DOI: 10.28919/ejma.2026.6.16

Eur. J. Math. Appl. (2026)6:16
URL: http://ejma.euap.org
© 2026 European Journal of Mathematics and Applications

EASE-DET: AN EFFICIENT ADAPTIVE SPARSE ENHANCEMENT ALGORITHM FOR STEEL SURFACE DEFECT DETECTION

YONG-TING HE AND JIN-SHAN ZHANG*

Abstract. Steel surface defects pose a persistent threat to product quality and structural integrity. Standard detection networks remain limited in three respects: the backbone struggles to discriminate defects from visually similar background textures; cross-scale fusion in the feature pyramid introduces spatial misalignment, degrading localization accuracy for small objects; and per-scale parameterization of the detection head causes redundancy while offering insufficient sensitivity to defect edge gradients. Taking YOLO11 as the baseline, this paper introduces an Adaptive Position-aware Feature Extraction Module to strengthen global context modeling in the backbone, an Adaptive Context-aware Feature Pyramid Network to improve cross-scale alignment, and an Efficient Shared Detection Head that combines parameter sharing with multi-directional differential convolution for lightweight edge enhancement. On the NEU-DET dataset, EASE-DET achieves 78.3% mAP@0.5, a 3.7-point improvement over the baseline. Cross-domain evaluation on GC10-DET further validates the generalization capability of the proposed method.

1. Introduction

Steel, as a fundamental raw material for modern manufacturing, is extensively employed in construction, automotive, aerospace, and mechanical engineering sectors, where its surface quality directly governs the mechanical performance and service life of end products [1]. During rolling, casting, and heat treatment processes, surface defects such as cracks, inclusions, and scratches inevitably arise. These localized anomalies not only compromise structural reliability but also incur severe economic losses and safety hazards [2]. Conventional manual visual inspection suffers from inherent limitations in throughput and consistency, rendering it inadequate for the high-speed automation demands of contemporary steel production lines. Against the backdrop of advancing intelligent manufacturing, pervasive industrial Internet of Things, and rapidly evolving digital twin technologies, developing efficient and accurate automated surface defect detection algorithms holds significant theoretical value and practical relevance for driving the intelligent transformation of the steel industry.

Early steel surface defect detection relied on handcrafted feature descriptors combined with conventional machine learning classifiers. Texture-based representations, including LBP,

College of Mathematical and Statistics, Sichuan University of Science and Engineering, Zigong 64300, China

*Corresponding author

Email addresses: amyzjyth@163.com, zjscdut@163.com.

Key words and phrases. defect detection; object detection; attention mechanism; multi-scale feature fusion; sparse attention.

GLCM, and Gabor filter banks, delivered reasonable accuracy under controlled imaging conditions with regular defect morphology; however, their discriminative capacity was constrained by the fixed parameterization of the feature extraction pipeline, limiting adaptability to the morphological diversity and textured backgrounds encountered in practice [3]. SVM classifiers with hand-tuned kernels offered partial robustness in low-complexity settings, but false positive and missed detection rates remained unacceptable under realistic production conditions where defect boundaries are ambiguous and surface textures exhibit significant spatial variation [4]. In essence, handcrafted feature pipelines are bounded by the dual limitations of intensive engineering effort and poor generalization across defect types and imaging conditions.

The adoption of convolutional neural networks shifted steel surface defect detection from handcrafted representations toward learned feature hierarchies. Classification architectures with deepened backbones and streamlined design improved inter-class discriminability by composing low-level edge responses into semantically meaningful defect representations [5]. Attention-augmented residual networks further enhanced robustness against complex background interference through adaptive saliency modeling across defect scales [6]. However, classification and segmentation paradigms are inherently limited to predicting either category labels or pixel-wise masks, lacking the capability to jointly output spatial coordinates and class confidence. Object detection, by unifying localization and classification within a single forward pass, has consequently become the dominant framework for this task.

Kong et al. [7] embedded attention mechanisms into a single-stage detection framework, improving fine-grained feature extraction and robustness against textured background interference. Zhao et al. [8] proposed RDD-YOLO, which combines a hierarchical multi-scale backbone with dual feature pyramids and decoupled regression-classification heads to strengthen multi-scale defect representation. Liu et al. [9] introduced a lightweight detection scheme in which channel gating and a streamlined neck structure jointly reduce computational overhead while preserving multi-scale perception. Huang et al. [10] integrated multi-path convolutional attention with local self-attention to enhance spatial sensitivity, while Ma et al. [11] embedded large-kernel attention within a bidirectional feature pyramid to expand contextual receptive fields. Liu et al. [12] designed a multi-scale edge enhancement module with a global-to-local spatial attention fusion network to improve joint detection across diverse defect types.

Existing approaches still face three limitations in steel surface defect detection. First, backbones predominantly rely on fixed convolutional stacks and lack efficient global context modeling with adaptive attention allocation, making it difficult to separate defects from visually similar rolling textures. Second, neck networks depend on bidirectional path aggregation, which yields insufficient cross-scale semantic alignment; shallow-layer localization cues tend to attenuate during upsampling, restricting performance on small objects. Third, detection heads instantiate independent convolutional branches for each scale, incurring substantial parameter redundancy, while standard convolutions are insensitive to the directional distribution of defect edge gradients, degrading localization accuracy under low-contrast conditions.

To address these limitations, this paper proposes EASE-DET (Efficient Adaptive Sparse Enhancement DEtection algorithm for steel surface defecT), which builds upon YOLO11 as the baseline and introduces improvements at three levels—backbone, feature fusion, and detection head. The main contributions are as follows:

(1) An Adaptive Position-aware Feature Extraction Module (APFEM) is proposed. Partitioned Channel Self-Attention (PCSA) confines self-attention computation to a low-dimensional channel subspace, while Dynamic Context Selection (DCS) adaptively sparsifies attention connections. Together, they strengthen defect-to-background discriminability under a lightweight budget.

(2) An Adaptive Context-aware Feature Pyramid Network (ACFPN) is designed. Multi-scale Context Aggregation (MCA) unifies multi-scale features at a shared spatial granularity, while Directional Coordinate Attention (DCA) explicitly encodes directional positional information. Dynamic Interpolation Fusion (DIF) and FuseBlockMulti (FBM) together perform gated fusion to eliminate spatial misalignment across scales.

(3) An Efficient Shared Detection Head (ESDH) is proposed. Cross-scale weight sharing compresses parameter redundancy, while Multi-directional Edge Enhancement Convolution (MEEC) employs five-directional differential convolutions to amplify sensitivity to edge gradients. At inference, the multi-branch weights are re-parameterized into a single convolution, incurring zero additional cost.

2. Related work

2.1. **Evolution and Limitations of Single-Stage Object Detection.** Single-stage detection frameworks eliminate the region proposal stage and perform localization and classification within a single forward pass, yielding faster inference than two-stage counterparts. Their detection accuracy, however, is directly governed by feature representation quality, placing heightened demands on both expressiveness and multi-scale coverage. Across successive YOLO generations, the backbone has evolved from shallow convolutional stacks to deep cross-stage partial architectures; the neck has progressed from unidirectional feature pyramids [13] to bidirectional path aggregation structures that incorporate a bottom-up enhancement pathway [14]; and the detection head has transitioned from coupled prediction to decoupled regression and classification. Weighted bidirectional feature fusion [15] further introduces learnable fusion weights to improve cross-scale information flow. Despite these advances, existing single-stage frameworks share three common limitations. The backbone relies on fixed-topology convolutional stacks and lacks dynamic global context modeling. The neck employs interpolation and concatenation as its primary fusion mechanisms, resulting in insufficient cross-scale spatial alignment. The detection head instantiates scale-specific parameter branches, incurring considerable redundancy. These issues are particularly pronounced in the low-signal-to-noise regime characteristic of steel surface defect detection.

2.2. **Research Status of Steel Surface Defect Detection.** The core challenge of steel surface defect detection lies in the large scale span of defects and their high visual similarity to background textures. He et al. [16] proposed DDN, an end-to-end multi-level feature fusion detection network that aggregates multi-stage backbone features into a hierarchical representation, demonstrating the critical role of cross-scale fusion in localization accuracy. Wang et al. [17] introduced ECA-Net, an efficient channel attention module that achieves lightweight attention through one-dimensional cross-channel interaction without dimensionality reduction, offering a valuable reference for lightweight attention design in detection backbones. Hou et al. [18] proposed the coordinate attention mechanism, which embeds positional information into

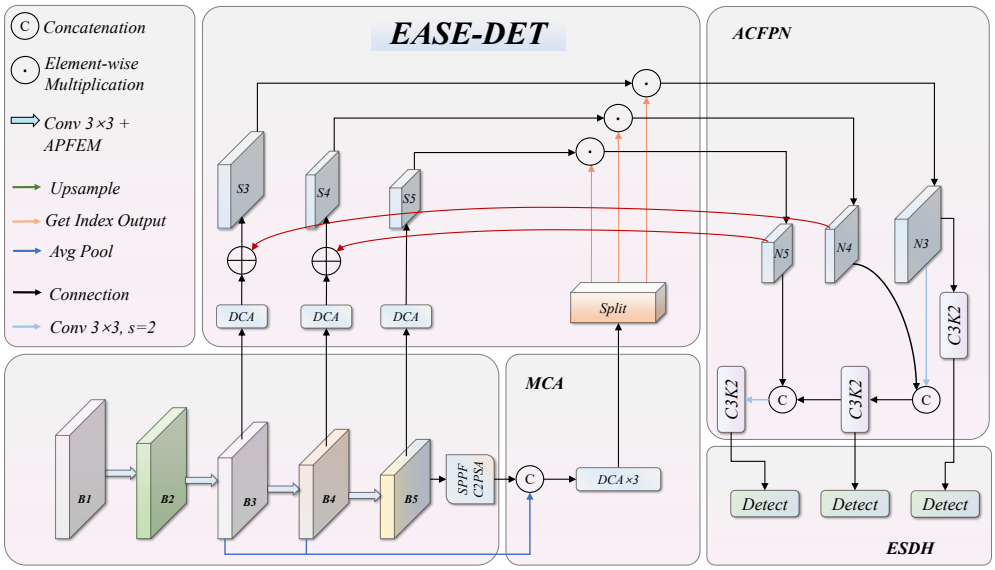


FIGURE 1. EASE-DET

channel attention via separate pooling along the horizontal and vertical directions, achieving attention modeling that balances long-range context awareness with precise position preservation on lightweight networks. Ding et al. [19] proposed RepVGG, a structural re-parameterization method that introduces a multi-branch topology during training to enhance feature representation and equivalently fuses the multi-branch weights into a single convolution at inference, improving both performance and efficiency without additional inference cost. Ge et al. [20] proposed YOLOX, which integrates a decoupled head into the YOLO framework, verifying the direct impact of detection head structure on accuracy. The above methods each address a single dimension—attention lightweighting, cross-scale fusion, or detection head efficiency—without achieving systematic co-optimization spanning the backbone, neck, and detection head. The proposed EASE-DET jointly optimizes all three levels, improving detection accuracy while maintaining a lightweight design.

3. METHOD

3.1. **EASE-DET Architecture.** EASE-DET adopts YOLO11 as its baseline and introduces improvements at three levels—backbone, neck, and detection head—as illustrated in Figure 1. In the backbone, C3k2 modules are replaced with APFEM, which incorporates sparse adaptive attention to strengthen global context modeling. In the neck, PAFPN is replaced with ACFPN to construct a context-aware cross-scale fusion pathway. In the detection head, the standard Detect module is replaced with ESDH, which combines cross-scale parameter sharing with multi-directional differential convolution for lightweight edge enhancement.

3.1.1. *APFEM Feature Extraction Module.* Surface defects such as cracks and inclusions exhibit local statistical distributions that are highly similar to those of rolling textures. Backbone networks composed of fixed convolutional stacks operate with limited receptive fields, restricting their capacity to establish discriminative responses that separate defects from background patterns. This paper proposes APFEM to replace the C3k2 modules in the backbone. Through the co-design of PCSA and DCS, APFEM achieves global context awareness under lightweight computational constraints. The architecture of APFEM is illustrated in Figure 2.

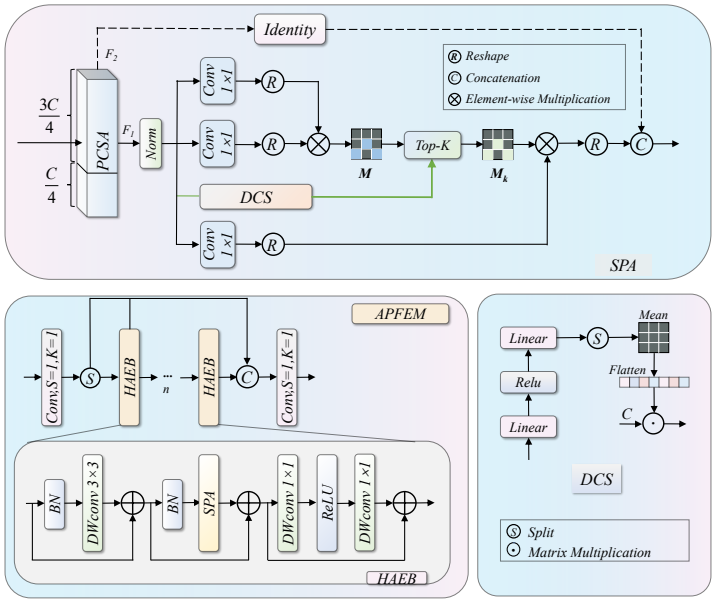


FIGURE 2. APFEM

APFEM adopts a cross-stage partial architecture that decouples feature extraction into a retention pathway and an attention-enhanced pathway. The input feature $X \in \mathbb{R}^{C_1 \times H \times W}$ is expanded to $2C$ channels via a 1×1 convolution and evenly split along the channel dimension into two streams: X_1 is propagated directly to preserve the original information, while X_2 passes sequentially through n stacked Hierarchical Attention Enhancement Blocks (HAEB) for attention-based refinement. All branches are subsequently concatenated and projected to the output dimension through a 1×1 convolution. Let $X_1, X_2 = \text{Split}(W_{\text{stem}}^{1\times1} * X, \text{dim} = 1)$ denote the channel-wise partition, and let H_i represent the output of the i-th HAEB. The overall mapping of APFEM is defined as:

$$Y_{\text{APFEM}} = \mathcal{W}_{\text{proj}} * \text{Concat}\big[X_1,\, X_2,\, H_1(X_2),\, H_2\big(H_1(X_2)\big),\, \ldots,\, H_n(\cdots)\big]$$

HAEB consists of three residual sub-modules stacked in sequence: a depthwise separable convolution that enhances local spatial features to capture defect edge details, Sparse Position-sensitive Attention (SPA) that models global context and long-range dependencies, and a feed-forward network employing an inverted bottleneck structure with two 1×1 convolution layers and SiLU activation for channel-wise transformation. Let $\mathcal{C}_{\text{DW}}^{3\times3}$ denote the depthwise separable convolution operator, $\mathcal{A}_{\text{SPA}}$ the sparse position-sensitive attention, and $\mathcal{F}_{\text{FFN}}$ the feed-forward network. The residual mapping of HAEB is defined as:

$$Y_{\text{HAEB}} = X + \mathcal{F}_{\text{FFN}}\Big(X + \mathcal{A}_{\text{SPA}}\big(X + \mathcal{C}_{\text{DW}}^{3\times3}(X)\big)\Big).$$

SPA integrates two mechanisms, PCSA and DCS, to achieve efficient global context modeling. The input feature $X \in \mathbb{R}^{C\times H\times W}$ is split along the channel dimension into $X_1 \in \mathbb{R}^{P\times H\times W}$ and $X_2 \in \mathbb{R}^{(C-P)\times H\times W}$, where $P = C/4$. X_1 undergoes Group Normalization followed by a 1×1 convolution that generates the query Q, key K, and value V, which are used to compute the attention affinity matrix. X_2 bypasses the attention computation via a shortcut connection. A lightweight gating network adaptively produces a selection ratio from the global feature distribution, guiding DCS to perform position-sensitive sparsification on the attention matrix.

Let $A \in \mathbb{R}^{N\times N}$ denote the affinity matrix, $M_{\mathrm{DCS}} \in \{0,1\}^{N\times N}$ the binary mask generated by DCS, and $N = H \times W$ the total number of spatial positions. The output of SPA is defined as:

$$Y_{\mathrm{SPA}} = \mathcal{W}_{\mathrm{proj}} * \sigma_{\mathrm{SiLU}}\Big(\mathrm{Concat}\big[V \cdot \mathrm{Softmax}(M_{\mathrm{DCS}} \odot A)^{\top},\, X_2\big]\Big).$$

PCSA restricts self-attention computation to one-quarter of the channels, substantially reducing the attention cost. In the low-dimensional subspace, let $Q, K \in \mathbb{R}^{0.5P\times N}$ and $V \in \mathbb{R}^{P\times N}$, with a scaling factor $s = (0.5P)^{-1/2}$ applied to prevent excessively large dot-product values. The affinity matrix A and the contextual feature Z_1 are computed as:

$$A = s \cdot Q^{\top}K \in \mathbb{R}^{N\times N}, \quad Z_1 = VA^{\top} \in \mathbb{R}^{P\times N}$$

DCS predicts a sample-level attention connection density through a lightweight gating network, enabling content-adaptive sparsification. The gating network consists of global average pooling, two 1×1 convolution layers, and activation functions, outputting a scalar $\alpha \in (0,1)$ that indicates the fraction of attention connections to retain under the current feature distribution:

$$\alpha = \sigma\left(W_2^{1\times 1} * \mathrm{ReLU}\big(W_1^{1\times 1} * \mathrm{GAP}(X)\big)\right)$$

The number of retained key-value pairs is $k = \lfloor \alpha \cdot N \rfloor$. A binary mask M_{DCS} is constructed by selecting the top-k entries in each row of the affinity matrix A:

$$M_{\mathrm{DCS}}(i,j) = \begin{cases} 1, & j \in \mathrm{TopK}(A_{i,:} \mid k), \\ 0, & \text{otherwise}, \end{cases} \qquad \tilde{A} = M_{\mathrm{DCS}} \odot A + (1 - M_{\mathrm{DCS}}) \odot (-\infty)$$

where $\mathrm{TopK}(A_{i,:}, k)$ returns the indices of the k largest attention scores in the i-th row. Non-top-k positions are set to $-\infty$, after which Softmax normalization completes the sparsification, concentrating attention exclusively on the most relevant contextual regions.

PCSA confines attention computation to a low-dimensional subspace to contain computational overhead. DCS adaptively sparsifies attention connections via a gating network, suppressing interference from background textures. The CSP-style split preserves local features from being overridden by the attention pathway. Together, these three designs enable APFEM to achieve effective global context awareness under a lightweight budget.

3.1.2. *ACFPN Feature Fusion Module.* The PAFPN neck relies on nearest-neighbor upsampling and channel-wise concatenation for cross-scale fusion. Semantic density discrepancies across different receptive-field levels introduce spatial misalignment into the fused features, degrading localization accuracy for small objects. Furthermore, PAFPN lacks unified perception of multi-scale global semantics. This paper proposes ACFPN to replace PAFPN, employing MCA for cross-scale global context modeling, DCA for enhancing spatial position sensitivity, and the coordinated gating of DIF and FBM to improve cross-scale alignment accuracy. The overall architecture of ACFPN is illustrated in Figure 3.

ACFPN receives three scale features, P3/8, P4/16, and P5/32, from the backbone. MCA first unifies all scales to a common spatial resolution via adaptive pooling, concatenates them along the channel dimension, models them through DCA modules, and then splits the result back into three scale-specific context features. Propagation proceeds top-down layer by layer: DCA performs spatial enhancement on the current layer's features, DIF interpolates and aligns

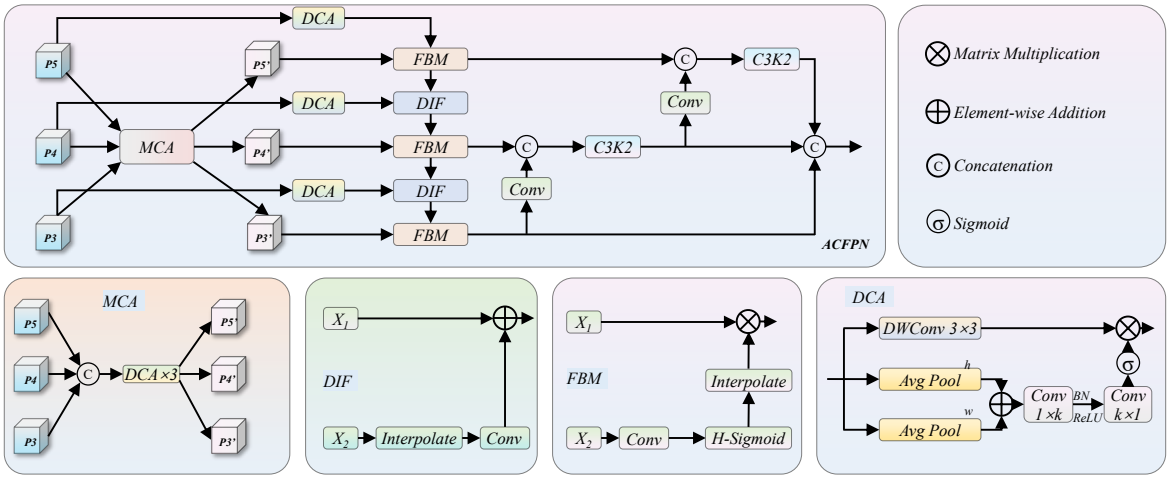


FIGURE 3. ACFPN

the upper-layer output to the current resolution, and FBM applies gated fusion over the DCA-enhanced feature, the MCA context feature, and the DIF-aligned feature. Let $F_i^{\mathrm{bb}} \in \mathbb{R}^{C_i \times H_i \times W_i}$ denote the original backbone feature at the i-th scale, F_i^{ctx} the context feature extracted by MCA, and F_i^{out} the final fused output. The recursive fusion relations are defined as:

$$\{F_3^{\mathrm{ctx}},\, F_4^{\mathrm{ctx}},\, F_5^{\mathrm{ctx}}\} = \mathrm{MCA}\big(\{F_3^{\mathrm{bb}},\, F_4^{\mathrm{bb}},\, F_5^{\mathrm{bb}}\}\big)$$

$$F_i^{\mathrm{out}} = \mathrm{FBM}\big(\mathrm{DCA}(F_i^{\mathrm{bb}}),\, F_i^{\mathrm{ctx}},\, \mathrm{DIF}(F_{i+1}^{\mathrm{out}},\, F_i^{\mathrm{bb}})\big), \quad i \in \{3,\, 4,\, 5\}$$

MCA achieves multi-scale global context extraction through pyramid pooling at a unified spatial granularity. Adaptive average pooling downsamples all scale features to the same resolution as the smallest scale. The three scale features are concatenated along the channel dimension to form an aggregated feature F_{agg}, which then passes through n sequential DCA modules for cross-scale context modeling, and is finally split evenly along the channel dimension into three context features. The mapping of MCA is defined as:

$$F_{\mathrm{agg}} = \mathrm{Concat}\big[\mathrm{GAP}(F_3^{\mathrm{bb}},\, H_5 \times W_5),\, \mathrm{GAP}(F_4^{\mathrm{bb}},\, H_5 \times W_5),\, F_5^{\mathrm{bb}}\big]$$

$$\{F_3^{\mathrm{ctx}},\, F_4^{\mathrm{ctx}},\, F_5^{\mathrm{ctx}}\} = \mathrm{Split}\left(\prod_{k=1}^{n} \mathrm{DCA}_k(F_{\mathrm{agg}}),\, \mathrm{dim} = 1\right)$$

DCA adopts a dual-branch parallel architecture that jointly models local features and directional global context. The local branch captures edge and texture details through a 3×3 depthwise separable convolution. The coordinate attention branch performs adaptive pooling along the vertical and horizontal directions to obtain a height encoding $H_{\mathrm{pool}} \in \mathbb{R}^{C \times H \times 1}$ and a width encoding $W_{\mathrm{pool}} \in \mathbb{R}^{C \times 1 \times W}$, which are summed and then passed through grouped stripe convolutions $1 \times k$ + ReLU and $k \times 1$ + Sigmoid to generate the spatial attention weight A_{coord}. The output of DCA is the Hadamard product of the local features and the attention weights:

$$H_{\mathrm{pool}} = \frac{1}{W}\sum_{j=1}^{W} X_{:,:,j} \in \mathbb{R}^{C \times H \times 1}, \quad W_{\mathrm{pool}} = \frac{1}{H}\sum_{i=1}^{H} X_{:,i,:} \in \mathbb{R}^{C \times 1 \times W}$$

$$A_{\mathrm{coord}} = \sigma\left(\mathrm{Conv}_{k\times 1}^{\mathrm{vert}}\left(\mathrm{ReLU}\left(\mathrm{Conv}_{1\times k}^{\mathrm{horz}}\left(H_{\mathrm{pool}} + W_{\mathrm{pool}}\right)\right)\right)\right)$$

$$Y_{\mathrm{DCA}} = \mathrm{DWConv}_{3\times 3}(X) \odot A_{\mathrm{coord}}$$

DIF dynamically aligns the upper-layer low-resolution semantic features to the current scale. Let F_i^{local} denote the local feature at the current layer and F_{i+1}^{up} the output from the upper layer. After channel alignment via a 1×1 convolution and spatial alignment via bilinear interpolation, the result is fused with the current-layer feature through element-wise summation:

$$F_{\text{DIF}} = F_i^{\text{local}} + \text{Interp}\big(W^{1\times1} * F_{i+1}^{\text{up}},\ |H_i|,\ |W_i|\big)$$

FBM performs selective fusion of multi-source features via a gating mechanism built upon the DIF alignment. The high-level semantic feature F_{high} passes through a 1×1 convolution and h-sigmoid activation to produce gating weights, which are upsampled to the low-level resolution and multiplied element-wise with the transformed low-level feature F_{low}:

$$G_{\text{gate}} = \text{Interp}\big(h\text{-}\sigma\big(W_2^{1\times1} * F_{\text{high}}\big),\ |H_{\text{low}}|,\ |W_{\text{low}}|\big)$$

$$Y_{\text{FBM}} = \big(W_1^{1\times1} * F_{\text{low}}\big) \odot G_{\text{gate}}$$

where $h\text{-}\sigma(x) = \min\big(1, , \max\big(0, , \frac{x+3}{6}\big)\big)$. Its bounded output avoids the gradient vanishing problem of standard Sigmoid in the saturation region.

ACFPN replaces the layer-by-layer sequential interaction of PAFPN with MCA's parallel cross-scale aggregation, establishing global semantic pathways. DCA explicitly encodes positional information into the feature representation through directional coordinate attention. DIF and FBM jointly suppress the feature misalignment introduced by concatenation-based fusion through interpolation alignment and gated selection, respectively, enabling the network to produce precise localization responses for steel surface defects of varying scales and morphologies.

3.1.3. *ESDH Detection Head.* Standard detection heads instantiate independent convolutional weights for the P3, P4, and P5 scales. Cross-scale feature patterns shared across scales are learned redundantly, incurring substantial parameter overhead. Moreover, standard convolutions lack sensitivity to directional gradient information, causing localization drift for defects with low contrast or ambiguous boundaries. This paper proposes ESDH, which compresses parameter scale through cross-scale weight sharing, enhances edge gradient awareness via the multi-directional differential design of MEEC, and adaptively adjusts per-scale prediction weights through learnable scale factors. The overall architecture of ESDH is illustrated in Figure 4.

ESDH receives three scale features from the neck network, $F_i^{\text{in}} \in \mathbb{R}^{C_i\times H_i\times W_i}, i \in 3,4,5$. Each layer passes through an independent 1×1 convolution that maps the channels to a unified hidden dimension h_{idc}, after which all scales share the same set of MEEC modules for feature enhancement. The enhanced features are fed into shared 1×1 convolutional branches for bounding box regression and class prediction, with the box regression branch applying per-layer learnable scale factors $s_i \in \mathbb{R}^+$ for adaptive scaling. Let $W_i^{1\times1}$ denote the channel alignment matrix for the i-th layer, $\mathcal{S}$share the shared MEEC module, and $W\text{box}^{1\times1}$ and $W_{\text{cls}}^{1\times1}$ the shared prediction heads. The output mapping of ESDH is defined as:

$$F_i^{\text{align}} = W_i^{1\times1} F_i^{\text{in}}, \quad F_i^{\text{enh}} = \mathcal{S}_{\text{share}}\big(F_i^{\text{align}}\big), \quad i \in \{3,4,5\}$$

$$P_i^{\text{box}} = s_i \odot W_{\text{box}}^{1\times1} F_i^{\text{enh}}, \quad P_i^{\text{cls}} = W_{\text{cls}}^{1\times1} F_i^{\text{enh}}, \quad Y_i = \text{Concat}\big(P_i^{\text{box}},\ P_i^{\text{cls}}\big)$$

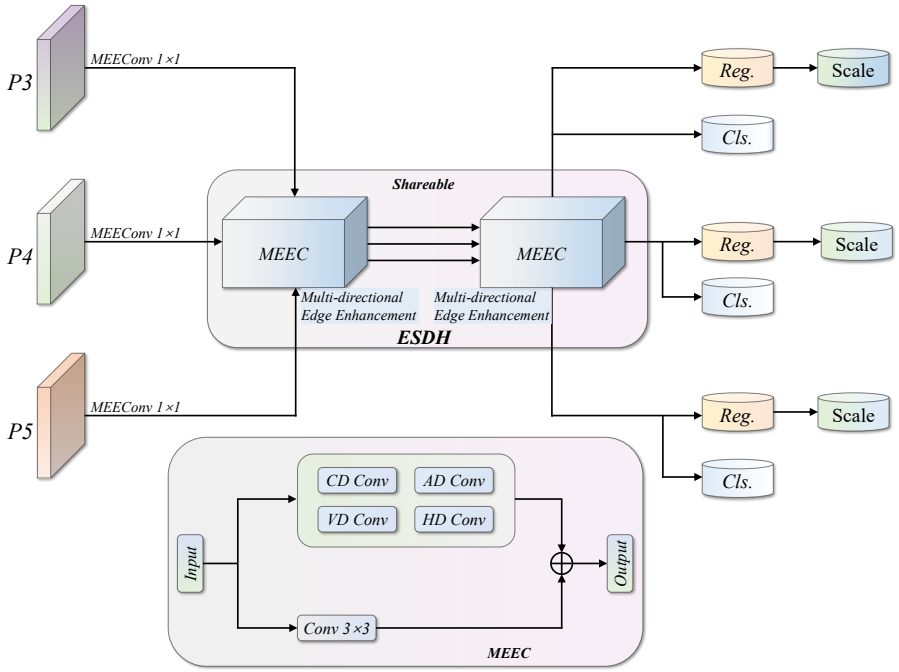


FIGURE 4. ESDH

MEEC adopts a five-branch parallel architecture that enhances defect edge perception through multi-directional differential convolution, with structural re-parameterization via post-training weight fusion. During training, central difference convolution (CDC), horizontal difference convolution (HDC), vertical difference convolution (VDC), angular difference convolution (ADC), and standard convolution separately extract multi-directional edge gradient features. The five branches are optimized independently. Let $W_{\text{cd}}, W_{\text{hd}}, W_{\text{vd}}, W_{\text{ad}}, W_{\text{std}} \in \mathbb{R}^{C_{\text{out}} \times C_{\text{in}} \times 3 \times 3}$ denote the weights of each branch and b_d, $(d \in \text{cd}, , \text{hd}, , \text{vd}, , \text{ad}, , \text{std})$ the corresponding biases. The linearity of the convolution operation ensures that the five branches can be equivalently fused into a single convolution:

$$W_{\text{fused}} = \sum_{d \in \{\text{cd,hd,vd,ad,std}\}} W_d, \quad b_{\text{fused}} = \sum_{d \in \{\text{cd,hd,vd,ad,std}\}} b_d$$

$$Y_{\text{MEEC}} = \sigma_{\text{act}}\big(\text{GN}(W_{\text{fused}} * X + b_{\text{fused}})\big)$$

During inference, switch_to_deploy() pre-fuses the weights of the five branches into a single convolution layer, thereby preserving the multi-directional edge enhancement capability at the computational cost of a single convolution operation.

ESDH compresses parameter redundancy into a single set of shared weights through cross-scale weight sharing. The multi-directional differential design of MEEC strengthens gradient responses to edges of low-contrast defects. Structural re-parameterization reduces the inference cost to that of a single convolution. The learnable scale factors adaptively balance the prediction contributions across detection levels according to the defect size distribution.

4. Experiments and Analysis

4.1. **Dataset Description.** Experiments are conducted on two steel surface defect datasets, NEU-DET and GC10-DET, with details summarized in Table 1. NEU-DET contains 1,800 images across 6 defect classes, split into training, validation, and test sets in an 8:1:1 ratio.

GC10-DET contains 2,294 images spanning 10 defect classes and is used for cross-dataset generalization evaluation.

TABLE 1. datasets

Dataset	Image-Type	Image-Size	Train-set	Val-set	Test-set	Classes	Instances
NEU-DET	Grayscale	200×200	1440	180	180	6	3351
CG10-DET	Grayscale	2048×1000	1835	229	230	10	3551

4.2. **Experimental Environment and Parameter Settings.** All experiments are conducted on Windows 11 using the PyTorch 2.4 deep learning framework with CUDA 12.6 and Python 3.10. The hardware configuration consists of a GeForce RTX 4090 GPU with 24 GB of video memory. For training, the model input size is fixed at 640×640, the optimizer is SGD with an initial learning rate of 0.01, the batch size is set to 8, and the total number of training epochs is 300.

4.3. **Evaluation Metrics.** Seven metrics are adopted to evaluate model performance: precision (P), recall (R), mAP50, mAP50-95, GFLOPs, Params, and FPS. P and R measure the confidence reliability of predicted boxes and the completeness of defect coverage, respectively. mAP50 evaluates overall detection accuracy at an IoU threshold of 0.5, while mAP50-95 averages across IoU thresholds from 0.5 to 0.95 to constrain bounding box localization precision. GFLOPs, Params, and FPS quantify floating-point operations, parameter scale, and real-time inference throughput, respectively, reflecting the deployment feasibility of the model under resource-constrained conditions.

4.4. **Ablation Studies.**

4.4.1. *Ablation Study on APFEM Core Parameter.* Table 2 presents the ablation results for the PCSA channel partition ratio P. As P narrows from $C/2$ to $C/8$, mAP50 exhibits an initial increase followed by a decline, peaking at 75.4% at $P = C/4$. This trend aligns with the information bottleneck principle: feature channels do not contribute equally to the detection task—redundant channels dilute gradient signals during backpropagation, interfering with the convergence of discriminative feature weights, while excessive compression sacrifices the receptive-field diversity essential for capturing multi-scale defects. $P = C/4$ achieves the optimal trade-off between computational efficiency and detection accuracy and is therefore adopted as the default configuration.

4.4.2. *Ablation Study on ACFPN.* Table 3 presents the ablation results for MCA and DCA. Introducing MCA alone raises mAP50 from the baseline 74.6% to 75.1%, while introducing DCA alone reaches 75.8%. MCA pools the three scale features to a unified spatial granularity and concatenates them along the channel dimension. This parallel aggregation establishes direct correspondence between deep semantics and shallow positional information in the concatenated feature space, breaking the restriction of PAFPN that limits information exchange to adjacent layers. DCA encodes positional information along the H and W directions through stripe pooling and subsequently generates spatial attention via grouped stripe convolutions. Rolling textures are suppressed due to the asymmetric gradient energy distribution along their

TABLE 2. Ablation results for the channel partition ratio P in APFEM

P	GFLOPs	Params	P	R	mAP50	mAP50-95
P=C/2	53.6	14.92	75.2	70.8	73.3	40.8
P=C/3	48.3	13.09	76.1	68.7	74.1	39.0
P=C/5	44.7	12.17	76.7	70.4	74.6	40.1
P=C/6	43.9	12.01	74.5	69.3	72.8	38.6
P=C/8	43.1	11.85	73.8	67.9	73.7	40.3
P=C/4	46.0	12.46	77.0	70.5	75.4	40.7

fixed process direction, while defects are preserved owing to their more isotropic directional distribution. When the two are jointly deployed, the directional encoding of DCA precisely compensates for the spatial orientation information lost during MCA pooling, driving mAP50 to 76.1%.

TABLE 3. Ablation results for MCA and DCA in ACFPN

method	MCA	DCA	GFLOPs	Params	P(%)	R(%)	mAP50(%)	mAP50-95(%)
YOLO11(base)			67.3	20.04	76.2	69.8	74.6	40.1
2	✓		61.6	18.92	74.7	71.3	75.1	41.2
3		✓	65.4	19.57	75.3	70.4	75.8	40.7
4(Ours)	✓	✓	59.1	18.26	76.9	71.1	76.1	41.5

4.4.3. *Overall Ablation Study.* Table 4 presents the overall ablation results. When introduced individually, APFEM, ACFPN, and ESDH raise mAP50 from the baseline 74.6% to 75.4%, 76.1%, and 75.8%, respectively, with ACFPN contributing the largest gain. Among pairwise combinations, ACFPN+ESDH and APFEM+ACFPN achieve 77.9% and 77.5%, respectively, both surpassing the best single-module result, indicating substantial forward propagation gain transfer across modules. With all three modules jointly deployed, mAP50 rises to 78.3% while both GFLOPs and Params decrease relative to the baseline.

TABLE 4. Overall ablation results

Model	APFEM	ACFP	ESDH	GFLOPs	Params	P(%)	R(%)	mAP50(%)	mAP50-95(%)
1.base				67.3	20.04	76.2	69.8	74.6	40.1
2	✓			46.0	12.46	77.0	70.5	75.4	40.7
3		✓		59.1	18.26	76.9	71.1	76.1	41.5
4			✓	48.6	14.29	77.5	70.9	75.8	41.2
5	✓	✓		54.6	15.64	79.3	71.3	77.5	43.2
6	✓		✓	48.5	13.17	78.9	71.6	76.8	41.9
7		✓	✓	57.9	17.85	78.2	72.7	77.9	42.6
8.Ours	✓	✓	✓	49.8	14.74	79.5	72.6	78.3	43.0

4.5. **Comparative Experiments.**

4.5.1. *Backbone Network Comparison.* Table 5 presents the comparison results of different backbone replacement schemes. APFEM surpasses all compared schemes with an mAP50 of 75.4% under a lightweight budget. InceptionNeXt and StarNet drop to 71.3% and 71.7%, respectively; the multi-branch heterogeneous convolutions suffer from receptive-field redundancy on small-scale defects. MobileNetV4 and MobileMamba yield mAP50 values at or below the baseline—their global aggregation properties struggle to produce concentrated activation in the spatially localized defect distribution, where background interference on defect features is pronounced. The PCSA channel partitioning and DCS adaptive sparsification of APFEM jointly optimize detection accuracy while maintaining a lightweight design.

TABLE 5. Backbone comparison results

Model	GFLOPs	Params	P(%)	R(%)	mAP50(%)	mAP50-95(%)
YOLO11(base)	67.3	20.04	76.2	69.8	74.6	40.1
InceptionNeXt [21]	47.8	24.53	73.7	67.1	71.3	39.7
MobileMamba [22]	64.1	20.18	75.5	70.4	73.8	41.3
MambaOut [23]	55.5	18.97	73.0	68.2	74.1	40.9
MobileNetV4 [24]	63.1	25.44	72.8	69.4	74.6	41.3
StarNet [25]	52.3	17.95	74.1	67.9	71.7	38.9
RMT [26]	41.8	14.56	77.3	66.9	74.5	38.4
APFEM(Ours)	46.0	12.46	77.0	70.5	75.4	40.7

4.5.2. *Neck Network Comparison.* Table 6 presents the comparison results of different neck networks. DPCF achieves the highest mAP50 at 76.3%, yet its mAP50-95 reaches only 40.3% with a large parameter count, indicating insufficient localization accuracy under strict IoU thresholds. Rethinking-FPN and HFFE obtain mAP50 values of 75.4% and 75.7%, respectively, both with larger parameter scales than ACFPN. LCA and MFPM fall below the baseline mAP50 of 74.6%, where aggressive lightweight compression leads to accuracy degradation. ACFPN preserves both mAP50 and mAP50-95 at high levels by enabling cross-scale semantic propagation without compromising spatial coordinate fidelity.

TABLE 6. Neck comparison results

Model	GFLOPs	Params	P(%)	R(%)	mAP50(%)	mAP50-95(%)
YOLO11(base)	67.3	20.04	76.2	69.8	74.6	40.1
LCA [27]	52.6	10.97	74.5	71.7	73.8	40.7
Rethinking-FPN [28]	79.1	27.42	76.6	70.5	75.4	41.6
HS-FPN [29]	41.8	28.36	75.1	68.6	74.9	41.0
DPCF [30]	76.7	27.19	77.3	70.1	76.3	40.3
MFPM [31]	60.4	21.27	73.7	71.3	74.2	39.8
HFFE [32]	64.5	20.53	75.9	72.4	75.7	40.9
ACFPN(Ours)	59.1	18.26	76.9	71.1	76.1	41.5

TABLE 7. Overall comparison results

Model	GFLOPs	Params/M	P(%)	R(%)	mAP50(%)	mAP50-95(%)	FPS
One-Stage Object Detector							
YOLOv12-M [33]	66.8	19.89	72.6	68.7	72.2	41.1	68.3
YOLOv10-M [34]	58.4	15.12	71.5	69.5	72.5	39.7	71.2
YOLOv8-M [35]	78.3	25.46	75.5	70.4	73.9	40.3	67.4
GOLD-YOLO-M [36]	84.9	38.62	73.7	70.2	74.4	39.2	71.5
FBRT-YOLO-M [37]	57.8	7.28	74.1	71.9	75.3	41.6	73.6
Two-Stage Object Detector							
Faster R-CNN [38]	208.1	41.39	72.9	69.4	72.3	39.8	33.0
TOOD-R50 [39]	199.3	32.05	73.4	68.5	73.6	39.5	30.2
DETR-Based Object Detector							
RT-DETR-R101 [40]	217.6	72.01	78.9	70.5	77.0	42.4	42.8
RT-DETR-R50 [40]	128.2	41.35	74.0	69.8	74.9	40.2	63.4
Deformable-DETR [41]	172.7	41.86	76.5	71.7	75.2	41.0	20.7
D-Fine-M [42]	55.8	19.04	76.6	69.4	75.8	41.3	37.1
DEIM-M [43]	269.5	45.97	78.2	73.6	76.1	42.5	24.9
YOLOv11-M(base)	67.3	20.04	76.2	69.8	74.6	40.1	76.9
EAST-DET(ours)	**49.8**	**14.74**	**79.5**	**72.6**	**78.3**	**43.0**	**82.5**

4.5.3. *Overall Comparison.* Table 7 presents the comparison results across different detection frameworks. Among single-stage methods, FBRT-YOLO-M achieves the best mAP50 at 75.3%. The DETR-based methods deliver relatively high accuracy, with RT-DETR-R101 and DEIM-M reaching 77.0% and 76.1%, respectively, though both incur substantial GFLOPs and parameter counts. EASE-DET surpasses all compared methods with an mAP50 of 78.3% and an mAP50-95 of 43.0% while maintaining a lightweight profile.

4.6. **Generalization Experiment.** Table 8 presents the cross-domain generalization results on GC10-DET. EASE-DET ranks first with an mAP50 of 62.4% and an mAP50-95 of 35.6%. D-Fine-M attains an mAP50 of 61.2%, albeit with a higher parameter count than EASE-DET. FBRT-YOLO-M incurs the lowest GFLOPs and Params among the compared methods, yet its mAP50 reaches only 60.5%.

TABLE 8. Generalization experiment

Model	GFLOPs	Params	P(%)	R(%)	mAP50(%)	mAP50-95(%)
YOLOv11-M(base)	67.3	20.04	63.7	56.8	58.9	33.4
FBRT-YOLO-M [37]	57.8	7.28	65.1	57.9	60.5	34.7
D-Fine-M [42]	55.8	19.04	65.8	59.6	61.2	33.9
EASE-DET(ours)	49.8	14.74	66.3	60.2	62.4	35.6

5. Conclusion

This paper proposes EASE-DET, a detection framework built upon YOLO11, targeting three core limitations in steel surface defect detection: insufficient global perception of fixed convolutional architectures, poor cross-scale feature alignment, and parameter redundancy in the detection head. APFEM achieves efficient global context modeling within a low-dimensional subspace through the co-design of partitioned-channel self-attention and dynamic context selection. ACFPN improves spatial alignment across semantic features at different scales via pyramid pooling aggregation and coordinate-aware gated fusion. ESDH strengthens edge responses to ambiguous-boundary defects through re-parameterized multi-directional differential convolution while reducing parameter counts via weight sharing. On the NEU-DET dataset, EASE-DET attains an mAP50 of 78.3%, and it maintains relatively robust transfer performance in the cross-domain generalization experiment on GC10-DET, corroborating the effectiveness of the synergistic improvements across the three modules.

Future work could proceed in two directions. First, the sparse attention mechanism proposed in this paper could be extended to large-size steel images captured by high-resolution industrial cameras to investigate the trade-off between inference efficiency and accuracy in real production-line deployment. Second, few-shot learning or self-supervised pre-training strategies could be incorporated to mitigate the practical constraints of high annotation cost and imbalanced sample distribution in industrial defect data, further enhancing the generalization capability of the model on novel defect categories.

References

[1] A.A.M.S. Ibrahim, J.-R. Tapamo, A Survey of Vision-Based Methods for Surface Defects' Detection and Classification in Steel Products, Informatics 11 (2024), 25. https://doi.org/10.3390/informatics11020025.

[2] B. Tang, L. Chen, W. Sun, Z.-K. Lin, Review of Surface Defect Detection of Steel Products Based on Machine Vision, IET Image Process. 17 (2023), 303–322. https://doi.org/10.1049/ipr2.12647.

[3] X. Wen, J. Shan, Y. He, K. Song, Steel Surface Defect Recognition: A Survey, Coatings 13 (2023), 17. https://doi.org/10.3390/coatings13010017.

[4] L. Li, T.-M. Ren, M. Feng, Research on Surface Defect Identification of Steel Balls Based on Improved K-CV Parameter Optimization Support Vector Machine, Adv. Mech. Eng. 15 (2023), 16878132231218586. https://doi.org/10.1177/16878132231218586.

[5] Z. Li, X. Wei, M. Hassaballah, Y. Li, X. Jiang, A Deep Learning Model for Steel Surface Defect Detection, Complex Intell. Syst. 10 (2024), 885–897. https://doi.org/10.1007/s40747-023-01180-7.

[6] H. Zhang, S. Li, Q. Miao, R. Fang, S. Xue, et al., Surface Defect Detection of Hot Rolled Steel Based on Multi-Scale Feature Fusion and Attention Mechanism Residual Block, Sci. Rep. 14 (2024), 7671. https://doi.org/10.1038/s41598-024-57990-3.

[7] C. You, H. Kong, Improved Steel Surface Defect Detection Algorithm Based on YOLOv8, IEEE Access 12 (2024), 99570–99577. https://doi.org/10.1109/access.2024.3429555.

[8] C. Zhao, X. Shu, X. Yan, X. Zuo, F. Zhu, RDD-YOLO: A Modified YOLO for Detection of Steel Surface Defects, Measurement 214 (2023), 112776. https://doi.org/10.1016/j.measurement.2023.112776.

[9] Y. Liu, Y. Liu, X. Guo, X. Ling, Q. Geng, Metal Surface Defect Detection Using SLF-YOLO Enhanced YOLOv8 Model, Sci. Rep. 15 (2025), 11105. https://doi.org/10.1038/s41598-025-94936-9.

[10] Y. Zhou, Z. Zhao, MPA-YOLO: Steel Surface Defect Detection Based on Improved YOLOv8 Framework, Pattern Recognit. 168 (2025), 111897. https://doi.org/10.1016/j.patcog.2025.111897.

[11] J. Tie, C. Zhu, L. Zheng, H. Wang, C. Ruan, et al., LSKA-YOLOv8: A Lightweight Steel Surface Defect Detection Algorithm Based on YOLOv8 Improvement, Alexandria Eng. J. 109 (2024), 201–212. https://doi.org/10.1016/j.aej.2024.08.087.

[12] Y. Feng, L. Jin, H. Yang, S. Liu, An Improved Steel Defect Detection Model Using Multi-Scale Feature Fusion Based on YOLO-MFD, Sci. Rep. 16 (2026), 313. https://doi.org/10.1038/s41598-025-29596-w.

[13] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, et al., Feature Pyramid Networks for Object Detection, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2017, pp. 936–944. https://doi.org/10.1109/cvpr.2017.106.

[14] S. Liu, L. Qi, H. Qin, J. Shi, J. Jia, Path Aggregation Network for Instance Segmentation, in: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, IEEE, 2018, pp. 8759–8768. https://doi.org/10.1109/cvpr.2018.00913.

[15] M. Tan, R. Pang, Q.V. Le, EfficientDet: Scalable and Efficient Object Detection, in: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2020, pp. 10778–10787. https://doi.org/10.1109/cvpr42600.2020.01079.

[16] Y. He, K. Song, Q. Meng, Y. Yan, An End-to-End Steel Surface Defect Detection Approach via Fusing Multiple Hierarchical Features, IEEE Trans. Instrum. Meas. 69 (2020), 1493–1504. https://doi.org/10.1109/tim.2019.2915404.

[17] Q. Wang, B. Wu, P. Zhu, P. Li, W. Zuo, et al., ECA-Net: Efficient Channel Attention for Deep Convolutional Neural Networks, in: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2020, pp. 11531–11539. https://doi.org/10.1109/cvpr42600.2020.01155.

[18] Q. Hou, D. Zhou, J. Feng, Coordinate Attention for Efficient Mobile Network Design, in: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2021, pp. 13708–13717. https://doi.org/10.1109/cvpr46437.2021.01350.

[19] X. Ding, X. Zhang, N. Ma, J. Han, G. Ding, et al., RepVGG: Making VGG-Style ConvNets Great Again, in: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2021, pp. 13728–13737. https://doi.org/10.1109/cvpr46437.2021.01352.

[20] Z. Ge, S. Liu, F. Wang, Z. Li, J. Sun, YOLOX: Exceeding YOLO Series in 2021, arXiv:2107.08430, 2021. https://doi.org/10.48550/arXiv.2107.08430.

[21] W. Yu, P. Zhou, S. Yan, X. Wang, InceptionNeXt: When Inception Meets ConvNeXt, in: 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2024, pp. 5672–5683. https://doi.org/10.1109/cvpr52733.2024.00542.

[22] H. He, J. Zhang, Y. Cai, H. Chen, X. Hu, et al., MobileMamba: Lightweight Multi-Receptive Visual Mamba Network, in: 2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2025, pp. 4497–4507. https://doi.org/10.1109/cvpr52734.2025.00424.

[23] W. Yu, X. Wang, MambaOut: Do We Really Need Mamba for Vision?, in: 2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2025, pp. 4484–4496. https://doi.org/10.1109/cvpr52734.2025.00423.

[24] D. Qin, C. Leichner, M. Delakis, M. Fornoni, S. Luo, et al., MobileNetV4: Universal Models for the Mobile Ecosystem, in: A. Leonardis, E. Ricci, S. Roth, O. Russakovsky, T. Sattler, et al., (eds) Computer Vision – ECCV 2024, Lecture Notes in Computer Science, Springer Nature Switzerland, Cham, pp. 78–96, (2025). https://doi.org/10.1007/978-3-031-73661-2_5.

[25] X. Ma, X. Dai, Y. Bai, Y. Wang, Y. Fu, Rewrite the Stars, in: 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2024, pp. 5694–5703. https://doi.org/10.1109/cvpr52733.2024.00544.

[26] Q. Fan, H. Huang, M. Chen, H. Liu, R. He, RMT: Retentive Networks Meet Vision Transformers, in: 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2024, pp. 5641–5651. https://doi.org/10.1109/cvpr52733.2024.00539.

[27] Q. Yan, Y. Feng, C. Zhang, G. Pang, K. Shi, et al., HVI: A New Color Space for Low-light Image Enhancement, in: 2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2025, pp. 5678–5687. https://doi.org/10.1109/cvpr52734.2025.00533.

[28] H. Li, Rethinking Features-Fused-Pyramid-Neck for Object Detection, in: A. Leonardis, E. Ricci, S. Roth, O. Russakovsky, T. Sattler, et al., (eds) Computer Vision – ECCV 2024, Lecture Notes in Computer Science, Springer Nature Switzerland, Cham, pp. 74–90, (2025). https://doi.org/10.1007/978-3-031-72855-6_5.

[29] Z. Shi, J. Hu, J. Ren, H. Ye, X. Yuan, et al., HS-FPN: High Frequency and Spatial Perception FPN for Tiny Object Detection, in: Proceedings of the AAAI Conference on Artificial Intelligence, Association for the Advancement of Artificial Intelligence (AAAI), 2025, pp. 6896–6904. https://doi.org/10.1609/aaai.v39i7.32740.

[30] W. Xu, S. Zheng, C. Wang, Z. Zhang, C. Ren, et al., SAMamba: Adaptive State Space Modeling with Hierarchical Vision for Infrared Small Target Detection, Inf. Fusion 124 (2025), 103338. https://doi.org/10.1016/j.inffus.2025.103338.

[31] F. Zhang, A. Pan, J. Yang, S. Deng, S. Zhao, et al., ISGLNet: Infrared Small Target Detection with Intrinsic Sensitivity and Guided Learning, IEEE Trans. Geosci. Remote Sens. 63 (2025), 1–17. https://doi.org/10.1109/tgrs.2025.3630246.

[32] Y. Zhang, W. Bao, Y. Yang, W. Wan, Q. Xiao, et al., HAFNet: Hierarchical Attention Fusion Network for Infrared Small Target Detection, IEEE Trans. Geosci. Remote Sens. 63 (2025), 1–16. https://doi.org/10.1109/tgrs.2025.3607732.

[33] Y. Tian, Q. Ye, D. Doermann, YOLOv12: Attention-Centric Real-Time Object Detectors, in: Advances in Neural Information Processing Systems 38, Neural Information Processing Systems Foundation, Inc. (NeurIPS), San Diego, California, USA, 2025, pp. 87151–87175. https://doi.org/10.52202/085713-2627.

[34] A. Wang, H. Chen, L. Liu, K. Chen, Z. Lin, et al., YOLOv10: Real-Time End-to-End Object Detection, in: Advances in Neural Information Processing Systems 37, Neural Information Processing Systems Foundation, Inc. (NeurIPS), San Diego, California, USA, 2024, pp. 107984–108011. https://doi.org/10.52202/079017-3429.

[35] M. Sohan, T. Sai Ram, C.V. Rami Reddy, A Review on YOLOv8 and Its Advancements, in: I.J. Jacob, S. Piramuthu, P. Falkowski-Gilski, (eds) Data Intelligence and Cognitive Informatics, Algorithms for Intelligent Systems, Springer Nature Singapore, pp. 529–545, (2024). https://doi.org/10.1007/978-981-99-7962-2_39.

[36] C. Wang, W. He, Y. Nie, J. Guo, C. Liu, et al., Gold-YOLO: Efficient Object Detector via Gather-and-Distribute Mechanism, in: Advances in Neural Information Processing Systems 36, Neural Information Processing Systems Foundation, Inc. (NeurIPS), San Diego, California, USA, 2023, pp. 51094–51112. https://doi.org/10.52202/075280-2224.

[37] Y. Xiao, T. Xu, Y. Xin, J. Li, FBRT-YOLO: Faster and Better for Real-Time Aerial Image Detection, in: Proceedings of the AAAI Conference on Artificial Intelligence, Association for the Advancement of Artificial Intelligence (AAAI), 2025, pp. 8673–8681. https://doi.org/10.1609/aaai.v39i8.32937.

[38] S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks, IEEE Trans. Pattern Anal. Mach. Intell. 39 (2017), 1137–1149. https://doi.org/10.1109/tpami.2016.2577031.

[39] C. Feng, Y. Zhong, Y. Gao, M.R. Scott, W. Huang, TOOD: Task-Aligned One-Stage Object Detection, in: 2021 IEEE/CVF International Conference on Computer Vision (ICCV), IEEE, 2021, pp. 3490–3499. https://doi.org/10.1109/iccv48922.2021.00349.

[40] Y. Zhao, W. Lv, S. Xu, J. Wei, G. Wang, et al., DETRs Beat YOLOs on Real-time Object Detection, in: 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2024, pp. 16965–16974. https://doi.org/10.1109/cvpr52733.2024.01605.

[41] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, et al., Deformable DETR: Deformable Transformers for End-to-End Object Detection, arXiv:2010.04159, 2020. https://doi.org/10.48550/arXiv.2010.04159.

[42] Y. Peng, H. Li, P. Wu, Y. Zhang, X. Sun, et al., D-FINE: Redefine Regression Task in DETRs as Fine-Grained Distribution Refinement, arXiv:2410.13842, 2024. https://doi.org/10.48550/arXiv.2410.13842.

[43] S. Huang, Z. Lu, X. Cun, Y. Yu, X. Zhou, et al., DEIM: DETR with Improved Matching for Fast Convergence, in: 2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2025, pp. 15162–15171. https://doi.org/10.1109/cvpr52734.2025.01412.