

ECD-YOLO: Multiscale Feature Analysis and Detail-Preserving Downsampling for Steel Surface Defect Detection

Zhang Xiaojian

School of Management, University of Shanghai for Science and Technology, Shanghai, 200093, China

Keywords: Steel surface defect detection; YOLO11; Differential attention; Multiscale feature fusion; Detail-preserving downsampling

Abstract: Automated inspection of steel surfaces is challenged by minute defects, weak contrast, repetitive rolling textures, and substantial scale variation. This paper presents ECD-YOLO, an enhanced YOLO11n detector for accurate and efficient defect localization. The proposed ETC-Down module decomposes downsampling into four complementary branches that preserve spatial structure, gradient edges, multiscale texture, and local contrast. The DAFB module employs differential attention to suppress redundant background responses and combines large-kernel positional encoding, dilated depthwise convolution, and channel gating to strengthen global-local discrimination. SSFF and TFE are incorporated into the neck to align and fuse P3-P5 features across spatial scales. Experiments on the NEU-DET dataset show that, relative to YOLO11n, ECD-YOLO improves precision, recall, mAP@0.5, and mAP@0.5:0.95 by 12.1, 2.6, 5.4, and 10.2 percentage points, respectively, while increasing the parameter count from 2.58 M to 3.28 M and computation from 6.3 to 7.4 GFLOPs. Ablation results confirm complementary gains in detail preservation, background suppression, and multiscale fusion. ECD-YOLO therefore provides a favorable accuracy-efficiency trade-off for online steel surface inspection.

1. Introduction

Steel surface defects, including cracks, scratches, inclusions, pitting, and rolled-in scale, arise from variations in raw materials, equipment condition, and rolling, cooling, and conveying processes. Their small size, weak contrast, irregular geometry, and texture similarity directly affect product quality. Manual inspection and magnetic flux leakage testing are constrained by operator fatigue, experience, and equipment conditions^[1], whereas vision-based deep learning is better suited to high-speed production^[2]. Two-stage detectors such as Faster R-CNN provide strong recognition performance but impose substantial computational cost^[3]. One-stage YOLO detectors offer faster inference and have consequently become a practical choice for online defect inspection^[4]. Recent studies have improved YOLO-based steel defect detectors through dynamic convolution and diffusion pyramids^[5], multiscale convolution and contextual attention^[6], lightweight feature and detection-head optimization^[7], residual structures and improved localization losses^[8], global attention and cascade

fusion^[9] knowledge distillation^[10] and global-local perception with hierarchical fusion^[11]. Nevertheless, three limitations remain. First, conventional downsampling can discard the edges of fine or low-contrast defects. Second, response-based attention may amplify rolling textures and reflections. Third, conventional necks provide insufficient scale alignment and feature selection. To address these issues, this paper proposes ECD-YOLO based on YOLO11n^[12]. The main contributions are as follows: (1) ETC-Down preserves spatial, gradient, texture, and contrast cues during downsampling; (2) DAFB combines differential attention, positional encoding, dilated convolution, and channel gating to suppress repetitive backgrounds; and (3) an SSFF-TFE neck aligns and fuses P3, P4, and P5 features to strengthen multiscale representation.

2. Related Work

YOLO11 is an Ultralytics one-stage detector developed after YOLOv5^[13] and YOLOv8^[17]. It provides n, s, m, l, and x variants that trade accuracy for computational cost. The backbone uses C3K2, C2PSA, and SPPF modules to extract hierarchical features; the neck integrates features from multiple stages; and the detection head predicts class probabilities and bounding boxes. This study adopts YOLO11n because it has the smallest parameter count and highest inference efficiency in the family, making it a suitable baseline for online steel surface inspection.

3. Methodology

3.1 Overall Architecture of ECD-YOLO

YOLO11n remains limited in fine-defect inspection because stride-based downsampling weakens edge detail, local convolution provides insufficient global context, and simple upsampling-concatenation does not fully exploit shallow details and deep semantics. ECD-YOLO addresses these limitations through three changes, as shown in Fig. 1. ETC-Down replaces selected downsampling operations to retain spatial, gradient, texture, and contrast information. DAFB is inserted into key backbone stages to model global relationships and suppress repetitive background responses. SSFF and TFE are added to the neck to align P3, P4, and P5 features and improve cross-scale interaction.

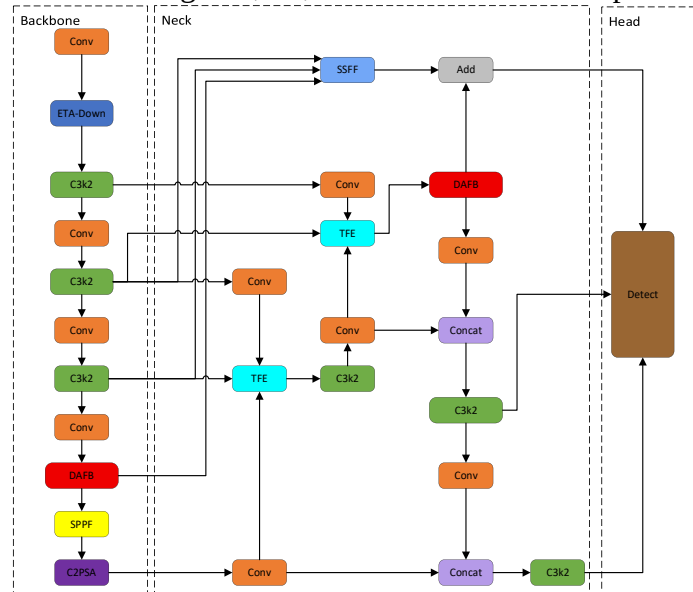


Fig. 1. Overall architecture of ECD-YOLO.

3.2 ETC-Down: Detail-Preserving Downsampling

Fine steel defects are characterized by elongated edges, local gray-level changes, and irregular textures. Because stride convolution performs spatial compression and feature transformation simultaneously, weak responses can be attenuated during early downsampling. ETC-Down therefore performs 2x downsampling through four complementary branches: Space-to-Depth (SPD)^[23], Gradient Aware (GA), Texture Aware (TA), and Contrast Aware (CA). These branches preserve spatial structure, gradient edges, multiscale texture, and local contrast, respectively. The module is illustrated in Fig. 2.

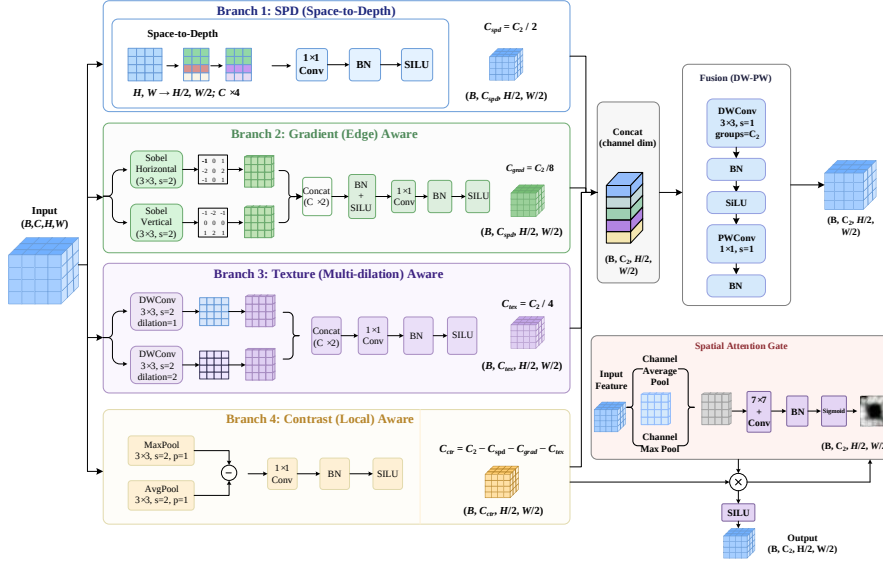


Fig. 2. Structure of the ETC-Down module.

Let the input be $X \in \mathbb{R}^{B \times C_1 \times H \times W}$ and the output be $Y_a \in \mathbb{R}^{B \times C_2 \times H/2 \times W/2}$

The SPD branch rearranges spatial samples into channels and applies a 1×1 convolution, batch normalization (BN), and SiLU activation to produce the spatial-detail feature F_1 .

$$F_1 = \text{SiLU} \left[\text{BN} \left(\text{Conv}_{1 \times 1} (\text{S2D}(X)) \right) \right] \quad (1)$$

The GA branch applies horizontal and vertical Sobel operators to obtain G_h and G_v , concatenates their responses, and uses channel projection to obtain the edge feature F_2 .

$$F_2 = \text{SiLU} \left[\text{BN} \left(\text{Conv}_{1 \times 1} \left(\text{SiLU} \left[\text{BN} \left(\text{Concat}(G_h, G_v) \right) \right] \right) \right) \right] \quad (2)$$

The TA branch uses depthwise convolutions with dilation rates 1 and 2. Their outputs, T_1 and T_2 , are concatenated and projected to produce the multiscale texture feature F_3 .

$$F_3 = \text{SiLU} \left[\text{BN} \left(\text{Conv}_{1 \times 1} (\text{Concat}(T_1, T_2)) \right) \right] \quad (3)$$

The CA branch defines local contrast as the difference between 3×3 max-pooled and average-pooled responses, and then applies channel projection to obtain F_4 .

$$F_4 = \text{SiLU} \left[\text{BN} \left(\text{Conv}_{1 \times 1} (F') \right) \right] \quad (4)$$

All four branches have spatial dimensions $H/2 \times W/2$. Four branches are concatenated. And a depthwise 3×3 convolution extracts local spatial information, and a 1×1 pointwise convolution fuses channels to produce F_{fuse} .

$$F_{dw} = \text{SiLU} \left[\text{BN} \left(\text{DWConv}_{3 \times 3} \left(\text{Concat}(F_1, F_2, F_3, F_4) \right) \right) \right] \quad (5)$$

$$F_{fuse} = \text{BN}(\text{PWConv}_{1 \times 1}(F_{dw})) \quad (6)$$

A spatial attention gate further suppresses background texture. Channel-wise average and maximum responses of F_{fuse} are concatenated and passed through a 7×7 convolution and sigmoid activation to obtain the spatial weight M_s .

$$M_s = \sigma[\text{BN}(\text{Conv}_{7 \times 7}[\text{Concat}(A_{max}, A_{avg})])] \quad (7)$$

Here, $\sigma(\cdot)$ denotes the sigmoid function.

The ETC-Down output is obtained by weighting F_{fuse} with M_s and applying SiLU activation.

$$Y_a = \text{SiLU}(F_{fuse} \odot M_s) \quad (8)$$

3.3 Differential Attention Feed-Forward Block

DAFB replaces selected C3k2 or feature-embedding layers to reduce repeated responses from rolling textures and reflections. It combines a split aggregation structure, differential attention, local positional encoding, and a gated lightweight feed-forward branch, as shown in Fig. 3.

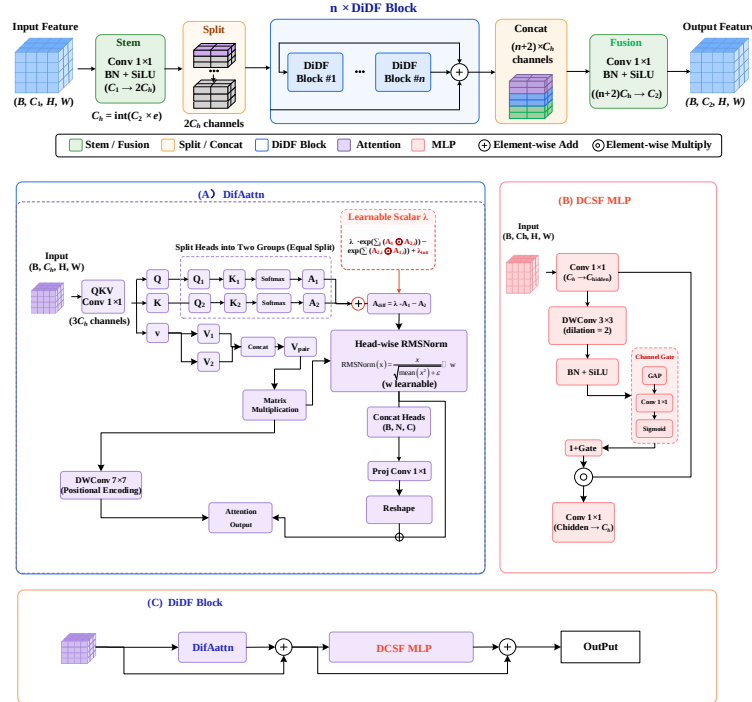


Fig. 3. Structure of the DAFB module.

For an input $X_b \in \mathbb{R}^{B \times C_1 \times H \times W}$, a 1×1 convolution, normalization, and SiLU activation produce the hidden feature F_s .

$$F_s = \text{SiLU}(\text{Conv}_{1 \times 1}(X_b)) \quad (9)$$

F_s is divided into a shallow retention branch X_1 and a branch X_2 processed by n DiDf blocks. X_1 , X_2 , and the intermediate DiDf outputs are concatenated and projected to obtain Y_b , preserving shallow texture while aggregating deeper relational information.

$$Y_b = \text{SiLU} \left(\text{BN} \left(\text{Conv}_{1 \times 1} \left(\text{Concat}(X_0, X_1, \dots, X_{n+1}) \right) \right) \right) \quad (10)$$

Where $Y_b \in \mathbb{R}^{B \times C_2 \times H \times W}$ denotes the output feature representation.

Each DIFD unit combines differential attention with a local feed-forward enhancement branch.

For a DIFD input $X \in \mathbb{R}^{B \times C \times H \times W}$, a 1×1 projection generates the query, key, and value features Q , K , and V .

For each attention head, Q and K are divided into two groups. The corresponding attention matrices A_1 and A_2 are computed as follows.

$$A_j = \text{SoftMax} \left(\frac{Q_1 K_1^T}{\sqrt{d}} \right), j \in \{1, 2\} \quad (11)$$

A learnable coefficient λ forms the differential attention matrix $A_{diff}^{[22]}$.

$$A_{diff} = \lambda A_1 - A_2 \quad (12)$$

$$\lambda = \exp(\lambda_{q1}^T \lambda_{k1}) - \exp(\lambda_{q2}^T \lambda_{k2}) + \lambda_{init} \quad (13)$$

The initial bias λ_{init} controls the differential-attention amplitude during early training.

A_{diff} is applied to V ; the head outputs are concatenated, normalized using RMSNorm^[15], and projected to produce the global relation feature Z_a .

$$Z_a = \text{Conv}_{1 \times 1} \left[\text{RMSNorm} \left(\text{ConcatHeads}(A_{diff} V) \right) \right] \quad (14)$$

To restore two-dimensional locality, a positional branch reshapes V and applies a 7×7 depthwise convolution, producing the positional feature P_v .

$$P_v = \text{DWConv}_{7 \times 7}(\text{Reshape}_{2D}(V)) \quad (15)$$

P_v is fused with the attention output and the residual input to obtain X_D . A DCSF-MLP branch then enlarges the local receptive field and selects informative channels.

The DCSF-MLP maps input F to a hidden space and applies dilated depthwise convolution, BN, and SiLU activation.

$$U = \text{SiLU} \left[\text{BN} \left(\text{Conv}_{3 \times 3, r=2}(\text{Conv}_{1 \times 1}(F)) \right) \right] \quad (16)$$

Global average pooling and a sigmoid gate generate channel weights that adaptively rescale the hidden feature U .

$$\tilde{U} = (1 + \sigma[\text{Conv}_{1 \times 1}(\text{GAP}(U))]) \odot U \quad (17)$$

After output projection, the gated feed-forward feature is fused with X_D to obtain the DAFB output Y_D .

$$Y_D = X_D + \text{Conv}_{1 \times 1}(\tilde{U}) \quad (18)$$

Differential attention suppresses repetitive background responses, while positional encoding and DCSF-MLP restore local continuity and enhance informative channels.

3.4 SSFF and TFE Multiscale Fusion

Steel defects vary substantially in scale, so the neck incorporates the Scale Sequence Feature Fusion (SSFF) and Triple Feature Encoder (TFE) modules from ASF-YOLO^[14]. SSFF models correlations across spatial and scale dimensions, whereas TFE aligns high-, medium-, and low-resolution features before fusion. Their structures are shown in Figs.4 and 5.

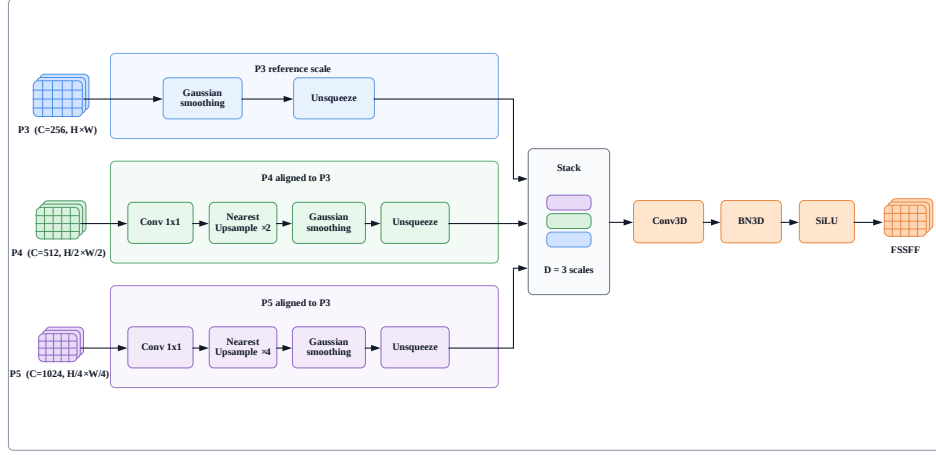


Fig. 4. Structure of the SSFF module.

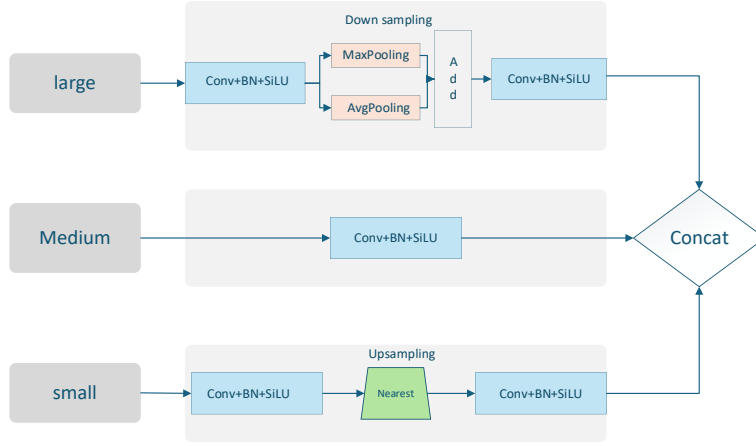


Fig. 5. Structure of the TFE module.

SSFF organizes P_3 , P_4 , and P_5 as a continuous scale sequence and uses three-dimensional convolution to jointly capture spatial and cross-scale dependencies.

The three backbone features contain complementary information: P_3 preserves fine edge and texture cues, whereas P_4 and P_5 provide progressively stronger semantics and larger receptive fields. Gaussian kernels G_{σ_i} with different scale parameters generate smoothed scale-space features F_{σ_i} before alignment.

$$\tilde{P}_i = \text{Align}(F_{\sigma} \times G_{\sigma}) \quad i=3,4,5 \quad (19)$$

After channel and spatial alignment, $\tilde{P}_3$, $\tilde{P}_4$, and $\tilde{P}_5$ are stacked along a new scale dimension. A 3D convolution, 3D batch normalization, and SiLU activation then produce the SSFF output F_{SSFF} .

$$F_{SSFF} = \text{SiLU} \left[\text{BN}_{3D} \left(\text{Conv}_{3D} \left(\text{Stack}(\tilde{P}_3, \tilde{P}_4, \tilde{P}_5) \right) \right) \right] \quad (20)$$

TFE jointly encodes large-, medium-, and small-resolution features to recover details lost during downsampling.

Let the three input features be F_l , F_m , and F_s . The large-resolution branch applies convolution followed by parallel max and average pooling; the small-resolution branch uses convolution and nearest-neighbor upsampling; and the medium-resolution branch requires only channel mapping. The three branches are thereby aligned to the same spatial resolution.

$$\begin{aligned}
\hat{F}_l &= \phi_l[MaxPool(F_l) \oplus AvgPool(F_l)] \\
\hat{F}_s &= \phi_s[Up(F_s)] \\
\hat{F}_m &= \phi_m(F_m)
\end{aligned} \tag{21}$$

Here, $\oplus$ denotes feature fusion, and $\phi_l(\cdot)$, $\phi_m(\cdot)$, and $\phi_s(\cdot)$ denote convolution-BN-activation mappings.

$Up(\cdot)$ denotes nearest-neighbor interpolation.

The aligned branches are concatenated and re-encoded by convolution to obtain F_{TFE} .

$$F_{TFE} = \phi_o(Concat(\hat{F}_l, \hat{F}_m, \hat{F}_s)) \tag{22}$$

SSFF captures continuous scale relationships, while TFE supplements local detail and contextual information. Their combination provides more discriminative multiscale features for defect detection.

4. Experimental Setup

4.1 Dataset and Preprocessing

Experiments were conducted on the NEU-DET steel surface defect dataset from Northeastern University. The dataset contains 1,800 images from six categories: crazing (Cr), inclusion (In), patches (Pa), pitted surface (Ps), rolled-in scale (Rs), and scratches (Sc). Each category contains 300 images. The data were divided into training, test, and validation subsets at a ratio of 7:2:1.

4.2 Implementation Details

All models were implemented in PyTorch and trained on an NVIDIA GeForce RTX 3060 (24 GB). The complete hardware and software configuration is listed in Table 1.

Table 1. Experimental environment.

Environment	Configuration
GPU	NVIDIA GeForce RTX 3060 (24 GB)
CPU	14 vCPU Intel Xeon Gold 6330
Python	3.10
PyTorch	2.1.0
CUDA	12.1
Operating System	Ubuntu 22.04

Stochastic gradient descent (SGD) was used for optimization with an initial learning rate of 0.01, a batch size of 32, and 400 training epochs.

4.3 Evaluation Metrics

Performance was evaluated using precision (P), recall (R), average precision (AP), and mean average precision (mAP). The metrics are defined as follows.

$$Precision = \frac{TP}{TP+FP} \tag{23}$$

$$Recall = \frac{TP}{TP+FN} \tag{24}$$

$$AP = \int_0^1 P(R) dR \tag{25}$$

$$mAP = \frac{1}{N} \sum_{i=1}^N AP_i \quad (26)$$

Here, TP, FP, and FN denote true positives, false positives, and false negatives, respectively. mAP@0.5 uses an IoU threshold of 0.50, whereas mAP@0.5:0.95 averages thresholds from 0.50 to 0.95 at intervals of 0.05. Params and GFLOPs measure model size and theoretical computational cost.

4.4 Comparative Experiments

Table 2 compares ECD-YOLO with representative detectors under identical experimental conditions. Such as RT-DETR^[21], YOLOv5n, YOLOv6n^[16], YOLOv8n, YOLOv9t^[18], YOLOv10n^[19], YOLOv11n and YOLOv12n^[20]. Relative to YOLO11n, ECD-YOLO improves precision, recall, mAP@0.5, and mAP@0.5:0.95 by 12.1, 2.6, 5.4, and 10.2 percentage points, respectively, while adding only 0.70 M parameters and 1.1 GFLOPs. RT-DETR achieves higher scores but requires 32 M parameters and 103.5 GFLOPs. ECD-YOLO therefore provides a more favorable accuracy-computation balance for online inspection.

Table 2. Comparison with representative detectors on NEU-DET.

Model	Precision (%)	Recall (%)	mAP@0.5 (%)	mAP@0.5:0.95 (%)	Parameters (M)	GFLOPs
RT-DETR	92.6	87.3	93.9	69.2	32	103.5
YOLOv5n	69.2	72.4	77.7	44	2.18	5.8
YOLOv6n	80.8	78.4	84.5	52.2	4.16	11.5
YOLOv8n	74.6	76.4	81.4	47.1	2.69	6.8
YOLOv9t	73.2	72.7	78	44.1	1.73	6.4
YOLOv10n	65.4	68.1	73.1	41	2.70	8.2
YOLOv11n	74.2	78.9	83.2	48.2	2.58	6.3
YOLOv12n	78.3	73.5	80.5	46.9	2.51	5.8
ECD-YOLO	86.3	81.5	88.6	58.4	3.28	7.4

This paper compares the baseline model with the improved model in terms of AP values and P-R curves to demonstrate the effectiveness of the overall improvements. The comparison of AP values is shown in Table 3. The P-R curves are shown in Figures 6 and 7. As shown in Table 3, the AP values of the model proposed in this paper have improved across all defect categories compared to the baseline model, with particularly significant increases observed for cracks and rolled-in scale. As shown in Figures 6 and 7, the ECD-YOLO model demonstrates a significant improvement in Rs and Cr compared with the baseline model.

Table 3. Class-wise AP comparison between YOLO11n and ECD-YOLO (%).

Algorithm	Cr	Ps	Sc	Pa	In	Rs
YOLO11n	93.7	89.9	51.8	97.7	86.1	80.1
ECD-YOLO	96.6	90.0	69.0	98.4	90.4	87.1

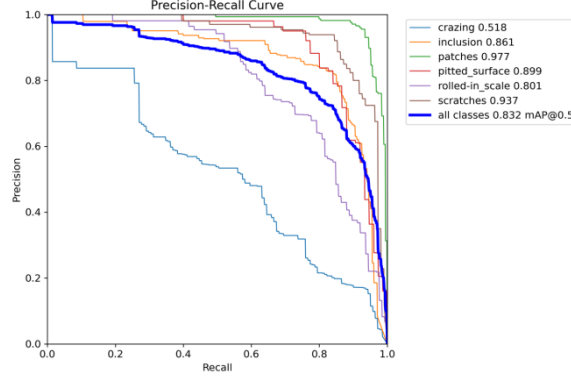


Fig. 6. Precision-recall curves of the YOLO11n baseline.

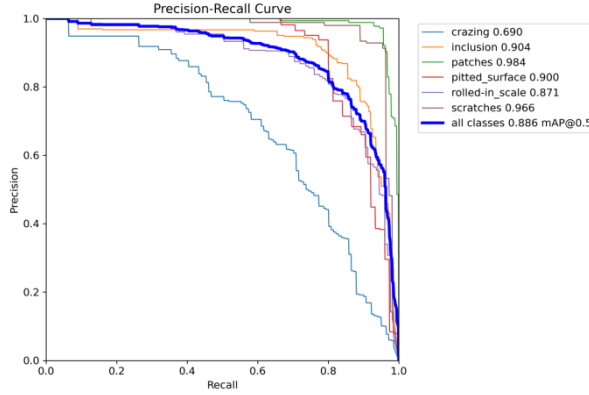


Fig. 7. Precision-recall curves of ECD-YOLO.

4.5 Ablation Study

Ablation experiments used YOLO11n as the baseline and introduced the proposed modules sequentially while keeping all training settings fixed. The results are reported in Table 4.

Table 4. Ablation results on NEU-DET

Model	Precision (%)	Recall (%)	mAP@0.5 (%)	mAP@0.5:0.95 (%)
YOLOv11n	74.2	78.9	83.2	48.2
+ETC-Down	80.7	78.5	84.9	48.2
+ETC-Down+DAFB	79.6	82.6	86.3	51.5
+ ETC-Down + DAFB + SSFF + TFE	86.3	81.5	88.6	58.4

Adding ETC-Down increases precision by 6.5 percentage points and mAP@0.5 by 1.7 points, although recall decreases by 0.4 points. This result indicates stronger preservation of fine details and suppression of background-induced false positives. Adding DAFB subsequently raises recall to 82.6% and mAP@0.5:0.95 to 51.5%, confirming the benefit of global relation modeling and background discrimination.

After SSFF and TFE are introduced, precision, mAP@0.5, and mAP@0.5:0.95 increase by 6.7, 2.3, and 6.9 percentage points, respectively, over the preceding configuration. Cross-scale sequence modeling and spatially aligned feature encoding therefore complement ETC-Down and DAFB,

enabling the final model to improve both classification accuracy and localization quality.

5. Conclusion

This paper presented ECD-YOLO for detecting small and texture-similar steel surface defects. Relative to YOLO11n, the final model improves precision, recall, mAP@0.5, and mAP@0.5:0.95 by 12.1, 2.6, 5.4, and 10.2 percentage points, respectively.

ETC-Down preserves edge, texture, and contrast information during downsampling; DAFB suppresses repetitive background responses while strengthening global-local discrimination; and SSFF-TFE improves interaction between shallow details and deep semantics. Comparative and ablation experiments confirm that the three components provide complementary gains with limited computational overhead.

The study is limited to the NEU-DET dataset and does not explore lightweight variants. Future work will evaluate cross-domain generalization across production lines and imaging conditions, and will reduce computation and memory use for edge deployment.

References

- [1] D. H. Wu, D. H. You, Z. L. Liu, et al., "Mechanism and experimental research on skin depth in AC magnetic flux leakage testing," *Chinese Journal of Scientific Instrument*, vol. 35, no. 2, pp. 327-336, 2014.
- [2] X. Y. Wei, W. Y. Liu, L. Y. Wang, et al., "Surface inspection of polished rail welds based on photometric stereo," *Journal of Mechanical Engineering*, vol. 60, no. 24, pp. 11-24, 2024.
- [3] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137-1149, 2017.
- [4] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, Las Vegas, NV, USA, 2016, pp. 779-788.
- [5] F. Q. Lei, L. W. Ma, P. Guan, et al., "Research on surface defect detection algorithm of steel equipment based on DFD-YOLOv11n," *Computer Engineering and Applications*, vol. 62, no. 2, pp. 92-102, 2026.
- [6] J. J. Ma, J. H. Zhang, Q. Wang, et al., "Strip surface defect detection algorithm based on improved YOLOv11," *Journal of Iron and Steel Research*, vol. 37, no. 10, pp. 1345-1358, 2025.
- [7] Z. Zeng, X. L. Shi, H. Yan, et al., "RGCE-YOLO: Lightweight steel surface defect detection algorithm," *Opto-Electronic Engineering*, vol. 53, no. 3, Art. no. 250240, 2026.
- [8] S. Gao, M. Chu, and L. Zhang, "A detection network for small defects of steel surface based on YOLOv7," *Digital Signal Processing*, vol. 149, Art. no. 104484, 2024.
- [9] G. Liu, M. Chu, R. Gong, et al., "Global attention module and cascade fusion network for steel surface defect detection," *Pattern Recognition*, vol. 158, Art. no. 110979, 2025.
- [10] T. H. Liu and Z. S. He, "Surface defect detection method based on auto-encoding and knowledge distillation," *Journal of Computer Applications*, vol. 41, no. 11, pp. 3200-3205, 2021.
- [11] Z. P. Ji, C. W. Lin, Z. W. He, et al., "Strip steel surface defect detection integrating global-local awareness and hierarchical feature fusion," *Computer Engineering and Applications*, vol. 62, no. 4, pp. 373-381, 2026, doi: 10.3778/j.issn.1002-8331.2412-0407.
- [12] G. Jocher and J. Qiu, "Ultralytics YOLO11, version 11.0.0," computer program, 2024.
- [13] G. Jocher, "Ultralytics YOLOv5: Version 7.0," Zenodo, 2022.
- [14] M. Kang, C. M. Ting, and F. F. Ting, "ASF-YOLO: A novel YOLO model with attentional scale sequence fusion for cell instance segmentation," *Image and Vision Computing*, vol. 147, Art. no. 105057, 2024.
- [15] B. Zhang and R. Sennrich, "Root mean square layer normalization," in *Advances in Neural Information Processing Systems*, vol. 32, 2019, pp. 12360-12371.
- [16] C. Li, L. Li, H. Jiang, et al., "YOLOv6: A single-stage object detection framework for industrial applications," *arXiv:2209.02976*, 2022.
- [17] G. Jocher, A. Chaurasia, and J. Qiu, "Ultralytics YOLOv8: Version 8.0.0," computer program, 2023.
- [18] C. Y. Wang, I. H. Yeh, and H. Y. M. Liao, "YOLOv9: Learning what you want to learn using programmable gradient information," in *Computer Vision-ECCV 2024*, Cham, Switzerland: Springer, 2024, pp. 1-21.
- [19] Wang, H. Chen, L. Liu, et al., "YOLOv10: Real-time end-to-end object detection," in *Advances in Neural Information Processing Systems*, vol. 37, 2024, pp. 107984-108011.
- [20] Y. Tian, Q. Ye, and D. Doermann, "YOLOv12: Attention-centric real-time object detectors," in *Advances in Neural*

Information Processing Systems, vol. 38, 2025, pp. 78433-78451.

[21] Y. Zhao, W. Lv, S. Xu, et al., "DETRs beat YOLOs on real-time object detection," in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition*, Seattle, WA, USA, 2024, pp. 16965-16974.

[22] T. Ye, L. Dong, Y. Xia, et al., "Differential transformer," in *Proc. International Conference on Learning Representations*, 2025.

[23] R. Sunkara and T. Luo, "No more strided convolutions or pooling: A new CNN building block for low-resolution images and small objects," in *Machine Learning and Knowledge Discovery in Databases*, Cham, Switzerland: Springer, 2023, pp. 443-459.