Computers, Materials & Continua

Tech Science Press

https://doi.org/10.32604/cmc.2026.085242

ARTICLE

EFAS-YOLO: A Lightweight Edge-Frequency Aware YOLOv11 Framework for Steel Surface Defect Detection

Jiahui Liu, Longzhen Dong* and Zeling Hou

School of Computer Science and Technology, Harbin University of Science and Technology, Harbin, China

*Corresponding Author: Longzhen Dong. Email: 2420400026@stu.hrbust.edu.cn

Received: 07 May 2026; Accepted: 31 July 2026; Published: 15 September 2026

ABSTRACT: Detecting surface defects on steel is challenging because many defect regions are visually weak, have blurred boundaries, and contain minimal pixel information. In detectors from the You Only Look Once (YOLO) family, these subtle cues may be weakened at the early feature extraction stage and further attenuated during repeated downsampling. To improve the preservation and utilization of such defect-related details, this paper proposes EFAS-YOLO, a lightweight YOLOv11-based detection framework for steel surface defect inspection. First, an Edge-Frequency Aware Stem (EFAS) is introduced before the backbone to explicitly extract Sobel-based gradient responses and fuse them with learnable shallow texture features, allowing edge-sensitive information to be retained from the input stage. Second, the backbone channel configuration and shallow receptive field are adjusted to better match the feature distribution produced by EFAS and to avoid unnecessary channel expansion. Third, a Small Defect Enhancement Path (SDEP) is constructed to transmit enhanced P3-level high-resolution features to the corresponding neck branch through a lightweight residual path, reducing the loss of spatial details for small defects. Experiments on NEU-DET show that EFAS-YOLO achieves 79.8% mean average precision at an intersection-over-union threshold of 0.5 (mAP@0.5), outperforming YOLOv11n by 3.6 percentage points while maintaining a lightweight scale of 2.1M parameters and 158 FPS on an RTX 3090 GPU. Additional validation on GC10-DET achieves 68.3% mAP@0.5, suggesting that the proposed design maintains stable performance across different steel surface defect datasets.

KEYWORDS: Real-time surface defect detection; strip steel inspection; convolutional neural network; YOLOv11-based object detection

1 Introduction

The steel industry is a fundamental pillar of high-end equipment manufacturing and infrastructure construction. During the rolling and transportation of strip steel, various surface defects, such as scratches, cracks, and rolled-in scale, are prone to occur on steel surfaces due to complex manufacturing processes and environmental conditions. These defects not only impair the appearance of steel products but may also induce stress concentration, thereby severely weakening the fatigue strength and safety performance of the material [1,2]. Therefore, achieving real-time and high-accuracy automatic surface defect detection on high-speed production lines is of great significance for improving product yield and ensuring production safety.

Manual inspection and traditional machine vision methods have long been applied in this field. However, the former suffers from low inspection efficiency and inconsistent inspection standards, while the latter relies heavily on hand-crafted features and thus exhibits limited generalization capability [3]. In recent years,

deep learning built upon convolutional neural networks (CNNs) has dominated industrial defect detection tasks, benefiting from its superior ability to learn discriminative feature representations automatically.

Current mainstream object detection frameworks can generally be categorized into three paradigms: two-stage detectors, such as Faster Region-based Convolutional Neural Network (Faster R-CNN) [4]; one-stage detectors, such as the YOLO series [5]; and end-to-end Transformer-based detectors, such as Real-Time Detection Transformer (RT-DETR) [6]. Two-stage object detection frameworks usually achieve favorable detection accuracy, but their relatively high inference latency limits their applicability to high-throughput industrial production lines with strict real-time requirements. End-to-end Transformer-based models exhibit strong global modeling capability, but they typically require substantial computational resources.

However, directly applying a standard YOLOv11 detector to steel surface inspection is not fully sufficient. The difficulty of this task does not only come from scale variation, but also from the fact that many defects appear as faint gray-level changes or fragmented edge structures. Once these weak responses are suppressed in the shallow convolutional stages, later feature pyramid fusion can only combine the remaining information and cannot fully reconstruct the lost spatial details. This is particularly unfavorable for defects such as crazing and scratches, whose boundaries are often narrow, discontinuous, and close to the background texture. Therefore, improving this task requires a design that considers not only multi-scale fusion in the Neck, but also how edge-related cues are generated, transmitted, and reused throughout the network [7].

Recent industrial defect detection studies have increasingly explored edge enhancement, high-frequency information modeling, and attention-based feature fusion to improve the perception of weak boundaries and small defects. Wang et al. introduced a Sobel-operator-based edge information extraction module for steel defect detection, showing that explicit contour cues are beneficial for distinguishing irregular and visually similar defect regions [8]. Jiang et al. proposed a joint attention-guided feature fusion network for surface defect saliency detection, where channel-spatial attention is used to emphasize low-contrast defect responses and suppress background interference [9]. More recently, LCED-YOLO incorporated an edge information enhancement module into a YOLOv11-based steel defect detector and demonstrated that edge-aware feature enhancement can improve defect representation while maintaining lightweight computation [10]. However, most existing methods still enhance edge or texture features after shallow convolution or within intermediate feature fusion stages, where part of the weak high-frequency information may already have been attenuated. Therefore, it is necessary to introduce stable edge-related priors at the input stage and maintain their effective transmission throughout the backbone and neck.

To address the above issues, we propose EFAS-YOLO, an improved YOLOv11 framework specifically designed for steel surface defect detection, as illustrated in Fig. 1. First, to alleviate the loss of weak-texture features, an Edge-Frequency Aware Stem (EFAS) module is designed. This module introduces a Sobel-operator-based high-frequency gradient prior at the input stage, thereby enhancing the model's perception of weak-texture defects. Second, to improve the propagation and utilization of edge-enhanced shallow features, the backbone channel configuration and shallow receptive field are adaptively adjusted. This design reduces abrupt channel expansion and improves the compatibility between EFAS-enhanced features and the subsequent feature extraction network. Third, to mitigate the loss of spatial details in small objects, a Small Defect Enhancement Path (SDEP) is constructed. By introducing a shallow high-resolution feature pathway, SDEP alleviates feature degradation caused by multi-level downsampling.

The main contributions are summarized as follows:

1. A lightweight EFAS-YOLO framework is proposed for weak-texture and small-scale steel surface defect detection. Different from methods that mainly enhance features in the Neck or Head, the proposed framework considers the complete information flow from input-level edge perception to backbone transmission and P3-level detail preservation.

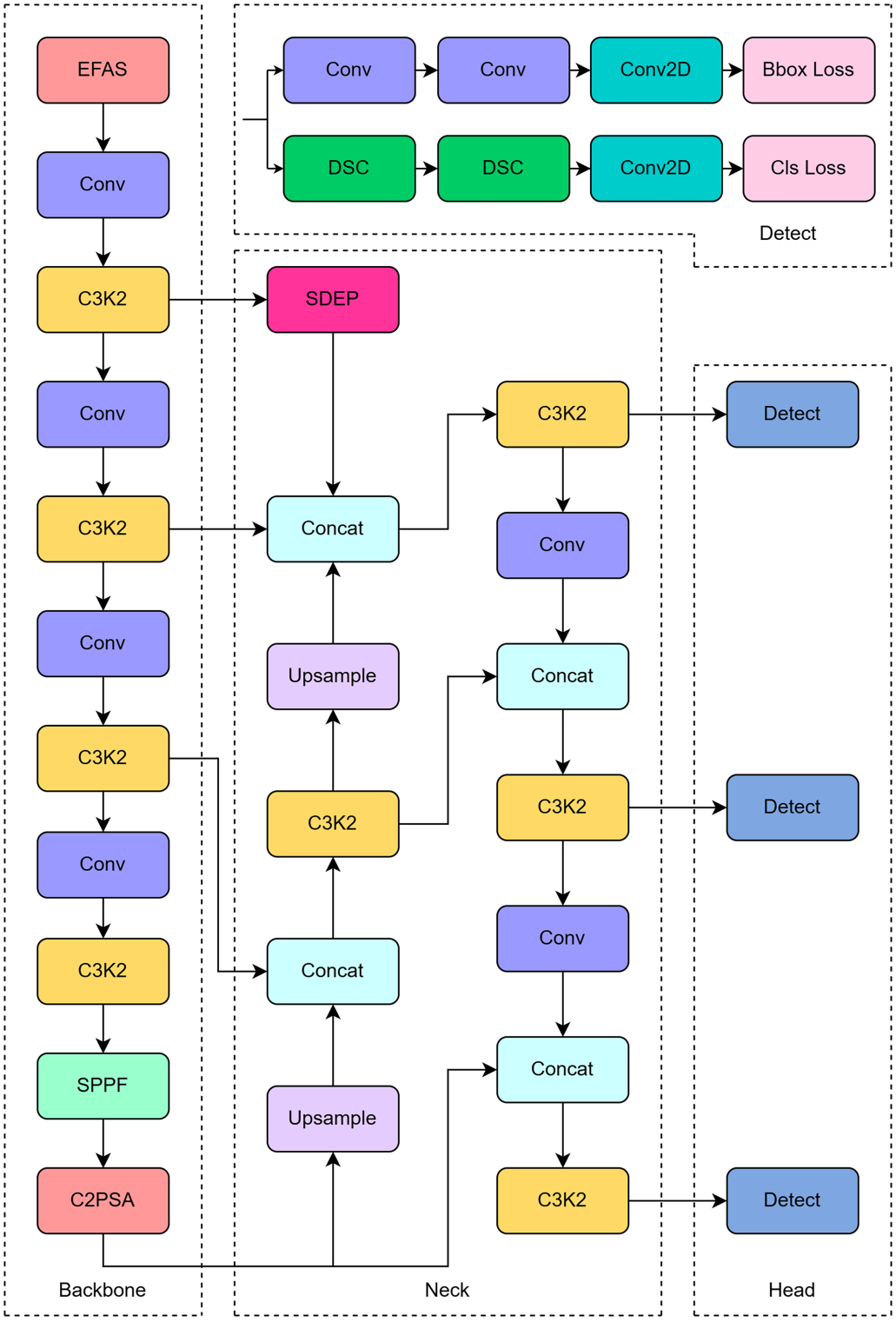

Figure 1: EFAS-YOLO model architecture.

2. An Edge-Frequency Aware Stem is designed at the input side of YOLOv11. It introduces fixed Sobel-gradient structural priors and learnable shallow texture features before repeated downsampling, improving the perception of weak defect boundaries.

3. A backbone adaptation strategy is introduced to improve the compatibility between EFAS-enhanced features and subsequent feature extraction. The progressive channel configuration and shallow receptive-field adjustment reduce abrupt feature transformation while maintaining lightweight computation.

4. A Small Defect Enhancement Path is constructed to provide a short residual route for P3-level high-resolution features, allowing small-defect details to participate more directly in Neck fusion. Experiments on NEU-DET and GC10-DET verify the effectiveness and lightweight characteristics of the proposed method.

2 Related Work

2.1 Research Status of Steel Surface Defect Detection

Steel surface defect detection is an important research topic in industrial visual inspection. Early studies mainly relied on manual inspection and traditional machine vision methods, in which defect recognition was achieved through threshold segmentation, edge detection, texture description, and classifier design. However, such methods are sensitive to illumination variations, noise interference, and changes in defect morphology, resulting in limited robustness and generalization capability [11]. With the development of deep learning, steel surface defect detection has gradually shifted from hand-crafted feature design to data-driven feature learning, and related studies have further expanded from defect classification and segmentation to object detection frameworks [12]. Existing reviews have shown that steel surface defect detection has formed a relatively complete research system, ranging from traditional vision-based methods to deep learning-based approaches. Among them, one-stage detectors have become an important technical route in this field because they can achieve a favorable balance between detection accuracy and inference speed [13].

Although deep learning methods have significantly improved the performance of steel surface defect detection, characteristics such as weak textures, small sizes, and imbalanced category distributions still pose adaptation challenges for general object detection frameworks in this scenario.

2.2 Small Object Detection and Multi-Scale Feature Fusion Methods

To address scale variations in object detection, multi-scale feature fusion has become a key technique. Feature Pyramid Network (FPN) realizes the fusion of high-level semantic information and shallow spatial details through a top-down pathway and lateral connections [7]. Path Aggregation Network (PANet) further introduces a bottom-up path augmentation mechanism based on FPN, thereby shortening the transmission path from low-level localization information to high-level semantic features [14]. EfficientDet employs Bidirectional Feature Pyramid Network (BiFPN) to achieve more efficient bidirectional feature fusion [15]. In recent years, studies on small object detection for steel surface defects have further strengthened this research direction. For example, SRN-YOLO [16], YOLOv8-BSPB [17], YOLO-SDS [18], and DSL-YOLO [19] improved detection performance from the perspectives of shallow feature preservation, multi-scale enhancement, lightweight design, and small-object representation, respectively. MPA-YOLO [20] further optimized multi-path feature extraction and multi-scale perception.

In addition to CNN-based feature pyramids, recent Transformer-based and remote sensing detectors have also revisited multi-scale feature representation and fusion. Liu et al. [21] rethought the multi-scale feature hierarchy in Detection Transformer (DETR) and showed that more effective hierarchical feature organization is important for improving detection performance in Transformer-based detectors. Wang et al. [22] proposed an attention-guided feature fusion strategy for drone-based remote sensing object

detection, demonstrating that attention-driven cross-scale feature interaction can enhance the recognition of small and complex objects. These studies indicate that multi-scale hierarchy design and adaptive feature fusion remain important issues in modern object detection. However, for steel surface defects with weak textures and blurred boundaries, feature fusion alone may still be insufficient if edge cues and fine spatial details have already been weakened in the early feature extraction stages.

However, most of the above methods are built upon feature representations obtained after progressive propagation through the backbone network. Their main focus lies in cross-layer feature fusion rather than suppressing the attenuation of small-object details during forward propagation. For tiny defects on steel surfaces that occupy only a few pixels, once their spatial details have been weakened through repeated convolution and downsampling operations, subsequent feature fusion modules often struggle to effectively recover them. Therefore, how to establish a shorter and lower-loss high-resolution feature transmission pathway remains an issue worthy of further investigation in steel small-defect detection.

2.3 Edge and High-Frequency Information Modeling Methods

For weak-texture and low-contrast targets, modeling edge and high-frequency information is an important strategy for improving detection performance. In traditional methods, edge operators such as Canny [23] can explicitly extract image gradients and boundary structure information, providing direct priors for object contours. In deep learning-based methods, Holistically-Nested Edge Detection (HED) [24] further formulates edge detection as an end-to-end learning problem and emphasizes the importance of multi-scale hierarchical features for edge modeling. In steel surface defect detection, recent studies have also paid increasing attention to edge and texture information enhancement. For example, RSTD-YOLOv7 [25] alleviates texture information loss by enlarging the receptive field and introducing an attention mechanism. An improved YOLOv9-based method [26] enhances detection capability in complex backgrounds through lightweight convolution and feature enhancement. LCED-YOLO [10] further explicitly incorporates an edge information enhancement module into the YOLOv11 framework to strengthen defect edge representation.

Nevertheless, existing edge or high-frequency enhancement methods still have certain limitations. On the one hand, many methods introduce edge enhancement branches in the middle or late stages of the network, where high-frequency information has already undergone multiple rounds of convolution and downsampling, making it difficult to fully recover. On the other hand, some methods rely heavily on complex modules or learnable branches, whose stability may still be limited in industrial defect scenarios involving small samples, imbalanced categories, and subtle texture differences. Therefore, explicitly injecting stable high-frequency priors from the input stage and collaboratively modeling them with shallow texture features remains of considerable research value.

2.4 Limitations of Existing Studies and Motivation of This Work

Although existing studies have improved steel surface defect detection from the perspectives of multi-scale fusion, attention mechanisms, and edge feature enhancement, two issues remain insufficiently addressed. First, many methods perform feature enhancement after several convolution and downsampling stages. At this point, part of the weak edge response and fine spatial detail may already have been weakened, especially for tiny or low-contrast defects. Second, feature fusion in the Neck can shorten the information path between different scales, but it does not directly solve the problem of detail loss before the shallow features arrive at the fusion stage.

Motivated by these observations, this work focuses on the continuity of defect-related information transmission. EFAS is placed at the front of the network to provide explicit gradient priors before early feature abstraction. The backbone is then adjusted to reduce the mismatch between edge-enhanced shallow features

and subsequent channel expansion. Finally, SDEP introduces a shorter P3-level detail transmission path so that high-resolution local cues can participate more directly in Neck fusion. Therefore, the proposed method is not a simple combination of independent modules, but a coordinated design for edge cue extraction, feature propagation, and small-defect detail preservation.

3 EFAS-YOLO Model

To address the attenuation of weak-texture and small-defect features, this paper proposes EFAS-YOLO based on YOLOv11. The framework follows the Backbone–Neck–Head paradigm and introduces three task-oriented components: an Edge-Frequency Aware Stem for input-level edge prior injection, an adapted backbone for smoother feature transmission, and a Small Defect Enhancement Path for P3-level detail preservation.

Compared with the original YOLOv11n, EFAS-YOLO differs in three aspects: the standard input stem is replaced by EFAS to introduce fixed Sobel-gradient priors and learnable shallow texture features; the shallow backbone is adjusted from rapid channel expansion to a progressive 64–128–256–512 configuration with a 5 × 5 first C3K2 block; and an SDEP residual path is added from the P3 backbone feature to the P3 neck fusion node to preserve high-resolution small-defect details.

3.1 Edge-Frequency Aware Stem

3.1.1 Adaptation Limitations of Existing Methods

Weak-texture defects mainly appear as subtle edge structures and local gray-level variations. To reduce the dependence on purely learnable shallow convolutions, EFAS introduces a fixed Sobel-based gradient branch at the input stage and combines it with a learnable shallow texture branch, as shown in Fig. 2.

3.1.2 Explicit Extraction of Non-Parametric High-Frequency Edges

Image edge gradients inherently reflect high-frequency structural information in images. For weak-texture defects, the gray-level contrast is often low, making it difficult to distinguish defective regions from the background using intensity information alone. However, directional gradient variations around defect boundaries usually remain discriminative, since they can characterize local structural changes in images. Therefore, explicitly introducing Sobel-based gradient priors at the input stage can provide the model with stable high-frequency structural information, thereby improving the separability between defect regions and the background.

In this paper, fixed Sobel operators are introduced to construct a non-learnable edge extraction branch. Let the input image be denoted as $I \in \mathrm{R}^{C\times H\times W}$. The horizontal Sobel kernel K_x and the vertical Sobel kernel K_y are used to calculate the directional gradient responses as follows:

$$G_x = K_x * I, \quad G_y = K_y * I \tag{1}$$

where $*$ denotes the convolution operation. The two directional gradient responses are then concatenated along the channel dimension:

$$G_0 = Concat\left(G_x, G_y\right) \tag{2}$$

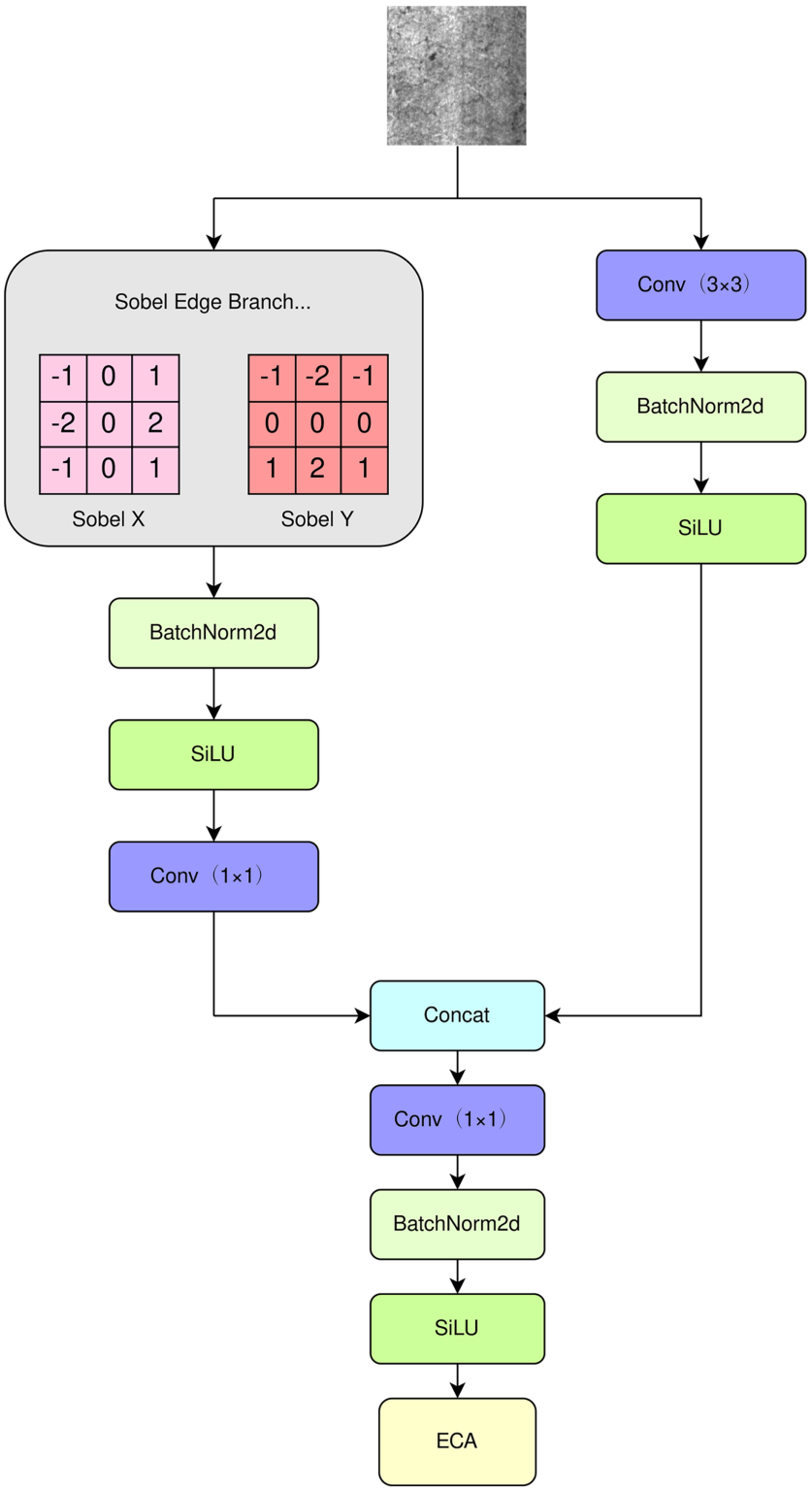

Figure 2: Structure of the EFAS module.

To improve the adaptability of the fixed gradient responses to subsequent feature extraction layers, the concatenated Sobel responses are further processed by BatchNorm, SiLU activation, and a 1 × 1 convolution:

$$G = Conv_{1\times1}\left(\delta\left(BN\left(G_0\right)\right)\right) \tag{3}$$

where $BN(\cdot)$ denotes batch normalization and $\delta(\cdot)$ denotes the SiLU activation function. In the network implementation, the Sobel convolution kernels are registered as fixed buffers and are not updated during backpropagation. This design provides the network with stable directional gradient responses and reduces the difficulty of learning edge structures from scratch in shallow layers. In addition, introducing high-frequency gradient priors at the input stage can, to some extent, alleviate the suppression of weak edge information caused by subsequent convolution and downsampling operations.

3.1.3 Multi-Source Feature Fusion and Channel Recalibration

Building upon the directional high-frequency edge extraction, EFAS feeds the input features into two parallel branches: a fixed Sobel edge branch and a learnable shallow texture branch. The Sobel edge branch extracts directional gradient responses using fixed Sobel-X and Sobel-Y operators, followed by BatchNorm, SiLU activation, and a 1 × 1 convolution for feature normalization, non-linear mapping, and channel embedding. The learnable shallow texture branch consists of a 3 × 3 convolution with stride 1, followed by BatchNorm and SiLU activation, which is used to complementarily model shallow textures, grayscale variations, and local structural responses in the original image.

In the implementation, the learnable texture branch adopts a 3 × 3 convolution with stride 1 and padding 1. The Sobel branch uses fixed 3 × 3 Sobel-X and Sobel-Y kernels, followed by BatchNorm, SiLU activation, and a 1 × 1 convolution for channel embedding. After concatenation, the fused feature is projected to 64 channels through a 1 × 1 convolution.

Let the output of the Sobel edge branch be denoted as $G \in \mathrm{R}^{C_g\times H\times W}$, and the output of the learnable shallow texture branch be denoted as $F \in \mathrm{R}^{C_f\times H\times W}$. Considering the differences in statistical distributions and semantic attributes between fixed gradient priors and learnable texture features, channel concatenation is adopted instead of element-wise addition to preserve the representational independence of the two branches. The fusion process can be expressed as:

$$Z = Concat\left(G, F\right) \in \mathbb{R}^{\left(C_g+C_f\right)\times H\times W} \tag{4}$$

After concatenation, a lightweight 1 × 1 convolutional projection is introduced to fuse the heterogeneous branch features and align the channel dimension:

$$\hat{Z} = \delta\left(BN\left(Conv_{1\times1}(Z)\right)\right) \tag{5}$$

where $\hat{Z} \in \mathbb{R}^{64\times H\times W}$. This projection performs lightweight cross-channel interaction between the Sobel-gradient priors and learnable shallow texture features, while providing a 64-channel edge-aware representation for the subsequent backbone.

The projected feature $\hat{Z}$ is then fed into the Efficient Channel Attention (ECA) module, as shown in Fig. 3, for adaptive channel-wise importance recalibration. Specifically, ECA generates channel weights through global average pooling, one-dimensional convolution, and sigmoid activation, and then reweights the projected feature channels. This process emphasizes defect-related edge and texture response channels while suppressing redundant background responses, finally producing the EFAS output feature X.

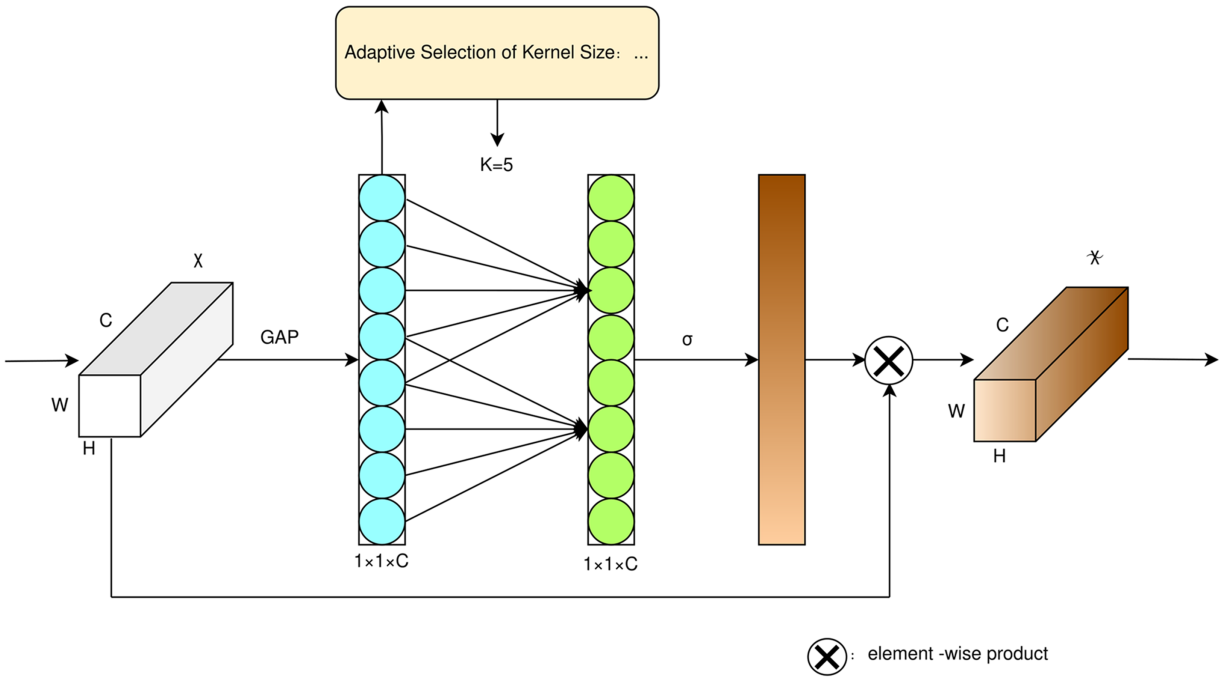

Figure 3: Architecture of the ECA module.

3.2 Adaptive Adjustment of the Backbone Network

The EFAS module changes the shallow feature distribution received by the backbone by introducing explicit gradient and local texture responses. Therefore, the subsequent backbone should not simply follow the original YOLOv11 configuration. In this work, the backbone is adapted from two aspects: progressive channel configuration and shallow receptive-field adjustment.

3.2.1 Progressive Channel Configuration Optimization

The original YOLOv11 adopts a relatively rapid channel expansion strategy in the shallow stages, where the 32-channel features output by the standard Stem are quickly transformed into high-dimensional feature representations in subsequent layers. This design is suitable for semantic feature extraction in general scenarios. However, for EFAS output features that already contain edge-enhanced information, excessively rapid channel expansion may introduce computational redundancy and increase the risk of abrupt changes in fine-grained feature distributions.

In this paper, the channel configuration of the backbone network is adjusted into a progressive doubling structure, namely EFAS(64) → C3K2(128) → C3K2(256) → C3K2(512), to achieve smooth channel transition and effectively control computational redundancy. While maintaining the hierarchical progression of the feature pyramid, this design reduces feature distribution mismatch and computational redundancy caused by abrupt channel changes in shallow layers, which is beneficial for the stable transmission of high-frequency features output by EFAS.

3.2.2 Differential Receptive Field Parameter Adaptation

The C3K2 module is a core feature extraction unit in YOLOv11, and the kernel size is one of the important factors affecting the local receptive field. In the original YOLOv11 backbone, the main convolutional operations in shallow C3K2 blocks are mainly based on 3 × 3 kernels. However, this configuration is

insufficiently adaptive to EFAS output features rich in defect edge structures, making it difficult to adequately aggregate contextual information around local edge responses and thereby limiting the construction of complete defect contour representations.

To address this issue, this paper adopts a differential receptive field configuration according to the feature characteristics at different network levels. Specifically, the convolution kernel size of the first C3K2 module is adjusted to 5 × 5, enlarging the shallow receptive field to facilitate contextual aggregation of edge features. The deeper C3K2 modules retain the default 3 × 3 convolution kernel configuration, thereby controlling the overall computational cost while maintaining semantic feature extraction capability.

Since shallow features still preserve relatively high spatial resolution and abundant edge details, moderately enlarging their receptive field is conducive to aggregating local edge context. In contrast, extensively applying large convolution kernels to deeper features would increase computational cost and weaken the lightweight advantage of the model. Therefore, this paper only applies differential receptive field adaptation to the shallow C3K2 module.

3.3 Small Defect Enhancement Path

To reduce the loss of small-defect details during long-path feature propagation, this paper introduces a Small Defect Enhancement Path (SDEP). The detailed structure of SDEP is illustrated in Fig. 4. In the original YOLO-style structure, P3 features need to pass through several convolution, upsampling, and downsampling operations before they are finally used by the detection branch. For small defects, this long route may weaken fine local textures and boundary cues.

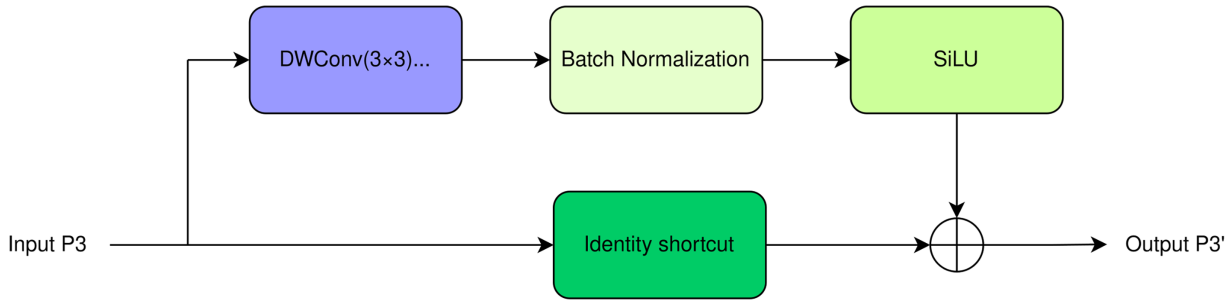

Figure 4: Architecture of the SDEP module.

SDEP builds a direct high-resolution path from the backbone P3 output to the corresponding P3 fusion node in the Neck. Since P3 retains 1/8 of the input resolution, it still contains relatively rich spatial information and is closely related to small-object localization. In the proposed path, the P3 feature is first enhanced by a lightweight channel-preserving depthwise convolution. A residual connection is then used to preserve the original shallow representation while supplementing local texture and edge responses. Specifically, the enhancement operation in SDEP is implemented by a 3 × 3 depthwise convolution with stride 1 and padding 1, where the number of groups is equal to the number of input channels. This design preserves the channel dimension of the P3 feature while introducing only a small amount of additional computation. Let the original P3 feature be denoted as F_{p3} and the depthwise enhancement operation be denoted as $H(\cdot)$. The SDEP output is formulated as $F_{sdep} = F_{p3} + H(F_{p3})$, where "+" denotes element-wise addition. After residual enhancement, F_{sdep} is concatenated with the corresponding Neck feature along the channel dimension, i.e., $F_{fuse} = \mathrm{Concat}(F_{sdep}, F_{neck})$. Through this design, SDEP does not replace the original Neck fusion process. Instead, it provides an additional low-loss route for shallow high-resolution information, allowing small-defect details to participate more directly in the final P3 detection branch.

4 Experiments and Results

4.1 Dataset

This work adopts the publicly accessible NEU-DET dataset released by Northeastern University. The dataset covers six typical surface defects of hot-rolled steel strips: crazing (Cr), inclusion (In), patches (Pa), pitted surface (Ps), rolled-in scale (Rs), and scratches (Sc), as illustrated in Fig. 5. In total, NEU-DET contains 1800 grayscale images, with 300 samples for each defect category. For model training, validation and performance evaluation, the entire dataset is randomly partitioned into three subsets at an 8:1:1 ratio, corresponding to the training set, validation set and test set, respectively. The split was performed by class-stratified random sampling, and the generated training, validation, and test lists were fixed and shared by all compared models to ensure fair evaluation.

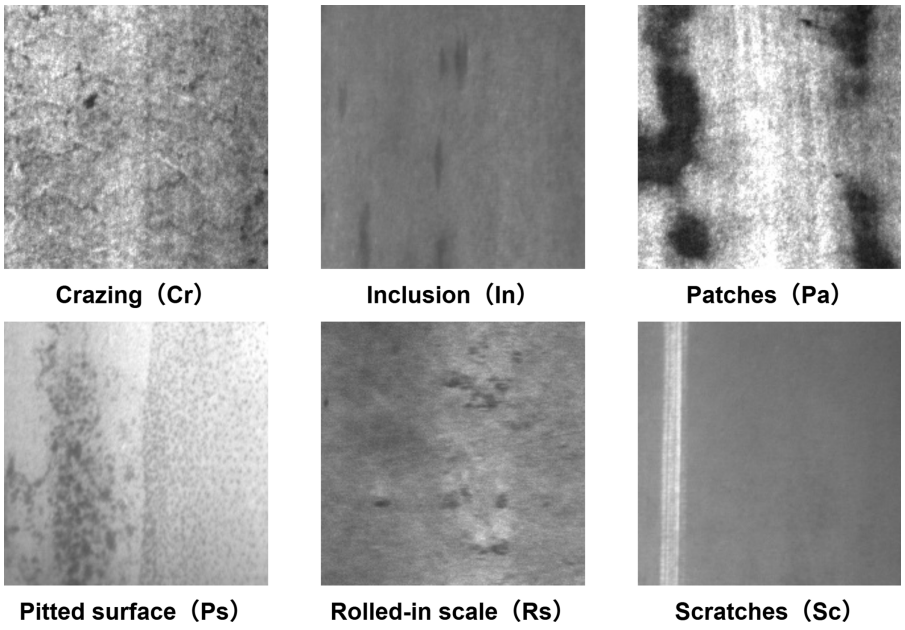

Figure 5: Six types of defects in the NEU-DET dataset.

4.2 Experimental Settings

All experiments were conducted on Linux with an NVIDIA GeForce RTX 3090 GPU, using PyTorch 2.1.0, Python 3.10, and CUDA 12.1. For fair comparison, all YOLO-based models were trained under the same data split, input size, optimizer, initial learning rate, training batch size, number of epochs, and data augmentation strategy, including random horizontal flipping, scaling, translation, and hue–saturation–value (HSV) intensity perturbation. For models implemented in different frameworks, the input resolution and epoch number were kept consistent, and the training settings were kept as close as possible. For inference speed evaluation, all models were tested on the same RTX 3090 GPU with an input size of 640 × 640 and an inference batch size of 1. TensorRT acceleration was not used, AMP-based mixed precision was enabled, several warm-up iterations were performed before timing, and FPS was calculated by averaging repeated inference runs under the same post-processing settings, including the NMS IoU threshold. The main training and inference parameters are summarized in Table 1.

Table 1: Training and inference parameter settings.

Parameter	Setting
Optimizer	SGD
Initial learning rate	0.01
Momentum	0.937
Training batch size	16
Inference batch size	1
Input image size	640 × 640
Training epochs	200
NMS IoU threshold	0.7
Random seed for comparative experiments	0
Random seeds for ablation experiments	0, 1, 2, 3, 4
Deterministic	TRUE
FP16	Mixed precision enabled via AMP, not pure FP16
TensorRT	Not used
Warm-up strategy	warmup_epochs = 3.0, warmup_momentum = 0.8, warmup_bias_lr = 0.1

4.3 Evaluation Metrics

To comprehensively validate the detection capability of the proposed model, six quantitative indicators are adopted for performance evaluation, including Precision, Recall, mAP, parameter count, FLOPs, and inference speed (FPS). Precision quantifies the prediction accuracy by calculating the proportion of correctly predicted positive samples in all detection outputs, whereas Recall characterizes the model's capacity to identify real defect targets. As a core evaluation indicator for object detection tasks, mAP provides a comprehensive assessment of detection results under diverse confidence thresholds. For fair comparison with existing mainstream methods, this study adopts mAP@0.5, i.e., mAP evaluated at an IoU threshold of 0.5, as the primary evaluation criterion. The corresponding mathematical formulas are defined in Eqs. (6)–(9).

$$Precision = \frac{TP}{TP + FP} \tag{6}$$

$$Recall = \frac{TP}{TP + FN} \tag{7}$$

$$AP = \int_0^1 Precision(Recall) d(Recall) \tag{8}$$

$$mAP = \frac{1}{n} \sum_{i=1}^{n} AP(i) \tag{9}$$

where TP, FP, and FN correspond to the counts of correctly localized targets, false predictions and missing targets, respectively; APi refers to the average precision of the i-th defect category, and n denotes the total number of defect classes. In Eq. (8), AP is expressed as the area under the precision–recall curve. In practical object detection evaluation, this value is calculated from discrete precision–recall points, while the integral form provides its continuous mathematical representation.

To further assess the practical deployment performance of the proposed model, this section investigates its computational complexity and inference efficiency. Specifically, the parameter count is adopted to quantify model memory overhead, while floating-point operations (FLOPs) are utilized to evaluate computational complexity. FLOPs reflect the total floating-point calculations consumed in the inference phase, and higher FLOP values correspond to greater computational consumption. Furthermore, frames per second (FPS) is

adopted as a critical metric to measure inference speed, which intuitively characterizes the model's real-time detection potential in actual industrial inspection scenarios.

4.4 Comparative Experiments

Tables 2 and 3 report the comparative results on NEU-DET.

Table 2: Overall comparison results of different detection methods on NEU-DET.

Method	mAP@0.5	mAP@0.5:0.95	Params (M)	FLOPs (G)	FPS
Faster R-CNN	70.9	37.6	134.7	83	11
YOLOv5n	71.3	38.8	2.6	7.7	104
YOLOv8n	74.8	42.6	3.2	8.7	117
YOLOv10n	75.5	43.2	2.3	6.7	113
YOLOv11n	76.2	44.3	2.6	6.5	135
YOLOv26n	76.7	44.1	2.4	5.4	143
RT-DETR-R18	71.5	38.2	20	60	39
RT-DETR-R50	73.3	39.5	42	136	23
RT-DETR-R101	74.2	41.7	76	259	14
EFAS-YOLO	79.8	47.2	2.1	5.7	158

Table 3: Comparison results of different detection methods on NEU-DET.

Method	Cr	In	Pa	Ps	Rs	Sc	ALL
YOLOv5n	36.3	83.7	86.0	68.6	58.8	94.2	71.3
YOLOv8n	40.9	83.1	89.4	78.9	61.0	95.5	74.8
YOLOv10n	41.7	85.0	90.5	84.0	57.1	94.7	75.5
YOLOv11n	41.3	82.2	89.6	86.9	63.1	94.1	76.2
YOLOv26n	41.5	84.7	90.7	84.8	63.6	94.7	76.7
RT-DETR-R18	38.2	81.2	85.7	73.9	59.7	90.3	71.5
RT-DETR-R50	38.3	82.8	89.3	75.1	62.2	92.1	73.3
RT-DETR-R101	39.5	83.5	90.2	75.7	64.0	92.3	74.2
EFAS-YOLO	45.8	85.5	91.9	91.0	66.8	97.6	79.8

EFAS-YOLO achieves 79.8% mAP@0.5, exceeding YOLOv11n by 3.6 percentage points. Compared with other lightweight one-stage detectors, the proposed method also performs better than YOLOv10n and YOLOv26n, indicating that the improvement is not only caused by the YOLOv11 baseline but also by the targeted feature enhancement design.

In terms of model complexity, EFAS-YOLO uses 2.1M parameters and 5.7 GFLOPs. Although EFAS and SDEP introduce additional branches, the adjusted backbone channel configuration reduces redundant channel expansion in shallow stages. Therefore, the final model remains lighter than YOLOv11n while achieving higher accuracy. The inference speed reaches 158 FPS on the RTX 3090 GPU, which is higher than YOLOv11n and YOLOv26n under the same experimental setting.

From the category-wise results, the improvement is more evident on defect types that rely strongly on local edge or texture cues. For example, EFAS-YOLO achieves 45.8% AP on Cr, which is higher than YOLOv11n by 4.5 percentage points. It also obtains 91.0% on Ps and 97.6% on Sc, showing that the proposed design improves both weak-boundary defects and relatively distinguishable texture defects. These results support the motivation that input-level edge priors and P3-level detail preservation are beneficial for steel surface defect detection.

4.5 Ablation Experiments

To verify the contribution of each improved module to model performance, ablation experiments are conducted on the NEU-DET dataset. To evaluate the stability of the proposed method, all ablation experiments were repeated under five random seeds, i.e., 0, 1, 2, 3, and 4. The mAP results are reported as mean ± standard deviation across the five runs. YOLOv11n is used as the baseline model, and the EFAS, SDEP, and backbone adaptation strategy are progressively introduced. The experimental results are shown in Table 4.

Table 4: Ablation study results of the proposed EFAS-YOLO on NEU-DET.

Model	EFAS	SDEP	Backbone Adapt	P (%)	R (%)	mAP@0.5	mAP@0.5:0.95
YOLOv11n	×	×	×	72.7	71.7	76.2 ± 0.22	44.3 ± 0.28
+EFAS	✓	×	×	73.2	72.6	77.2 ± 0.34	45.3 ± 0.32
+Backbone Adapt	×	×	✓	73.2	72.6	76.6 ± 0.27	45.0 ± 0.24
+SDEP	×	✓	×	74.5	73.2	77.4 ± 0.15	45.4 ± 0.18
+EFAS (Backbone Adapt)	✓	×	✓	75.0	73.5	78.0 ± 0.28	45.9 ± 0.30
+EFAS + SDEP	✓	✓	×	75.1	73.6	78.6 ± 0.22	46.5 ± 0.26
Full Model	✓	✓	✓	75.5	74.1	79.8 ± 0.24	47.2 ± 0.22

As shown in Table 4, all proposed components contribute positively to the detection performance. Compared with YOLOv11n, adding EFAS improves mAP@0.5 from 76.2% to 77.2%, indicating that input-level edge-frequency priors are beneficial for weak-texture defect representation. SDEP achieves 77.4% mAP@0.5, showing that the shallow high-resolution path helps reduce small-defect detail loss during feature propagation.

The improvement of the full EFAS-YOLO over YOLOv11n is much larger than the corresponding standard deviations across five random seeds, indicating that the performance gain is stable rather than being caused by random initialization.

The individual introduction of EFAS, SDEP, and backbone adaptation all improves the baseline, indicating that edge prior extraction, shallow detail preservation, and feature transmission adaptation are beneficial to steel surface defect detection. Among them, EFAS mainly improves the early perception of weak edge structures, while SDEP contributes to the preservation of high-resolution local details. The backbone adaptation alone brings a relatively limited gain, but when combined with EFAS, the performance further increases, suggesting that the adapted backbone improves the compatibility and propagation of EFAS-enhanced features. The full model achieves the best result, demonstrating that the performance gain comes from the coordinated design rather than a single isolated component.

4.5.1 Detailed Analysis of EFAS Components

To isolate the contribution of the internal EFAS design, all variants in Table 5 retain SDEP and Backbone Adapt, while only the EFAS/input-stem configuration is varied. Specifically, in the "EFAS removed" variant, the entire EFAS module, including the Sobel branch, learnable texture branch, ECA attention, and feature-fusion operation, is removed, and the original YOLOv11 stem is restored. The remaining variants use the indicated partial or complete EFAS configurations. All other network components and training settings remain unchanged.

Table 5: Ablation study results of EFAS components with SDEP and Backbone adapt retained on NEU-DET.

Variant	Sobel	Texture Branch	ECA	Fusion	mAP@0.5	mAP@0.5:0.95
EFAS removed	×	×	×	Original Stem	78.4 ± 0.14	45.0 ± 0.12
Texture only	×	✓	×	Single branch	78.6 ± 0.16	45.2 ± 0.16
Sobel only	✓	×	×	Single branch	79.0 ± 0.22	45.9 ± 0.16
Sobel + Texture w/o ECA	✓	✓	×	Concat + 1 × 1 Conv	79.4 ± 0.20	46.5 ± 0.22
Sobel + Texture + Add	✓	✓	×	Element-wise Add	79.2 ± 0.22	46.2 ± 0.15
Full EFAS	✓	✓	✓	Concat + 1 × 1 Conv	79.8 ± 0.24	47.2 ± 0.22

The results show that both branches of EFAS are useful, but their effects are different. The texture-only variant improves mAP@0.5 from 78.4% to 78.6%, while the Sobel-only variant reaches 79.0%, indicating that explicit gradient priors provide stronger guidance for weak and blurred defect boundaries. Combining Sobel and learnable texture features further increases mAP@0.5 to 79.4%, which verifies the complementarity between fixed edge priors and learnable shallow texture features. Compared with element-wise addition, concatenation followed by a 1×1 convolution obtains better results, suggesting that learnable channel mixing is more suitable for fusing heterogeneous edge and texture responses. The full EFAS with ECA attention achieves the best performance, confirming that channel reweighting helps suppress redundant shallow responses and strengthen defect-relevant cues.

4.5.2 Analysis of Edge Operator and Shallow Receptive Field

To examine the choice of the fixed edge operator in EFAS, Sobel is compared with Laplacian, Canny, and a learnable edge filter under the same backbone adaptation and SDEP settings. As shown in Table 6, Sobel achieves 79.8% mAP@0.5 and 47.2% mAP@0.5:0.95, which is comparable to the learnable edge filter in mAP@0.5 and slightly better in mAP@0.5:0.95. Canny and Laplacian produce lower results, suggesting that their edge responses are less compatible with the lightweight end-to-end detection framework used in this study. Therefore, Sobel is adopted as a simple and stable gradient prior.

Table 6: Ablation study results of different edge operators in EFAS on NEU-DET.

Edge Operator in EFAS	Backbone Adapt	SDEP	mAP@0.5	mAP@0.5:0.95
Laplacian	✓	✓	79.3 ± 0.22	46.0 ± 0.18
Canny	✓	✓	78.8 ± 0.18	45.6 ± 0.12
Learnable edge filter	✓	✓	79.8 ± 0.26	47.1 ± 0.16
Sobel	✓	✓	79.8 ± 0.24	47.2 ± 0.22

The shallow receptive-field setting in the adapted backbone is also evaluated. As shown in Table 7, using a 5 × 5 convolution in the first shallow C3K2 module and keeping the remaining C3K2 modules at 3 × 3 achieves a favorable balance between detection performance and model complexity. Although the K7 setting obtains a marginally higher mAP@0.5, its mAP@0.5:0.95 decreases from 47.2% to 47.0% and the FLOPs increase from 5.67 to 5.75 G. Therefore, the proposed K5 setting is retained as the final configuration.

Table 7: Ablation study results of different kernel-size settings in Backbone Adapt on NEU-DET.

Setting	C3K2-1	C3K2-2	C3K2-3	C3K2-4	mAP@0.5	mAP@0.5:0.95	Params (M)	FLOPs (G)
K3	3 × 3	3 × 3	3 × 3	3 × 3	79.3 ± 0.18	46.5 ± 0.20	2.132	5.62
K5 (proposed)	5 × 5	3 × 3	3 × 3	3 × 3	79.8 ± 0.24	47.2 ± 0.22	2.133	5.67
K7	7 × 7	3 × 3	3 × 3	3 × 3	79.9 ± 0.22	47.0 ± 0.18	2.134	5.75

4.6 Performance Curve Analysis

To further analyze the detection performance of the model, this paper compares the F1-Confidence curves, as shown in Fig. 6, and Precision-Recall curves, as shown in Fig. 7, of YOLOv11n and EFAS-YOLO on the NEU-DET dataset.

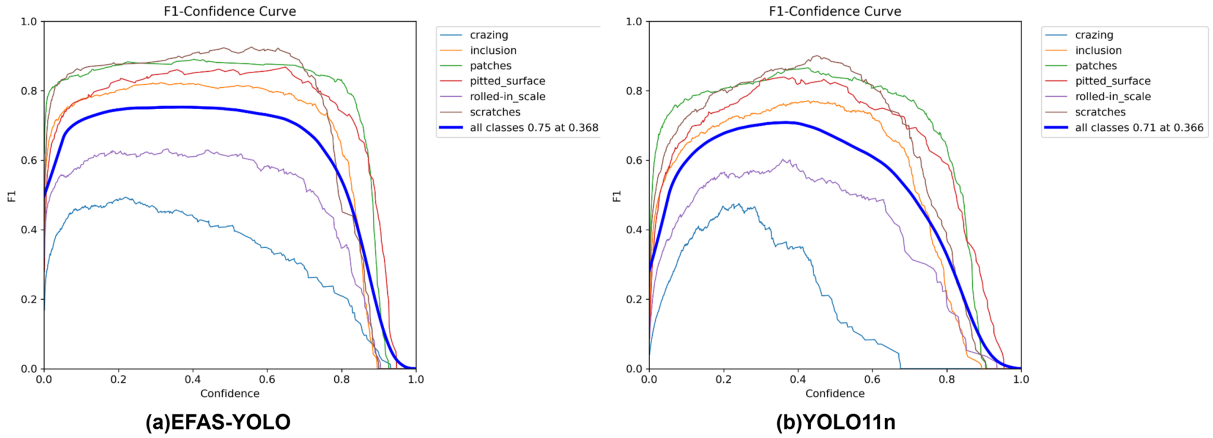

Figure 6: F1 score curve.

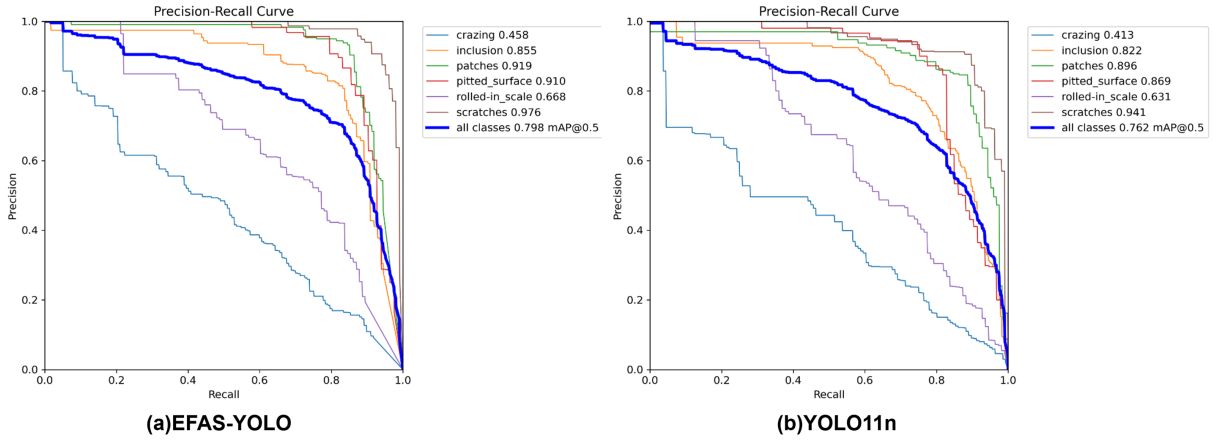

Figure 7: Precision-Recall curve.

Fig. 6 shows that EFAS-YOLO maintains a higher and more stable F1 score over a wider confidence range, indicating better robustness to confidence-threshold variations. As shown in Fig. 7, the PR curves of most categories shift toward the upper-right region, and the overall mAP@0.5 increases from 76.2% to 79.8%. The improvements are especially evident for crazing and scratches, suggesting that the proposed edge-frequency enhancement is effective for weak-texture and boundary-sensitive defects.

4.7 Visualization and Error Analysis

As shown in Fig. 8, the main errors of both models are concentrated between defect categories and background, while inter-class confusion is relatively limited. Compared with YOLOv11n, EFAS-YOLO improves the diagonal values of scratches, rolled-in scale, and pitted surface, and reduces their misclassification as background, indicating fewer missed detections for these categories. However, slight decreases are observed for crazing and patches, suggesting that extremely weak-texture defects with ambiguous boundaries remain challenging. Since the confusion matrix is computed under a fixed confidence threshold, its category-level trend may not be fully consistent with mAP, which reflects the overall precision–recall performance across different thresholds.

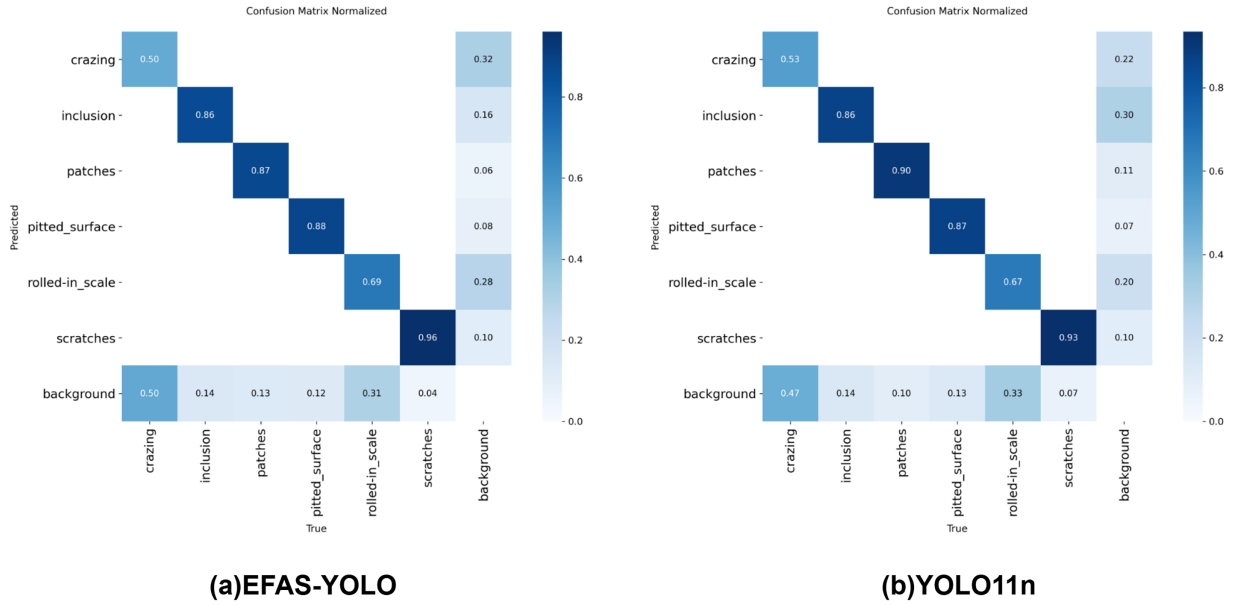

Figure 8: Confusion matrices of the models on the NEU-DET dataset.

Discussion on Industrial Limitations. Although EFAS-YOLO improves the detection of weak-boundary and small-scale defects, some challenging cases may still occur in practical industrial inspection scenarios. For extremely low-contrast defects such as Cr and Pa, the gray-level difference between the defect region and the surrounding steel surface is very small, and the boundary may be discontinuous or partially mixed with background texture. In such cases, the Sobel-based branch can enhance local gradient responses, but when the original gradient variation is too weak, the extracted edge prior may still be insufficient to form a complete defect representation.

Moreover, real production lines often involve more complex imaging conditions than public datasets. Illumination variation may change the gray-level distribution, motion blur may weaken high-frequency boundary details, and surface contamination such as oil stains or water marks may introduce pseudo-edge

responses. To further improve robustness, future work will consider illumination normalization, motion-blur-oriented data augmentation, low-contrast sample reweighting, and domain adaptation across different production lines.

4.8 Supplementary Validation on GC10-DET

NEU-DET is used as the main evaluation dataset in this study, while GC10-DET is adopted as a supplementary validation dataset to examine the stability of the proposed method under a more complex category setting.

GC10-DET is one of the commonly used datasets in the field of steel surface defect detection. It contains 10 typical types of surface defects, with a total of 2306 images. To ensure consistency and comparability of the experimental settings, the dataset is divided into training, validation, and test sets at a ratio of 8:1:1. The same class-stratified random splitting strategy was adopted for GC10-DET, and the split files were kept unchanged for all comparative and ablation experiments. The training strategy is kept as consistent as possible with that used for NEU-DET, including the input resolution, optimizer, learning rate, and number of training epochs.

4.8.1 Comparative Experiments and Ablation Experiments

To verify the overall detection performance of the proposed method on the GC10-DET dataset, Faster R-CNN, YOLOv5n, YOLOv8n, YOLOv10n, YOLOv11n, YOLOv26n, and three RT-DETR variants, namely RT-DETR-R18, RT-DETR-R50, and RT-DETR-R101, are selected as comparative models. The comparative experimental results are shown in Table 8, and the ablation experimental results are shown in Table 9.

Table 8: Overall comparison results of different detection methods on GC10-DET.

Method	mAP@0.5	mAP@0.5:0.95	Params (M)	FLOPs (G)	FPS
Faster R-CNN	55.4	29.2	134.7	83	10
YOLOv5n	63.4	31.7	2.6	7.7	100
YOLOv8n	63.5	33.5	3.2	8.7	109
YOLOv10n	64.9	34.1	2.3	6.7	105
YOLOv11n	65.3	34.6	2.6	6.5	130
YOLOv26n	65.7	34.6	2.4	5.4	133
RT-DETR-R18	58.7	32.0	20	60	37
RT-DETR-R50	60.4	32.5	42	136	22
RT-DETR-R101	62.5	33.1	76	259	13
EFAS-YOLO	68.3	36.4	2.1	5.7	151

Table 9: Ablation study results of the proposed method on GC10-DET.

Model	EFAS	SDEP	Backbone Adapt	P (%)	R (%)	mAP@0.5	mAP@0.5:0.95
YOLOv11n	×	×	×	66.1	65.7	65.3 ± 0.15	34.6 ± 0.12
+EFAS	✓	×	×	66.2	65.8	66.5 ± 0.24	35.4 ± 0.18
+Backbone Adapt	×	×	✓	66.7	66.1	65.9 ± 0.17	35.1 ± 0.15
+EFAS(Backbone Adapt)	✓	×	✓	67.0	66.4	67.1 ± 0.24	35.6 ± 0.20
+SDEP	×	✓	×	66.8	66.2	66.7 ± 0.22	35.3 ± 0.18
+EFAS + SDEP	✓	✓	×	67.7	66.2	67.5 ± 0.19	35.9 ± 0.16
Full Model	✓	✓	✓	67.9	66.3	68.3 ± 0.25	36.4 ± 0.22

As shown in Table 8, EFAS-YOLO achieves an mAP@0.5 of 68.3% on GC10-DET, representing an improvement of 3.0 percentage points over YOLOv11n, while maintaining only 2.1M parameters and an inference speed of 151 FPS. The ablation results in Table 9 indicate that EFAS, SDEP, and Backbone Adapt still bring stable performance gains on GC10-DET, with the complete model achieving the best performance. These results demonstrate that the proposed method is effective not only on NEU-DET but also exhibits a certain degree of stability on steel defect datasets with different category settings.

4.8.2 Qualitative Visualization Analysis

As shown in Fig. 9, the qualitative results on selected representative samples show that EFAS-YOLO achieves more stable localization for slender, low-contrast, and boundary-ambiguous defects. Compared with the selected comparison models, EFAS-YOLO produces fewer missed detections and redundant boxes in visually weak regions. These results are consistent with the design motivation of EFAS and SDEP: input-level Sobel-gradient priors help enhance weak structural responses, while the P3-level enhancement path helps preserve shallow localization details for small defects.

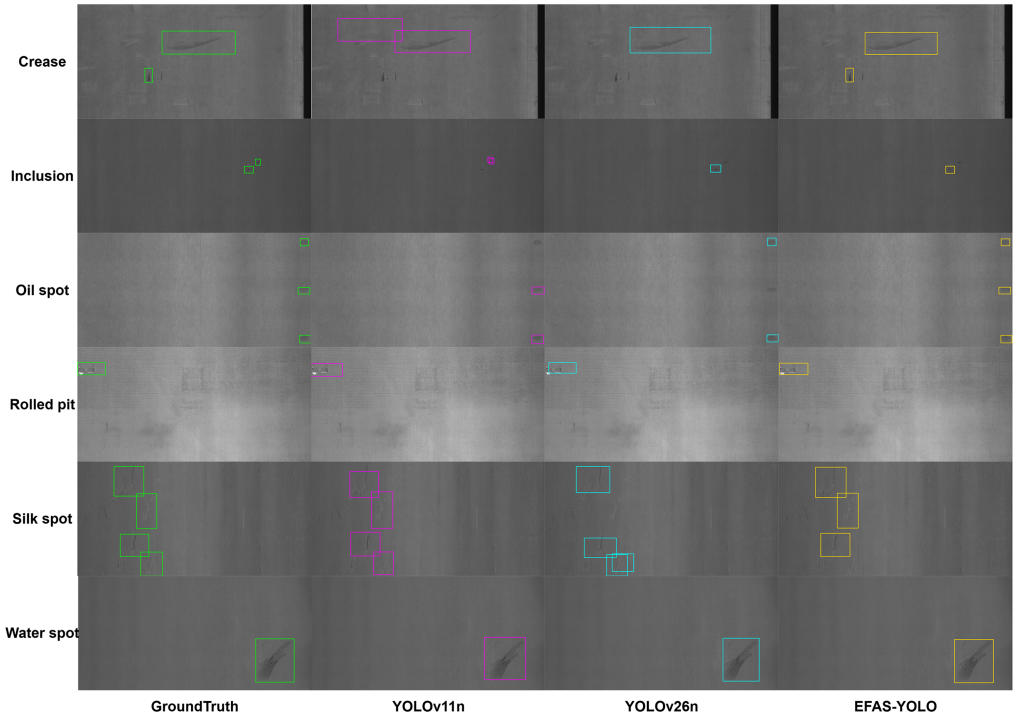

Figure 9: Qualitative comparison of detection results on the GC10-DET dataset.

5 Conclusion

This paper proposes EFAS-YOLO, a lightweight YOLOv11-based framework for steel surface defect detection. The method is designed around the problem that weak edge responses and small-defect details are easily suppressed during early convolution and multi-level downsampling. EFAS introduces fixed Sobel gradient priors and learnable shallow texture features at the input stage, Backbone Adaptation improves the transmission of edge-enhanced features through a smoother channel configuration, and SDEP provides a short residual path for P3-level high-resolution details. Experiments on NEU-DET and GC10-DET demonstrate that EFAS-YOLO improves detection accuracy while maintaining low parameter count and

high inference speed. With only 2.1M parameters, 5.7 GFLOPs, and an inference speed of 158 FPS on an RTX 3090 GPU, EFAS-YOLO provides a lightweight and real-time detection solution for high-throughput steel surface inspection, showing practical potential for industrial scenarios where both detection accuracy and inference efficiency are required.

However, this study is still mainly validated on public offline datasets. Real industrial production lines may involve more complex illumination changes, motion blur, surface contamination, and domain shifts between different steel products. Therefore, future work will focus on validating the proposed method under real production-line scenarios and improving the robustness of extremely low-contrast defect localization.

Acknowledgement: Not applicable.

Funding Statement: The authors received no specific funding for this study. The article processing charge (APC) for this open-access publication was covered by the corresponding author's personal funds.

Author Contributions: Jiahui Liu: Conceptualization, methodology, software, validation, formal analysis, investigation, visualization, and writing—original draft. Longzhen Dong: Conceptualization, supervision, project administration, resources, writing—review and editing, and correspondence. Zeling Hou: Data curation, experimental validation, result analysis, visualization, and writing—review and editing. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: The NEU-DET and GC10-DET datasets used in this study are publicly available. The source code, dataset configuration files, trained model weights, and experimental outputs will be made available at [https://github.com/zhenshuai222/EFAS-YOLO] upon acceptance of the paper. Additional materials are available from the corresponding author upon reasonable request.

Ethics Approval: Not applicable.

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

References

1. Chazhoor AAP, Ho ESL, Gao B, Woo WL. A review and benchmark on state-of-the-art steel defects detection. SN Comput Sci. 2023;5(1):114. doi:10.1007/s42979-023-02436-2.
2. 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.
3. Ma Y, Yin J, Huang F, Li Q. Surface defect inspection of industrial products with object detection deep networks: a systematic review. Artif Intell Rev. 2024;57(12):333. doi:10.1007/s10462-024-10956-3.
4. Ren S, He K, Girshick R, Sun J. Faster R-CNN: towards real-time object detection with region proposal networks. In: Proceedings of the 29th International Conference on Neural Information Processing Systems-Volume 1; 2015 Dec 7–12; Montreal, QC, Canada. p. 91–9.
5. 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. p. 779–88.
6. 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 IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2024 Jun 16–22; Seattle, WA, USA. p. 16965–74.
7. Lin TY, Dollár P, Girshick R, He K, Hariharan B, Belongie S. Feature pyramid networks for object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2017 Jul 21–26; Honolulu, HI, USA. p. 2117–25. doi:10.1109/CVPR.2017.106.
8. Wang Y, Yin T, Chen X, Hauwa AS, Deng B, Zhu Y, et al. A steel defect detection method based on edge feature extraction via the Sobel operator. Sci Rep. 2024;14(1):27694. doi:10.1038/s41598-024-79205-5.

9. Jiang X, Yan F, Lu Y, Wang K, Guo S, Zhang T, et al. Joint attention-guided feature fusion network for saliency detection of surface defects. arXiv:2402.02797. 2024. doi:10.48550/arxiv.2402.02797.
10. Zheng A, Jiang X, Liu W. An efficient lightweight method for steel surface defect detection. Sensors. 2025;25(24):7527. doi:10.3390/s25247527.
11. Neogi N, Mohanta DK, Dutta PK. Review of vision-based steel surface inspection systems. EURASIP J Image Video Process. 2014;2014(1):50. doi:10.1186/1687-5281-2014-50.
12. Ibrahim AAMS, Tapamo JR. A survey of vision-based methods for surface defects' detection and classification in steel products. Informatics. 2024;11(2):25. doi:10.3390/informatics11020025.
13. Luo Q, Fang X, Liu L, Yang C, Sun Y. Automated visual defect detection for flat steel surface: a survey. IEEE Trans Instrum Meas. 2020;69(3):626–44. doi:10.1109/TIM.2019.2963555.
14. Liu S, Qi L, Qin H, Shi J, Jia J. Path aggregation network for instance segmentation. In: Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2018 Jun 18–23; Salt Lake City, UT, USA. p. 8759–68.
15. Tan M, Pang R, Le QV. EfficientDet: scalable and efficient object detection. In: Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2020 Jun 13–19; Seattle, WA, USA. p. 10781–90.
16. Gao S, Chu M, Zhang L. A detection network for small defects of steel surface based on YOLOv7. Digit Signal Process. 2024;149(6):104484. doi:10.1016/j.dsp.2024.104484.
17. Wang J, Chen T, Xu X, Zhao L, Yuan D, Du Y, et al. An improved YOLOv8 model for strip steel surface defect detection. Appl Sci. 2025;15(1):52. doi:10.3390/app15010052.
18. 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.
19. Wang Z, Zhao L, Li H, Xue X, Liu H. Research on a metal surface defect detection algorithm based on DSL-YOLO. Sensors. 2024;24(19):6268. doi:10.3390/s24196268.
20. Zhou Y, Zhao Z. MPA-YOLO: steel surface defect detection based on improved YOLOv8 framework. Pattern Recognit. 2025;168(6):111897. doi:10.1016/j.patcog.2025.111897.
21. Liu F, Zheng Q, Tian X, Shu F, Jiang W, Wang M, et al. Rethinking the multi-scale feature hierarchy in object detection transformer (DETR). Appl Soft Comput. 2025;175(3):113081. doi:10.1016/j.asoc.2025.113081.
22. Wang H, Li Y, Zhang Y, Shang J, Meng D, Moon H. Drone-based high-precision object detection in remote sensing with attention-guided feature fusion. Tsinghua Sci Technol. 2026;31(2):1263–81. doi:10.26599/TST.2025.9010091.
23. Canny J. A computational approach to edge detection. IEEE Trans Pattern Anal Mach Intell. 1986;8(6):679–98. doi:10.1109/TPAMI.1986.4767851.
24. Xie S, Tu Z. Holistically-nested edge detection. In: Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV); 2015 Dec 7–13; Santiago, Chile. p. 1395–403.
25. Song H. RSTD-YOLOv7: a steel surface defect detection based on improved YOLOv7. Sci Rep. 2025;15(1):19649. doi:10.1038/s41598-025-04811-w.
26. Chen C, Lee H, Chen M. Steel surface defect detection method based on improved YOLOv9. Sci Rep. 2025;15(1):25098. doi:10.1038/s41598-025-10647-1.