in Steel Surfaces Using Multi-Scale Feature Enhancement

Yu Wang[1], Chao Mao[1], Fengling Jiang[2], Le Zou[3,*] and Jian Liu[4,*]

[1]School of Energy Materials and Chemical Engineering, Hefei University, Hefei, China

[2]School of Computer and Artificial Intelligence, Hefei Normal University, Hefei, China

[3]School of Artificial Intelligence and Big Data, Hefei University, Hefei, China

[4]School of Chemical and Pharmaceutical Engineering, Hefei Normal University, Hefei, China

ABSTRACT

Traditional methods for detecting surface defects in steel typically rely on manual visual inspection, eddy current testing, magnetic particle inspection, and machine vision. These methods often struggle to adapt to defects of varying scales and complex shapes, resulting in insufficient feature extraction, false detections, missed detections, and low detection accuracy. To address these issues, we propose an enhanced YOLOv8n model named MAM-YOLO, which aims to optimize defect detection on steel surfaces through multi-scale feature enhancement. By redesigning the convolution modules, the model improves accuracy in identifying surface defects, crazing, scratches, and patches. Furthermore, the design of a new module enhances processing capabilities for diverse surface features. A new loss function optimization is also applied to better identify crack characteristics. Experimental results on the NEU-DET dataset demonstrate that MAM-YOLO increases mAP by 5.4% and improves accuracy by 5.9% compared to the original YOLOv8n. Additionally, the model's generalization capability was verified on the KolektorSDD dataset, achieving performance superior to the YOLOv8n baseline. Therefore, the proposed MAM-YOLO model offers significant advantages in precision and robust feature extraction for identifying steel surface defects.

OPEN ACCESS

Received: 25/11/2025

Accepted: 27/01/2026

Published: 21/07/2026

DOI
10.23967/j.rimni.2026.10.76720

Keywords:
Defect detection
multi-scale convolution
attention mechanism
MPD-IoU loss function

1 Introduction

Information technology plays a pivotal role as China accelerates the development of emerging industries while advancing progress in the industrial domain. In steel production, detecting surface defects is crucial for ensuring product quality and safety. Steel is widely used across sectors such as aerospace, oil, automotive, shipping, and large equipment manufacturing, which are experiencing rapid growth in China [1]. Detecting surface defects in steel is essential for these industries since defects can directly impact final product specifications and quality, potentially leading to significant economic losses. Hence, improving the accuracy of steel surface defect detection has become imperative.

*Correspondence: Le Zou, Jian Liu (zoule@hfuu.edu.cn, liujian@hfnu.edu.cn).

Current detection methods primarily include deep learning-based technologies [2], traditional detection methods, and machine vision technologies. These methods integrate techniques such as eddy current inspections, visual inspections, and manual inspections. Manual visual inspection [3] involves examining the appearance or performance of products through vision or touch. Although straightforward, manual visual inspections are highly subjective, inefficient, and prone to environmental interference. Eddy current detection [4] identifies defects by monitoring changes in eddy current distribution and magnetic field characteristics caused by imperfections. However, its sensitivity drops for complex defects, and it requires skilled technicians, leading to higher costs. Magnetic particle inspection is routinely used in steel-dependent industries and is capable of detecting cracks, inclusions, pores, and fatigue cracks. While fast, cost-effective, and simple to operate, it is limited to magnetic materials and struggles with complex shapes. Machine vision-based surface defect detection [5] combines algorithms and visual technologies, allowing for the quick localization of defects through image input. It commonly identifies imperfections using optical devices to capture material images. While fast detection is advantageous for single-object inspections, it heavily relies on heuristic rules, often yielding relatively low accuracy. Compared to traditional methods, machine vision offers speed and adaptability, reducing human interference, which is especially valuable for large-scale automated inspections. However, it requires manual feature extraction and further improvements in detection precision. Consequently, deep learning-based detection technology holds promising potential in smart manufacturing, necessitating ongoing exploration and innovation.

Deep learning automates recognition and feature extraction, moving beyond manual efforts and bolstering the rapid evolution of object recognition technology. While U-Net [6] addresses defect segmentation through pixel-level classification, object detection methods follow a different paradigm: two-stage approaches first generate region proposals before classifying them through the network to acquire prediction outcomes, whereas single-stage techniques such as YOLO [7] and SSD [8] directly predict bounding boxes and class probabilities. Typical two-stage algorithms include R-CNN [9], Fast R-CNN [10], RetinaNet [11], Faster R-CNN [12], and Mask R-CNN [13]. These algorithms typically generate candidate regions in the first stage before classifying such regions and adjusting their bounding boxes to obtain final object detection outcomes. Additionally, methods such as transformers [14] and unsupervised learning [15] have been introduced into the field of computer vision to address related issues in object detection. YOLOv8 [16] optimized its predecessor, making significant improvements in the feature extraction network and introducing multi-scale feature fusion techniques. YOLOv11 [17], proposed by Glenn Jocher et al., first combined the C3K2 module with the Partial Spatial Attention Convolution Block (C2PSA), significantly enhancing the receptive field for small objects. YOLOv12 [18], proposed by Tian et al, fully transitions to a CNN-Transformer hybrid architecture, introducing the regional attention module A2, FlashAttention, and the Residual Efficient Layer Aggregation Network (R-ELAN). Recent studies in 2024 and 2025 have increasingly focused on balancing the trade-off between model complexity and real-time performance, with a trend towards integrating lightweight attention mechanisms and efficient convolution operators [18,19].

Many studies have explored the application of deep learning algorithms for detecting defects in steel surfaces. Ren et al. [20] proposed the ECA-SimSPPFSIoU-Yolov5 model, which enhances weight significance through effective channel attention. You and Kong [21] utilized attention mechanisms to calculate attention weights, aiding in the extraction of specific feature regions. They also improved SPPF to optimize target detection and enhance the receptive field. Guo et al. [22] proposed MSFT-YOLO, which integrates TRANS with data augmentation techniques to improve defect detection accuracy in steel. Cui and Jiao [23] designed the MCB-FAH-YOLOv8, introducing an improved Convolutional Block Attention Module (CBAM) and a replaceable four-head ASFF predictor to

minimize false positives and negatives, as well as to improve detection capabilities for small objects and densely packed targets. The aforementioned methods focus on the core scenario of steel surface defect detection and are implemented based on YOLO series algorithms. Zhang et al. [24] proposed the DDI-YOLO model, which combines the Dilated Wide Residual module (DWR) and C2f to yield C2f_DWR, enhancing feature extraction capabilities and detection performance. Zhu et al. [25] used an efficient Swin Transformer to detect defects in steel surfaces, improving the connections between feature mapping channels and addressing the issue of image information retention. He et al. [26] designed the multilevel feature fusion network (MFN), which strengthened the representation of steel defects through cross-layer feature fusion. Initially utilizing a CNN backbone for multi-level feature extraction, MFN compresses them into a unified representation and relies on an RPN to generate candidate boxes, ultimately achieving a 74.8% mAP on the ResNet-34 baseline.

Despite designs based on transformers and improvements to CNNs, existing methods for steel surface defect detection still face limitations. The size and shape of steel surface defects vary greatly, and traditional feature processing modules struggle to deal with extremely small targets. Although such approaches have somewhat improved object detection performance, there remains room for further optimization of mAP metrics. Therefore, this paper proposes MAM-YOLO, a novel detection algorithm for steel surface defects that addresses the inadequate feature extraction of the original Conv module in YOLOv8 and its inability to extract multi-scale contextual information. The contributions of this study are as follows:

(1) We design a new multi-scale convolutional module called MSConvBlock and apply a residual structure to enhance the feature representation and information flow capabilities of the model.
(2) We designed a C2f_ABlock module, which utilizes the AdditiveBlock to further enhance feature processing capabilities.
(3) We replace the original CIoU loss function in YOLOv8 with the MPD-IoU loss function, which achieves faster convergence and higher precision.

The paper is organized as follows: Section 2 reviews related work on YOLOv8 and steel defect detection. Section 3 details the proposed MAM-YOLO architecture and its core components. Section 4 presents the experimental results and analysis. Finally, Section 5 concludes the paper and discusses future work.

2 Related Work

YOLOv8, released by Ultralytics, represents a novel version based on improvements and upgrades of prior YOLO models, aiming to further improve performance and adaptability. It supports multi-platform operations from CPU to GPU and incorporates many improvements, including an anchor-free detection head, a new backbone network, and new loss algorithms. It has five versions: YOLOv8n, YOLOv8s, YOLOv8l, YOLOv8m, and YOLOv8x, differing in feature map size and model depth. The network architecture consists of the following components: input, backbone, neck, and head. Input image resizing to the required training size, including operations such as image scaling and hue adjustment, is performed at the input stage. The backbone extracts the main information and contains convolutional modules like the spatial pyramid pooling module (originating from YOLOv5), C2f, and CSPLayer_2Conv modules. By effectively merging features of different dimensions through the Feature Pyramid Network (FPN) and its path aggregation structure, the neck enhances efficiency and speed. Similar to YOLOv6 [27] and YOLOX [28], the head applies a decoupled structure, whereas previous versions like YOLOv3 [29], YOLOv4 [30], and YOLOv5 used a coupled head. Each of the

three output branches of the YOLOv8 head has two components for bounding box classification and regression.

3 Methodology

3.1 MAM-YOLO Based on Improved YOLOv8 Network Architecture

As shown in Fig. 1, this study presents an enhanced YOLOv8n model MAM-YOLO to solve problems associated with previous approaches for detecting defects on steel surfaces, such as poor detection accuracy, high rates of false and missed detections, and inadequate feature extraction capabilities. MAM-YOLO enhances detection by introducing three key modules: first, the MSConv module, which improves themulti-scale feature expression capacity of the model; second, the C2f_ABlock module, which further improves feature processing capability by adopting the AdditiveBlock module; and finally, the MPD-IoU loss function, which optimizes the CIoU loss function in YOLOv8n to enhance detection accuracy and accelerate model convergence. The following sections provide detailed introductions to these modules.

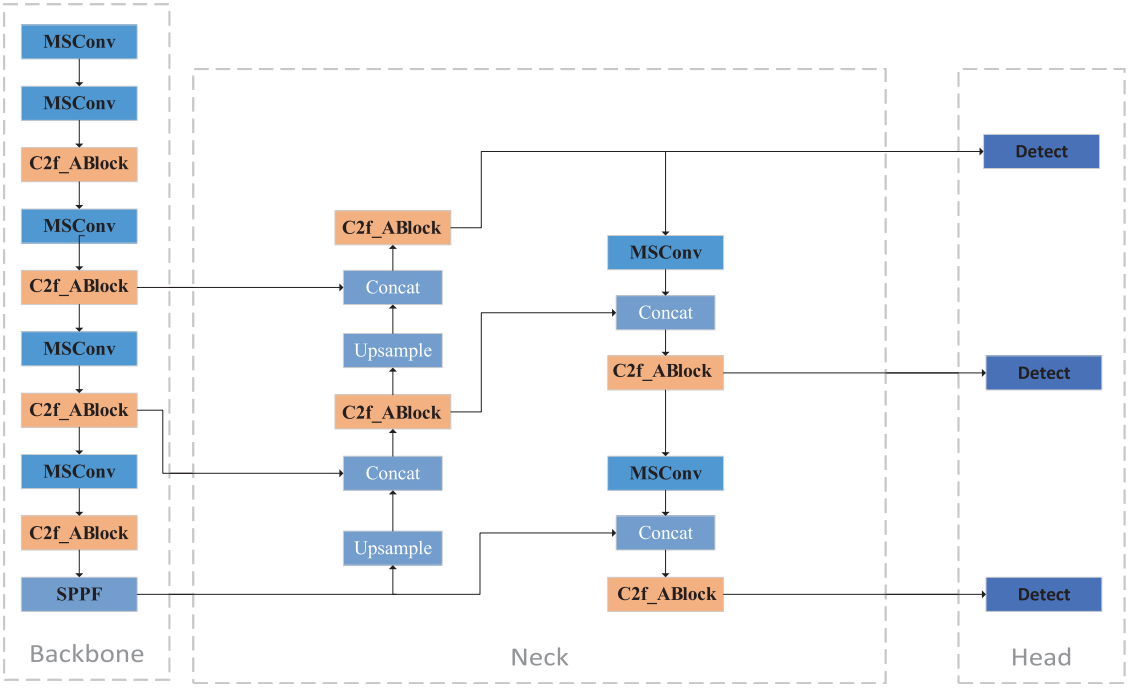


Figure 1: MAM-YOLO network architecture

3.2 MSConvBlock

The multi-scale convolutional module, MSConvBlock, is employed to improve target detection. Refer to Fig. 2 for details. First, the input features are processed by three parallel branches: a 3×3 Conv kernel, a 3×3 DWConv kernel [31], and a 5×5 DWConv kernel.

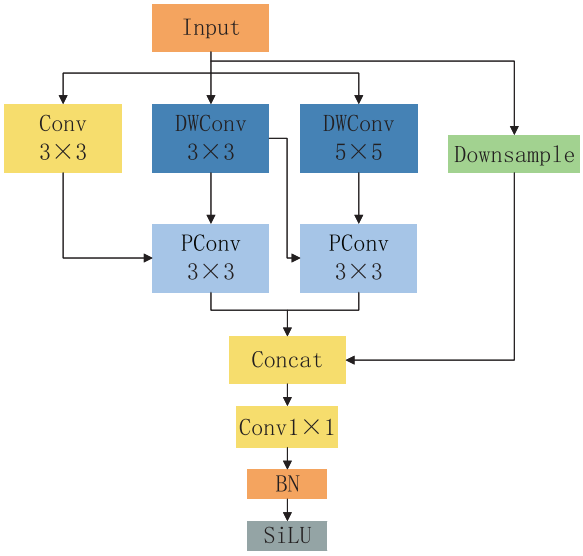


Figure 2: Architecture of the proposed MSConvBlock module

Steel surface defects often exhibit extreme scale variations, ranging from microscopic cracks to large patches. Standard single-scale convolutions struggle to simultaneously capture these diverse features. Therefore, our MSConvBlock is specifically designed to address this issue. Since the size and shape of steel surface defects vary greatly, a single-scale convolutional kernel is insufficient to effectively capture features at different scales. By incorporating 3×3 and 5×5 convolutional kernels, MSConvBlock is capable of capturing large-scale and detailed contextual information simultaneously, thereby enhancing the model's ability to represent multi-scale features. Depthwise separable convolution (DWConv) significantly decreases the computational load and number of parameters by separating convolution operations along the spatial and channel dimensions to enhance model efficiency and speed. Subsequently, the features output by the 3×3 Conv kernel and the 3×3 DWConv kernel are fed into a 3×3 PConv kernel [32] for processing. Meanwhile, the features output by the 3×3 DWConv kernel and the 5×5 DWConv kernel are processed by another 3×3 PConv kernel. The MSConvBlock also incorporates a residual structure. The features input to the residual branch undergo a downsampling operation. Finally, these features are combined and passed through a standard 1×1 convolution to adjust the number of output channels. The output is then normalized via a Batch Normalization (BN) layer [33] and activated by SiLU. As a multi-scale convolutional structure, the MSConvBlock utilizes 3×3 convolutional kernels for detail capture and 5×5 convolutional kernels for larger-scale context capture. By improving the model's capacity to represent information, this multi-scale convolution can adapt to different patterns in the input data. In addition to facilitating information flow, the residual structure aids in solving the vanishing gradient problem in deep networks.

One popular convolution procedure in neural networks is depthwise separable convolution. It consists of two parts: the depthwise convolution and pointwise convolution. In depthwise convolution, each input channel is convolved with its own filter (kernel). This implies that there will be an identical number of output channels for every input channel. The primary use case for depthwise convolution is capturing spatial information in the incoming data. The pointwise convolution involves convolving all input channels in a 1×1 operation at each location. It can be viewed as a convolution on the channel

dimension of the input data, without spatial information. It is used to linearly integrate the feature maps produced by the depthwise convolution.

The primary benefits of the depthwise separable convolution include decreasing the parameter count and computational load while simultaneously enhancing model efficiency and speed. In this convolution, every channel is convolved with a single filter, as opposed to traditional convolution where every channel is convolved with all filters. Pointwise convolution further decreases the parameter count by reducing input channel dimensionality. This structure can substantially decrease computational requirements and model size while sustaining relatively high performance, particularly for mobile devices and resource-constrained settings.

Partial Convolution (PConv) is a novel operation for improving computational speed and neural network efficiency. By cutting down unnecessary calculations and memory requests, PConv increases floating-point operations per second (FLOPS), thereby addressing the low FLOPS issue present in depthwise convolution.

PConv takes advantage of redundancy in feature maps. Usually, substantial similarity exists among different channels of a feature map. By applying regular convolution to only a subset of input channels while leaving the remaining channels unprocessed, PConv reduces memory access and computation. Moreover, while retaining the capacity of regular convolution to extract spatial features, PConv reduces redundant calculations. Compared to regular convolution, PConv has lower FLOPs.

3.3 C2f_ABlock

The YOLOv8 architecture integrates the C2f module, a cross-stage partial feature fusion mechanism that enables efficient parameter compression while maintaining channel redundancy. It compresses the number of parameters while maintaining channel redundancy, thereby achieving a balance between accuracy and efficiency in general object detection tasks. However, due to the specificity of steel surface defect detection scenarios—where there exist large differences in shape, position, and size, especially for crack defects, rolled scale defects, and pitting surface defects—the defect distribution range is extensive. The multi-scale and multi-form characteristics of these defects are vastly different from the structured features of conventional natural targets. This leads to the standard C2f's receptive field configuration and fusion weights being unable to adaptively capture the narrow-band features of tiny cracks or fully aggregate large-scale contextual information. The original C2f module is insufficient for extracting features of steel surface defects and fails to capture multi-scale contextual information. Both convolutional branches in the C2f module have a fixed 3×3 receptive field, which poorly adapts to the large scale variations of steel defects.

To enhance feature processing capability, as shown in Fig. 3, the AdditiveBlock module replaces the original Bottleneck in C2f module. The AdditiveBlock is particularly suitable for steel defect detection because it captures long-range dependencies necessary for identifying extended defects like cracks, while maintaining linear computational complexity. The AdditiveBlock module, proposed by Zhang et al. [34] as a core component of the CAS-VIT model, achieves efficient integration of contextual information through multi-dimensional information interaction and a simplified additive similarity function, avoiding the complex computations of traditional Multi-Head Self-Attention (MSA) mechanisms.

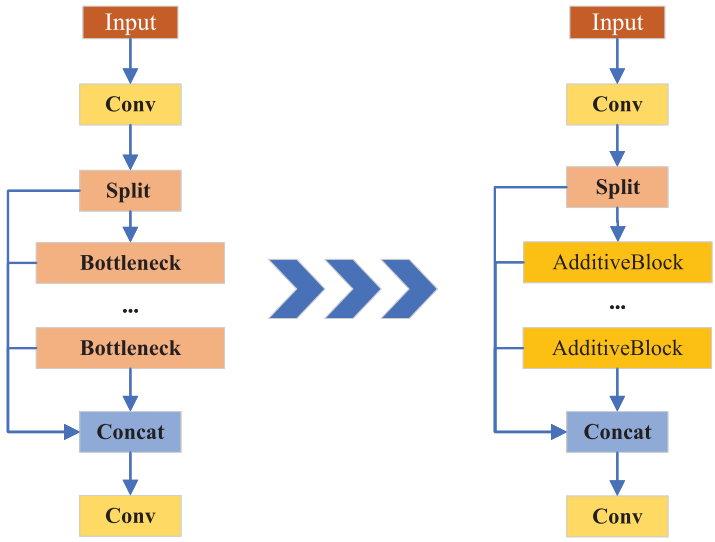


Figure 3: Comparison of (**Left**) original C2f module and (**Right**) proposed C2f_ABlock module

Fig. 4 illustrates the AdditiveBlock structure, where the left panel (a) shows the similarity calculation and the right panel (b) depicts the context mapping function. As shown in Fig. 4b, $\Gamma(\cdot) \in \mathbb{R}^N \times \mathcal{C}$ represents a linear transformation to integrate contextual information. First, local token information is aggregated through a 3×3 convolutional layer. Then, the dimension is reduced to 1 using a 1×1 convolutional layer. Subsequently, the spatial domain undergoes Sigmoid activation to generate an attention map. The specific representation is as follows in Eq. (1):

$$x_s = \text{Sigmoid}\left(D_{1\times1}\left(D_{3\times3,\,\text{ReLU,BN}}\right)\right) \odot x \tag{1}$$

For channel interaction, referring to SENet but using 1×1 convolution to integrate information between channels rather than channel reduction Eq. (2):

$$x_c = \text{Sigmoid}(D_{1\times1}(P_{1\times1}(x))) \odot x \tag{2}$$

where $P_{1\times1}$ denotes 1×1 average pooling, and $D_{1\times1}$ represents the pointwise convolution layer with groups equal to channel number (depthwise convolution). By stacking these two operations, we obtain the feature map following the interaction between the spatial and channel domains, denoted as $\Phi(x)$.

As shown in Fig. 4a, the similarity function is defined as the sum of the context scores of $Q, K \in \mathbb{R}^{N\times C}$, as shown in Eq. (3).

$$Sim(Q, K) = \Phi(Q) + \Phi(K) \tag{3}$$

The Query, Key, and Value are obtained through separate independent linear transformations, specifically in the form of $Q = W^Q x$, $K = W^K x$, $V = W^V x$, where $W^Q x$, $W^K x$, $W^V x$, are learnable projection matrices, x is the input feature, and Φ denotes the context mapping function responsible for information interaction. The advantage of such generalization is that it does not rely on manually designed contexts, thus providing the possibility for information interaction through convolution operations. These operations are combined through addition, employing a linear method that avoids highly complex matrix multiplication and effectively retains key information. Therefore,

AdditiveBlock's output may be represented as:

$$O = \Gamma(\Phi(Q) + \Phi(K)) \odot V \tag{4}$$

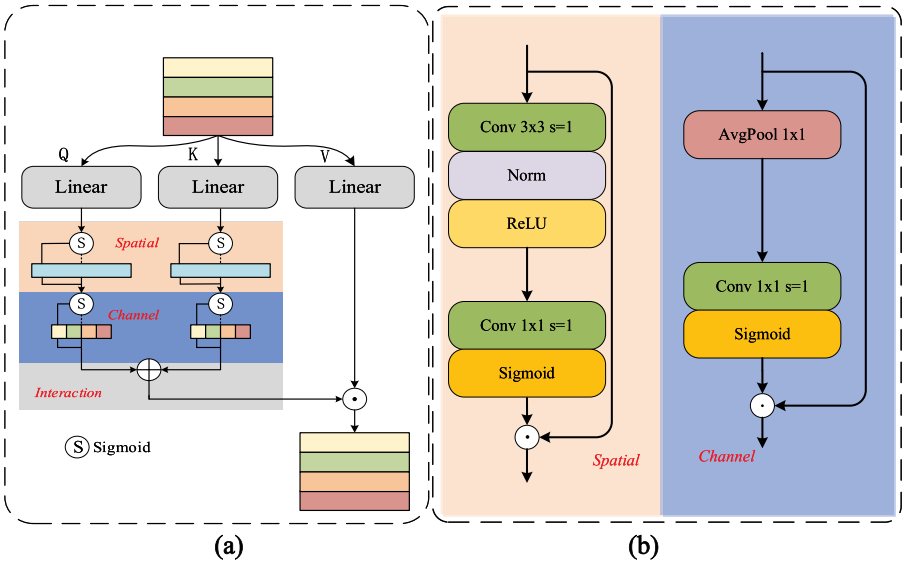


Figure 4: AdditiveBlock structure: (a) Similarity calculation; (b) Context mapping

3.4 Minimum Point Distance (MPD)-IoU Loss Function

In object detection, precise localization of bounding boxes is vital for improving detection performance. The regression loss function used in YOLOv8 is CIoU, which optimizes the localization of bounding boxes through center point distance, aspect ratio inconsistency, and overlap area. Although CIoU performs well in most cases, in practical applications involving the detection of defects in steel surfaces, the model optimization is divided. Classification loss employs Binary Cross-Entropy Loss (BCEL), while regression uses Bounding Box Regression Loss (BBRL) and Distribution Focal Loss (DFL). Eq. (5) details the entire loss function.

$$f_{\text{Loss}} = \lambda_1 f_{\text{BCEL}} + \lambda_2 f_{\text{DFL}} + \lambda_3 f_{\text{BBRL}} \tag{5}$$

where f_{Loss} is the total loss; λ_1, λ_2, and λ_3 are weight factors assigned to each loss term, respectively; and f_{BCEL}, f_{DFL}, and f_{BBRL} represent the individual loss functions for Binary Cross-Entropy Loss, Distribution Focal Loss, and Bounding Box Regression Loss, respectively. In classification, BCEL measures the discrepancy between actual labels and expected class probabilities. Distribution Focal Loss (DFL) is applied to regression, focusing on the distribution of predicted bounding box values and prioritizing more difficult samples. Finally, BBRL aims to minimize the difference between true bounding box coordinates and predicted ones. Bounding Box Regression Loss (BBRL) is crucial in instance segmentation and object detection, as it is responsible for accurately locating targets. However, traditional bounding box regression loss functions have limitations, especially if the width and height vary from the ground-truth box despite the predicted bounding box maintaining the same aspect ratio, failing to optimize effectively. To resolve this issue, unlike YOLOv8's CIoU [35] loss function, we employ the Minimum Point Distance Intersection over Union (MPD-IoU) [36] loss function.

The MPD-IoU loss simplifies calculation by minimizing the distance between the top-left and bottom-right points of the predicted and ground truth boxes. The mathematical definition is given as follows:

$$d_1^2 = (x_1^{prd} - x_1^{gt})^2 + (y_1^{prd} - y_1^{gt})^2 \tag{6}$$

$$d_2^2 = (x_2^{prd} - x_2^{gt})^2 + (y_2^{prd} - y_2^{gt})^2 \tag{7}$$

$$w^2 + h^2 = w_{img}^2 + h_{img}^2 \tag{8}$$

$$\text{MPD-IoU} = \text{IoU} - \frac{d_1^2}{w^2 + h^2} - \frac{d_2^2}{w^2 + h^2} \tag{9}$$

$$L_{\text{MPD-IoU}} = 1 - \text{MPD-IoU} \tag{10}$$

$$f_{\text{Loss1}} = \lambda_1 f_{\text{BCEL}} + \lambda_2 f_{\text{DFL}} + \lambda_3 \mathcal{L}_{\text{MPD-IoU}} \tag{11}$$

where (x_1^{prd}, y_1^{prd}) and (x_2^{prd}, y_2^{prd}) denote the coordinates of the top-left and bottom-right corners of the predicted bounding box, respectively. Similarly, (x_1^{gt}, y_1^{gt}) and (x_2^{gt}, y_2^{gt}) correspond to the ground truth box. w_{img} and h_{img} represent the width and height of the input image.

The overall loss function, denoted as f_{Loss1}, is weighted (λ_1, λ_2, λ_3) to balance the Binary Cross-Entropy Loss f_{BCEL}, the distance-based Focal Loss f_{DFL}, and the Bounding Box Regression Loss (BBRL). Using just the coordinates of the upper-left and lower-right locations, we can derive all the variables that go into current loss functions, such as the distance between centers, width and height deviations, and non-overlapping area, thus underscoring the expressiveness of the MPD-IoU measure. Therefore, the MPD-IoU metric we propose not only takes all these factors into account comprehensively but also further simplifies calculation. Remarkably, if the predicted bounding box's aspect ratio and the ground-truth bounding box's aspect ratio are the same, the MPD-IoU measure can ensure bounding box regression accuracy. Unlike the situation where the predicted bounding box exceeds the ground-truth bounding box, the value becomes lower when the ground-truth bounding box entirely contains the predicted one. Such an attribute improves bounding box regression accuracy and eventually reduces redundant predictions.

4 Results

4.1 Implementation Details

All experiments were conducted on a system equipped with an NVIDIA GeForce RTX 3090 GPU (24 GB). The software environment included PyTorch 1.12.1 and Python 3.9. The input images were resized to 640 × 640. The proposed MAM-YOLO model was trained for 200 epochs using the SGD optimizer with an initial learning rate of 0.01, momentum of 0.937, and weight decay of 0.0005. The batch size was set to 16. Standard data augmentation techniques, including mosaic and mixup, were applied during training to enhance model robustness.

4.2 Image Dataset

This study employs the NEU-DET dataset [37], provided by Northeastern University, which contains images of steel surface defects. This dataset consists of 1800 images. Fig. 5 shows six defect types: inclusions (In), rolled-in scale (Cr), patches (Pa), pitting surface (Ps), scratches (Sc), and rolled scale (Rs). Each type is represented by 300 images of size 200 × 200. For training and validation

purposes, the dataset was randomly divided into proportions of 80%, 10%, and 10%, with 1440 images allocated for training, 180 for testing, and another 180 for validation.

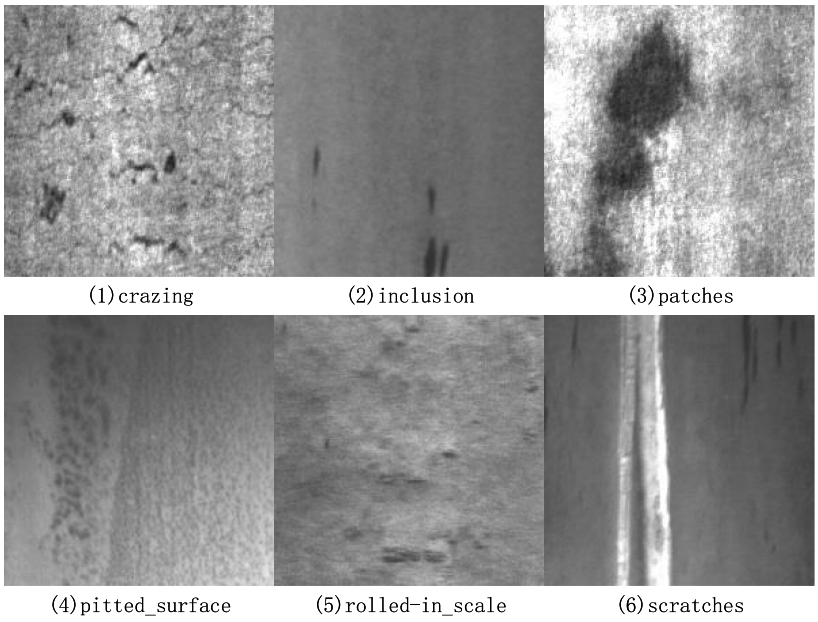


Figure 5: Schematic diagram of six defect samples

To further verify the generalization capability of the proposed MAM-YOLO model on different industrial scenarios, we additionally employed the KolektorSDD dataset [38]. This dataset is constructed from images of electrical commutators containing microscopic surface cracks and fractures. Since the dataset focuses on a single type of surface discontinuity, we unified the label as "Defect". Consistent with the NEU-DET experiment, we randomly divided the dataset into a training set and a validation set in an 8:2 ratio to evaluate detection performance.

4.3 Experimental Metrics

This paper's evaluation metrics include the number of parameters, precision-recall curve (P-R), mean average precision (mAP), frames per second (FPS), and Average Precision (AP) for each defect type. Among these, AP is determined according to recall (R) and precision (P). The P-R curve is plotted via precision and recall obtained from the experiments, and the area under this curve denotes mAP. In this study, mAP refers to mAP@0.5, and Precision/Recall are calculated as the macro-average across all classes. The relevant calculation formulas are as follows:

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

where True Positives (TP) is the number of samples predicted as positive that are actually positive, i.e., correctly identified positive samples. False Positives(FP) is the number of samples predicted as positive that are actually negative, i.e., falsely reported positive samples.

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

where False Negatives (FN) represents missed positive samples or samples predicted as negative but are actually positive.

$$AP = \int_0^1 p(r)\, dr \tag{14}$$

where Average Precision (AP) is the precision for each type of detection.

$$\text{mAP} = \frac{1}{n} \sum_{i=1}^{n} \text{AP}_i \tag{15}$$

where mean Average Precision (mAP) represents the average of all class values, where n is the number of classes.

4.4 Analysis of Experimental Results

4.4.1 Parameter Sensitivity Analysis

To thoroughly validate the effectiveness of the proposed improvements in MAM-YOLO, we conducted a comprehensive set of ablation studies on the NEU-DET dataset. These experiments aim to evaluate two critical aspects: (1) the sensitivity of the model to hyperparameter configurations, specifically the kernel sizes in the MSConvBlock, and (2) the compatibility and individual contributions of each proposed module (MSConvBlock, AdditiveBlock, and MPD-IoU) under various combination scenarios.

The MSConvBlock is designed to capture multi-scale features through parallel convolution branches. The choice of kernel size is pivotal for balancing the receptive field and computational efficiency. We compared three configurations: utilizing only small kernels (3×3 & 3×3), the proposed hybrid combination (3×3 & 5×5), and a larger kernel combination (3×3 & 7×7). The results are presented in Table 1.

Table 1: Impact of different kernel size configurations in MSConvBlock on detection performance

Kernel configuration	Params/M	GFLOPs/G	mAP50%	FPS
3×3 & 3×3	**2.88**	**7.8**	77.2	**142**
3×3 & 5×5 (Ours)	2.95	8.0	78.9	138
3×3 & 7×7	3.12	8.5	**79.0**	125

Note: Bold values indicate the best performance.

As observed in Table 1, employing only small kernels (3×3) yields the lowest parameter count but limits the receptive field, resulting in a suboptimal mAP of 77.2%. Conversely, introducing a larger 7×7 kernel marginally increases the mAP to 79.0% but incurs a significant penalty in computational cost (GFLOPs rise to 8.5) and reduces inference speed (FPS drops to 125). The proposed configuration (3×3 & 5×5) achieves the optimal balance between detection accuracy (78.9%) and inference efficiency, justifying its selection for the final MAM-YOLO architecture.

4.4.2 Ablation Studies

To verify the independent contribution of each proposed component and rule out potential dependencies on the sequence of module integration, we conducted a comprehensive combination

analysis. Instead of a sequential addition study, we tested the baseline YOLOv8n with individual modules and different pairwise combinations. The results are summarized in Table 2.

Table 2: Comprehensive analysis of module combinations and their impact on detection performance

Modules			Metrics		
MSConvBlock	C2f_ABlock	MPD-IoU	Precision (%)	Recall (%)	mAP50 (%)
–	–	–	64.3	69.9	73.5
✓	–	–	66.8	72.5	75.8
–	✓	–	65.9	71.8	75.2
–	–	✓	64.9	70.5	74.3
✓	✓	–	68.5	74.2	78.1
✓	–	✓	69.4	72.7	76.6
–	✓	✓	66.5	72.1	75.9
✓	✓	✓	**69.7**	**75.8**	**78.9**

Note: Bold values indicate the best performance.

As shown in Table 2, the integration of any single module yields a performance improvement over the baseline YOLOv8n, confirming their individual effectiveness. Specifically, the MSConvBlock provides the most significant standalone boost (+2.3% mAP), validating its critical role in multi-scale feature extraction. Furthermore, pairwise combinations (e.g., MSConvBlock + C2f_ABlock) consistently outperform single-module configurations, achieving an mAP of 78.1%. This trend indicates that the modules function orthogonally and complement each other. Most importantly, the consistent performance across different combinations demonstrates that the improvements are robust and not sensitive to the specific order of module integration.

4.4.3 Comparative Experiments

To validate the effectiveness of the MAM-YOLO algorithm, this study uses YOLOv8n as a benchmark and tests both on the NEU-DET dataset. From Table 3, the results indicate that MAM-YOLO improved the mAP by 5.4% compared to YOLOv8n. Specifically, MAM-YOLO improved the mAP by 23.6% for crazing (Cr) defects and by 6.9% for rolled-in_scale (Rs) defects. Additionally, the use of MSConvBlock and the introduction of AdditiveBlock effectively expanded the network's receptive field and enhanced its feature extraction capabilities. This improvement not only enhances the model's capture of multi-scale features but also further optimizes feature processing efficiency. Furthermore, when detecting scratches (Sc) defects, the accuracy of MAM-YOLO increased from 69.4% to 81.7%, and for patches (Pa) defects, it increased from 71.0% to 81.4%, representing improvements of 12.3% and 10.4%, respectively. The outcomes fully exhibit the significant advantages of MAM-YOLO in enhancing detection accuracy and precision, reflecting the effectiveness of the proposed methods. To ensure fairness, all comparison models were retrained under identical experimental settings.

Table 3: The detection performance of MAM-YOLO using the NEU-DET datasets

Dataset	Methods	Defect Type	P%	R%	mAP%
NEU-DET	*YOLOv8*	*Cr*	*54.6*	*31.6*	*39.2*
		In	70.0	79.8	86.2
		Pa	71.0	88.1	85.2
		Ps	70.4	73.5	81.6
		Rs	50.8	67.8	57.1
		Sc	69.4	88.9	92.0
	MAM-YOLO	Cr	65.5	47.4	52.8(↑ 23.6)
		In	78.7	81.0	89.3(↑ 3.1)
		Pa	81.4	88.1	88.7(↑ 3.5)
		Ps	77.4	79.4	85.2(↑ 3.6)
		Rs	58.2	55.9	64.0(↑ 6.9)
		Sc	81.7	91.8	93.5(↑ 1.5)

Furthermore, to demonstrate the robustness and generalization of MAM-YOLO, we conducted a comparative experiment on the KolektorSDD dataset. As shown in Table 4, the proposed method achieved an mAP of 77.8%, outperforming the baseline YOLOv8n by 1.1%. Specifically, MAM-YOLO shows significant advantages in Precision (87.5%) and Recall (76.9%), which are 13.5% and 7.7% higher than YOLOv8n, respectively. These results indicate that the multi-scale feature enhancement modules (MSConvBlock) and the attention mechanism (AdditiveBlock) designed in this paper are not only effective for steel surfaces but also possess strong feature extraction capabilities for microscopic defects on other industrial components, even with a limited number of training samples.

Table 4: The detection performance comparison on the KolektorSDD dataset

Dataset	Methods	Defect type	P%	R%	mAP%
KolektorSDD	YOLOv8n	Defect	74.0	69.2	76.7
	MAM-YOLO	Defect	87.5	76.9	77.8(↑ 1.1)

According to the comprehensive comparison in Table 5, the proposed MAM-YOLO achieves the highest mAP of 78.9%, surpassing not only the baseline YOLOv8n but also recent state-of-the-art models, including YOLOv11, YOLOv12, and YOLOv13. Notably, while the latest iterations such as YOLOv12 and YOLOv13 have significantly increased model complexity (exceeding 43M parameters), MAM-YOLO maintains a lightweight architecture with only 2.95M parameters. This indicates that our method provides a superior balance between detection accuracy and computational efficiency, achieving higher precision with significantly fewer processing resources than these larger models.

Table 5: Comparative experimental results on NEU-DET dataset

Model	P%	R%	mAP%	Params/M	GFLOPs/G
SSD	75.50	61.12	68.43	21.5	31.5
YOLOv3-tiny	57.9	72	67.1	33.6	18.9
YOLOv5n	69.7	70.7	75.5	4.27	7.1
YOLOv6n	64	74.1	74.6	4.2	11.8
YOLOv7-tiny	70.5	61.2	66.8	6.03	13.2
YOLOv8n	64.3	69.9	73.5	3.01	8.1
RT-DETR	69.1	69.2	72.9	41	100.6
YOLOv11	71.1	72.8	76.4	6.3	6.3
YOLOv12	71.5	69.0	75.7	43.7	5.8
YOLOv13	75.8	72.1	76.3	43.9	6.2
MAM-YOLO (Ours)	69.7	75.8	78.9	2.95	8.0

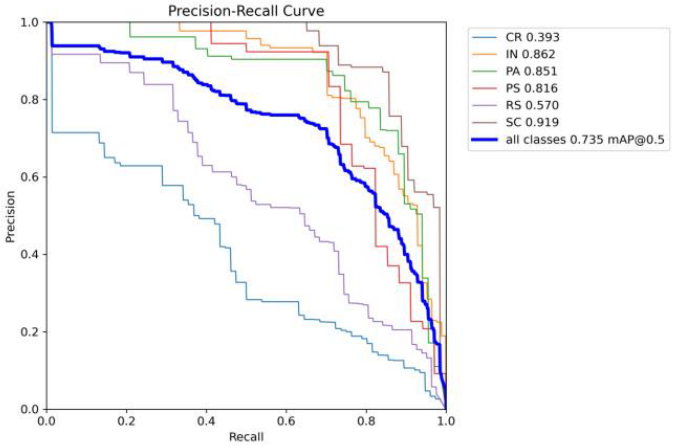
Furthermore, the consistent improvement across multiple metrics and datasets is supported by a Wilcoxon signed-rank test analysis ($p < 0.05$). As detailed in Table 6, we performed a 10-fold cross-validation comparison. The p-value obtained (0.002) is well below the significance level of $\alpha = 0.05$, confirming that the performance gains of MAM-YOLO are statistically significant and not due to random chance. Consequently, the proposed method demonstrates strong robustness and is highly suitable for real-time industrial applications.

Table 6: Wilcoxon signed-rank test analysis comparing mAP scores across 10 validation folds

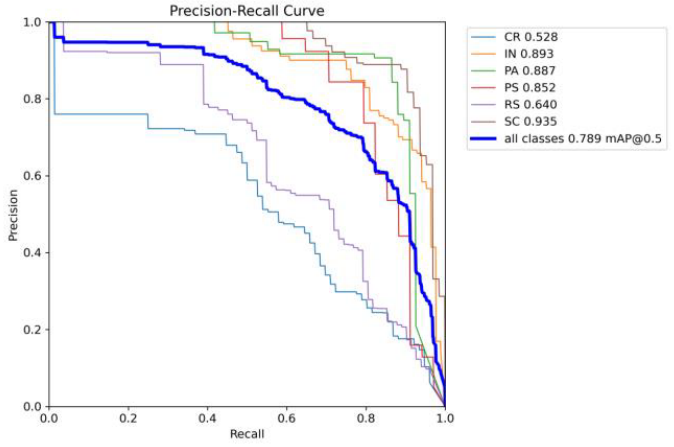
Fold Index	YOLOv8n (mAP)	MAM-YOLO (mAP)	Diff	Rank
1	73.1	78.2	+5.1	4
2	72.8	79.1	+6.3	9
3	73.9	78.5	+4.6	2
4	74.2	78.8	+4.6	3
5	73.4	79.3	+5.9	7
6	73.0	78.4	+5.4	5
7	73.6	79.0	+5.4	6
8	74.0	78.7	+4.7	10
9	73.2	79.5	+6.3	8
10	73.8	79.4	+5.6	1
Mean	**73.5**	**78.9**	–	–
***p*-value**	–	–	–	**0.002**

To gain a deeper understanding of the detection capabilities of our proposed MAM-YOLO and the baseline model YOLOv8n, the P-R (Precision-Recall) curves of the two methods are shown in Fig. 6a,b. The area enclosed by the curve of our proposed MAM-YOLO is larger than that enclosed by YOLOv8n. The overall mAP on the NEU-DET dataset is 78.9%, which is a significant

improvement of 5.4 percentage points over YOLOv8n. The efficacy of the network enhancements proposed in this research was assessed using comparative tests between the improved YOLOv8n algorithm model and other models. This paper selected the classic SSD, YOLOv3-tiny, YOLOv5n (official implementation), YOLOv6, YOLOv7-tiny [39], YOLOv12n, YOLOv8n (baseline), RT-DETR [40] and the more recent YOLOv13 [41] for comparative studies. These experiments were conducted using the same environment, dataset, and equipment. Finally, the experimental detection results of several different algorithms were compared. As can be observed in Fig. 7, other algorithm models are prone to incorrect detections and missed detections, while the algorithm model proposed in this paper demonstrates better detection performance.



(a)



(b)

Figure 6: Comparison of different loss functions: (a) CIoU P-R curve; (b) MPD-IoU P-R curve

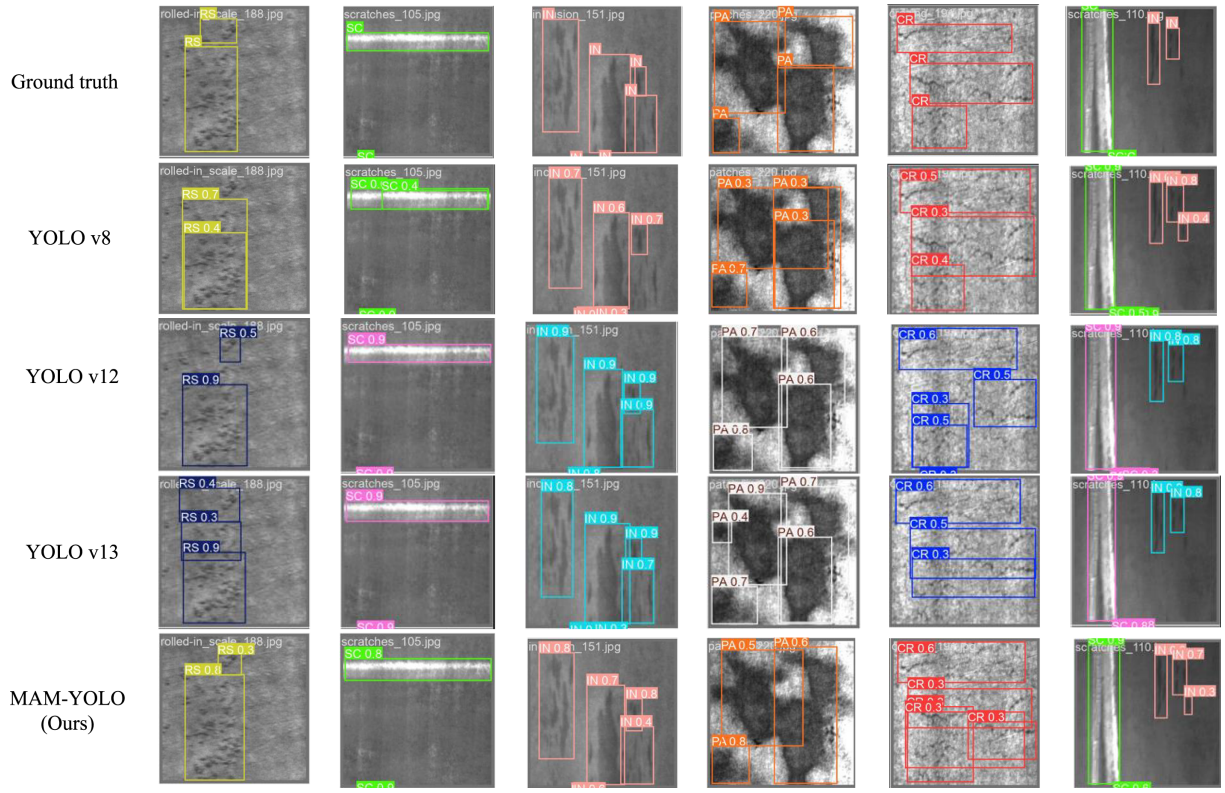


Figure 7: Qualitative results comparing MAM-YOLO with baseline models

5 Conclusions

This study addresses the issues of insufficient feature extraction capability, false detections, missed detections, and low detection accuracy in steel surface defect detection in actual production with the proposed MAM-YOLO algorithm. By introducing the MSConvBlock module, C2f_ABlock module, and MPD-IoU loss function, the algorithm significantly enhances the model's feature extraction capability, resolves the vanishing gradient problem, and accelerates the training process and detection accuracy. The MSConvBlock module strengthens the feature expression ability for defects of varying sizes and shapes through multi-scale convolution operations, while the C2f_ABlock module further optimizes feature processing efficiency. The MPD-IoU loss function improves bounding box regression accuracy and speeds up model convergence. From a practical standpoint, MAM-YOLO achieves improvements in mAP and accuracy, reaching 78.9% and 69.7%, respectively, while maintaining high computational efficiency, making it suitable for real-time industrial scenarios.

However, this study also has limitations, such as the potential impact of dataset dependency on model generalization and room for further optimization of the loss function. Although we verified performance on KolektorSDD, cross-domain adaptation remains a challenge. Future research directions include expanding the dataset to enhance generalization capability, exploring better loss functions to boost model performance, and conducting extensive testing in real industrial environments to optimize real-time deployment outcomes. Furthermore, we plan to apply the MAM-YOLO algorithm to other types of industrial defect detection tasks to verify its versatility and effectiveness. We believe that

through these future endeavors, the MAM-YOLO algorithm will bring more value and advancement to the field of industrial defect detection.

Acknowledgement: Not applicable.

Funding Statement: This work was supported by Key Project of Natural Science Research of University in Anhui Provence (No. 2025AHGXZK30903).

Author Contributions: All authors contributed to the study conception. Yu Wang: investigation, data curation, formal analysis. Chao Mao: methodology,writing-original draft. Fengling Jiang: supervision, validation. Le Zou: conceptualisation, supervision. Jian Liu: methodology, supervision. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: The data underlying this article will be shared on reasonable request to the corresponding author.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Wen X, Shan J, He Y, Song K. Steel surface defect recognition: a survey. Coatings. 2022;13(1):17. doi:10.3390/coatings13010017.
2. LeCun Y, Bengio Y, Hinton G. Deep learning. Nature. 2015;521(7553):436–44 doi: 10.1038/nature14539.
3. Yasuda YD, Cappabianco FA, Martins LEG, Gripp JA. Aircraft visual inspection: a systematic literature review. Comput Ind. 2022;141:103695.
4. Mohseni E, Habibzadeh Boukani H, Ramos França D, Viens M. A study of the automated eddy current detection of cracks in steel plates. J Nondestruct Eval. 2020;39(1):6. doi:10.1007/s10921-019-0647-9.
5. Tang B, Chen L, Sun W, Zk Lin. Review of surface defect detection of steel products based on machine vision. IET Image Process. 2023;17(2):303–22 doi: 10.1049/ipr2.12647.
6. Hussain M. YOLOv1 to v8: unveiling each variant—a comprehensive review of YOLO. IEEE Access. 2024;12:42816–33.
7. Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu CY, et al. SSD: single shot multibox detector. In: Proceedings of the Computer Vision—ECCV 2016: 14th European Conference; 2016 Oct 11–14; Amsterdam, The Netherlands. p. 21–37.
8. Ronneberger O, Fischer P, Brox T. U-net: convolutional networks for biomedical image segmentation. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. Berlin/Heidelberg, Germany: Springer; 2015. p. 234–41.
9. Xie X, Cheng G, Wang J, Yao X, Han J. Oriented R-CNN for object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision; 2021 Oct 11–17; Montreal, BC, Canada. p. 3520–9.
10. Ren S, He K, Girshick R, Sun J. Faster R-CNN: towards real-time object detection with region proposal networks. IEEE Trans Pattern Anal Mach Intell. 2016;39(6):1137–49 doi: 10.1109/tpami.2016.2577031.
11. Cheng X, Yu J. RetinaNet with difference channel attention and adaptively spatial feature fusion for steel surface defect detection. IEEE Trans Instrum Meas. 2020;70:1–11 doi: 10.1109/tim.2020.3040485.
12. Girshick R. Fast R-CNN. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV); 2015 Dec 7–13; Santiago, Chile. p. 1440–8.

13. He K, Gkioxari G, Dollár P, Girshick R. Mask R-CNN. In: Proceedings of the IEEE International Conference on Computer Vision; 2017 Oct 22–29; Venice, Italy. p. 2961–9.
14. Wolf T, Debut L, Sanh V, Chaumond J, Delangue C, Moi A, et al. Transformers: state-of-the-art natural language processing. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations; 2020 Nov 16–20; Online. p. 38–45.
15. Barlow HB. Unsupervised learning. Neural Comput. 1989;1(3):295–311 doi: 10.7551/mitpress/7011.003.0002.
16. Varghese R, Sambath M. YOLOv8: a novel object detection algorithm with enhanced performance and robustness. In: Proceedings of the 2024 International Conference on Advances in Data Engineering and Intelligent Computing Systems (ADICS); 2024 Apr 18–19; Chennai, India. p. 1–6.
17. Khanam R, Hussain M. YOLOv11: an overview of the key architectural enhancements. arXiv:241017725. 2024.
18. Tian Y, Ye Q, Doermann D. YOLOv12: attention-centric real-time object detectors. arXiv:2502.12524. 2025.
19. Hozhabr SH, Giorgi R. A survey on real-time object detection on FPGAs. IEEE Access. 2025;13:38195–238 doi: 10.1109/access.2025.3544515.
20. Ren F, Fei J, Li H. Fusion lightweight steel surface defect detection algorithm based on improved deep learning. Int J Adv Comput Sci Appl. 2024;15(5):377. doi:10.14569/ijacsa.2024.0150537.
21. You C, Kong H. Improved steel surface defect detection algorithm based on YOLOv8. IEEE Access. 2024;12:99570–7 doi: 10.1109/access.2024.3429555.
22. Guo Z, Wang C, Yang G, Huang Z, Li G. Msft-YOLO: improved yolov5 based on transformer for detecting defects of steel surface. Sensors. 2022;22(9):3467.
23. Cui K, Jiao J. Steel surface defect detection algorithm based on MCB-FAH-YOLOv8. J Graph. 2024;45(1):112. doi:10.1109/icicml63543.2024.10957993.
24. Zhang T, Pan P, Zhang J, Zhang X. Steel surface defect detection algorithm based on improved YOLOv8n. Appl Sci. 2024;14(12):5325. doi:10.1117/12.2670493.
25. Zhu W, Zhang H, Zhang C, Zhu X, Guan Z, Jia J. Surface defect detection and classification of steel using an efficient Swin Transformer. Adv Eng Inform. 2023;57:102061. doi:10.1016/j.aei.2023.102061.
26. He Y, Song K, Meng Q, Yan Y. An end-to-end steel surface defect detection approach via fusing multiple hierarchical features. IEEE Trans Instrum Meas. 2019;69(4):1493–504 doi: 10.1109/tim.2019.2915404.
27. Li C, Li L, Jiang H, Weng K, Geng Y, Li L, et al. YOLOv6: a single-stage object detection framework for industrial applications. arXiv:2209.02976. 2022.
28. Ge Z, Liu S, Wang F, Li Z, Sun J. YOLOx: exceeding yolo series in 2021. arXiv:2107.08430. 2021.
29. Redmon J, Farhadi A. YOLOv3: an incremental improvement. arXiv:1804.02767. 2018.
30. Bochkovskiy A, Wang CY, Liao HYM. YOLOv4: optimal speed and accuracy of object detection. arXiv:2004.10934. 2020.
31. Howard AG, Zhu M, Chen B, Kalenichenko D, Wang W, Weyand T, et al. Mobilenets: efficient convolutional neural networks for mobile vision applications. arXiv:1704.04861. 2017.
32. Liu G, Reda FA, Shih KJ, Wang TC, Tao A, Catanzaro B. Image inpainting for irregular holes using partial convolutions. In: Proceedings of the European Conference on Computer Vision (ECCV); 2018 Sep 8–14; Munich, Germany. p. 85–100.
33. Ioffe S, Szegedy C. Batch normalization: accelerating deep network training by reducing internal covariate shift. Int Conf Mach Learn. Proc Mach Learn Res. 2015;37:448–56.
34. Zhang T, Li L, Zhou Y, Liu W, Qian C, Hwang JN, et al. CAS-ViT: convolutional additive self-attention vision transformers for efficient mobile applications. arXiv:2408.03703. 2024.
35. Zheng Z, Wang P, Liu W, Li J, Ye R, Ren D. Distance-IoU loss: faster and better learning for bounding box regression. Proc AAAI Conf Artif Intell. 2020;34:12993–3000 doi: 10.1609/aaai.v34i07.6999.

36. Ma S, Xu Y. Mpdiou: a loss for efficient and accurate bounding box regression. arXiv:2307.07662. 2023.
37. Lv X, Duan F, Jiang Ji, Fu X, Gan L. Deep metallic surface defect detection: the new benchmark and detection network. Sensors. 2020;20(6):1562. doi:10.3390/s20061562.
38. Tabernik D, Šela S, Skvarč J, Skočaj D. Segmentation-based deep-learning approach for surface-defect detection. J Intell Manuf. 2020;31(3):759–76 doi: 10.1007/s10845-019-01476-x.
39. Wang CY, Bochkovskiy A, Liao HYM. YOLOv7: trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2023); 2023 Jun 17–24; Vancouver, BC, Canada. p. 7464–75.
40. Zhao Y, Lv W, Xu S, Wei J, Wang G, Dang Q, et al. DETRs beat YOLOs on real-time object detection. In: Proceedings of the 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2024 Jun 17–21; Seattle, WA, USA. p. 16965–74.
41. Lei M, Li S, Wu Y, Hu H, Zhou Y, Zheng X, et al. YOLOv13: real-time object detection with hypergraph-enhanced adaptive visual perception. arXiv:2506.17733. 2025.