technologies

MDPI

Article

Improved YOLOv8n for Lightweight Rail Surface Defect Detection

Lei Wang [1], Yuan Si [1], Jun Wang [1,2], Liqing Liao [3] and Wensheng Xie [3,*]

[1] Hangzhou Hanggang Metro Line 5 Co., Ltd., Hangzhou 310018, China; ylh_wl@163.com (L.W.); siyuan_3344@163.com (Y.S.); 19955030997@163.com (J.W.)
[2] College of Economics and Management, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China
[3] School of Automation, Central South University, Changsha 410083, China; zdh-dqkz@csu.edu.cn
* Correspondence: 254612068@csu.edu.cn

Abstract

Rail-surface defects are often small, low-contrast, and confused with steel texture or reflections, making it difficult to improve accuracy without increasing model cost. This study proposes ESiV-YOLOv8, a compact detector that assigns complementary modifications to feature selection, box regression, and multiscale fusion in YOLOv8n. EffectiveSE recalibrates deep backbone features, SIoU provides direction-aware regression, and a VoV-GSCSP/GSConv neck reduces redundant computation. Evaluation used a self-built four-class dataset of 4020 images, a near-duplicate-aware train-validation-test split, and matched seven-seed experiments. ESiV-YOLOv8 achieved 97.7% precision, 94.6% recall, 97.2% mAP@0.5, and 68.6% mAP@0.5:0.95. Relative to YOLOv8n, mAP@0.5:0.95 increased by 5.0 percentage points, while parameters and GFLOPs decreased by 17.1% and 12.2%, respectively. On the combined natural-condition subset, ESiV-YOLOv8 achieved 59.6% mAP@0.5:0.95, 5.6 percentage points above the baseline. The annotation audit yielded 98.0% class agreement and a mean box IoU of 0.89. Model-only and end-to-end latency increased by 1.4% and 2.2%, respectively. Overall, ESiV-YOLOv8 improves detection accuracy and reduces model scale with limited latency overhead on the tested backend.

Keywords: rail surface defect detection; YOLOv8n; EffectiveSE; SIoU; VoV-GSCSP; lightweight object detection; held-out test set

1. Introduction

Rail surface defects develop under repeated rolling contact, impact, wear, and environmental exposure. If not detected early, localized damage can grow and compromise operational safety. Machine-vision inspection provides non-contact, high-throughput localization, but field images remain difficult to interpret because defects can be small, low-contrast, irregular, and visually similar to grinding marks, contamination, or specular reflections. Traditional rail-vision systems rely on handcrafted contrast, morphology, filtering, or saliency cues, whose thresholds and descriptors are sensitive to illumination, texture, and noise. The coarse-to-fine model of Yu et al. processes rail images progressively at the subimage, region, and pixel levels, whereas the multilevel system of Yang et al. combines rail extraction, defect segmentation, and localization [1,2].

Deep convolutional detectors learn task-specific representations and enable direct end-to-end localization. One-stage detectors are particularly suitable for rail inspection because they eliminate a separate proposal stage and offer a practical balance between

Academic Editor: Valeri Mladenov

Received: 21 August 2026
Revised: 5 September 2026
Accepted: 10 September 2026
Published: 14 September 2026

detection accuracy and computational cost. YOLO formulates detection as a unified regression problem, and the Ultralytics YOLOv8 implementation uses a C2f-based backbone and neck with an anchor-free decoupled head [3,4]. Subsequent rail-surface studies have addressed limited annotations through ensemble learning, strengthened multiscale representation in RSDNet, and improved learning under class imbalance [5–7]. These results established the feasibility of deep rail-defect detection, while leaving the accuracy and resource requirements of compact inspection models closely coupled.

Recent work has focused on feature extraction, multiscale fusion, attention, and bounding-box regression. GD-YOLOv8 redesigns the neck and replaces CIoU with SIoU, while PerMSCA-YOLO combines a lightweight backbone with perceptual multiscale convolutional attention [8,9]. Other studies have pursued high-precision detection for heavy-haul railways, StarNet and bidirectional feature-pyramid fusion for track damage, a lightweight YOLOv8n configuration for complex railway environments, and adaptive convolution with hierarchical multiscale fusion [10–13]. This body of work confirms that rail defects benefit from task-specific feature enhancement, but the reported architectures, datasets, and evaluation protocols vary substantially.

Related railway-vision tasks share similar design priorities. MSIA-YOLOv8 combines multiscale fusion, an additional small-object head, frequency-domain enhancement, and an improved regression loss for obstacle intrusion [14]. C5LS removes the P5 branch and introduces large-kernel attention and SIoU for densely distributed railway insulators [15]. RS-YOLOv8 uses a hybrid real-synthetic dataset with context-guided features, attention-enhanced bidirectional fusion, and a dynamic head for multi-hazard detection [16]. For internal rail B-scan images, CSP-YOLO combines Ghost-style blocks, coordinate attention, a shared detection head, and pruning [17]. BCD-YOLO instead targets perimeter intrusion with dynamic sparse attention, content-aware upsampling, and tracking [18]. Because these studies address different objects or sensing modalities, their numerical scores are not direct benchmarks for rail-surface defects. Nevertheless, their designs show that small-target response, feature fusion, localization, and deployment cost must be considered together in railway applications.

The recent literature leaves the interaction among these design choices insufficiently resolved for compact rail-surface detection. Feature strengthening, box regression, and neck simplification are often evaluated in different models or on different datasets, making it difficult to isolate their combined effect. In addition, fewer parameters or GFLOPs do not necessarily translate to lower latency on a given inference backend. EffectiveSE provides channel recalibration with a compact gating operation [19]; SIoU introduces direction-aware center optimization without adding inference-time layers [20]; and GSConv with VoV-GSCSP reduces redundant computation in multiscale feature fusion [21]. These complementary roles motivate a constrained architecture that applies attention only to the deepest backbone features, changes the regression objective, and simplifies the neck while retaining the YOLOv8n detection head.

Building on this design, we develop ESiV-YOLOv8 to detect corrugation, spalling, squat, and wheel-burn defects. The central hypothesis is that coordinated modifications to feature selection, box localization, and multiscale fusion can improve rail-defect localization while reducing model size. The evaluation combines a near-duplicate-aware train-validation-test split with a held-out test set [22,23], seven-seed single-module and pairwise ablations, class-wise and cross-architecture comparisons, controlled perturbations and natural-condition subsets, and model-size and runtime measurements. Annotation consistency and split-threshold sensitivity are also examined. Together, these analyses link architectural choices to detection accuracy, robustness, data quality, and computational behavior.

The main contributions of this study are as follows. First, we develop a task-oriented co-design of ESiV-YOLOv8 for rail-surface defect detection. EffectiveSE is inserted only in the deepest backbone stage to strengthen semantically meaningful defect responses; SIoU is used for direction-aware box regression; and VoV-GSCSP/GSConv is introduced in the neck to reduce redundant multiscale fusion. These explicit integration choices assign each modification a distinct role while preserving the YOLOv8n anchor-free decoupled head. Second, we establish a controlled evaluation protocol that isolates individual effects from their joint benefit. The protocol combines a near-duplicate-aware train-validation-test split with a held-out test set, matched seven-seed runs, and ablations at the individual, pairwise, and complete-model levels under the same training and evaluation settings. Third, we assess the accuracy-efficiency trade-off from both model and runtime perspectives. In addition to class-wise detection metrics, the study reports parameters, GFLOPs, checkpoint size, GPU memory, stage-wise and end-to-end latency, and performance under controlled image perturbations.

Section 2 describes the ESiV-YOLOv8 architecture and the placement of its components. Section 3 presents the dataset, experimental protocol, and results. Section 4 concludes the paper.

2. ESiV-YOLOv8 Method

2.1. Design Motivation and Overall Architecture

ESiV-YOLOv8 is motivated by the appearance of rail-surface defects in inspection images and by the constraints of compact deployment. Corrugation, spalling, squat, and wheel-burn regions can occupy only a few pixels and have weak contrast against textured steel. The backbone therefore needs to preserve defect-sensitive responses without applying a costly attention block at every stage. Irregular defect boundaries and aspect ratios also make box-center and shape alignment important during training. At the same time, multi-scale prediction repeatedly fuses features across resolutions, so redundant convolution and concatenation in the neck can offset the efficiency of a small backbone. These observations define three design targets: selective feature recalibration, more informative box regression, and lower-cost multiscale fusion.

The three targets are mapped to distinct functional locations in the detector. EffectiveSE is placed in the final deep-backbone C2f block, where feature maps have stronger semantic context and the attention operation can reweight defect-relevant channels after low-level texture has been encoded [19]. SIoU replaces the box-regression term, incorporating direction-aware center optimization and shape alignment during training without adding a new inference layer [20]. VoV-GSCSP and GSConv are used in the neck to reduce the standard-convolution workload in multiscale fusion while preserving cross-scale feature exchange [21]. These modifications therefore address feature selection, localization, and fusion at non-overlapping locations. This placement also makes their individual, pairwise, and joint effects measurable under the same prediction head.

In the implemented configuration, the final deep-backbone C2f block at model layer 8 is replaced with C2f-EffectiveSE. The original box-regression term is replaced with SIoU. In the neck, the fusion blocks at layers 12, 15, 18, and 21 are replaced with VoV-GSCSP, and the scale-transition convolutions at layers 16 and 19 are replaced with GSConv. The classification and distribution-focal-loss terms, the three prediction scales, and the anchor-free decoupled detection head remain unchanged. This localized modification preserves the baseline prediction interface and provides a consistent reference for ablation experiments. The resulting network architecture is shown in Figure 1.

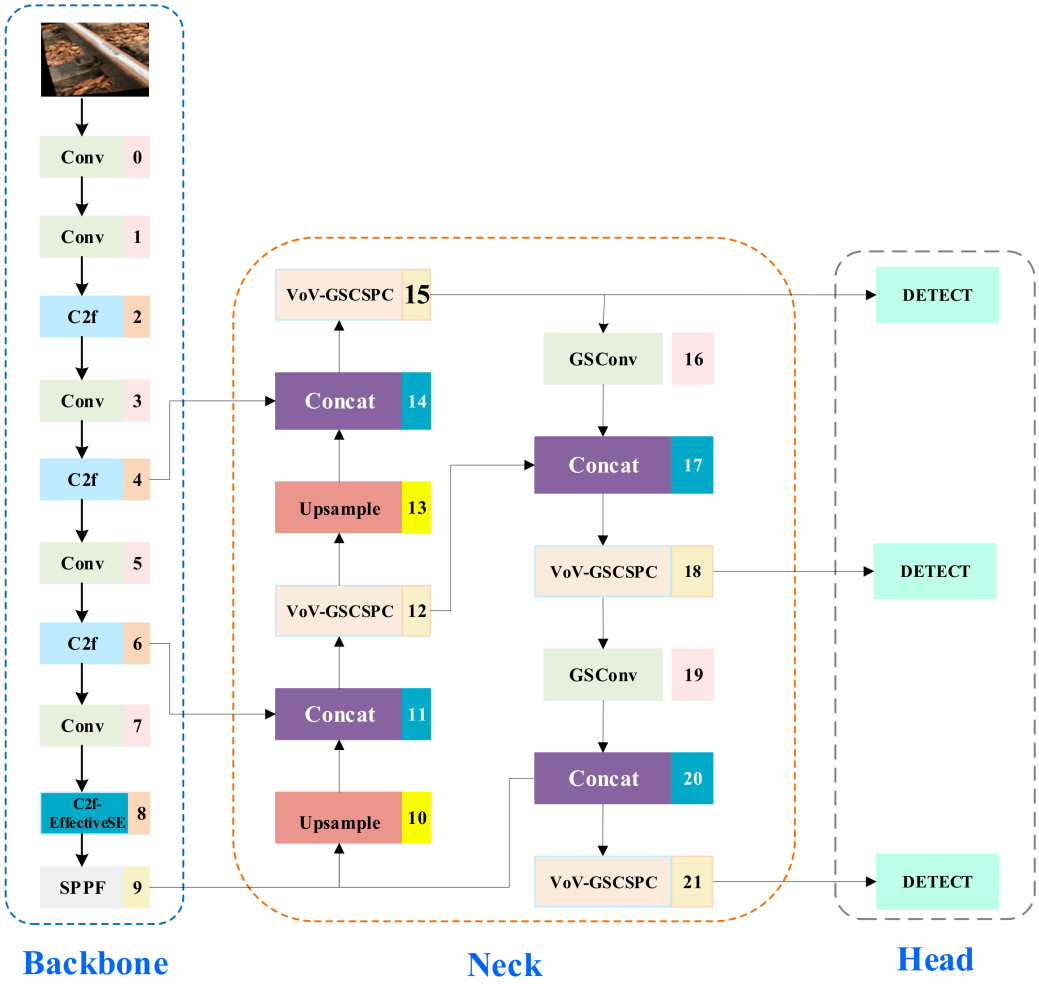

Figure 1. Network architecture of ESiV-YOLOv8.

2.2. *EffectiveSE Channel Recalibration*

EffectiveSE performs channel recalibration by applying global average pooling, a 1×1 channel projection, and a hard-sigmoid gate to the input feature map [19]. As shown in Figure 2, the gate is multiplied channel-wise with the original feature map within the final deep-backbone C2f block. This placement supplies semantically strong features to the subsequent SPPF and neck while adding 0.016 M parameters and 0.0055 GFLOPs. Its independent effect is quantified in the ablation analysis.

Let $X \in \mathbb{R}^{C \times H \times W}$ denote the input feature map, where C, H, and W are the channel count, height, and width, respectively. EffectiveSE computes the channel descriptor z via global average pooling and generates channel weights using a 1×1 projection and a hard-sigmoid gate:

$$z_c = \frac{1}{HW} \sum_{i=1}^{H} \sum_{j=1}^{W} X_c(i,j) \tag{1}$$

$$Y_c(i,j) = X_c(i,j) \cdot \mathrm{hsigmoid}(W_z z + b_z) \tag{2}$$

where z_c is the pooled descriptor for channel c, and W_z and b_z are the projection weight and bias. The multiplication in Equation (2) is channel-wise. The implemented activation is $\mathrm{Hsigmoid}(u) = \max(0, \min(1, (u + 3)/6))$, consistent with torch.nn.Hardsigmoid. This fixed piecewise-linear activation introduces no additional trainable parameters.

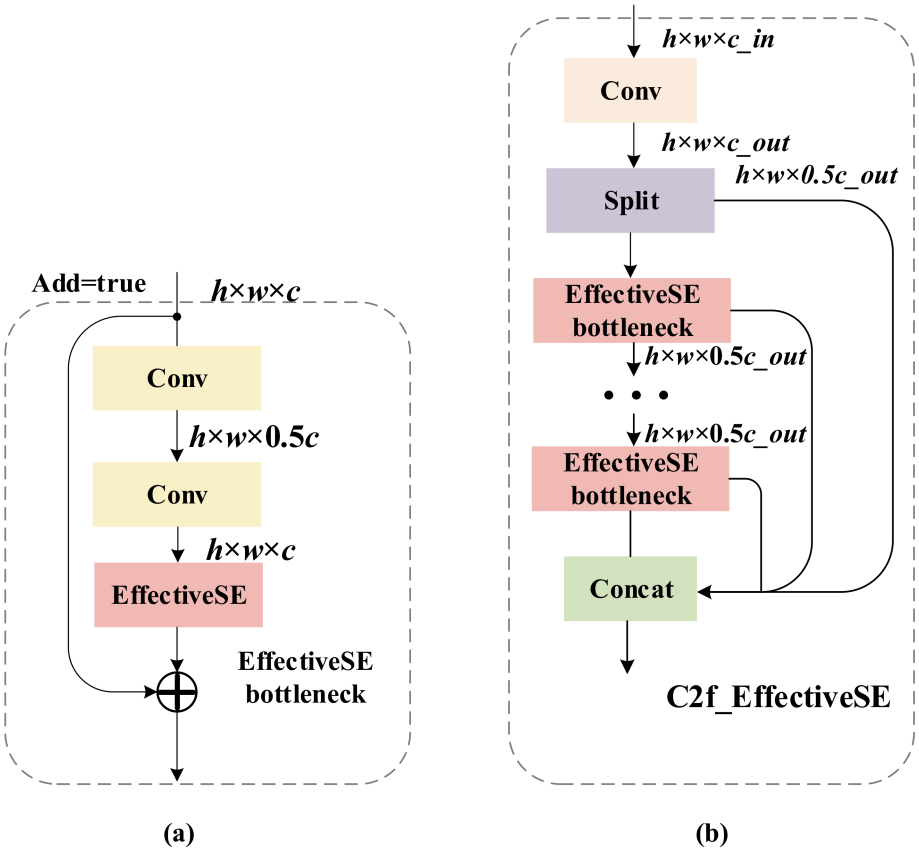

Figure 2. C2f-EffectiveSE structure. (**a**) EffectiveSE bottleneck; (**b**) C2f-EffectiveSE.

2.3. SIoU Bounding-Box Regression

Rail-surface defects often form elongated or irregular regions, making localization sensitive to center displacement and box shape. SIoU (Scylla Intersection over Union) extends IoU-based regression with angle-aware distance and shape costs [20]. Let B and B^{gt} denote the predicted and ground-truth boxes. Their centers are (x_c, y_c) and $(x_c{}^{gt}, y_c{}^{gt})$, their widths and heights are (w, h) and (w^{gt}, h^{gt}), and c_w and c_h are the width and height of the smallest enclosing box. Equations (3)–(6) define the angle cost, distance cost, shape cost, and the final SIoU loss used in this study.

$$\Lambda = 1 - 2\sin^2\left(\sin^{-1}(s) - \frac{\pi}{4}\right) \tag{3}$$

$$\Delta = [1 - \exp(-\gamma\rho_x)] + [1 - \exp(-\gamma\rho_y)] \tag{4}$$

$$\Omega = [1 - \exp(-\omega_w)]^\theta + [1 - \exp(-\omega_h)]^\theta,\ \theta = 4 \tag{5}$$

$$L_{\mathrm{SIoU}} = 1 - \mathrm{IoU} + \frac{\Delta + \Omega}{2} \tag{6}$$

The center offsets are $d_x = x_c{}^{gt} - x_c$ and $d_y = y_c{}^{gt} - y_c$, with $\sigma = \sqrt{(d_x{}^2 + d_y{}^2 + \varepsilon)}$ and $\varepsilon = 10^{-7}$. The direction variable $s = \min(|d_x|, |d_y|)/\sigma$ selects the smaller center-offset component, so $0 \le s \le 1/\sqrt{2}$. Equation (3) is algebraically equivalent to $\Lambda = \cos(2\arcsin(s) - \pi/2)$, which is the form evaluated by the implementation. For Equation (4), $\gamma = 2 - \Lambda$, $\rho_x = [d_x/(c_w + \varepsilon)]^2$, and $\rho_y = [d_y/(c_h + \varepsilon)]^2$. The shape ratios in Equation (5) are $\omega_w = |w - w^{gt}|/[\max(w, w^{gt}) + \varepsilon]$ and $\omega_h = |h - h^{gt}|/[\max(h, h^{gt}) + \varepsilon]$. The exponent θ is fixed at 4, within the range of 2–6 discussed in the original SIoU study [20].

Equation (6) combines overlap with the angle-aware distance and shape penalties, where IoU denotes the intersection-over-union between B and B^{gt}. During training, this

per-box SIoU term replaces the CIoU component of the YOLOv8 localization loss and is aggregated over foreground predictions using the baseline target-score weighting. The classification and distribution focal loss terms remain unchanged. Because SIoU modifies only the training objective, it adds no parameters, GFLOPs, or inference-time layers.

2.4. VoV-GSCSP Lightweight Neck

GSConv combines a standard-convolution branch with a depthwise branch and then mixes their channels [21]. Slim-Neck organizes GSConv-based bottlenecks into VoV-GSCSP blocks for feature fusion. In this work, VoV-GSCSP replaces four neck fusion blocks, and GSConv replaces two neck downsampling convolutions. The single-module ablation implements this complete neck substitution, matching the full-model implementation.

The implemented GSConv includes a standard convolution branch, a 5×5 depthwise branch, concatenation, and channel shuffle. Parameters and GFLOPs are computed for the instantiated baseline and modified networks at the same input resolution and reported in the ablation analysis.

The adopted block includes a transformed GSConv bottleneck branch and a cross-stage shortcut branch, followed by feature aggregation and channel unification.

3. Experiments and Results

3.1. Dataset, Annotation, and Data Partition

The self-built Rail Surface Defect Dataset contains 4020 images at a native resolution of 416×416 pixels. Bounding-box annotations are stored in normalized YOLO format and cover four defect categories: corrugation, spalling, squat, and wheel burn. The dataset includes 7155 annotated instances. Before partitioning, the annotation files were screened for missing image-label pairs, invalid class identifiers, non-finite or out-of-range normalized coordinates, and boxes with non-positive dimensions. Table 1 reports the number of instances and positive images for each class. Because an image may contain multiple classes, the class-wise image counts are not mutually exclusive. A stratified annotation audit to quantify class and box consistency is described in Section 3.9.2. In Tables 1 and 2, counts are reported as n, with percentages in parentheses where shown.

Table 1. Class distribution of the rail-surface defect dataset.

Defect Class	Images	Instances	Instances per Positive Image
Corrugation	987 (24.6)	1452 (20.3)	1.47
Spalling	1506 (37.5)	2208 (30.9)	1.47
Squat	1710 (42.5)	2949 (41.2)	1.72
Wheel burn	474 (11.8)	546 (7.6)	1.15
Total	4020	7155	1.78 per dataset image

Table 2. Training, validation, and held-out test partition.

Subset	Images	Instances	Corrugation	Spalling	Squat	Wheel Burn
Training	2819 (70.1)	5013 (70.1)	1019	1526	2097	371
Validation	599 (14.9)	1088 (15.2)	222	353	418	95
Held-out test	602 (15.0)	1054 (14.7)	211	329	434	80
Total	4020 (100)	7155 (100)	1452	2208	2949	546

As summarized in Table 2, a 64-bit perceptual hash was computed for each image to reduce optimistic bias from visually similar images appearing across subsets. Images with a Hamming distance of four bits or fewer were grouped before partitioning, and each group was assigned entirely to one subset. Four bits correspond to 6.25% of the hash

length and provide a conservative screening radius for small global appearance changes while limiting excessive grouping of images that merely share rail texture. Because the appropriate perceptual-hash cutoff depends on the image source, Section 3.9.3 evaluates thresholds of 2, 4, 6, and 8 bits [22]. The resulting 70/15/15 split contains 2819 training images, 599 validation images, and 602 held-out test images. The split seed was 20260811; 48 groups contained more than one image, the largest group contained three images, and no candidate pair identified at the four-bit threshold crossed subset boundaries. The held-out test set was excluded from training, validation, hyperparameter tuning, and checkpoint selection; the term does not imply independence across acquisition sessions, rail sections, routes, or source domains. Acquisition-session and rail-section identifiers were not recorded, so visual similarity was used to construct the near-duplicate groups.

3.2. Experimental Setup and Evaluation

The evaluation proceeded in five stages. First, all images and YOLO annotations were checked for readability, class validity, and duplicate identifiers. Second, near-duplicate groups were assigned intact to the training, validation, or held-out test subset. Third, the baseline and ablation variants were initialized from the same pretrained checkpoint and trained with identical optimization settings. Fourth, checkpoint selection relied solely on validation performance. Fifth, the selected checkpoints were evaluated on the held-out test set. Detection, class-wise, robustness, natural-condition, and qualitative results were computed using this fixed test partition; model complexity was computed from the instantiated networks, while latency and memory were measured under the batch-size-1 deployment protocol described below.

All formal ablation experiments used the same dataset split, COCO-pretrained YOLOv8n initialization, augmentation policy, optimizer, input resolution, batch size, stopping rule, checkpoint-selection criterion, and evaluation code. Only the component specified by each ablation condition was modified. Training was performed at the native resolution of 416 × 416 pixels. Each configuration was trained with seven deterministic seeds. Automatic mixed precision was enabled, while RAM image caching was disabled to preserve deterministic data handling. For cross-architecture comparisons, each detector retained its native model definition and COCO-pretrained initialization. The dataset split, input resolution, training budget, augmentation settings, checkpoint-selection criterion, and test procedure remained fixed. The complete settings are summarized in Table 3.

Table 3. Training and evaluation settings.

Setting	Value Used for All Formal Experiments
Framework/hardware	Python 3.11.15; PyTorch 2.11.0+cu12; Ultralytics 8.4.37; RTX 5060 Laptop GPU
Initialization/input	COCO-pretrained YOLOv8n; 416 × 416 pixels; batch size 32
Training budget	100 epochs; patience = 100; AMP enabled
Optimizer	SGD; lr0 = 0.01; lrf = 0.01; momentum = 0.937; weight decay = 0.0005
Loss weights	box = 7.5; cls = 0.5; DFL = 1.5
Augmentation	HSV; translation = 0.1; scale = 0.5; horizontal flip = 0.5; mosaic = 1.0; mixup = 0
Repeated runs	Seven deterministic seeds; deterministic mode enabled
Checkpoint selection	Highest validation mAP@0.5:0.95; test set excluded from selection
Test settings	Framework default confidence; IoU = 0.7; maximum detections = 300

Detection performance on the held-out test set was quantified using precision (P), recall (R), mAP@0.5, and mAP@0.5:0.95. P and R denote TP/(TP + FP) and TP/(TP + FN), respectively, where TP, FP, and FN are true positives, false positives, and false negatives. mAP@0.5 is the mean class-wise average precision at an IoU threshold of 0.5, whereas mAP@0.5:0.95 averages the metric over IoU thresholds from 0.5 to 0.95 in 0.05 increments.

For experiments on the self-built dataset, accuracy values are arithmetic means across seven deterministic seeds, and the reported SD is the sample SD of the seven seed-level test results. Absolute differences between percentages are expressed in percentage points (pp). Parameter counts and GFLOPs are deterministic quantities computed from the instantiated models at an input resolution of 416 × 416. Unless stated otherwise, tabulated P, R, and mAP values are percentages; parameter counts are in millions, checkpoint and memory values are in MiB, and latency is in milliseconds. Two-sided 95% confidence intervals (CIs) for seven-seed means were calculated using Student's t distribution with six degrees of freedom. The primary statistical comparison used a two-sided paired *t*-test on seed-matched mAP@0.5:0.95 results from YOLOv8n and ESiV-YOLOv8; statistical significance was assessed at alpha = 0.05.

Deployment benchmarking used the selected checkpoints with identical 416 × 416 inputs, a batch size of 1, and the same preprocessing, postprocessing, warm-up, and repeated timing procedures on an NVIDIA GeForce RTX 5060 Laptop GPU running PyTorch 2.11.0+cu12. Model-only latency excludes preprocessing and postprocessing, whereas end-to-end latency includes them. Peak allocated GPU memory was recorded using CUDA statistics. Stage-wise latency is reported as the mean ± SD across five independent timing sessions conducted after warm-up, with CUDA synchronization at each measured boundary. Robustness was evaluated without retraining by applying the fixed darkening, brightening, Gaussian blur, motion blur, and Gaussian noise transformations described in Section 3.7 to the held-out test images.

3.3. Component Effectiveness: Ablation Results

Table 4 presents a hierarchical ablation design comprising the YOLOv8n baseline, three single-module variants, three pairwise combinations, and the complete ESiV-YOLOv8 model. All variants use the same split, initialization, training settings, checkpoint-selection rule, and seven seeds. EffectiveSE improved mAP@0.5:0.95 by 1.3 percentage points; SIoU improved it by 1.6 percentage points without changing inference-time complexity; and VoV-GSCSP improved it by 0.6 percentage points while reducing parameters and GFLOPs. The EffectiveSE + SIoU, EffectiveSE + VoV-GSCSP, and SIoU + VoV-GSCSP combinations reached 66.5%, 65.6%, and 66.2% mAP@0.5:0.95, respectively. Their gains over the baseline were 2.9, 2.0, and 2.6 percentage points. The complete model achieved 68.6% mAP@0.5:0.95, 5.0 percentage points above the baseline. It exceeded the strongest pairwise variant by 2.1 percentage points, supporting the complementary use of the three modifications under the matched protocol. The corresponding 95% CIs for mean mAP@0.5:0.95 were 63.16–64.04% for YOLOv8n and 68.26–68.94% for ESiV-YOLOv8. In the seed-matched comparison, the paired mean difference was 5.0 percentage points (95% CI: 4.68–5.33 percentage points; paired t(6) = 36.1; $p < 0.001$).

Table 4. Component-wise ablation results.

Variant	P	R	mAP@0.5	mAP@0.5:0.95	Params	GFLOPs
YOLOv8n baseline	94.2 ± 0.41	91.2 ± 0.55	94.2 ± 0.43	63.6 ± 0.48	3.012	3.4634
EffectiveSE only	95.8 ± 0.35	92.6 ± 0.54	95.4 ± 0.31	64.9 ± 0.46	3.028	3.4689
SIoU only	96.2 ± 0.32	93.2 ± 0.51	96.6 ± 0.26	65.2 ± 0.41	3.012	3.4634
VoV-GSCSP only	94.8 ± 0.45	91.9 ± 0.55	95.4 ± 0.39	64.2 ± 0.52	2.480	3.0360
EffectiveSE + SIoU	97.1 ± 0.30	94.1 ± 0.49	97.1 ± 0.29	66.5 ± 0.37	3.028	3.4689
EffectiveSE + VoV-GSCSP	96.4 ± 0.32	93.5 ± 0.43	96.2 ± 0.38	65.6 ± 0.45	2.496	3.0415
SIoU + VoV-GSCSP	96.3 ± 0.38	93.6 ± 0.42	96.8 ± 0.30	66.2 ± 0.41	2.480	3.0360
ESiV-YOLOv8	97.7 ± 0.27	94.6 ± 0.39	97.2 ± 0.28	68.6 ± 0.37	2.496	3.0415

3.4. Class-Wise Detection Results

Table 5 reports class-wise seven-seed means for the 602-image test set, with SD included for the primary mAP@0.5:0.95 metric. Class-wise image counts are not mutually exclusive because a test image may contain more than one defect category; the overall row therefore reports the 602 unique test images rather than the sum of the class rows. ESiV-YOLOv8 improved mAP@0.5:0.95 over YOLOv8n by 5.3 percentage points for corrugation, 4.4 for spalling, 5.1 for squat, and 4.8 for wheel burn. Wheel-burn recall increased from 89.4% to 93.5%, and its mAP@0.5:0.95 increased from 62.5% to 67.3%. The gains across all four categories indicate that the overall improvement is not confined to the more frequent classes.

Table 5. Class-wise detection performance.

Model	Class	Test Support	P	R	mAP@0.5	mAP@0.5:0.95
YOLOv8n	Corrugation	140/211	94.9	90.3	93.5	64.2 ± 0.64
	Spalling	226/329	94.4	92.2	94.7	64.2 ± 0.52
	Squat	260/434	95.5	93.1	96.0	64.1 ± 0.48
	Wheel burn	71/80	93.2	89.4	92.5	62.5 ± 0.95
	Overall	602/1054	94.2	91.2	94.2	63.6 ± 0.48
ESiV-YOLOv8	Corrugation	140/211	97.6	93.7	97.8	69.5 ± 0.59
	Spalling	226/329	98.2	94.7	97.8	68.6 ± 0.50
	Squat	260/434	98.7	95.8	98.4	69.2 ± 0.42
	Wheel burn	71/80	96.4	93.5	95.6	67.3 ± 0.87
	Overall	602/1054	97.7	94.6	97.2	68.6 ± 0.37

3.5. Comparison with Recent Rail-Defect Detectors

Table 6 contains two comparison groups. The first four rows reproduce results from the corresponding publications and provide context for recent rail-defect detectors. Because their datasets, class definitions, splits, input sizes, and training procedures differ, these values are not used for direct ranking. The remaining six rows report detectors evaluated on the self-built dataset using the common split, 416 × 416 input, training budget, checkpoint-selection rule, and test procedure described in Section 3.2. Their accuracy values are presented as the mean ± sample SD across seven deterministic seeds and therefore form the direct comparison group. Within this group, ESiV-YOLOv8 exceeded YOLOv5n, YOLOv9s, and YOLOv10n by 6.3, 1.1, and 1.9 percentage points in mean mAP@0.5:0.95, respectively. It achieved the highest mean precision and both mean mAP measures, whereas RT-DETR-R18 achieved the highest mean recall.

Table 6. External and same-protocol detector comparisons.

Method	Dataset/Task Scope	P	R	mAP@0.5	mAP@0.5:0.95
Improved YOLOv8n [13] (published)	Railway-track defects; external dataset	90.2	84.5	90.2	73.2
SNBF-YOLO [11] (published)	Rail-track and component damage; external dataset	82.2	76.2	77.2	44.5
Lightweight YOLOv8n [12] (published)	ProRail rail-surface defects; external dataset	96.3	77.5	85.7	51.6
RSD-YOLO [24] (published)	Four-class rail-surface defects; external dataset	66.7	83.8	79.7	56.2
YOLOv5n (same protocol)	Self-built rail-surface dataset	93.2 ± 0.42	90.5 ± 0.62	93.4 ± 0.45	62.3 ± 0.56
YOLOv8n (same protocol)	Self-built rail-surface dataset	94.2 ± 0.41	91.2 ± 0.55	94.2 ± 0.43	63.6 ± 0.48

Table 6. *Cont.*

Method	Dataset/Task Scope	P	R	mAP@0.5	mAP@0.5:0.95
YOLOv9s (same protocol)	Self-built rail-surface dataset	96.1 ± 0.37	93.2 ± 0.42	96.2 ± 0.32	67.5 ± 0.43
YOLOv10n (same protocol)	Self-built rail-surface dataset	95.5 ± 0.32	93.1 ± 0.53	96.0 ± 0.38	66.7 ± 0.46
RT-DETR-R18 [25] (same protocol)	Self-built rail-surface dataset	89.8 ± 0.51	95.0 ± 0.44	95.3 ± 0.41	59.4 ± 0.53
ESiV-YOLOv8 (same protocol)	Self-built rail-surface dataset	97.7 ± 0.27	94.6 ± 0.39	97.2 ± 0.28	68.6 ± 0.37

Note: Published results are reproduced from the cited studies and are provided only for the literature context because their datasets and experimental protocols differ. Only rows marked "same protocol" are directly comparable under the common experimental settings of this study.

3.6. Structural Complexity and Runtime Efficiency

Table 7 evaluates structural complexity and measured runtime under a common batch-size-1 protocol. Relative to YOLOv8n, ESiV-YOLOv8 reduces parameters by 17.1%, GFLOPs by 12.2%, checkpoint size by 15.8%, and peak allocated GPU memory by 9.7%. Across five timing sessions, model-only latency increased from 8.10 ± 0.17 to 8.21 ± 0.24 ms, and end-to-end latency increased from 10.19 ± 0.24 to 10.41 ± 0.48 ms. These changes correspond to increases of 1.4% and 2.2%, respectively; the ESiV-YOLOv8 end-to-end value is equivalent to approximately 96.1 images per second. Thus, the proposed model reduces model scale and memory allocation with a small latency overhead in the tested PyTorch GPU configuration.

Table 7. Model complexity and measured runtime.

Model	Params	GFLOPs	Checkpoint	Model-Only Latency	End-to-End Latency	Peak GPU Memory
YOLOv8n	3.012	3.4634	5.96	8.10 ± 0.17	10.19 ± 0.24	30.61
ESiV-YOLOv8	2.496	3.0415	5.02	8.21 ± 0.24	10.41 ± 0.48	27.65

Parameters and GFLOPs measure model scale and arithmetic workload, whereas latency also depends on operator type, memory access, kernel launches, and backend optimization. The modified neck includes depthwise operations, branch concatenation, and channel shuffling, which can offset some of the arithmetic reduction at batch size 1. In this study, lightweight therefore denotes the measured reductions in model scale, theoretical computation, checkpoint size, and memory allocation; runtime is reported separately for the tested backend. The stage-wise results in Section 3.9.4 quantify the small latency differences across the processing pipeline.

3.7. Robustness to Controlled Perturbations

Table 8 reports the seven-seed mean performance after applying fixed image perturbations to the held-out test set without retraining; SD is reported for each mAP@0.5:0.95 value. Each drop is computed from the corresponding clean and perturbed means. Across the five perturbations, mAP@0.5:0.95 averaged 56.5% for YOLOv8n and 62.4% for ESiV-YOLOv8, corresponding to mean decreases of 7.1 and 6.2 percentage points from their clean results. ESiV-YOLOv8 retained an absolute advantage of 5.1–6.8 percentage points across the individual conditions. Motion blur produced the largest decrease for both models, while brightening produced the smallest. Compared with YOLOv8n, the reduction in degradation ranged from 0.1 percentage points under brightening to 1.8 percentage points under motion blur.

Table 8. Robustness under image perturbations.

Condition	Perturbation	YOLOv8n	Drop	ESiV-YOLOv8	Drop
Clean	None	63.6 ± 0.48	Ref.	68.6 ± 0.37	Ref.
Darkening	Intensity factor = 0.70	59.3 ± 0.51	4.3	64.9 ± 0.47	3.7
Brightening	Intensity factor = 1.30; clip to [0, 255]	60.6 ± 0.55	3.0	65.7 ± 0.42	2.9
Gaussian blur	Kernel = 5 × 5; sigma = 1.5	54.8 ± 0.72	8.8	60.6 ± 0.60	8.0
Motion blur	Length = 9 px; angle = 0°	50.6 ± 0.81	13.0	57.4 ± 0.75	11.2
Gaussian noise	Mean = 0; sigma = 0.03; seed = 2026	57.3 ± 0.56	6.3	63.3 ± 0.51	5.3

3.8. Qualitative Detection Results

Figure 3 compares detector outputs for coexisting wheel-burn and spalling defects under pronounced specular reflection. All models in the figure were trained and evaluated using the same dataset split, 416 × 416 input resolution, training budget, augmentation settings, checkpoint-selection criterion, and test procedure. ESiV-YOLOv8 and RT-DETR-R18 localized both defects without redundant boxes; their confidence scores were 0.90 and 0.89 for ESiV-YOLOv8 and 0.87 and 0.88 for RT-DETR-R18. YOLOv8n detected both defects with scores of 0.74 and 0.76, while YOLOv5n and YOLOv10n produced an additional wheel-burn prediction near the spalling region.

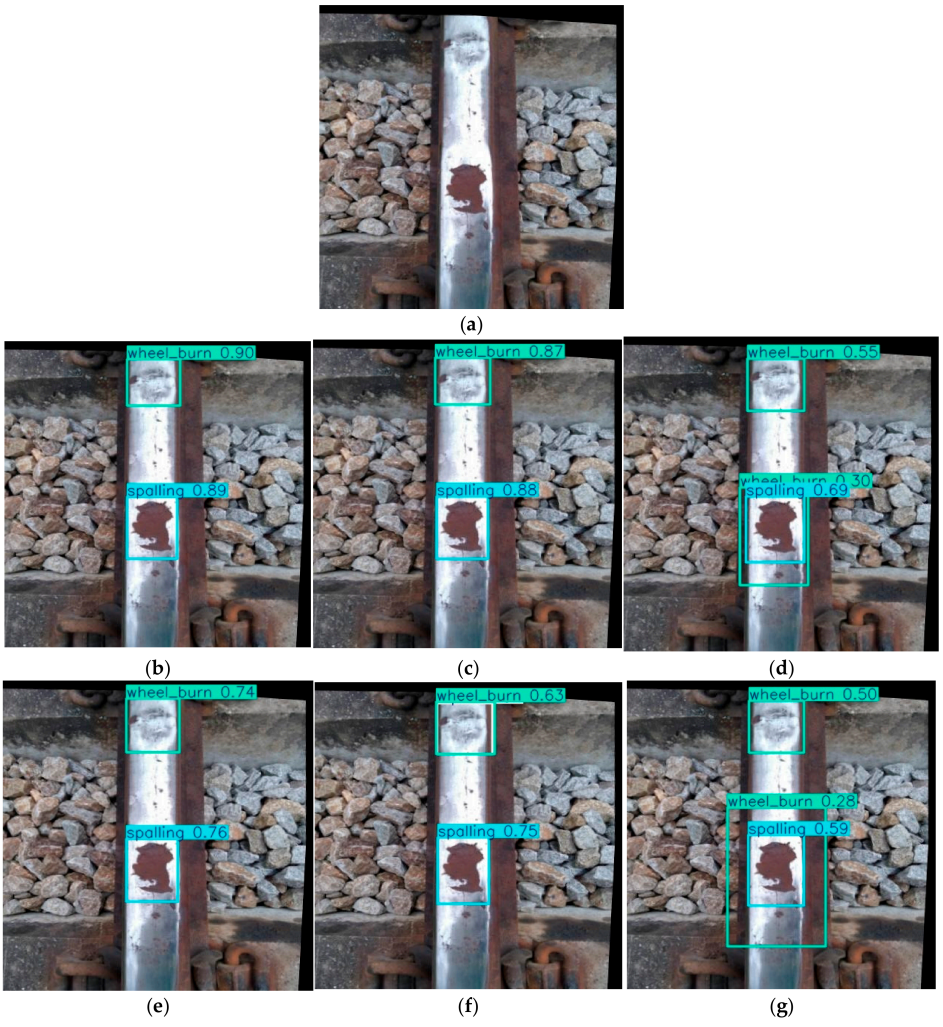

Figure 3. Detection comparison under specular reflection. (**a**) Original Image; (**b**) ESiV-YOLOv8; (**c**) RT-DETR-R18; (**d**) YOLOv5n; (**e**) YOLOv8n; (**f**) YOLOv9s; (**g**) YOLOv10n. Extra boxes in (**d**,**g**) indicate additional predictions near the spalling region.

3.9. Additional Validation Analyses

3.9.1. Evaluation Under Natural Field Conditions

Natural difficult cases were selected from the held-out test set before detector scores were examined. Strong illumination variation was defined as image intensity outside the 10th-90th percentiles; surface contamination required visible material covering at least 10% of the rail region; partial occlusion required a non-defect object to overlap at least 10% of an annotated defect box; and an alternative rail or environment required a distinct rail profile or surrounding scene, independently confirmed by two annotators. An image could satisfy more than one condition, so the four condition-specific subsets overlap; the combined subset is their union, with each image counted once. All subsets used the checkpoints, confidence threshold, and IoU settings from Section 3.2. Across the four condition-specific subsets, ESiV-YOLOv8 improved mAP@0.5:0.95 by 5.2–6.2 percentage points. On the 282-image combined subset, it achieved 59.6 ± 0.84%, compared with 54.0 ± 0.65% for YOLOv8n, an improvement of 5.6 percentage points. The detailed results are summarized in Table 9.

Table 9. Detection performance under natural field conditions.

Field Condition	Selection Criterion	Images	Instances	YOLOv8n	ESiV-YOLOv8	Difference
Strong illumination variation	Global grayscale mean below the 10th or above the 90th percentile	120	210	56.8 ± 0.81	62.2 ± 0.87	5.4
Surface contamination	Visible material covers at least 10% of the rail region	82	147	56.3 ± 1.12	61.5 ± 0.81	5.2
Partial occlusion	Non-defect object overlaps at least 10% of a defect box	67	118	50.1 ± 1.18	56.3 ± 1.10	6.2
Alternative rail/environment	Distinct rail profile or surrounding scene confirmed by two annotators	88	154	54.7 ± 1.17	60.4 ± 1.22	5.7
Combined field subset	Union of eligible subsets	282	494	54.0 ± 0.65	59.6 ± 0.84	5.6

3.9.2. Annotation-Quality Audit

The annotation-quality audit used 300 unique images, assigned to four 75-image class strata, and 500 sampled boxes distributed across the training, validation, and test subsets. A second annotator, trained on the project labeling guide, reviewed each class assignment and independently redrew each sampled box without access to model predictions. Class agreement was the proportion of the 500 boxes assigned the same category by both annotators, while box consistency was the mean IoU between the original and reviewed boxes. A box was counted as revised when the review changed its coordinates; a class correction was counted when the assigned category changed. The audit yielded 98.0% class agreement and a mean box IoU of 0.89. The review resulted in coordinate revisions for 23 boxes (4.6%) and class corrections for 10 boxes. The corrected annotation set was used to calculate the reported test metrics. The audit results are summarized in Table 10.

Table 10. Annotation-quality audit results.

Class	Audited Images	Audited Boxes	Class Agreement	Mean Box IoU	Boxes Revised	Class Corrections
Corrugation	75	100	98.0	0.89	5 (5.0)	2
Spalling	75	150	98.7	0.90	6 (4.0)	2
Squat	75	200	98.0	0.89	8 (4.0)	4
Wheel burn	75	50	96.0	0.87	4 (8.0)	2
Overall	300	500	98.0	0.89	23 (4.6)	10

3.9.3. Sensitivity to the Perceptual-Hash Threshold

Threshold sensitivity was evaluated by recomputing image pairs and connected groups at Hamming-distance thresholds of 2, 4, 6, and 8 bits, using the same 64-bit perceptual hash. Table 11 counts every image pair whose distance falls within the specified threshold and every resulting connected group containing at least two images. The number of qualifying pairs increased from 16 at two bits to 53, 119, and 324 at four, six, and eight bits, respectively. At four bits, 100 images formed 48 connected groups, and the largest group contained three images. Increasing the threshold to six and eight bits expanded the grouped set to 220 and 576 images and increased the largest group size to four and six. The four-bit threshold was retained because it captured more close image similarities than the two-bit setting without the much broader grouping produced by the six- and eight-bit settings. For each threshold, every connected group was assigned entirely to one subset using the same split seed; the small differences in subset counts result from assigning intact groups while maintaining the target 70/15/15 proportions.

Table 11. Sensitivity of near-duplicate grouping to the Hamming-distance threshold.

Threshold	Image Pairs	Connected Groups	Grouped Images	Largest Group	Resulting Split
2	16	16	32	2	2816/602/602
4 (used)	53	48	100	3	2819/599/602
6	119	106	220	4	2819/601/600
8	324	265	576	6	2816/601/603

3.9.4. Stage-Wise Runtime Profiling

Stage-wise profiling used the hardware, software, batch size, input resolution, warm-up, and the five-session timing procedure specified in Section 3.2, with CUDA synchronization at each measured boundary. The ESiV-YOLOv8 backbone, neck, and detection head required 2.80, 3.15, and 2.31 ms, compared with 2.72, 3.11, and 2.30 ms for YOLOv8n. The corresponding increases were 0.08, 0.04, and 0.01 ms. Model-only latency increased by 0.11 ms, from 8.10 to 8.21 ms, while postprocessing increased by 0.11 ms and preprocessing was unchanged. Consequently, end-to-end latency increased by 0.22 ms, from 10.19 to 10.41 ms. Complete-path totals were measured separately from the component timings. The difference column is ESiV-YOLOv8 minus YOLOv8n, and the relative change is this difference divided by the YOLOv8n value. The stage-wise measurements are summarized in Table 12.

Table 12. Stage-wise latency profiling.

Processing Stage	YOLOv8n	ESiV-YOLOv8	Difference	Relative Change
Preprocessing	1.00 ± 0.03	1.00 ± 0.05	0.00	0.0
Backbone	2.72 ± 0.09	2.80 ± 0.13	0.08	2.9
Neck	3.11 ± 0.12	3.15 ± 0.22	0.04	1.3

Table 12. *Cont.*

Processing Stage	YOLOv8n	ESiV-YOLOv8	Difference	Relative Change
Detection head	2.30 ± 0.07	2.31 ± 0.09	0.01	0.4
Postprocessing	1.09 ± 0.11	1.20 ± 0.32	0.11	10.1
Model-only total	8.10 ± 0.17	8.21 ± 0.24	0.11	1.4
End-to-end total	10.19 ± 0.24	10.41 ± 0.48	0.22	2.2

4. Conclusions

This study developed ESiV-YOLOv8 for four-class rail-surface defect detection by introducing complementary modifications to feature selection, box regression, and multiscale fusion. On the held-out, near-duplicate-aware test set, ESiV-YOLOv8 achieved 97.7% precision, 94.6% recall, 97.2% mAP@0.5, and 68.6% mAP@0.5:0.95. Compared with YOLOv8n, these results represent gains of 3.5, 3.4, 3.0, and 5.0 percentage points, respectively.

The single-module ablations clarify the role of each modification. EffectiveSE improved mAP@0.5:0.95 by 1.3 percentage points; SIoU improved it by 1.6 percentage points without changing inference-time model complexity; and VoV-GSCSP improved it by 0.6 percentage points while reducing parameters and GFLOPs. The three pairwise variants achieved 65.6–66.5% mAP@0.5:0.95, whereas the complete model reached 68.6%. Its 2.1 percentage-point advantage over the strongest pairwise variant supports the joint design rather than attributing the final gain to a single component.

ESiV-YOLOv8 reduced parameters, GFLOPs, checkpoint size, and peak allocated GPU memory by 17.1%, 12.2%, 15.8%, and 9.7%, respectively. Model-only latency increased from 8.10 to 8.21 ms, and end-to-end latency increased from 10.19 to 10.41 ms on the tested PyTorch GPU backend. Stage-wise profiling showed small increases of 0.04 ms in the neck and 0.11 ms in postprocessing. Under five controlled perturbations, ESiV-YOLOv8 retained a 5.1–6.8 percentage-point absolute advantage over YOLOv8n, while its mean decrease from the clean baseline was 6.2 percentage points, compared with 7.1 percentage points for YOLOv8n. It also retained a 5.6-percentage-point advantage on the combined natural-condition subset.

The annotation audit achieved 98.0% class agreement and a mean box IoU of 0.89. The threshold analysis identified 16, 53, 119, and 324 candidate pairs at two, four, six, and eight bits, respectively, supporting the four-bit cutoff as a conservative grouping threshold. The remaining evaluation scope is limited to one self-built dataset, one laptop GPU, and the PyTorch 2.11.0+cu12 inference backend. Future work will preserve acquisition metadata, expand the wheel-burn class, evaluate transfer to public rail-defect datasets with harmonized labels, and benchmark optimized inference on edge hardware.

Supplementary Materials: The following supporting information can be downloaded at: https://www.mdpi.com/article/10.3390/technologies14090583/s1, Representative non-confidential samples from the self-built Rail Surface Defect Dataset and their corresponding YOLO-format annotations are provided with this submission.

Author Contributions: Conceptualization, L.W. and W.X.; methodology, Y.S. and J.W.; software, Y.S.; validation, L.W. and Y.S.; formal analysis, J.W.; investigation, L.W.; resources, L.W. and W.X.; data curation, Y.S. and J.W.; writing—original draft preparation, L.W. and Y.S.; writing—review and editing, J.W., L.L., and W.X.; visualization, Y.S.; supervision, L.L. and W.X.; project administration, L.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The complete Rail Surface Defect Dataset contains proprietary field imagery and is not publicly available. Access may be requested from the corresponding author and is subject to permission from the data owner. Representative non-confidential samples from the self-built dataset and their YOLO-format annotations are provided in the Supplementary Materials.

Conflicts of Interest: Authors Lei Wang, Yuan Si, and Jun Wang were employed by Hangzhou Hanggang Metro Line 5 Co., Ltd. The company had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results. The remaining authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

YOLO	You Only Look Once
ESiV-YOLOv8	EffectiveSE-SIoU-VoV-GSCSP YOLOv8
EffectiveSE	Effective Squeeze-and-Excitation
SIoU	Scylla Intersection over Union
VoV-GSCSP	VoVNet-based GSConv Cross-Stage Partial module
C2f	CSP Bottleneck with Two Convolutions
SPPF	Spatial Pyramid Pooling-Fast
PAN	Path Aggregation Network
mAP	mean Average Precision
GFLOPs	giga floating-point operations

References

1. Yu, H.; Li, Q.; Tan, Y.; Gan, J.; Wang, J.; Geng, Y.-A.; Jia, L. A coarse-to-fine model for rail surface defect detection. *IEEE Trans. Instrum. Meas.* **2018**, *68*, 656–666. [CrossRef]
2. Yang, H.; Wang, Y.; Hu, J.; He, J.; Yao, Z.; Bi, Q. Deep learning and machine vision-based inspection of rail surface defects. *IEEE Trans. Instrum. Meas.* **2021**, *71*, 5005714. [CrossRef]
3. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*; IEEE: New York, NY, USA, 2016; pp. 779–788.
4. Jocher, G.; Chaurasia, A.; Qiu, J. *Ultralytics Yolo*; Zenodo: Geneva, Switzerland, 2023.
5. Li, H.; Wang, F.; Liu, J.; Song, H.; Hou, Z.; Dai, P. Ensemble model for rail surface defects detection. *PLoS ONE* **2022**, *17*, e0268518. [CrossRef]
6. Du, J.; Zhang, R.; Gao, R.; Nan, L.; Bao, Y. RSDNet: A new multiscale rail surface defect detection model. *Sensors* **2024**, *24*, 3579. [CrossRef]
7. Ding, Y.; Zhao, Q.; Li, T.; Lu, C.; Tao, L.; Ma, J. A rail defect detection framework under class-imbalanced conditions based on improved you only look once network. *Eng. Appl. Artif. Intell.* **2024**, *138*, 109351. [CrossRef]
8. Xu, C.; Liao, Y.; Liu, Y.; Tian, R.; Guo, T. Lightweight rail surface defect detection algorithm based on an improved YOLOv8. *Measurement* **2025**, *242*, 115922. [CrossRef]
9. Zhang, J.; Zhang, R.; Luan, F.; Zhang, H. PerMSCA-YOLO: A perceptual multi-scale convolutional attention enhanced YOLOv8 model for rail defect detection. *Appl. Sci.* **2025**, *15*, 3588. [CrossRef]
10. Cao, Y.; Ma, L.; Sun, Y.; Wang, F.; Su, S. Improved YOLOv8 for high-precision detection of rail surface defects on heavy-haul railways. *Chin. J. Electron.* **2025**, *34*, 802–815. [CrossRef]
11. Pang, Y.; Wang, X.; Tang, Y.; Lou, B.; Ning, L.; Xiong, H.; Teng, X.; Yuan, Q.; Bao, C.; Chen, J.; et al. A multi-scale adaptive framework for high-precision rail track damage detection via StarNet and bidirectional feature pyramid network. *Sci. Rep.* **2025**, *15*, 44099. [CrossRef]
12. Zhu, W.; Wang, W.; Shi, W.; Wang, Z.; Yang, Y. A lightweight YOLOv8n-based method for rail surface defect detection in complex railway environments. *Sci. Rep.* **2026**. [CrossRef]
13. Zhang, Z.; Zhang, L.; Lu, X.; Ma, T.; Huang, F.; Zhong, S. An improved YOLOv8n with multi-scale feature fusion for real time and high precision railway track defect detection. *Front. Artif. Intell.* **2025**, *8*, 1711309. [CrossRef]
14. Hu, T.; Gao, F.; Zhou, F. Railway obstacle intrusion detection and risk assessment based on MSIA-YOLOv8 and DALNet. *Expert Syst. Appl.* **2025**, *299*, 130132. [CrossRef]
15. Zhou, X.; Xu, M.; Pan, P. C5LS: An Enhanced YOLOv8-Based Model for Detecting Densely Distributed Small Insulators in Complex Railway Environments. *Appl. Sci.* **2025**, *15*, 10694. [CrossRef]

16. Deng, R.; Zou, Y.; Liu, S.; Ni, H.; Wang, R.; Ma, Y. Enhanced YOLOv8 for multi-hazard detection on railways using hybrid datasets. *Meas. Sci. Technol.* **2026**, *37*, 115403. [CrossRef]
17. Wu, X.; Yu, S. CSP-YOLO: An efficient and lightweight real-time algorithm for internal rail defect detection. *J. Supercomput.* **2025**, *81*, 1450. [CrossRef]
18. Wang, X.; Han, C.; Jin, W. BCD-YOLO: A railway perimeter foreign body intrusion detection method based on YOLOv8. *Meas. Sci. Technol.* **2025**, *36*, 056007. [CrossRef]
19. Lee, Y.; Park, J. Centermask: Real-time anchor-free instance segmentation. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: New York, NY, USA, 2020; pp. 13903–13912.
20. Gevorgyan, Z. SIoU loss: More powerful learning for bounding box regression. *arXiv* **2022**, arXiv:2205.12740.
21. Li, H.; Li, J.; Wei, H.; Liu, Z.; Zhan, Z.; Ren, Q. Slim-neck by GSConv: A lightweight-design for real-time detector architectures. *J. Real-Time Image Process.* **2024**, *21*, 62. [CrossRef]
22. Adimoolam, Y.K.; Poullis, C.; Averkiou, M. Data Leakage Detection and De-duplication in Large Scale Geospatial Image Datasets. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*; IEEE: New York, NY, USA, 2026; pp. 72–81.
23. Li, H.; Li, J.; Wei, H.; Liu, Z.; Zhan, Z.; Ren, Q. Intelligent computational methods for damage detection of laminated composite structures for mobility applications: A comprehensive review. *Arch. Comput. Methods Eng.* **2025**, *32*, 441–469. [CrossRef]
24. Pan, C.X. RSD-YOLO: An Improved YOLOv11n-Based Algorithm for Railway Surface Defect Detection. *Acad. J. Emerg. Technol.* **2026**, *3*, 93–102. [CrossRef]
25. Zhao, Y.; Lv, W.; Xu, S.; Wei, J.; Wang, G.; Dang, Q.; Liu, Y.; Chen, J. DETRs beat YOLOs on real-time object detection. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: New York, NY, USA, 2024; pp. 16965–16974.