Computers, Materials & Continua

Tech Science Press

https://doi.org/10.32604/cmc.2026.088008

ARTICLE

MLQ-YOLO: A Multi-Scale Gated Feature Fusion and Local Structure-Aware Attention Network for Steel Surface Defect Detection

Changzhi Lv[1], Jiaxuan Guo[1], Yu Chen[1], Di Fan[2,*], Zonglei Mou[1] and Shaoshao Liu[3]

[1]College of Electrical Engineering and Automation, Shandong University of Science and Technology, Qingdao, China
[2]College of Electronic and Information Engineering, Shandong University of Science and Technology, Qingdao, China
[3]Research and Development Department, Shandong Lingyu Information Technology Co., Ltd., Jining, China

*Corresponding Author: Di Fan. Email: fandi93@126.com

Received: 26 June 2026; Accepted: 09 September 2026; Published: 22 October 2026

ABSTRACT: Accurate detection of steel surface defects remains challenging because defect patterns often exhibit small spatial extent, elongated morphology, weak visual contrast, and strong interference from complex surface textures. This study presents Multi-Scale, Local-Structure, and Quality-Aware YOLO (MLQ-YOLO), a lightweight detector developed from YOLO11n to improve feature representation and prediction reliability for steel surface inspection. The proposed Multi-Scale Feature Gating Mapping (MFGM) employs asymmetric feature decomposition, cross-guided spatial–channel modulation, and adaptive gated fusion to strengthen complementary defect representations. A Local Structural Perception Attention (LSPA) module further models localized structural relationships through resolution-adaptive overlapping-window attention and lightweight convolutional refinement. In addition, a parameter-free Quality-Aware Classification Head (QAC-Head) uses regression-response information to recalibrate classification logits, introducing one-way interaction between the regression and classification branches without an auxiliary quality-prediction branch. Offline augmentation is applied to the NEU-DET training subset to increase sample diversity. On the augmented NEU-DET dataset, MLQ-YOLO achieves a mean Average Precision (mAP)@0.5 of 85.6%, exceeding YOLO11n by 5.2 percentage points. On GC10-DET, it obtains a mAP@0.5 of 74.2%. The final model contains 2.61 M parameters and requires 6.7 GFLOPs, indicating that the proposed modifications improve detection accuracy while retaining a compact computational profile.

KEYWORDS: Deep learning; YOLO11n; steel surface defects; object detection

1 Introduction

Steel is one of the most important industrial materials and is extensively used in a wide range of fields, including mechanical manufacturing, construction, automotive engineering, and aerospace. The surface quality of steel has a direct impact on the reliability and service life of steel products. Surface defects, such as crazing, inclusions, pits, and rolled-in scale, can induce stress concentration, leading to structural degradation or even failure. In addition, these defects may accelerate corrosion during service, thereby increasing potential risks to engineering safety. Therefore, accurate and efficient automated detection of steel surface defects is essential for ensuring product quality and enhancing industrial production efficiency [1].

Traditional steel surface defect detection methods primarily rely on physical inspection techniques and manual expertise. While these methods can be effective under controlled conditions, they often suffer from low detection efficiency, limited robustness, and poor adaptability to complex industrial environments.

Copyright © 2026 The Authors. Published by Tech Science Press. This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

With the rapid advancement of deep learning, object detection methods have shown superior performance in automatically identifying and localizing defects from image data. Consequently, deep learning-based approaches have become a key research direction in steel surface defect detection.

Deep learning-based object detection methods are generally classified into two-stage and one-stage architectures. Two-stage detectors, exemplified by Faster R-CNN [2], generally achieve high localization accuracy, particularly for small objects and complex scenes. However, their sequential region proposal and refinement process introduces considerable computational overhead, resulting in relatively low inference efficiency. In contrast, one-stage detectors, such as the YOLO series [3–6] and SSD [7], directly predict object categories and bounding boxes from feature maps, thereby significantly improving inference efficiency and making them well suited for real-time applications. Recent YOLO-based studies in aerial and remote sensing imagery and wood surface inspection have explored multi-scale feature representation, attention mechanisms, and lightweight architectural design [8–10]. These studies offer methodological insights into small-target representation and background discrimination. In steel surface defect detection, related work has investigated multi-scale feature sensing, adaptive and attention-guided feature fusion, and lightweight backbones to improve detection accuracy and computational efficiency [11–14]. YOLO11n provides a favorable balance between detection accuracy and computational complexity and is therefore adopted as the baseline in this work. However, steel surface defect detection still faces several challenges. Small and elongated defects are prone to fine-grained information loss during feature extraction, while complex surface textures can interfere with the representation of localized defect structures [15]. Moreover, the largely independent classification and regression branches limit task interaction, which may affect prediction reliability for defects with ambiguous boundaries. The limited scale and diversity of available steel-defect datasets further constrain effective representation learning.

Motivated by the above challenges, this work develops Multi-Scale, Local-Structure, and Quality-Aware YOLO (MLQ-YOLO) based on YOLO11n. The main contributions are summarized as follows.

(1) A Multi-Scale Feature Gating Mapping (MFGM) module is introduced to enhance intra-level feature representation. Through asymmetric feature decomposition, cross-branch spatial–channel guidance, and adaptive gated fusion, MFGM improves the interaction of complementary defect features while limiting redundant background responses.
(2) A Local Structural Perception Attention (LSPA) module is developed to strengthen local structural modeling. It combines resolution-adaptive overlapping-window self-attention with lightweight convolutional refinement to better preserve localized and structurally continuous defect patterns.
(3) A Quality-Aware Classification Head (QAC-Head) is proposed to establish lightweight regression-to-classification interaction. It reuses existing regression responses to recalibrate classification logits without introducing an auxiliary quality-prediction branch or additional learnable parameters.

Experiments on NEU-DET and GC10-DET, together with controlled ablation studies, are conducted to evaluate the effectiveness and complementary roles of the proposed components.

2 Related Work

2.1 Traditional Methods for Steel Surface Defect Detection

Physical non-destructive testing (NDT) methods are used to detect surface and subsurface defects in steel components. Representative techniques include infrared thermography, eddy current testing, and magnetic flux leakage (MFL) testing [16]. Usamentiaga et al. [17] investigated the feasibility of active infrared thermography for detecting rolled-in material defects in manufactured rails, examining the effects of heating power and duration on defect detectability. Qiu et al. [18] utilized the polarity characteristics of

signals generated by a positively magnetized eddy current probe to identify surface cracks in steel plates through permeability perturbation analysis. To mitigate the skin-effect limitation and enable backside defect detection, Wang et al. [19] proposed a multi-frequency alternating-current magnetic flux leakage method that combines low- and high-frequency excitation signals, thereby balancing penetration depth and detection resolution. Although these methods are effective for specific defect types, their performance is generally dependent on sensing conditions, excitation configurations, and inspection environments, which can limit their flexibility in automated production-line inspection.

In addition to physical inspection techniques, machine vision-based methods have gradually become a promising alternative for automated defect detection. Early machine vision approaches primarily relied on conventional image processing techniques, including edge detection, texture analysis, and threshold-based segmentation. These methods can effectively characterize visually salient defects under controlled conditions, but their performance is sensitive to illumination variations, noise, and morphological changes in surface defects [20]. Traditional machine-learning methods subsequently combined handcrafted visual descriptors with shallow classifiers for defect recognition [21]. However, their dependence on manually designed features limits their ability to represent complex and subtle defect patterns. Overall, traditional machine vision methods laid an important foundation for steel surface defect detection. Nevertheless, their inherent limitations in feature representation, adaptability, and robustness have driven the development of deep learning-based approaches.

2.2 Detection Methods Based on Deep Learning

Benefiting from the powerful feature learning capability of convolutional neural networks (CNNs), deep learning-based methods have achieved remarkable success in image detection and recognition tasks. Deep learning-based object detection frameworks are generally divided into two categories: two-stage detectors, represented by R-CNN [22], Fast R-CNN [23], and Faster R-CNN [2], and one-stage detectors, such as SSD [7] and the YOLO series [3–6]. Although two-stage detectors often achieve superior localization accuracy, their multi-stage processing pipeline introduces considerable computational overhead, resulting in relatively low inference efficiency. In contrast, one-stage detectors offer a more favorable trade-off between detection accuracy and inference speed, making them particularly suitable for real-time industrial defect detection applications.

Steel surface defects exhibit substantial variations in category, scale, and appearance, while complex industrial backgrounds further increase the difficulty of accurate defect detection. To address these challenges, numerous studies have developed specialized improvements based on general-purpose object detection frameworks, primarily focusing on multi-scale feature representation, feature fusion optimization, attention mechanism design, and lightweight network construction. MSFT-YOLO [24] improves global contextual modeling and multi-scale feature fusion by integrating a Transformer-based module and a BiFPN architecture into YOLOv5, while MPA-YOLO [25] strengthens multi-scale feature representation through a Multi-path Convolution Attention (MPCA) module and exploits hierarchical information using an auxiliary detection head. Similarly, ELG-YOLO [26] enhances feature representation for complex backgrounds and multi-scale defects by incorporating collaborative attention and adaptive feature fusion mechanisms. Feature fusion has also received considerable attention as an effective means of improving defect representation. Huang et al. [27] introduced a Multi-level Alignment Module (MAM) and a Fusion-Redistribution Module (FRM) to facilitate adaptive cross-layer information interaction and improve detection performance. RDD-YOLO [28] further strengthens feature aggregation by employing a Double Feature Pyramid Network (DFPN), enabling more effective feature reuse across different semantic levels. In addition, Zheng et al. [29] designed a lightweight feature fusion framework based on YOLO11n, which improves the perception of

cross-region and small-scale defects through the integration of relation-aware global attention and a dilated convolution detection head. To improve feature extraction capability and contextual modeling, attention mechanisms have been widely incorporated into steel defect detection frameworks. MPA-YOLO [25] introduces a Partial Self-Attention (PSA) mechanism to capture long-range dependencies, whereas ELA-YOLO [30] employs a linear attention module to enhance feature extraction and designs a selective feature pyramid network to improve cross-layer information fusion. These studies demonstrate the effectiveness of attention-based designs in improving defect perception under complex industrial conditions. Meanwhile, lightweight network design has become an important research direction to meet the real-time requirements of industrial inspection systems. GDM-YOLO [31] reduces information loss and computational cost during downsampling through depthwise ghost convolution while improving small-defect detection performance using enhanced feature extraction modules. DCAM-Net [32] combines deformable convolution with attention mechanisms to improve feature extraction for rapid strip-steel surface defect detection. Recent studies have further explored multi-scale representation and lightweight feature fusion for steel surface defect detection. Liao et al. [33] proposed APG-YOLO, a YOLO11s-based detector that integrates attention-guided intra-scale feature interaction, poly-kernel multi-scale feature extraction, and a generalized feature pyramid network to strengthen cross-scale feature representation and defect localization. The method was validated on both NEU-DET and GC10-DET, showing improved detection performance, particularly for low-contrast and visually ambiguous defect patterns. Zhou and Cao [34] developed CTG-YOLO based on YOLOv8s, incorporating a parallel CBY feature-extraction module, a TFF-PANet for multi-scale feature interaction, and a lightweight GS-Head. Experiments on NEU-DET and GC10-DET demonstrated that the proposed architecture achieves a favorable trade-off between feature representation capability and computational efficiency.

Although existing studies have achieved considerable progress in multi-scale feature extraction, feature fusion, attention modeling, and lightweight network design, several challenges remain. Fine-grained information from small and elongated defects can still be weakened during feature extraction, while complex surface textures make local structural patterns difficult to distinguish. Moreover, the interaction between classification and localization-related information has received comparatively less attention in lightweight steel defect detectors. These limitations motivate the feature remapping, local structural modeling, and regression-guided classification strategies investigated in this work.

3 Methods

3.1 YOLO11

YOLO11 is a representative one-stage object detection framework that provides a favorable balance between detection accuracy, computational efficiency, and deployment flexibility. Its lightweight architecture, efficient feature fusion strategy, and anchor-free detection mechanism enable strong performance in real-time vision tasks. Owing to these advantages, YOLO11 has demonstrated considerable potential for industrial defect detection applications and is therefore adopted as the baseline framework in this work.

3.2 MLQ-YOLO

To address the challenges of insufficient small-defect perception and limited local feature modeling in steel surface defect detection while preserving computational efficiency, this paper proposes MLQ-YOLO, an enhanced detection framework based on YOLO11n. The overall architecture of the proposed model is illustrated in Fig. 1. Specifically, MLQ-YOLO incorporates a Multi-Scale Feature Gating Mapping (MFGM) module to strengthen multi-scale feature representation, a Local Structural Perception Attention (LSPA)

module to enhance local structural modeling, and a Quality-Aware Classification Head (QAC-Head) to introduce regression-guided modulation of classification logits.

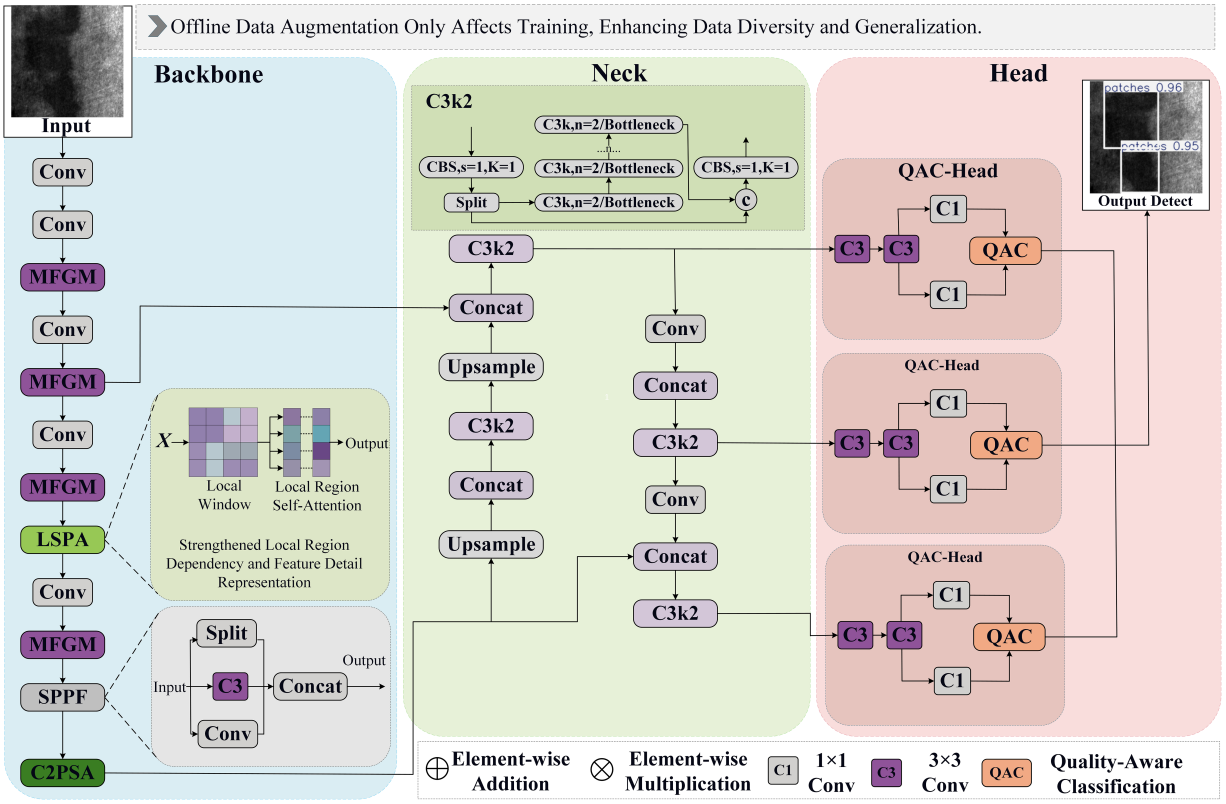


Figure 1: Architecture of MLQ-YOLO.

3.2.1 MFGM: Multi-Scale Feature Gating Mapping

Steel surface defects often exhibit substantial variations in scale and morphology. In particular, small, elongated, and low-contrast defects may occupy only a limited portion of the feature map, while their appearance can be easily confused with irregular background textures. Under such conditions, simply increasing the receptive field or directly aggregating feature responses may introduce redundant background information and weaken subtle defect cues. To address this issue, we design a Multi-Scale Feature Gating Mapping (MFGM) module, shown in Fig. 2, that reorganizes feature extraction within a single feature level through three coupled operations: asymmetric channel decomposition, cross-guided spatial–channel modulation, and input-adaptive gated fusion. Rather than treating multi-scale processing as an independent branch-selection problem, MFGM allows complementary feature branches to guide each other before their final aggregation.

Given an input feature tensor $X \in R^{B\times C\times H\times W}$, MFGM first performs an asymmetric partition along the channel dimension. The channel numbers assigned to the two branches are defined as:

$$C_1 = C - \left[\frac{C}{4}\right], C_2 = \left[\frac{C}{4}\right] \tag{1}$$

and the input is decomposed as:

$$[X_1, X_2] = Split_c(X), X_1 \in R^{B\times C_1\times H\times W}, X_2 \in R^{B\times C_2\times H\times W} \tag{2}$$

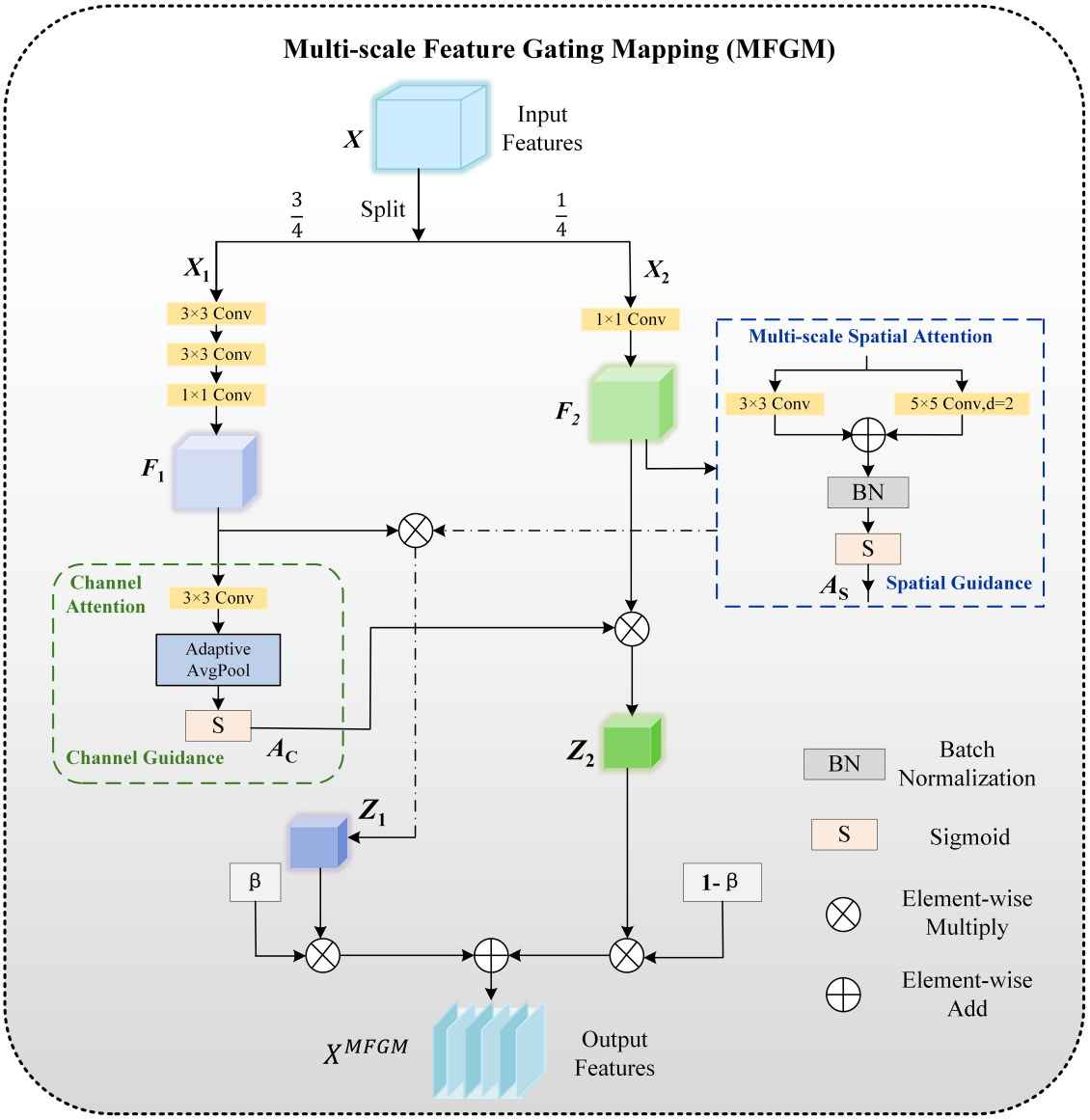


Figure 2: Architecture of multi-scale feature gating mapping module.

Thus, approximately three-quarters of the channels are retained in the primary feature-transformation branch, whereas the remaining one-quarter form a lightweight guidance branch. This asymmetric allocation avoids duplicating expensive transformations over the entire input while reserving most of the representation capacity for local defect structures. The two branches are transformed as:

$$F_1 = Conv_{1\times1}\left(Conv_{3\times3}\left(Conv_{3\times3}\left(X_1\right)\right)\right), F_2 = Conv_{1\times1}\left(X_2\right) \tag{3}$$

where $F_1, F_2 \in R^{B\times C\times H\times W}$. The first branch therefore performs a relatively richer local transformation, while the second branch provides a computationally economical complementary representation. Both branches are projected to the same channel dimension before interaction.

A key characteristic of MFGM is that the two branches are not independently recalibrated by their own attention responses. Instead, spatial and channel cues are exchanged across branches. Specifically, a multi-scale spatial attention map is generated from the lightweight branch F_2:

$$A_s = \sigma\left[BN\left(Conv_{3\times3}\left(F_2\right) + Conv_{5\times5}^{d=2}\left(F_2\right)\right)\right] \tag{4}$$

where $A_s \in R^{B\times1\times H\times W}$, and $\sigma[.]$ denotes the Sigmoid function. The two spatial paths correspond to effective receptive fields of 3×3 and 9×9, respectively, since the 5×5 convolution is implemented with a dilation rate of 2. This combination allows the spatial response to incorporate both localized defect evidence and a broader neighborhood without introducing an additional high-resolution feature-pyramid level.

In the opposite direction, channel-wise guidance is extracted from the locally transformed branch F_1 using a 3×3 depthwise convolution followed by global average pooling:

$$A_c = \sigma\left[GAP\left(DWConv_{3\times3}\left(F_1\right)\right)\right] \tag{5}$$

where $A_c \in R^{B\times C\times1\times1}$. The two attention responses are then applied crosswise:

$$Z_1 = F_1 \otimes A_s, Z_2 = F_2 \otimes A_c \tag{6}$$

where $\otimes$ denotes element-wise multiplication with broadcasting when necessary. Consequently, the spatial distribution inferred from the lightweight branch is used to modulate the locally enriched representation, whereas the channel statistics extracted from the local branch recalibrate the lightweight representation. This reciprocal interaction differs from conventional self-reweighting attention, in which an attention map is typically generated and applied within the same feature stream. In MFGM, the two branches instead provide complementary guidance to each other before fusion.

After cross-guided modulation, MFGM determines the contribution of the two enhanced representations according to the current input. The gate is generated from their concatenated global representation:

$$\beta = \sigma\left[W_2\varnothing\left(W_1GAP\left(\left[Z_1; Z_2\right]\right)\right)\right] \tag{7}$$

where $[\cdot;\cdot]$ denotes channel concatenation, $\varnothing(\cdot)$ is the sigmoid linear unit (SiLU) activation, W_1 denotes a 1×1 projection from $2C$ channels to $C/4$, and W_2 further projects the representation to a single gating coefficient. Therefore, $\beta \in R^{B\times1\times1\times1}$ indicating that the implemented gate is an input-dependent sample-wise scalar, rather than a fixed coefficient or an additional spatial/channel attention map. The final output is given by:

$$X_{MFGM} = \beta Z_1 + (1-\beta) Z_2 \tag{8}$$

This complementary weighting constrains the two fusion coefficients to sum to one while allowing their relative contribution to vary with each input sample.

It should be emphasized that the basic operators used in MFGM, including convolution, depthwise convolution, global pooling, attention, and Sigmoid gating, are established building blocks. The contribution of MFGM lies instead in how these operations are organized and coupled for steel surface defect representation. First, the asymmetric $\frac{3}{4}-\frac{1}{4}$ channel allocation separates a representation-intensive local branch from a lightweight guidance branch. Second, the proposed cross-guided interaction exchanges spatial and channel information between the two branches rather than performing isolated self-attention. Third, the sample-adaptive gate jointly evaluates the two cross-modulated responses and determines their relative contribution before output. Unlike feature-pyramid fusion schemes that primarily combine features

from different resolution levels, MFGM operates on a single input feature tensor and performs adaptive intra-level feature remapping. Within MLQ-YOLO, MFGM therefore focuses on strengthening multi-scale feature representation.

3.2.2 LSPA: Local Structural Perception Attention

Steel surface defects often exhibit localized, irregular, and low-contrast structures, while repetitive background textures may generate strong irrelevant responses. Global feature interaction can therefore introduce unnecessary background information, whereas a fixed local receptive field may be unsuitable for feature maps at different backbone stages. To improve local structural modeling, we introduce a Local Structural Perception Attention (LSPA) module consisting of three coordinated components: learnable residual feature scaling, resolution-adaptive overlapping-window self-attention, and convolution-enhanced lightweight refinement. The overall architecture is shown in Fig. 3.

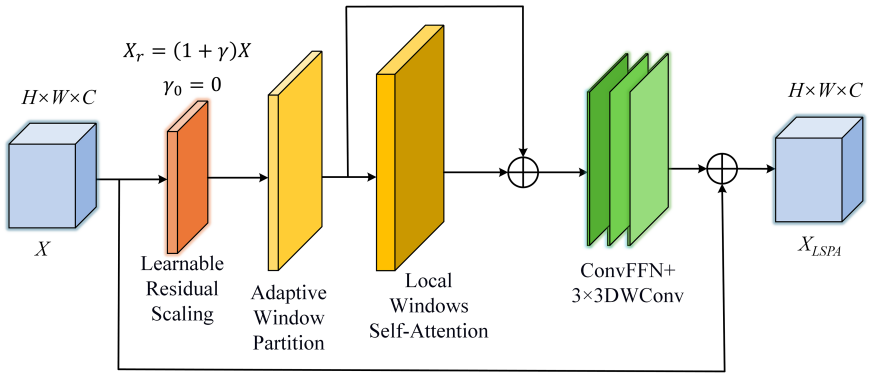


Figure 3: Structure of local structural perception attention module.

Given an input feature tensor $X \in R^{B\times C\times H\times W}$, LSPA first applies a learnable residual scaling:

$$X_r = X + \gamma X = (1+\gamma)\, X, \gamma_0 = 0 \tag{9}$$

where γ is a single learnable scalar initialized to zero. Hence, the operation starts as an identity mapping and introduces only one additional parameter, allowing the subsequent local modeling process to be incorporated without strongly perturbing the incoming representation at the beginning of training. The implementation explicitly defines γ in this form.

LSPA then constructs a resolution-adaptive overlapping local perception region. Instead of using a uniform window size, the local window size p and stride s are determined by:

$$p = \min\left(16, \max\left(8, \left[\frac{\min(H, W)}{4}\right]\right)\right), s = \max(1, p-2) \tag{10}$$

Accordingly, p is constrained to $[8, 16]$, and adjacent windows generally overlap by two feature positions. This design differs from fixed non-overlapping window partitioning and does not require an alternating shifted-window strategy. The adaptation is resolution-dependent, enabling feature maps at different backbone stages to operate with different local perception ranges. Replicate padding is applied when necessary to maintain valid window extraction.

Each $p \times p$ region is reshaped into a local token sequence and processed independently by multi-head self-attention. After Layer Normalization, query, key, and value representations are generated through linear

projections. The current implementation employs four attention heads with a total query/key dimension of 36, while the value projection preserves the input feature dimension. The local attention is computed as:

$$Attn\left(Q,K,V\right)=Soft\max\left(\frac{QK^{T}}{\sqrt{D_{qk}}}\right)V,D_{qk}=36 \tag{11}$$

The outputs of the four heads are concatenated, linearly projected, and combined with the window input through a residual connection. Restricting self-attention to overlapping local regions reduces unnecessary interaction with distant background textures while preserving structural continuity across neighboring windows, which is particularly useful for elongated or fragmented defects. The implementation confirms the use of linear Q/K/V projections and four-head scaled dot-product self-attention.

After attention, the local windows are reconstructed to their original spatial positions, with overlapping responses accumulated by element-wise addition. The reconstructed feature is then processed by a convolution-enhanced feed-forward network (ConvFFN). Specifically, the feature is first projected from C channels to a hidden dimension of 96, followed by Gaussian error linear unit (GELU) activation and a 3×3 depthwise convolution, and is finally projected back to C channels. This design supplements token interaction with explicit short-range spatial modeling while retaining a lightweight structure.

A 3×3 depthwise kernel is adopted instead of a wider 5×5 kernel for local spatial refinement. Under identical feature dimensions, the kernel-related computation and parameter cost scale with k^2, giving:

$$\frac{C_{3\times3}}{C_{5\times5}}=\frac{9}{25} \tag{12}$$

which corresponds to approximately 64% fewer kernel-related operations and parameters for this depthwise convolutional operator. This reduction refers only to the depthwise refinement operation rather than to the total FLOPs or memory consumption of the complete LSPA module.

Finally, an outer residual connection combines the refined feature with the original input, preserving the backbone representation while introducing locally modeled structural information. Overall, LSPA combines resolution-adaptive overlapping local attention for neighborhood-level structural interaction with a lightweight depthwise ConvFFN for local spatial refinement. The contribution does not lie in the individual use of self-attention or depthwise convolution, but in their task-oriented organization for steel surface defect representation, complementing the multi-scale feature interaction performed by MFGM.

3.2.3 QAC-Head

In steel surface defect detection, ambiguous boundaries and similar surface textures may lead to inconsistent responses between the classification and regression branches. In the conventional decoupled detection head, these two branches produce their predictions largely in parallel, so the classification score does not explicitly reflect the state of the regression branch. To introduce a lightweight interaction between the two tasks, we design a Quality-Aware Classification Head (QAC-Head) that uses the existing regression response to recalibrate the classification logits without introducing an additional prediction branch.

For each detection level, let B_l denote the raw regression output before Distribution Focal Loss (DFL) decoding. In the implemented YOLO11n detection head, each spatial location contains $4R$ regression channels with $R=16$. Instead of predicting an additional Intersection-over-Union (IoU) or localization-quality score, QAC directly summarizes the response strength of these regression channels as:

$$Q_l=\sigma\left(\mathrm{Mean}_c\left(\left|sg\left(B_l\right)\right|\right)\right) \tag{13}$$

where $\mathrm{Mean}_c(\cdot)$ denotes averaging over the regression channels, $sg(\cdot)$ denotes stop-gradient, and $\sigma(\cdot)$ is the Sigmoid function. Because the mean absolute regression response is nonnegative, $Q_l \in [0.5, 1)$. This scalar is shared across classes at each spatial location and serves as a heuristic modulation factor derived from raw regression outputs before DFL decoding, rather than an explicit IoU estimate.

The resulting quality proxy is then directly applied to the classification logits:

$$C'_l = C_l \otimes Q_l \tag{14}$$

where C_l and C'_l denote the original and modulated classification logits, respectively, and $\otimes$ denotes element-wise multiplication with broadcasting across classes. For a fixed logit, multiplication by Q_l reduces its magnitude toward zero. After Sigmoid, positive logits yield lower probabilities and negative logits yield higher probabilities, both moving toward 0.5. Larger Q_l values preserve more of the original logit magnitude. This operation introduces regression-dependent scaling into classification scoring.

An important aspect of QAC is the stop-gradient operation in Eq. (13). The regression output is detached before constructing Q_l, so classification gradients do not propagate back to the regression branch through the QAC pathway. Consequently, QAC establishes a one-way regression-to-classification guidance mechanism while leaving the original regression optimization path unchanged. Moreover, because the modulation is performed before the training/inference branch of the detection head, the same operation is applied during both training and inference.

QAC differs from approaches that explicitly predict localization quality or introduce additional quality supervision. It requires neither an auxiliary quality branch nor an additional IoU target; instead, it reuses the existing regression responses through several parameter-free operations. Therefore, the main contribution of QAC lies in providing a minimal regression-to-classification coupling strategy rather than constructing a new localization-quality estimator.

3.2.4 Offline Data Augmentation

To evaluate the effectiveness of the proposed MLQ-YOLO model, experiments are conducted on the NEU-DET dataset, which contains six typical categories of hot-rolled steel strip surface defects, including crazing (Cr), inclusions (In), patches (Pa), pitted surfaces (Ps), rolled-in scale (Rs), and scratches (Sc). The dataset consists of 1800 grayscale images covering these six representative defect types, with 300 samples collected for each category. All images have an original resolution of 200 × 200 pixels.

To avoid information leakage caused by augmented variants of the same source image appearing in different subsets, the original images are first divided into training, validation, and test sets at a ratio of 8:1:1. Offline augmentation is subsequently performed only on the training subset. The validation and test sets contain exclusively the original, non-augmented images, and no transformed derivative of an evaluation image is included in the training set. This split-before-augmentation protocol ensures that model evaluation is performed on images that are independent of the offline augmentation procedure. Representative defect samples are shown in Fig. 4.

As illustrated in Fig. 4, two major issues can be observed in the dataset. On the one hand, significant intra-class variation exists within the same defect category. Taking scratches (Sc) as an example, the defects may appear in various forms, such as horizontal scratches or inclined scratches. Furthermore, factors including illumination variation, occlusion, and material inconsistencies can lead to noticeable grayscale fluctuations within defect images of the same category, thereby adversely affecting detection performance. On the other hand, several defect categories exhibit highly similar inter-class characteristics. For instance,

defects such as crazing (Cr), patches (Pa), and rolled-in scale (Rs) share similar visual patterns and texture features, which can easily cause category confusion and further degrade detection accuracy.

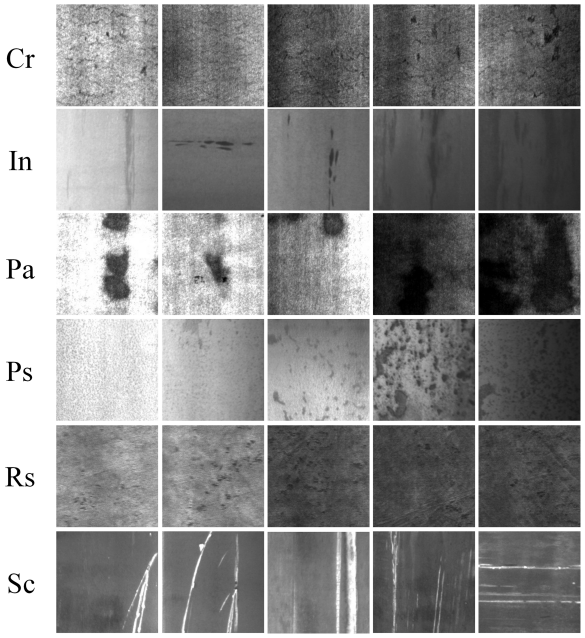


Figure 4: Representative steel surface defects from NEU-DET. Each original image contains 200 × 200 pixels. Physical scale bars are omitted because the official dataset description does not provide pixel-to-length calibration.

Considering the aforementioned issues, as well as the relatively poor detection performance observed for crazing (Cr) and rolled-in scale (Rs) during preliminary experiments, offline data augmentation was employed to increase data diversity and improve detection accuracy. Specifically, each image was randomly rotated around its center within an angle range of $[-15°, 15°]$ using a rotation step of 1°. Bilinear interpolation was adopted for pixel filling to preserve image clarity after rotation. In addition, contrast adjustment was applied, and random rectangular occlusions were introduced at different positions of the original images to simulate partial defect shielding. The occlusion area was controlled within 5%–20% of the image size. The corresponding bounding-box annotations are transformed consistently with the geometric operations to preserve the spatial correspondence between defects and labels.

Through these augmentation operations, only the original training subset is expanded to five times its initial size, from 1440 to 7200 images, whereas the 180 validation images and 180 test images remain unchanged. Thus, the augmented images are used exclusively for model optimization rather than performance evaluation. In addition, when this offline-augmented training set is adopted, the default online augmentation strategy is disabled so that the contribution of the predefined offline transformations is not confounded by additional stochastic augmentation during training. Fig. 5 presents representative examples of the offline-augmented training samples.

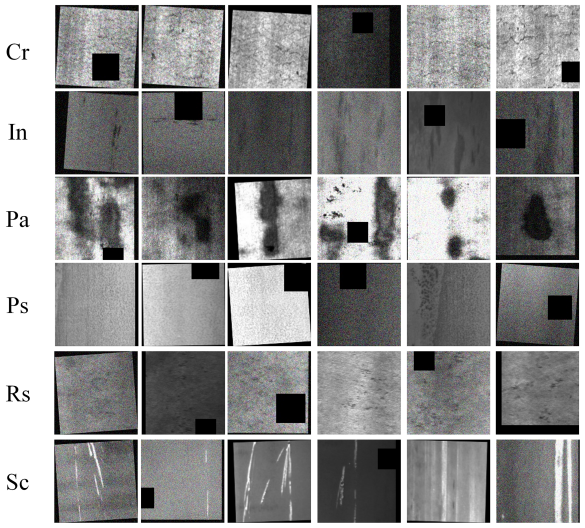


Figure 5: Representative examples of the offline-augmented NEU-DET training samples.

4 Experiments

4.1 Experimental Setup

All experiments were conducted on a workstation equipped with a single NVIDIA GeForce RTX 4060 Ti GPU with 16 GB of memory and a 12th-generation Intel Core i5 processor under Windows 11. The proposed network was implemented using PyTorch 2.0.0+cu118 with CUDA 11.8 and cuDNN 8.7.0. The detailed hardware and software environment is summarized in Table 1.

Table 1: Hardware and software environment used in the experiments.

Parameter	Specification
GPU model	NVIDIA GeForce RTX 4060 Ti
Number of GPUs	1
GPU memory	16 GB
GPU driver	591.86
Processor name	Intel Core i5 12th Gen
Operating system	Windows 11
PyTorch	2.0.0+cu118
CUDA	11.8
cuDNN	8.7.0
Ultralytics	8.3.85
Python	3.9.20

For model training, all input images were resized to 640 × 640. The training batch size was set to 32, and each model was trained for 200 epochs using the stochastic gradient descent (SGD) optimizer with an initial learning rate of 0.01 and a momentum of 0.937. Eight data-loading workers were used, and the random seed was fixed at 0 for the reported experiments. The main training hyperparameters are listed in Table 2.

Table 2: Training hyperparameter settings.

Parameter	Value
Image size	640
Training batch size	32
Epochs	200
Initial learning rate	0.01
Optimizer	SGD
Momentum	0.937
Workers	8
Random seed	0

The GC10-DET dataset was additionally employed as a second benchmark to evaluate the proposed model under different defect categories and surface conditions. Unless otherwise specified, consistent experimental settings were maintained within each controlled comparison. Inference efficiency was evaluated separately from model training using a unified batch size of 1 and 32-bit floating-point (FP32) precision; the detailed timing protocol and efficiency results are presented in the corresponding efficiency comparison experiment.

4.2 Evaluation Metrics

The performance of the detection models is evaluated from two aspects: detection accuracy and computational efficiency. Detection accuracy is primarily assessed using mean Average Precision (mAP)@0.5 and mAP@0.5:0.95, while model complexity and inference efficiency are characterized by the number of parameters, Floating-Point Operations (FLOPs), inference latency, and Frames Per Second (FPS). Precision and Recall are introduced as the basic quantities used to construct the Precision–Recall curve and calculate Average Precision (AP).

Precision represents the proportion of correct detections among all positive predictions, whereas Recall measures the proportion of ground-truth objects successfully detected. They are defined as:

$$P = \frac{TP}{TP + FP} \tag{15}$$

$$R = \frac{TP}{TP + FN} \tag{16}$$

where (TP), (FP), and (FN) denote true-positive detections, false-positive detections, and missed ground-truth objects, respectively. For the c-th defect category, Average Precision (AP_c) is obtained from the area under the corresponding Precision–Recall curve:

$$AP_c = \sum_{j=1}^{K} P_{c,j} \left(R_{c,j} - R_{c,j-1} \right) \approx \int_0^1 P_c(R) dR \tag{17}$$

where $P_{c,j}$ and $R_{c,j}$ denote the Precision and Recall values at the j-th point of the c-th category, respectively, and K is the number of sampled points on the Precision–Recall curve. The mean Average Precision over all C defect categories is calculated as:

$$mAP = \frac{1}{C} \sum_{c=1}^{C} AP_c \tag{18}$$

where C denotes the total number of defect categories.

In this study, mAP@0.5 denotes the mean AP evaluated at an IoU threshold of 0.5. To provide a stricter assessment of localization accuracy, mAP@0.5:0.95 is additionally reported by averaging the mAP values over IoU thresholds ranging from 0.50 to 0.95 with an interval of 0.05. Compared with mAP@0.5, this metric places progressively stricter requirements on the overlap between predicted and ground-truth bounding boxes and therefore provides a more comprehensive evaluation of detection and localization performance.

For computational evaluation, the parameter count indicates the number of learnable model parameters, while FLOPs characterize the theoretical computational cost under the specified input resolution. Inference efficiency is further evaluated using average latency and *FPS*. Under the standardized inference benchmark described in the corresponding efficiency experiment, *FPS* is calculated as:

$$FPS = \frac{1000}{T_{\text{inf}}} \tag{19}$$

where T_{inf} denotes the average model-forward latency in milliseconds per image. The benchmark is conducted with a batch size of 1 and FP32 precision using identical 640×640 inputs. Image loading, preprocessing, host-to-device data transfer, and Non-Maximum Suppression (NMS) post-processing are excluded from the timing process so that the reported latency and FPS reflect the forward-computation efficiency of the models.

4.3 Comparative Experiments

To verify the effectiveness of the proposed MLQ-YOLO model, comparative experiments were conducted with several representative state-of-the-art object detection methods. In addition to conventional detectors and different YOLO variants, the recent steel surface defect detectors CTG-YOLO and APG-YOLO were included for comparison with more recent task-specific architectures. For a fair comparison, CTG-YOLO and APG-YOLO were reproduced using the same experimental settings and training hyperparameters adopted in this study. For all experiments on the NEU-DET dataset, offline data augmentation was applied exclusively to the training set, while the validation and test sets remained unchanged to ensure unbiased performance evaluation. In addition, when the offline-augmented dataset was used for training, the default online augmentation strategy was disabled to avoid potential interference between different augmentation schemes. All other training configurations were kept consistent across experiments. This setting enabled a fair comparison and provided a reliable assessment of the generalization capability of the proposed method. The comparison results are summarized in Tables 3 and 4.

Offline augmentation is regarded as a training-data strategy rather than a component of MLQ-YOLO, and its effect is therefore examined separately from that of the architectural modifications. Using the original training data, YOLO11n and MLQ-YOLO achieve mAP@0.5 values of 74.1% and 81.4%, respectively, corresponding to an architectural gain of 7.3 percentage points. After applying the same offline augmentation protocol, their mAP@0.5 values increase to 80.4% and 85.6%, respectively, while MLQ-YOLO retains a 5.2-point advantage over the baseline. From the perspective of augmentation, the performance improvements are 6.3 points for YOLO11n and 4.2 points for MLQ-YOLO. These results indicate that increased training-sample diversity benefits both models, while the proposed architectural modifications remain effective under both training-data conditions. The final performance gain should therefore be interpreted as the combined contribution of data diversity and network design rather than being attributed to either factor alone.

Table 3: Detection results of various models on the original NEU-DET dataset.

Methods	Cr	In	Pa	Ps	Rs	Sc	mAP@0.5 (%)	mAP@0.5:0.95 (%)	Params (M)	FLOPs (G)
SSD	38.2	76.3	88.4	77.9	65.1	77.1	70.5	38.8	26.30	30.8
Faster R-CNN	37.7	77.5	91.2	80.5	60.7	89.2	72.8	37.9	134.30	83.5
RT-DETR	32.0	77.7	88.7	86.1	57.6	92.3	72.4	39.8	33.10	108.7
YOLOv5n	42.5	78.2	87.1	80.6	67.1	86.7	73.7	42.7	2.51	7.2
YOLOv7-tiny	44.3	78.0	92.6	81.4	63.7	85.8	74.3	41.6	6.03	13.2
YOLOv8n	42.5	79.2	90.5	76.1	65.3	90.4	74.0	42.2	3.01	8.2
YOLOv9t	44.8	78.2	91.2	75.9	64.6	89.9	74.1	44.5	2.01	7.9
YOLOv10n	45.3	79.3	91.5	70.4	68.7	87.6	73.8	42.1	2.71	8.4
YOLO11n	44.3	78.5	90.1	76.2	65.3	90.2	74.1	44.2	2.59	6.4
CTG-YOLO	48.2	80.1	90.4	79.3	67.2	92.6	76.3	44.3	13.67	27.2
APG-YOLO	57.5	86.2	92.8	87.3	73.5	95.3	82.1	46.8	16.03	42.8
MLQ-YOLO	59.6	85.4	92.2	85.1	72.6	93.5	81.4	46.4	2.61	6.7

Table 4: Detection results of models trained on the data-augmented NEU-DET training set.

Methods	Cr	In	Pa	Ps	Rs	Sc	mAP@0.5 (%)	mAP@0.5:0.95 (%)	Params (M)	FLOPs (G)
SSD	44.2	80.1	90.1	81.8	70.2	81.2	74.6	40.1	26.30	30.8
Faster R-CNN	40.7	81.8	91.5	83.6	62.0	91.1	75.1	38.3	134.30	83.5
RT-DETR	36.4	81.6	90.7	90.3	61.7	92.7	75.6	40.4	33.10	108.7
YOLOv5n	49.5	86.3	90.1	85.6	73.4	93.0	79.7	45.3	2.51	7.2
YOLOv7-tiny	50.3	84.0	91.0	87.4	68.7	90.8	78.7	42.9	6.03	13.2
YOLOv8n	51.6	89.2	90.9	85.3	73.3	90.9	80.2	44.7	3.01	8.2
YOLOv9t	51.7	87.3	91.6	82.0	70.1	92.1	79.1	46.1	2.01	7.9
YOLOv10n	53.3	86.2	91.6	78.1	75.4	91.3	79.3	44.4	2.71	8.4
YOLO11n	54.2	87.4	90.5	85.3	74.3	90.7	80.4	43.4	2.59	6.4
CTG-YOLO	56.7	88.4	91.1	86.6	74.7	92.7	81.7	46.1	13.67	27.2
APG-YOLO	63.2	93.1	92.2	89.3	76.5	97.5	85.3	47.6	16.03	42.8
MLQ-YOLO	65.4	94.0	92.7	90.0	77.4	94.1	85.6	47.5	2.61	6.7

The comparison with CTG-YOLO and APG-YOLO further extends the evaluation to recent steel-specific detection architectures. Compared with YOLO11n, MLQ-YOLO introduces only a limited increase in computational complexity, with 2.61 M parameters and 6.7 GFLOPs, while providing a clear improvement in detection accuracy. Representative detection results are shown in Fig. 6. The revised visualization includes challenging samples with small spatial extent, elongated morphology, weak contrast, or complex background textures, allowing the localization and classification behavior of YOLO11n and MLQ-YOLO to be compared more clearly.

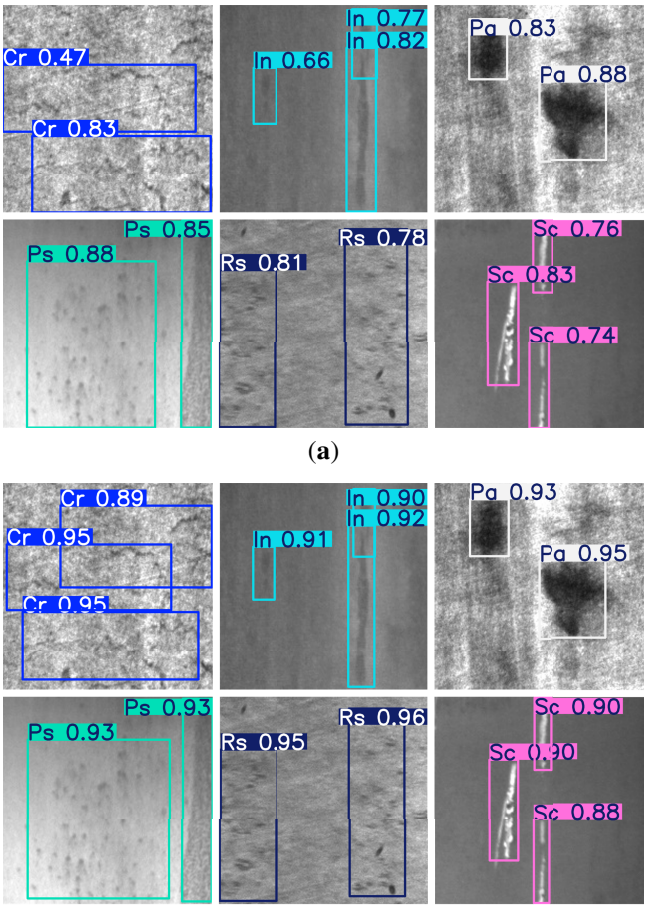


Figure 6: Comparative detection results for YOLO11n and MLQ-YOLO. (**a**) YOLO11n; (**b**) MLQ-YOLO.

To evaluate the proposed model on a second steel surface defect benchmark, experiments were conducted on the GC10-DET dataset. Compared with NEU-DET, GC10-DET presents a more challenging detection scenario, as multiple defect categories may coexist within a single image and defect instances exhibit substantial variations in scale, appearance, and spatial distribution. These characteristics make the dataset more representative of practical industrial inspection environments. As shown in Fig. 7, MLQ-YOLO accurately detects defects in both single-defect and multi-defect scenarios. The model maintains reliable detection performance even when different defect categories with diverse visual characteristics appear simultaneously in the same image. Furthermore, the quantitative results in Table 5 indicate that MLQ-YOLO achieves an mAP@0.5 of 74.2%, which is 4.0 percentage points higher than that of the baseline YOLO11n. The consistent performance improvement across datasets demonstrates the effectiveness of the proposed MFGM, LSPA, and QAC-Head designs and further validates the robustness of MLQ-YOLO for steel surface defect detection.

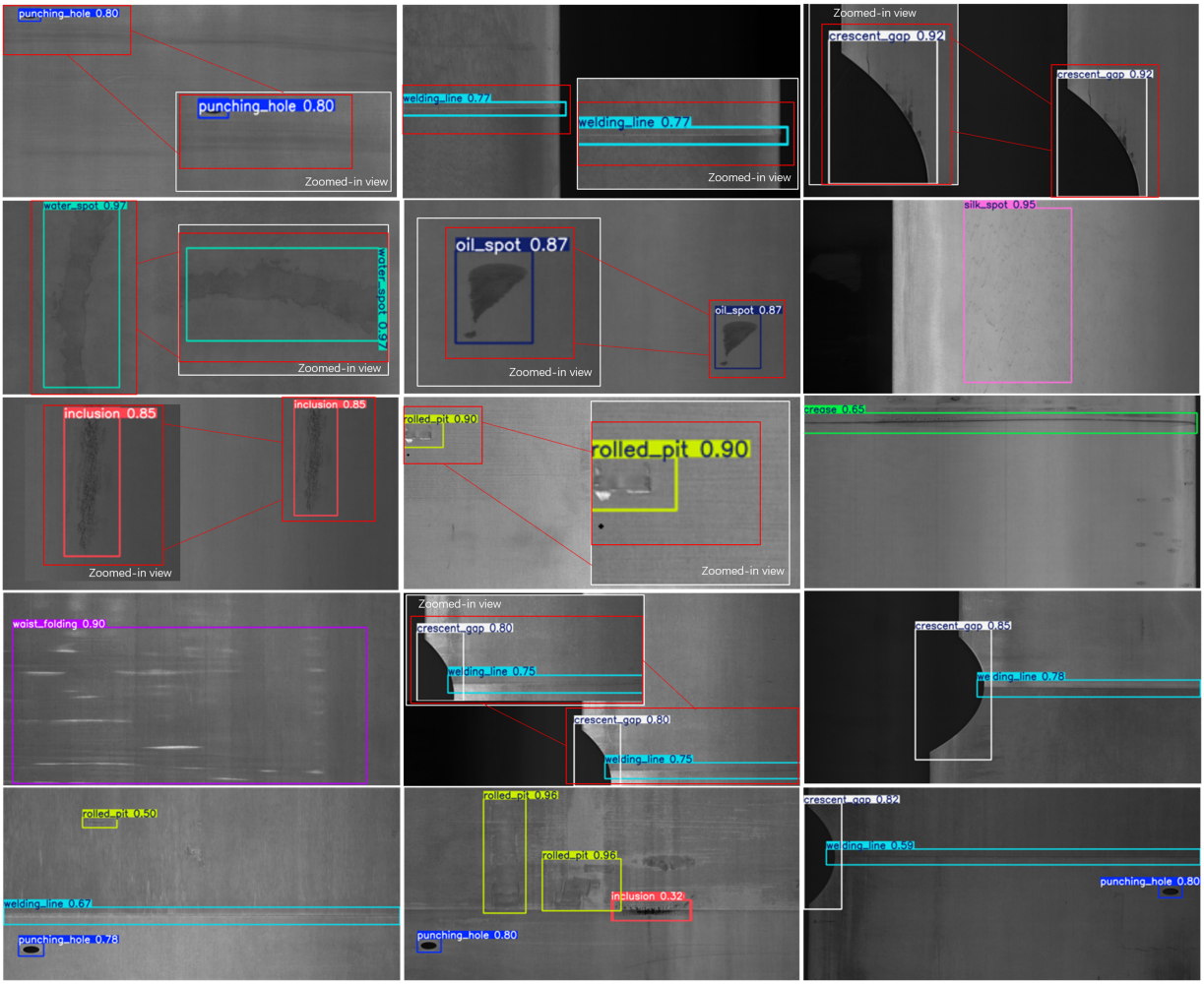


Figure 7: Detection results of MLQ-YOLO on the GC10-DET dataset.

Table 5: Detection results of various models on the GC10-DET dataset.

Methods	mAP@0.5 (%)	mAP@0.5:0.95 (%)	Params (M)	FLOPs (G)
Faster R-CNN	68.7	35.2	134.3	83.5
YOLOv5n	70.2	34.9	2.51	7.2
YOLOv7-tiny	71.4	37.6	6.03	13.2
YOLOv8n	70.4	34.1	3.01	8.2
YOLOv9t	69.2	36.8	2.01	7.9
YOLOv10n	69.1	34.8	2.71	8.4
YOLO11n	70.2	36.4	2.59	6.4
CTG-YOLO	70.0	34.2	13.67	27.2
APG-YOLO	69.6	36.5	16.03	42.8
MLQ-YOLO	74.2	39.8	2.61	6.7

Inference efficiency is evaluated separately from the dataset-dependent detection experiments. Representative models are benchmarked using identical synthetic inputs with a batch size of 1 and FP32 precision.

After 100 warm-up iterations, 1000 forward passes are timed with CUDA synchronization enabled. The measurement covers model forward computation only and excludes image loading, preprocessing, host-to-device transfer, and NMS. The resulting latency and FPS, together with model complexity, are reported in Table 6 to provide a consistent comparison of inference efficiency.

Table 6: Inference efficiency and resource consumption of different models.

Methods	Params (M)	FLOPs (G)	Latency (ms/Image)	Peak GPU Memory (MB)	Model Size (MB)	FPS
SSD	26.30	30.8	20.7	190.82	53.70	48.3
Faster R-CNN	134.30	83.5	76.9	772.54	275.62	13.0
RT-DETR	33.10	108.7	33.6	347.62	69.03	29.8
YOLOv5n	2.51	7.2	5.7	42.49	5.04	175.4
YOLOv7-tiny	6.03	13.2	9.2	89.78	11.70	108.7
YOLOv8n	3.01	8.2	4.5	46.39	5.97	222.2
YOLOv9t	2.01	7.9	17.9	34.81	4.44	55.9
YOLOv10n	2.71	8.4	8.4	79.46	5.50	119.0
YOLO11n	2.59	6.4	6.9	77.51	5.24	144.9
CTG-YOLO	13.67	27.2	17.5	193.51	28.12	57.1
APG-YOLO	16.03	42.8	26.2	281.49	34.05	38.2
MLQ-YOLO	2.61	6.7	9.3	79.74	5.31	107.5

4.4 Ablation Experiments

4.4.1 Ablation Study of Attention Mechanisms

To evaluate the effectiveness of the proposed LSPA module, several representative attention mechanisms, including Squeeze-and-Excitation (SE) [35], Convolutional Block Attention Module (CBAM) [36], Efficient Multi-Scale Attention (EMA) [37], Efficient Channel Attention (ECA) [38], and Local-Region Self-Attention (LRSA) [39], were integrated into the network for comparison under identical experimental settings. The quantitative results are summarized in Table 7. LRSA refers to the local-region self-attention module described in CATANet [39], which follows the overlapping-patch attention design of HPINet [40]. To ensure a controlled comparison, MFGM and QAC-Head are retained in all configurations, and only the attention module is replaced at the same insertion position, with the corresponding channel setting kept unchanged. Therefore, the LSPA configuration in Table 7 represents YOLO11n equipped with MFGM, LSPA, and QAC-Head, rather than YOLO11n with LSPA alone.

Table 7: Ablation study of different attention mechanisms on the augmented NEU-DET dataset.

Methods	mAP@0.5 (%)	mAP@0.5:0.95 (%)	Params (M)	FLOPs (G)
YOLO11n+MFGM+SE+QAC-Head	79.9	41.3	3.73	7.5
YOLO11n+MFGM+CBAM+QAC-Head	81.0	43.1	3.79	7.6
YOLO11n+MFGM+EMA+QAC-Head	80.4	42.6	3.87	7.9
YOLO11n+MFGM+ECA+QAC-Head	79.2	44.1	3.09	7.3
YOLO11n+MFGM+LRSA+QAC-Head	81.5	44.8	3.16	8.2
YOLO11n+MFGM+LSPA+QAC-Head	85.6	47.5	2.61	6.7

As shown in Table 7, LSPA achieves the best detection performance among all compared attention mechanisms. Compared with LRSA, LSPA obtains an additional 4.1 percentage-point gain. This result suggests that the resolution-adaptive overlapping-window strategy is better suited to preserving localized structural relationships under the present network configuration. In addition, the convolution-enhanced refinement stage supplements local self-attention with spatial inductive bias while maintaining relatively low computational complexity. Among the compared configurations, LSPA also yields the lowest parameter count and FLOPs while achieving the highest detection accuracy, indicating a favorable accuracy–complexity trade-off under the controlled attention-replacement setting.

It should be noted that the LSPA row in Table 7 corresponds to the complete MLQ-YOLO configuration (Model 8 in Table 8). In contrast, Model 3 in Table 8 contains LSPA alone without MFGM or QAC-Head. The different results of these two configurations therefore reflect different enabled-module combinations rather than inconsistent measurements.

Table 8: Ablation study of the proposed modules on the augmented NEU-DET dataset.

Methods	MFGM	LSPA	QAC-Head	mAP@0.5 (%)	mAP@0.5:0.95 (%)	Params (M)	FLOPs (G)
Model 1	×	×	×	80.4	43.4	2.59	6.4
Model 2	√	×	×	82.6	42.1	2.26	6.2
Model 3	×	√	×	81.5	43.2	2.87	7.2
Model 4	×	×	√	81.2	46.3	2.59	6.4
Model 5	√	√	×	84.2	45.9	2.61	6.7
Model 6	√	×	√	83.5	46.8	2.26	6.2
Model 7	×	√	√	83.2	46.6	2.87	7.2
Model 8	√	√	√	85.6	47.5	2.61	6.7

Note: "√" indicates that the corresponding module is enabled, whereas "×" indicates that it is disabled.

4.4.2 Overall Model Ablation Study

To evaluate the contribution of each proposed component, an ablation study was conducted using YOLO11n as the baseline model. All configurations were trained and evaluated using the same augmented NEU-DET split and experimental settings, with only the indicated modules enabled or disabled. The experimental results are summarized in Table 8. The baseline model (Model 1) achieves an mAP@0.5 of 80.4%. After replacing the original feature extraction module with MFGM (Model 2), the mAP@0.5 increases to 82.6%, indicating that the proposed multi-scale gated feature modeling strategy effectively enhances feature representation and improves the detection of small-scale defects. When only the LSPA module is introduced (Model 3), the mAP@0.5 reaches 81.5% with modest increases in model parameters and computational complexity, demonstrating the effectiveness of adaptive local structural perception for defect representation learning. Incorporating only the QAC-Head (Model 4) increases mAP@0.5 to 81.2%, indicating that regression-guided logit modulation benefits detection performance under the evaluated setting. In the subsequent combination experiments, integrating MFGM and LSPA simultaneously (Model 5) increases the mAP@0.5 to 84.2%. This improvement indicates that the two modules provide complementary advantages, with MFGM enhancing multi-scale feature extraction and LSPA strengthening local structural perception. Furthermore, introducing MFGM and LSPA into the YOLO11n framework equipped with QAC-Head results in additional performance gains, demonstrating the compatibility and effectiveness of the proposed modules when used jointly. When all three components are incorporated simultaneously (Model 8), the proposed MLQ-YOLO achieves the highest mAP@0.5 of 85.6%, representing a 5.2 percentage-point improvement over the baseline model. Although the parameter count and computational complexity

increase slightly, the additional overhead remains limited, and the model maintains high inference efficiency. The ablation study verifies the effectiveness of each proposed component. MFGM improves the representation of multi-scale defect features, LSPA enhances the perception of local structural information, and QAC-Head introduces regression-dependent modulation of classification logits.

To further examine how the proposed feature-processing modules affect spatial representation, feature-response maps are visualized for representative defect samples, as shown in Fig. 8. Feature responses are extracted from semantically corresponding intermediate stages of the compared models and converted into spatial response maps by averaging the absolute activation values across channels. For visualization, each response map is normalized independently; therefore, the comparison focuses on the spatial distribution of responses rather than absolute activation magnitude.

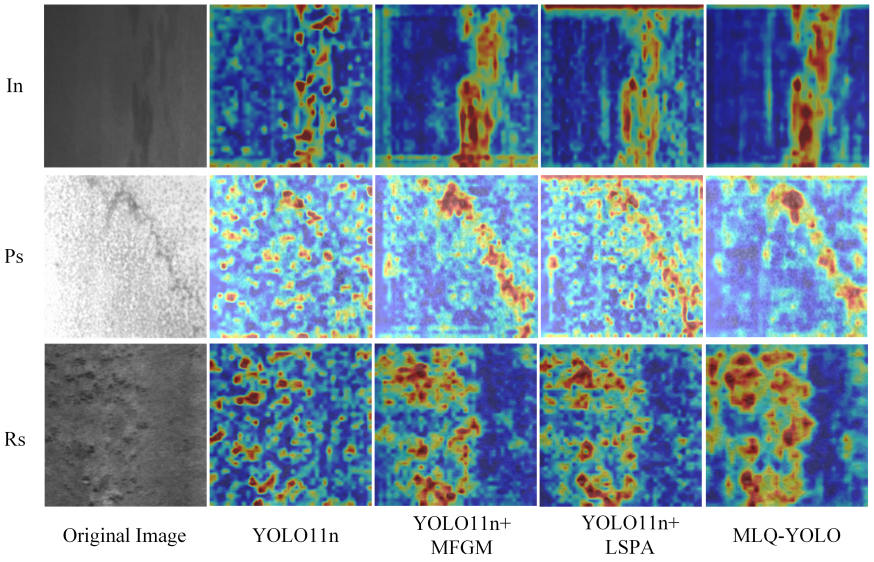


Figure 8: Feature-response maps generated by the models on the NEU-DET dataset.

Compared with YOLO11n, the configurations incorporating the proposed modules exhibit more concentrated responses around defect-related regions and fewer scattered activations over irrelevant background textures. This tendency is particularly visible for elongated, low-contrast, and locally distributed defect patterns. The visualization provides qualitative support for the quantitative ablation results, suggesting that MFGM and LSPA help preserve defect-related structural responses while reducing interference from complex surface textures.

Additional controlled analyses, including the separation of augmentation and architectural effects, class-wise augmentation gains, overall component ablation results and five-seed stability analysis are provided in the Supplementary Material (Tables S1–S5).

5 Conclusion

This paper presented MLQ-YOLO, an improved YOLO11n-based model for steel surface defect detection. To address the challenges of multi-scale defects, insufficient local structural perception, and limited interaction between classification and regression branches, three dedicated components were developed: the MFGM module, the LSPA module, and the QAC-Head. MFGM strengthens multi-scale feature representation through gated feature modulation, LSPA enhances local structural modeling, and QAC-Head introduces

regression-response-guided recalibration of classification predictions. Extensive experiments on the NEU-DET and GC10-DET datasets demonstrated the effectiveness of the proposed method. MLQ-YOLO achieved an mAP@0.5 of 85.6% on NEU-DET and 74.2% on GC10-DET, outperforming the baseline YOLO11n while introducing only limited additional model complexity. The ablation studies further confirmed the individual contributions of the proposed components and demonstrated their complementary effects on overall detection performance. The influence of offline data augmentation was also evaluated separately, showing that the architectural improvements remain effective under both original and augmented training settings. Overall, the proposed method provides an effective lightweight framework for steel surface defect detection. Nevertheless, the current evaluation is limited to publicly available benchmark datasets and does not yet constitute validation in real industrial production environments. Future work will therefore focus on embedded-platform deployment and further evaluation using large-scale production-line data under practical operating conditions.

Acknowledgement: None.

Funding Statement: This research was funded by the National Key Research and Development Program of China (https://service.most.gov.cn), Grant No. 2024YFB3310701 (C. L.); the National Natural Science Foundation of China (https://www.nsfc.gov.cn), Grant Nos. 62201327 and 62371274 (C. L.); and the Shandong Provincial Graduate Education Quality Teaching Resource Project, Grant No. SDYAL2024037 (C. L.).

Author Contributions: The authors confirm their contributions to the paper as follows: conceptualization, Changzhi Lv and Jiaxuan Guo; methodology, Changzhi Lv; software, Yu Chen; validation, Jiaxuan Guo, Yu Chen and Di Fan; formal analysis, Zonglei Mou; investigation, Shaoshao Liu; resources, Changzhi Lv; data curation, Jiaxuan Guo; writing—original draft preparation, Jiaxuan Guo; writing—review and editing, Changzhi Lv; visualization, Jiaxuan Guo; supervision, Changzhi Lv; project administration, Changzhi Lv; funding acquisition, Changzhi Lv. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: The data that support the findings of this study are available from the corresponding author, Di Fan, upon reasonable request.

Ethics Approval: Not applicable.

Conflicts of Interest: Shaoshao Liu is a full-time employee of Shandong Lingyu Information Technology Co., Ltd., which provided equipment and other resources for this study. The authors affirm that they have no additional financial or personal conflicts of interest that could have influenced the work reported in this manuscript.

Supplementary Materials: The supplementary material is available online at https://www.techscience.com/doi/10.32604/cmc.2026.088008/s1.

References

1. Xu Y, Yang S, Zhang X, Xie Z. Steel surface defects detection based on improved faster R-CNN. In: Proceedings of the 2022 4th International Conference on Control and Robotics (ICCR); 2022 Dec 2–4; Guangzhou, China. New York, NY, USA: IEEE; 2022. p. 353–7. doi:10.1109/ICCR55715.2022.10053878.
2. 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. 2017;39(6):1137–49. doi:10.1109/TPAMI.2016.2577031.
3. Redmon J, Divvala S, Girshick R, Farhadi A. You only look once: unified, real-time object detection. In: Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2016 Jun 27–30; Las Vegas, NV, USA. New York, NY, USA: IEEE; 2016. p. 779–88. doi:10.1109/CVPR.2016.91.

4. Redmon J, Farhadi A. YOLO9000: better, faster, stronger. In: Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2017 Jul 21–26; Honolulu, HI, USA. New York, NY, USA: IEEE; 2017. p. 6517–25. doi:10.1109/CVPR.2017.690.
5. Bochkovskiy A, Wang CY, Liao HYM. YOLOv4: optimal speed and accuracy of object detection. arXiv:2004.10934. 2020.
6. 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 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2023 Jun 17–24; Vancouver, BC, Canada. New York, NY, USA: IEEE; 2023. p. 7464–75. doi:10.1109/CVPR52729.2023.00721.
7. Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu CY, et al. SSD: single shot MultiBox detector. In: Computer vision—ECCV 2016. Cham, Switzerland: Springer; 2016. p. 21–37. doi:10.1007/978-3-319-46448-0_2.
8. Xiao Y, Xu T, Xin Y, Li J. FBRT-YOLO: faster and better for real-time aerial image detection. Proc AAAI Conf Artif Intell. 2025;39(8):8673–81. doi:10.1609/aaai.v39i8.32937.
9. Zhang Y, Ye M, Zhu G, Liu Y, Guo P, Yan J. FFCA-YOLO for small object detection in remote sensing images. IEEE Trans Geosci Remote Sens. 2024;62:5611215. doi:10.1109/TGRS.2024.3363057.
10. Zhang Q, Liu L, Yang Z, Yin J, Jing Z. WLSD-YOLO: a model for detecting surface defects in wood lumber. IEEE Access. 2024;12:65088–98. doi:10.1109/ACCESS.2024.3395623.
11. Mi Z, Gao Y, Xu X, Tang J. Steel strip surface defect detection based on multiscale feature sensing and adaptive feature fusion. AIP Adv. 2024;14(4):045005. doi:10.1063/5.0196580.
12. Xia Y, Lu Y, Jiang X, Xu M. Enhanced multiscale attentional feature fusion model for defect detection on steel surfaces. Pattern Recognit Lett. 2025;188:15–21. doi:10.1016/j.patrec.2024.11.024.
13. Wang Z, Zhou W, Li Y. MSAF-YOLO: an efficient multi-scale attention fusion network for high-precision steel surface defect detection. Measurement. 2026;257:118640. doi:10.1016/j.measurement.2025.118640.
14. Chu Y, Yu X, Rong X. A lightweight strip steel surface defect detection network based on improved YOLOv8. Sensors. 2024;24(19):6495. doi:10.3390/s24196495.
15. Hao Z, Li Z, Ren F, Lv S, Ni H. Strip steel surface defects classification based on generative adversarial network and attention mechanism. Metals. 2022;12(2):311. doi:10.3390/met12020311.
16. Khedmatgozar Dolati SS, Caluk N, Mehrabi A, Khedmatgozar Dolati SS. Non-destructive testing applications for steel bridges. Appl Sci. 2021;11(20):9757. doi:10.3390/app11209757.
17. Usamentiaga R, Sfarra S, Fleuret J, Yousefi B, García DF. Rail inspection using active thermography to detect rolled-in material. In: Proceedings of the 2018 International Conference on Quantitative InfraRed Thermography; 2018 Jun 25–29; Berlin, Germany. Québec, QC, Canada: QIRT Council; 2018. p. 845–52. doi:10.21611/qirt.2018.107.
18. Qiu G, Kang Y, Tang J, Feng B, Cai X, Ma H. Normal magnetizing-based eddy current testing method for surface crack and internal delamination of steel plate. In: Proceedings of the 2023 IEEE International Instrumentation and Measurement Technology Conference (I2MTC); 2023 May 22–25; Kuala Lumpur, Malaysia. New York, NY, USA: IEEE; 2023. p. 1–6. doi:10.1109/I2MTC53148.2023.10175887.
19. Wang G, Xiao Q, Gao Z, Li W, Jia L, Liang C, et al. Multifrequency AC magnetic flux leakage testing for the detection of surface and backside defects in thick steel plates. IEEE Magn Lett. 2022;13:8102105. doi:10.1109/LMAG.2022.3142717.
20. Sun Q, Cai J, Sun Z. Detection of surface defects on steel strips based on singular value decomposition of digital image. Math Probl Eng. 2016;2016:5797654. doi:10.1155/2016/5797654.
21. Tang B, Chen L, Sun W, Lin ZK. 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.
22. Girshick R, Donahue J, Darrell T, Malik J. Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition; 2014 Jun 23–28; Columbus, OH, USA. New York, NY, USA: IEEE; 2014. p. 580–7. doi:10.1109/CVPR.2014.81.
23. Girshick R. Fast R-CNN. In: Proceedings of the IEEE International Conference on Computer Vision; 2015 Dec 7–13; Santiago, Chile. p. 1440–8. doi:10.1109/ICCV.2015.169.

24. 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. doi:10.3390/s22093467.
25. Zhou Y, Zhao Z. MPA-YOLO: steel surface defect detection based on improved YOLOv8 framework. Pattern Recognit. 2025;168:111897. doi:10.1016/j.patcog.2025.111897.
26. Chen Z, Fan W, Guo B. ELG-YOLO: multi-scale feature fusion and attention mechanisms for enhanced steel surface defect detection. IEEE Access. 2026;14:5660–74. doi:10.1109/ACCESS.2025.3649757.
27. Huang J, Zhang X, Jia L, Zhou Y. An improved you only look once model for the multi-scale steel surface defect detection with multi-level alignment and cross-layer redistribution features. Eng Appl Artif Intell. 2025;145:110214. doi:10.1016/j.engappai.2025.110214.
28. Zhao C, Shu X, Yan X, Zuo X, Zhu F. RDD-YOLO: a modified YOLO for detection of steel surface defects. Measurement. 2023;214:112776. doi:10.1016/j.measurement.2023.112776.
29. Zheng T, Yu L, Shi Y, Niu F. RAGA-YOLO: enhancing global structural perception for accurate and efficient surface defect detection. J Comput Des Eng. 2026;13(1):324–51. doi:10.1093/jcde/qwaf138.
30. Ma R, Chen J, Feng Y, Zhou Z, Xie J. ELA-YOLO: an efficient method with linear attention for steel surface defect detection during manufacturing. Adv Eng Inform. 2025;65:103377. doi:10.1016/j.aei.2025.103377.
31. Zhang T, Pang H, Jiang C. GDM-YOLO: a model for steel surface defect detection based on YOLOv8s. IEEE Access. 2024;12:148817–25. doi:10.1109/ACCESS.2024.3476908.
32. Chen H, Du Y, Fu Y, Zhu J, Zeng H. DCAM-net: a rapid detection network for strip steel surface defects based on deformable convolution and attention mechanism. IEEE Trans Instrum Meas. 2023;72:5005312. doi:10.1109/TIM.2023.3238698.
33. Liao J, Wang D, Liang J, Yang M, Liang C. Enhanced multi-scale feature fusion for accurate steel surface defect detection. Vis Comput. 2026;42(9):401. doi:10.1007/s00371-026-04575-y.
34. Zhou Z, Cao Y. Enhanced multi-scale defect detection in steel surfaces *via* innovative deep learning architecture. Sensors. 2026;26(6):2001. doi:10.3390/s26062001.
35. Hu J, Shen L, Sun G. Squeeze-and-excitation networks. In: Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2018 Jun 18–23; Salt Lake City, UT. p. 7132–41. doi:10.1109/cvpr.2018.00745.
36. Woo S, Park J, Lee JY, Kweon IS. CBAM: convolutional block attention module. In: Computer vision—ECCV 2018. Cham, Switzerland: Springer; 2018. p. 3–19. doi:10.1007/978-3-030-01234-2_1.
37. Ouyang D, He S, Zhang G, Luo M, Guo H, Zhan J, et al. Efficient multi-scale attention module with cross-spatial learning. In: Proceedings of the ICASSP 2023—2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP); 2023 Jun 4–10; Rhodes Island, Greece. New York, NY, USA: IEEE; 2023. p. 1–5. doi:10.1109/ICASSP49357.2023.10096516.
38. Wang Q, Wu B, Zhu P, Li P, Zuo W, Hu Q. ECA-net: efficient channel attention for deep convolutional neural networks. In: Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2020 Jun 13–19; Seattle, WA, USA. p. 11531–9. doi:10.1109/cvpr42600.2020.01155.
39. Liu X, Liu J, Tang J, Wu G. CATANet: efficient content-aware token aggregation for lightweight image super-resolution. In: Proceedings of the 2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2025 Jun 10–17; Nashville, TN, USA. New York, NY, USA: IEEE; 2025. p. 17902–12. doi:10.1109/CVPR52734.2025.01668.
40. Liu J, Chen C, Tang J, Wu G. From coarse to fine: hierarchical pixel integration for lightweight image super-resolution. Proc AAAI Conf Artif Intell. 2023;37(2):1666–74. doi:10.1609/aaai.v37i2.25254.