



ARTICLE

SteelSpec-RTDETR: Defect-Aware Multi-Scale Transformer Detection for Low-Contrast Steel Surface Inspection

Mien Ka Duong¹ and Hoanh Nguyen^{2,*}

¹Faculty of Electrical Engineering Technology, Industrial University of Ho Chi Minh City, Ho Chi Minh City, Vietnam

²Robotics, Intelligent Control and Computer Vision Research Group, Faculty of Electrical Engineering Technology, Industrial University of Ho Chi Minh City, Ho Chi Minh City, Vietnam

*Corresponding Author: Hoanh Nguyen. Email: nguyenhoanh@iuh.edu.vn

Received: 10 July 2026; Accepted: 23 September 2026; Published: 21 October 2026

ABSTRACT: Accurate and efficient detection of steel surface defects is essential for automated industrial quality inspection. However, steel defects often exhibit small sizes, low contrast, irregular shapes, and strong similarity to background textures, making them difficult to localize reliably under real-time deployment constraints. To address these challenges, this paper proposes SteelSpec-RTDETR, an accuracy-efficiency-balanced multi-scale Transformer detector for industrial steel surface defect detection. The proposed model is developed based on the end-to-end detection paradigm of RT-DETR, while redesigning the feature extraction and feature fusion stages to better capture defect-sensitive representations. Specifically, a defect-oriented backbone is introduced, consisting of a Surface Background Smoothing Block, Local-Spectral Defect Mixing Blocks, and an Agent-based Defect Context Attention module. These components suppress background-dominant surface responses, enhance local and high-frequency defect cues, and capture compact long-range contextual dependencies. Furthermore, a Defect-Aware BiFPN and an Intra-Level Defect Aggregation module are designed to improve cross-scale feature interaction and adaptively integrate complementary representations from different fusion depths. The refined multi-scale features are then used as memory tokens for uncertainty-minimal query selection and decoding, enabling accurate end-to-end detection without non-maximum suppression. Experiments on the NEU-DET and GC10-DET datasets demonstrate that SteelSpec-RTDETR achieves a favorable balance between detection accuracy and computational cost. On NEU-DET, the proposed model achieves 81.6% mAP, 88.5% mAP₅₀, and 74.2% mAP₇₅. On GC10-DET, it obtains 76.4% mAP, 88.4% mAP₅₀, and 68.1% mAP₇₅. The complete model contains 35.6 M parameters, requires 59.4 G FLOPs, and achieves an inference latency of 22.7 ms.

KEYWORDS: Steel surface defect detection; industrial inspection; RT-DETR; Transformer detector; multi-scale feature fusion; object detection

1 Introduction

Surface defect detection is a critical task in modern steel manufacturing, where product quality, production efficiency, and safety requirements depend heavily on reliable inspection systems. Defects such as crazing, inclusions, scratches, pits, rolled-in scale, patches, oil spots, and weld lines may occur during rolling, cooling, transportation, or other production stages. If these defects are not detected accurately and in time, they can reduce material strength, affect downstream processing, and lead to economic losses. Therefore, automated steel surface defect detection has become an important research topic in industrial visual inspection.

Traditional steel defect inspection methods mainly rely on manual examination or hand-crafted image processing techniques. Manual inspection is labor-intensive, subjective, and difficult to maintain under high-speed production conditions. Conventional image processing methods, such as thresholding, edge detection, texture descriptors, and morphological analysis, can be effective in controlled environments but usually suffer from limited robustness when surface illumination, defect contrast, texture background, or defect morphology changes. Steel surface defects often appear as small, low-contrast, irregular, and texture-like regions. In many cases, the visual difference between defects and background patterns is subtle, making it difficult for traditional methods to distinguish true defects from rolling traces, scratches, illumination streaks, and material noise.

With the development of deep learning, convolutional neural networks (CNN) have significantly improved the performance of industrial defect detection. CNN-based object detectors, including two-stage and one-stage detection frameworks, can automatically learn discriminative representations from data and have been widely applied to surface inspection tasks. In particular, YOLO-series detectors have attracted attention due to their high inference speed and deployment-friendly design. However, CNN-based detectors still face several limitations in steel surface defect detection. First, repeated downsampling may weaken small and weak defect responses. Second, local convolutional operations may have limited ability to model long-range surface dependencies. Third, generic feature extraction backbones are usually designed for natural object detection and may not fully capture the characteristics of industrial steel defects, such as fine boundaries, weak grayscale transitions, and repetitive background interference. Recently, Transformer-based object detectors have provided an alternative detection paradigm by using attention mechanisms and set prediction. DETR [1] and its variants introduce object queries and bipartite matching to formulate object detection as an end-to-end prediction problem, reducing the dependence on hand-designed post-processing procedures such as non-maximum suppression. Deformable DETR [2] improves convergence and efficiency by attending to sparse sampling locations, while RT-DETR [3] further enhances real-time detection performance by combining efficient hybrid encoding, uncertainty-minimal query selection, and Transformer decoding. These methods have shown strong performance in general object detection. Nevertheless, directly applying generic Transformer detectors to steel surface defect inspection may still be suboptimal. Steel defects are often small, sparse, low-contrast, and visually similar to non-defective textures. Therefore, high-quality query initialization and defect-aware multi-scale feature representation are essential for accurate detection.

To address these challenges, this paper proposes SteelSpec-RTDETR, a deployment-oriented multi-scale Transformer detection framework that balances detection accuracy and computational efficiency for industrial steel surface defect inspection. The proposed model follows the end-to-end detection paradigm of RT-DETR, but redesigns the backbone and feature fusion stages to better adapt to steel surface inspection. Specifically, a defect-oriented backbone is introduced to extract hierarchical defect-sensitive features. A Surface Background Smoothing Block (SBS) is first used to suppress background-dominant surface responses while retaining foreground-sensitive defect cues. Then, Local-Spectral Defect Mixing (LSDM) blocks are designed to enhance mid-level representations by combining local spatial modeling, reduced agent spectral attention, and high-frequency feature enhancement. At the highest stage, an Agent-based Defect Context Attention (ADCA) module captures compact long-range contextual dependencies with limited computational overhead. In addition to the backbone design, SteelSpec-RTDETR introduces a defect-aware multi-scale feature fusion neck. Since steel defects vary significantly in size, shape, and contrast, robust detection requires both fine spatial information and high-level semantic context. To this end, a Defect-Aware BiFPN (DA-BiFPN) is developed to perform bidirectional cross-scale feature fusion while preserving the finest feature level. This design helps retain weak defect boundaries and small local anomalies that may disappear after repeated downsampling. Furthermore, an Intra-Level Defect Aggregation (ILDA) module

adaptively integrates same-level features from different fusion depths, allowing the model to combine local texture details and contextual representations more effectively. The refined multi-scale features are then used as memory tokens for uncertainty-minimal query selection and RT-DETR decoding, enabling accurate end-to-end prediction without non-maximum suppression. Extensive experiments are conducted on two steel surface defect detection datasets, namely NEU-DET and GC10-DET. The experimental results demonstrate that SteelSpec-RTDETR achieves a favorable balance between detection accuracy and computational efficiency. It is important to distinguish the architectural contributions of SteelSpec-RTDETR from the established operations on which they are built. The contribution of this work lies instead in their defect-oriented restructuring and coordinated integration: selectively smoothing mask-estimated background responses while retaining foreground-sensitive features; combining reduced-resolution agent context with local and high-frequency defect representations; retaining the finest pyramid level throughout bidirectional fusion; and aggregating same-level features across different fusion depths before RT-DETR query selection. Therefore, the novelty is positioned at the level of task-specific architectural design and information-flow organization for weak, small, and texture-like steel defects, rather than at the level of inventing new primitive attention or gating operators. The main contributions of this paper are summarized as follows:

- SteelSpec-RTDETR is presented as a defect-oriented reconfiguration of the RT-DETR framework for real-time steel surface inspection. The original RT-DETR query-selection, decoder, detection head, and set-prediction objective are retained, while the generic backbone and hybrid encoder are replaced with a task-specific feature extraction and multi-scale fusion pipeline.
- A Surface Background Smoothing Block is developed to decompose shallow features using a learned soft mask and apply smoothing selectively to the background-dominant component. Unlike indiscriminate early-stage smoothing, this asymmetric pathway retains foreground-sensitive responses and residual information while reducing repetitive texture interference. The novelty lies in the selective background-processing arrangement rather than in the individual gating or convolution operations.
- A heterogeneous defect-oriented backbone is constructed by combining the proposed selective background pathway with a Local-Spectral Defect Mixing configuration and a high-stage contextual-attention configuration. The proposed backbone integrates the SBS, LSDM, and ADCA modules to spatially reduced features and combine the resulting context with local spatial and high-frequency branches through content-dependent branch weighting, thereby balancing fine-defect preservation and efficient contextual reasoning.
- A defect-aware multi-scale fusion strategy is introduced based on an adapted BiFPN and the proposed Intra-Level Defect Aggregation configuration. The DA-BiFPN retains the established bidirectional weighted-fusion principle of BiFPN but keeps the finest feature level active throughout all fusion repeats. ILDA subsequently aggregates lateral, early-fusion, and late-fusion representations at the same pyramid level, allowing the model to select complementary information across fusion depths before uncertainty-minimal query selection and RT-DETR decoding.
- Comprehensive experiments are conducted to validate the effectiveness of the proposed method. Experimental results on the NEU-DET and GC10-DET datasets demonstrate that SteelSpec-RTDETR outperforms representative CNN-based and Transformer-based detectors in detection accuracy while maintaining practical real-time inference efficiency.

The remainder of this paper is organized as follows. [Section 2](#) reviews related work on steel surface defect detection, CNN-based detectors, and Transformer-based object detection. [Section 3](#) presents the proposed SteelSpec-RTDETR framework in detail. [Section 4](#) reports the experimental settings, comparative results, visualization analysis, and ablation studies. Finally, [Section 5](#) concludes the paper and discusses future research directions.

2 Related Work

2.1 Deep Learning-Based Object Detection

Deep learning-based object detection has developed rapidly in recent years and has become the dominant paradigm for visual inspection, autonomous perception, remote sensing, and industrial defect detection. Different from traditional methods based on hand-crafted features, deep detectors learn hierarchical representations directly from data and can jointly perform feature extraction, object classification, and bounding-box regression. Existing object detectors can generally be divided into CNN-based detectors and Transformer-based detectors. CNN-based detectors rely mainly on convolutional feature extraction and dense prediction, whereas Transformer-based detectors introduce attention mechanisms, object queries, and set prediction to improve global reasoning and end-to-end detection.

Early CNN-based object detectors are commonly represented by two-stage detection frameworks. R-CNN [4] and Fast R-CNN [5] first generate candidate regions and then classify each region using deep convolutional features. Faster R-CNN [6] further integrates a Region Proposal Network into the detection pipeline, allowing region proposals and object classification to share convolutional features. This design significantly improved detection accuracy and efficiency compared with earlier proposal-based methods. Later, feature pyramid networks were introduced to enhance multi-scale representation by combining high-level semantic features with low-level spatial details [7–10]. Two-stage detectors generally provide strong localization accuracy, but their region proposal and refinement procedures often result in higher computational cost, which may limit their suitability for real-time industrial inspection. To improve inference efficiency, one-stage CNN-based detectors directly predict object classes and bounding boxes from dense feature maps. SSD [11] uses multi-scale feature maps to detect objects of different sizes in a single forward pass, while RetinaNet [12] introduces focal loss to address the class imbalance between foreground and background samples. The YOLO series has become one of the most widely used families of real-time object detectors because of its favorable balance between accuracy and speed. Recent YOLO variants have continuously improved network design, feature aggregation, label assignment, and training strategies. YOLOv7 [13] introduces trainable bag-of-freebies to improve accuracy without increasing inference cost. YOLOv8 [14] adopts a modern anchor-free detection framework and provides flexible model scales for detection, segmentation, and related vision tasks. YOLOv9 [15] introduces programmable gradient information and GELAN to improve information preservation and parameter utilization. YOLOv10 [16] further moves toward end-to-end real-time detection by reducing dependence on non-maximum suppression and optimizing the architecture for efficiency. More recent YOLO11 models [17] continue this trend by improving the accuracy-efficiency trade-off for practical real-time detection. These CNN-based detectors are attractive for industrial deployment because of their compact structures and fast inference speed. However, their local convolutional operations may be insufficient for modeling long-range dependencies, and repeated downsampling can weaken small or low-contrast defect responses.

Transformer-based object detectors have recently attracted increasing attention due to their ability to model global context and object-level interactions. DETR reformulates object detection as a direct set prediction problem by using a Transformer encoder–decoder architecture, learnable object queries, and bipartite matching loss. This design removes the need for hand-crafted anchor generation and non-maximum suppression, providing a simple end-to-end detection framework. Nevertheless, the original DETR suffers from slow convergence and limited performance on small objects because global attention over image features is computationally expensive and spatially coarse. To address these limitations, Deformable DETR introduces deformable attention, where each query attends only to a sparse set of sampling points around reference locations. This greatly accelerates convergence and improves multi-scale detection, especially for small objects. Subsequent DETR variants further improved query formulation,

training stability, and detection accuracy. Conditional DETR [18] and DAB-DETR [19] improve query design by introducing spatial prior information into object queries. DN-DETR [20] and DINO [21] use denoising training and improved query initialization to stabilize optimization and accelerate convergence. In particular, DINO combines contrastive denoising, mixed query selection, and improved box prediction to achieve strong end-to-end detection performance. These methods demonstrate the potential of Transformer-based detectors for accurate object localization and global contextual reasoning. However, many high-performance DETR variants still require relatively high computational resources, which can be challenging for real-time industrial inspection systems. RT-DETR was proposed to bridge the gap between Transformer-based detection accuracy and real-time inference efficiency. It introduces an efficient hybrid encoder, uncertainty-minimal query selection, and a lightweight Transformer decoder, enabling real-time end-to-end detection without non-maximum suppression. Later RT-DETR variants [22,23] further improve training strategies, multi-scale feature sampling, and deployment flexibility. Compared with CNN-based real-time detectors, RT-DETR-style models provide stronger global reasoning and avoid post-processing dependency. Compared with conventional DETR variants, they are more suitable for practical real-time applications. Therefore, RT-DETR provides a strong baseline framework for industrial defect detection.

Despite the progress of both CNN-based and Transformer-based detectors, directly applying generic object detectors to steel surface defect detection remains challenging. Steel defects are often small, sparse, low-contrast, irregularly shaped, and visually similar to background textures such as rolling traces, scratches, illumination streaks, and material noise. CNN-based detectors may miss weak defects due to downsampling and limited contextual modeling, whereas Transformer-based detectors may introduce unnecessary computation if global attention is applied without considering the special characteristics of surface defects. Therefore, a detector for steel surface inspection should preserve fine local defect details, suppress background-dominant responses, exploit multi-scale information, and maintain efficient inference. Motivated by these observations, the proposed SteelSpec-RTDETR is developed based on the end-to-end RT-DETR detection paradigm, while redesigning the feature extraction and feature fusion stages for steel surface defect detection. Unlike generic object detectors, the proposed model introduces defect-oriented feature extraction, defect-aware multi-scale fusion, and refined multi-scale query selection. This design aims to retain the real-time and NMS-free advantages of RT-DETR while improving sensitivity to weak, small, and texture-like industrial steel defects.

2.2 Surface Defect Detection in Industrial Inspection

Surface defect detection is an important branch of industrial visual inspection, aiming to identify and localize abnormal regions such as scratches, cracks, pits, inclusions, stains, rolled-in scale, weld lines, and surface deformation. Compared with general object detection, surface defect detection is more challenging because defects often exhibit irregular shapes, large scale variations, low contrast, weak boundaries, and high similarity to background textures. In addition, industrial environments usually impose strict requirements on inference speed, model size, and robustness across different production conditions. Recent studies have therefore focused on designing defect-oriented feature extraction modules, multi-scale fusion strategies, attention mechanisms, lightweight architectures, and data-efficient learning methods.

For steel strip and metal surface defect detection, many recent studies have improved YOLO-based detectors by introducing defect-aware feature extraction and multi-scale representation learning. Ma and Zhang [24] proposed PCT-Net, a position-guided hybrid CNN-Transformer network based on YOLOv5s for steel strip surface defect detection. The method introduces a CNN and Transformer Block to construct a hybrid backbone, an attention-based local and global feature fusion module to combine multi-scale features, and a target position-guided module to improve defect localization. Similarly, IAMF-YOLO [25]

improves YOLOv8 for metal surface defect detection by introducing an information augmentation network and a parallel multi-scale feature pyramid network. This design aims to reduce information loss during feature downsampling and strengthen the representation of small surface defects. These studies demonstrate that standard YOLO architectures require defect-specific modifications to better handle weak and small industrial anomalies. Lightweight and efficient detection has also become a major concern for industrial deployment. Zhang et al. [26] proposed LHMB-Net, a lightweight metal surface defect detection network with hierarchical multi-branch feature extraction and group decision attention. The hierarchical multi-branch feature extraction module captures defect features across multiple scales, while the partial group decision attention mechanism adaptively emphasizes important features and alleviates the influence of class imbalance. CSSI-YOLO [27] further addresses real-time rail surface defect detection by integrating C2f-LESKA attention, SPD-Conv, multi-scale feature fusion, and an improved Inner-CIoU loss into a YOLOv10n-based framework. These lightweight designs indicate that practical surface inspection systems require not only high detection accuracy but also low computational cost and stable real-time performance.

Attention mechanisms and multi-scale feature fusion are widely used to enhance defect representation. Du et al. [28] proposed AFF-Net for strip steel surface defect detection, where an adaptive focusing feature block and a focused feature pyramid network are designed to capture spatial and morphological variations of defects. The method adaptively assigns weights to defect-related features and improves the fusion of multi-scale representations. Chen and Liu [29] proposed MSANet, a lightweight multi-scale attention network for steel surface defect recognition. Although formulated as a recognition task, MSANet shows that multi-scale feature extraction and attention-based feature refinement can effectively emphasize defect-relevant regions while maintaining computational efficiency. HVA-YOLO [30] extends this direction to real-time instance segmentation for steel plate surface defects by incorporating hybrid visual attention, multi-head self-attention, hybrid pooling, and depthwise separable convolution. Unlike bounding-box detection, instance segmentation provides pixel-level defect regions, which can support more detailed quality assessment and process monitoring.

Beyond CNN- and YOLO-based architectures, recent studies have explored Transformer, state-space, and graph-based designs for surface defect inspection. Liu et al. [31] proposed LGP-DETR, an end-to-end detection model for light guide plate surface defects. It integrates a deformable Transformer fusion layer for multi-scale feature capture, dynamic upsampling for edge preservation, and a channel attention module for background noise suppression. Gao et al. [32] proposed CBH-YOLO, which combines cross-stage Mamba-enhanced multi-scale convolution, a binary amplification matrix, and a hierarchical semantic graph fusion network for steel surface defect detection. These works suggest that more advanced contextual modeling mechanisms are increasingly being adopted to overcome the limited receptive field and insufficient feature discrimination of conventional convolutional detectors. Data scarcity is another important challenge in industrial surface defect detection. Defect samples are often rare, expensive to annotate, and highly imbalanced across categories. To address this issue, several studies have explored few-shot, semi-supervised, generative, and foundation-model-based approaches. Yan et al. [33] proposed EFINet for few-shot metal surface defect segmentation. The method introduces prior-guided bidirectional feature interaction, context-aware attention-guided feature aggregation, and prototype-loss compensation to improve generalization to unseen defect categories. Liu et al. [34] designed an attention-guided few-shot learning framework for metal surface defect classification, using dual-branch attention and cross-set guided attention to enhance discriminative feature learning under limited data. Semi-SSDDet [35] uses unlabeled data for semi-supervised steel surface defect detection through adaptive pseudo-label thresholding, dynamic label assignment, dilated convolution, and Haar wavelet downsampling. These methods reduce dependence on fully labeled datasets and improve model adaptability in data-scarce industrial scenarios.

Generative and foundation-model-enhanced methods have also been investigated to improve defect inspection under limited annotations. Wang et al. [36] proposed a progressive feature injection and global-local discrimination-based image generation network for rail surface defect inspection. The method first learns background features from defect-free samples and then injects defect foreground features into generated mask regions, improving both image diversity and downstream inspection performance. Cao et al. [37] introduced a Segment Anything Model-enhanced method for rail surface defect detection in heavy-haul railway scenarios. The method extracts rail regions to reduce background interference, generates prompt information using an unsupervised student-teacher similarity network, and adapts SAM for task-specific rail defect segmentation. More recently, ID-RAG [38] reinterprets retrieval-augmented generation for industrial defect detection by retrieving image-internal evidence, including semantic classification information and statistical anomaly priors, to ground vision-language model outputs. These works indicate that generative learning and foundation models may become useful complements to conventional supervised detection, especially when labeled defect samples are limited. Overall, existing methods have achieved significant progress by improving YOLO-based detectors, introducing Transformer or attention mechanisms, enhancing boundary localization, and reducing dependence on labeled data. However, several limitations remain. Many CNN-based methods still rely heavily on local convolutional operations and may have limited ability to model long-range surface dependencies. Some Transformer- or foundation-model-based methods provide stronger contextual reasoning but may introduce high computational cost or require additional adaptation for real-time deployment. Moreover, small, weak, and texture-like steel defects can still be suppressed by downsampling or confused with repetitive background patterns. Motivated by these observations, the proposed SteelSpec-RTDETR is designed to provide an effective accuracy-efficiency trade-off for steel surface defect detection. Unlike generic object detectors, it integrates defect-oriented feature extraction, background response suppression, local-spectral defect mixing, compact contextual attention, defect-aware multi-scale fusion, and refined query selection within an end-to-end RT-DETR-based framework. This design aims to improve sensitivity to weak and small defects while maintaining practical inference efficiency for industrial inspection.

2.3 Methodological Gap

Existing attention and feature-fusion mechanisms provide useful building blocks, but they do not directly resolve the combination of failure modes encountered in low-contrast steel defect detection. Generic channel- or spatial-weighting mechanisms emphasize informative responses but usually reweight an entire feature tensor. When repetitive steel textures, illumination streaks, rolling traces, and genuine defects generate similar activations, such reweighting may amplify both defect and background responses. Uniform smoothing presents the opposite problem: it can suppress surface noise but may simultaneously remove the weak grayscale discontinuities associated with small scratches, pits, and inclusions. Therefore, the shallow stage requires selective background processing rather than uniform enhancement or suppression.

Efficient attention mechanisms address the limited receptive field of convolution, but their use alone does not guarantee defect-sensitive representation. Agent Attention [39], for example, efficiently transmits global information through a compact set of agent tokens. However, it is a generic contextual operator and does not explicitly preserve local boundaries, high-frequency defect cues, or background-foreground asymmetry. Applying attention directly to high-resolution features would also introduce unnecessary computation. The proposed design therefore uses Agent Attention in two restricted roles: as a reduced-resolution contextual branch within LSDM at the middle stages and as a compact high-level contextual operator within ADCA.

Existing feature pyramids exhibit related limitations. FPN [7] mainly propagates high-level semantics through a top-down path, whereas BiFPN [10] adds bidirectional propagation and normalized learnable fusion weights. These are established and effective cross-scale mechanisms. Nevertheless, generic fusion designs are not specifically organized to preserve the finest steel-defect feature throughout every fusion repeat. Moreover, repeated pyramid fusion normally retains only the final output of each scale, potentially discarding complementary texture information available before or during the earlier fusion stages. DA-BiFPN and ILDA address these two separate issues: DA-BiFPN keeps the finest pyramid level active in both fusion directions, while ILDA aggregates representations at the same scale but from different fusion depths.

Recent defect-specific methods also combine attention and multi-scale fusion. PCT-Net [24] uses a hybrid CNN-Transformer backbone, local-global feature fusion, and position guidance. AFF-Net [28] employs adaptive focusing and a focused feature pyramid. GC-Net [40] combines global attention with cascade feature fusion. These methods confirm the value of contextual reasoning and multi-scale interaction for surface inspection. SteelSpec-RTDETR differs in the organization of its representation pathway: selective background-only smoothing is performed before mid-level enhancement; reduced contextual attention is explicitly coupled with local and high-frequency branches; the finest feature level remains active through bidirectional fusion; and same-level representations are retained across fusion depths before four-level RT-DETR query selection. Thus, the proposed method is not differentiated by the mere presence of attention or feature fusion, but by how these mechanisms are assigned to specific stages and defect-related failure modes.

3 Methodology

3.1 Overall Model Architecture

Fig. 1 illustrates the overall architecture of the proposed SteelSpec-RTDETR for real-time steel surface defect detection. The model is developed based on the end-to-end detection paradigm of RT-DETR, but its feature extraction and feature fusion stages are redesigned to better address the characteristics of steel surface defects. In industrial steel inspection, defects are often small, low-contrast, texture-like, and easily confused with repetitive background patterns such as scratches, rolling traces, illumination streaks, and material noise. Therefore, the proposed architecture aims to enhance defect-sensitive local details while maintaining sufficient contextual modeling ability and real-time inference efficiency.

Given an input steel surface image, the model first employs a defect-oriented backbone to extract hierarchical feature representations. The backbone consists of a convolutional stem layer, a Surface Background Smoothing Block (SBS), two Local-Spectral Defect Mixing (LSDM) stages, and an Agent-based Defect Context Attention (ADCA) stage. The stem layer extracts shallow texture features from the input image. The SBS module is then used to suppress background-dominant surface responses while preserving foreground-sensitive defect cues. The following LSDM stages enhance mid-level defect representations by jointly modeling local spatial patterns, reduced spectral-context dependencies, and high-frequency defect responses. Finally, the ADCA module is applied at the highest stage to capture compact long-range context with limited computational overhead.

After backbone feature extraction, the multi-scale feature fusion module aggregates the hierarchical representations for robust defect detection. Since steel defects vary significantly in size, shape, contrast, and spatial extent, effective detection requires both fine localization information and high-level discriminative semantics. To this end, the proposed model employs a Defect-Aware BiFPN (DA-BiFPN) followed by an Intra-Level Defect Aggregation (ILDA) module. DA-BiFPN performs bidirectional cross-scale fusion so that high-level semantic information can be propagated to shallow feature maps, while fine-grained spatial details can also be transferred back to deeper levels. ILDA further aggregates same-level representations from different fusion depths to generate refined feature maps.

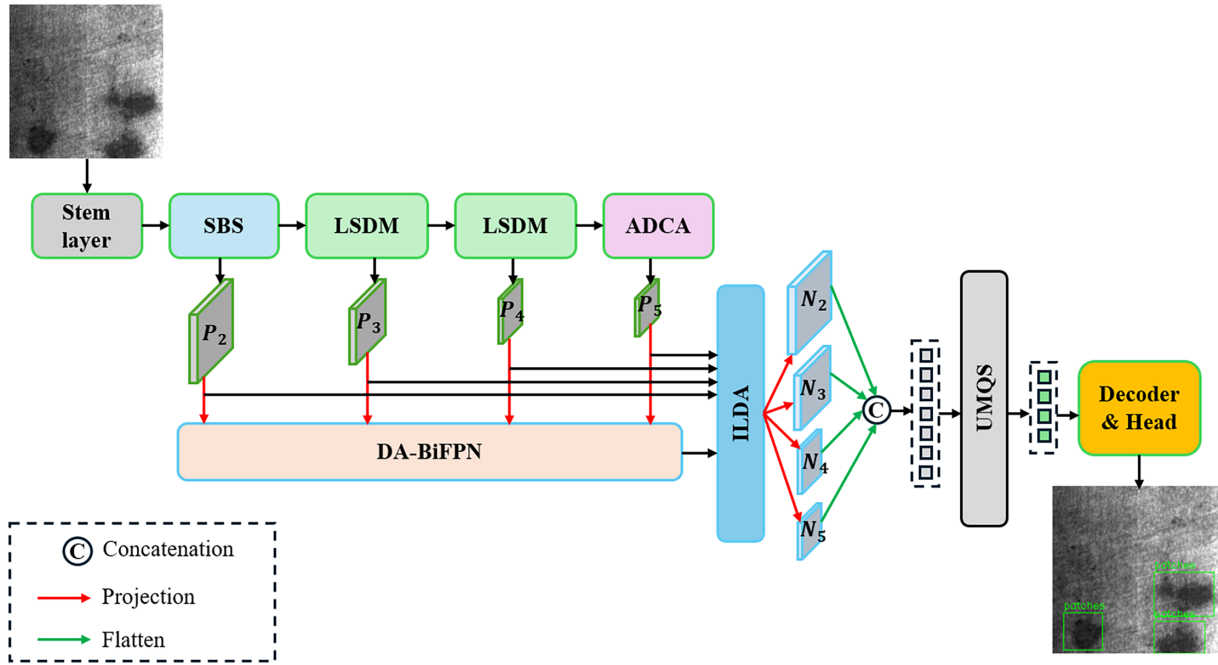


Figure 1: Overall architecture of the proposed SteelSpec-RTDETR framework. The model consists of a defect-oriented backbone, a defect-aware multi-scale feature fusion neck, UMQS-based query selection, and an RT-DETR decoder for end-to-end steel surface defect detection.

The refined multi-scale features are then flattened and used as memory tokens for query selection and decoding. Uncertainty-Minimal Query Selection (UMQS) selects high-quality candidate tokens from the multi-scale memory as initial object queries. This design allows the detector to initialize queries from both fine-resolution defect-sensitive features and high-level semantic representations. The selected queries are subsequently processed by decoder and detection head. The decoder performs query interaction, deformable cross-attention with the multi-scale memory, and iterative bounding-box refinement. The final head predicts a fixed set of defect categories and bounding boxes in an end-to-end manner. Hungarian matching is adopted during training to assign predictions to ground-truth defects, and the optimization objective consists of classification loss, bounding-box regression loss, and generalized IoU loss. During inference, the model directly outputs final detection results without requiring non-maximum suppression, which is beneficial for real-time industrial deployment.

3.2 Backbone

The backbone of the proposed SteelSpec-RTDETR is designed to extract defect-sensitive hierarchical features from steel surface images while maintaining real-time efficiency. Unlike generic classification backbones, which mainly emphasize semantic abstraction, steel surface inspection requires the preservation of weak local textures, thin defect boundaries, and subtle intensity discontinuities. Therefore, the proposed backbone follows a heterogeneous design: an initial convolutional stem extracts shallow surface features, a background-smoothing block suppresses repetitive non-defect textures, local-spectral mixing blocks enhance mid-level defect representations, and an agent-based attention block captures compact high-level context. Given an input image $X \in \mathbb{R}^{H \times W \times C_{in}}$, the backbone first generates a shallow feature map P_1 using a stem layer:

$$P_1 = \text{Stem}(X) \quad (1)$$

The stem contains two convolutional layers. The first layer is a 3×3 convolution with stride 2 that maps the three-channel input to 32 channels, followed by batch normalization and SiLU activation. The second layer is a 3×3 convolution with stride 1 that expands the representation to 64 channels, also followed by batch normalization and SiLU. For a 640×640 input, the stem therefore produces a $64 \times 320 \times 320$ shallow feature map with an effective stride of 2. SBS subsequently applies its stride-2 transition and generates P_2 at 160×160 , corresponding to an effective stride of 4. The following LSDM and ADCA transitions generate P_3 , P_4 , and P_5 at effective strides of 8, 16, and 32, respectively. The final backbone outputs are denoted as:

$$\{P_2, P_3, P_4, P_5\} = \mathcal{B}(X) \quad (2)$$

where P_2 retains fine defect texture, P_3 and P_4 encode mid-level local-spectral information, and P_5 provides compact semantic context. The backbone design follows the principle that early stages should remain strongly convolutional, while attention operations should be applied only after sufficient spatial reduction. The overall backbone transformation is formulated as:

$$P_2 = SBS(P_1) \quad (3)$$

$$P_3 = LSDM_3^2(\varphi_3(P_2)) \quad (4)$$

$$P_4 = LSDM_4^2(\varphi_4(P_3)) \quad (5)$$

$$P_5 = ADCA(\varphi_5(P_4)) \quad (6)$$

where $\varphi_3(\cdot)$, $\varphi_4(\cdot)$, and $\varphi_5(\cdot)$ denote stride-2 downsampling transition layers, SBS denotes Surface Background Smoothing Block, $LSDM^2(\cdot)$ denotes two consecutive Local-Spectral Defect Mixing Blocks, and $ADCA$ denotes the Agent-based Defect Context Attention stage.

3.2.1 Surface Background Smoothing Block

Steel surface images usually contain strong background patterns, including rolling traces, brushed textures, illumination streaks, and repetitive material noise. These background responses may have high-frequency characteristics similar to true defects, especially for small scratches, pits, and inclusions. Directly preserving all high-frequency responses at early stages can therefore increase false positives and make weak defects less distinguishable. To address this issue, we introduce the Surface Background Smoothing Block (SBS) after the stem layer. In the proposed backbone, SBS receives the shallow feature map P_1 and generates the first output pyramid level P_2 . As shown in Fig. 2a, SBS is implemented as a downsampling-refinement block. It first applies a stride-2 convolutional transition to obtain a raw second-stage feature:

$$\bar{P}_2 = \varphi_2(P_1) \quad (7)$$

where $\varphi_2(\cdot)$ denotes a lightweight convolutional downsampling layer. A soft background-aware mask is then predicted from $\bar{P}_2$:

$$M = \sigma(\text{Conv}_{1 \times 1}(\bar{P}_2)) \quad (8)$$

where $\sigma(\cdot)$ is the sigmoid function. The feature is decomposed into foreground-sensitive and background-dominant components:

$$P_{fg} = \bar{P}_2 \odot M \quad (9)$$

$$P_{bg} = \bar{P}_2 \odot (1 - M) \quad (10)$$

where $\odot$ denotes element-wise multiplication. The background component is processed by a lightweight smoothing operator:

$$S(P_{bg}) = PWConv_{up}(\delta(DWConv_{k \times k}(PWConv_{down}(P_{bg})))) \quad (11)$$

where $PWConv_{down}$ and $PWConv_{up}$ are point-wise convolutional layers for channel reduction and expansion, $DWConv_{k \times k}$ is a depth-wise smoothing convolution, and δ denotes a non-linear activation. The final P_2 feature is computed as:

$$P_2 = P_{fg} + S(P_{bg}) \quad (12)$$

This formulation preserves foreground-sensitive defect cues while smoothing background-dominant responses. The residual retention of P_{bg} prevents excessive information loss, while the smoothing term reduces the effect of repetitive surface noise.

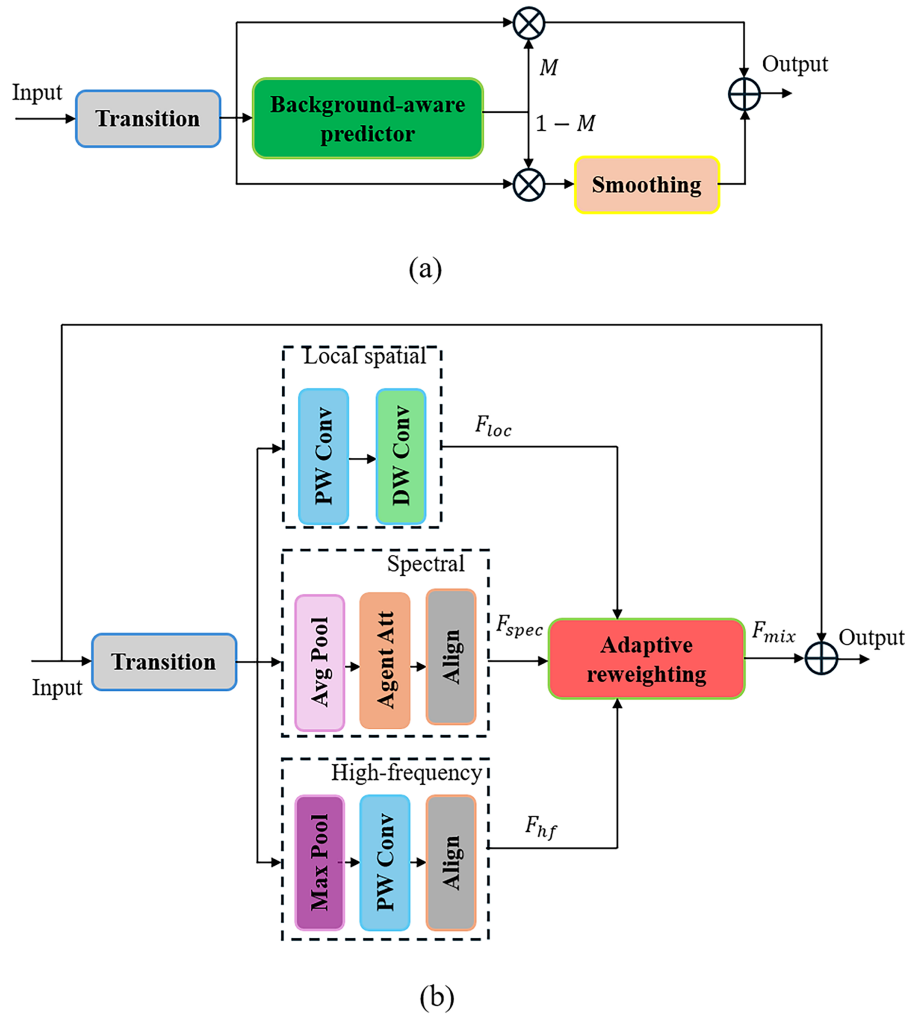


Figure 2: Structures of the proposed backbone modules. (a) Surface background smoothing block (SBS), which suppresses background-dominant surface responses while preserving defect-sensitive cues. (b) Local-Spectral Defect Mixing Block (LSDM), which integrates local spatial modeling, reduced spectral-context attention, and high-frequency enhancement.

3.2.2 Local-Spectral Defect Mixing Block

The middle stages of the backbone use the Local-Spectral Defect Mixing Block (LSDM) to enhance defect representations at P_3 and P_4 . These stages are important because the feature maps are sufficiently semantic to benefit from contextual reasoning, but still retain enough spatial resolution for defect localization. Each stage first applies a stride-2 transition and then stacks two LSDM Blocks. The LSDM Block contains three complementary branches: a local spatial branch, a reduced agent spectral attention branch, and a high-frequency enhancement branch, as shown in Fig. 2b.

Given an input feature $F \in \mathbb{R}^{H \times W \times C}$, the local spatial branch extracts short-range texture information using point-wise and depth-wise convolutions:

$$F_{loc} = DWConv_{3 \times 3}(PWConv(F)) \quad (13)$$

This branch is responsible for preserving local defect patterns such as scratch edges, pit boundaries, small cracks, and weak grayscale transitions.

The spectral branch is designed to capture broader surface dependencies without applying expensive attention to the full-resolution feature map. First, the input feature is spatially reduced:

$$F_r = AvgPool_r(F) \quad (14)$$

where r is the reduction ratio. Then, Reduced Agent Spectral Attention (RASA) is applied to the reduced feature map:

$$F_{spec} = Up_r(RASA(F_r)) \quad (15)$$

RASA uses Agent Attention [39] as its attention operator. Agent Attention introduces a compact set of agent tokens between query tokens and key-value tokens. The agent tokens first aggregate global information from the feature sequence and then broadcast the aggregated context back to the query tokens. Since the number of agent tokens can be much smaller than the number of original tokens, this mechanism reduces the cost compared with standard softmax attention while preserving global context modeling ability.

For a flattened reduced feature sequence $F_r \in \mathbb{R}^{N \times C}$, the RASA operation can be written as:

$$A = PoolAgent(F_r) \quad (16)$$

$$Z_A = Attn(A, K, V) \quad (17)$$

$$Z_Q = Attn(Q, A, Z_A) \quad (18)$$

where A denotes the agent tokens, Q , K , and V denote the query, key, and value projections, and Z_Q is the context-enhanced output sequence. After reshaping and upsampling, the spectral branch output F_{spec} is aligned with the original spatial resolution.

The high-frequency branch enhances edge-like and impulse-like defect responses:

$$F_{hf} = Up_r(PWConv(MaxPool_{3 \times 3}(F_r))) \quad (19)$$

The max-pooling operation emphasizes locally strong activations and helps retain narrow scratches, small pits, and sharp boundaries that may otherwise be weakened by repeated convolutional smoothing.

The three branches are fused using an adaptive branch reweighting mechanism. For each branch $b \in \{loc, spec, hf\}$, a global descriptor is obtained by global average pooling:

$$s_b = GAP(F_b) \quad (20)$$

A lightweight scoring function predicts branch importance:

$$u_b = W_2 \delta(W_1 s_b) \quad (21)$$

and normalized branch weights are computed by softmax:

$$\alpha_b = \frac{\exp(u_b)}{\sum_{j \in \{loc, spec, hf\}} \exp(u_j)} \quad (22)$$

The mixed feature is then obtained as:

$$F_{mix} = \alpha_{loc} F_{loc} + \alpha_{spec} F_{spec} + \alpha_{hf} F_{hf} \quad (23)$$

Finally, the LSDM Block adopts a residual output:

$$Y = F + PWConv(F_{mix}) \quad (24)$$

This residual formulation stabilizes optimization and preserves the original representation. Compared with direct concatenation of all branches, adaptive weighted fusion avoids excessive channel growth and is more suitable for real-time deployment. In the proposed backbone, two LSDM Blocks are used at P_3 and two LSDM Blocks are used at P_4 , providing sufficient mid-level defect enhancement without introducing excessive latency. In the default implementation, the P_3 and P_4 LSDM stages contain 128 and 256 channels, respectively, and each stage contains two consecutive LSDM blocks. The spatial-reduction ratio of the RASA branch is set to 2. RASA uses four attention heads at P_3 and eight heads at P_4 , with 49 agent tokens generated by adaptive pooling to a 7×7 agent grid. The local branch uses a point-wise convolution followed by a 3×3 depth-wise convolution. The high-frequency branch applies a 3×3 max-pooling operation to the reduced feature, processes the pooled response using a 1×1 point-wise convolution, and upsamples the resulting feature to the original spatial resolution. Each branch retains the stage channel dimension. For adaptive branch reweighting, every branch descriptor is processed by an independent two-layer scoring MLP with a channel-reduction ratio of 4, after which a softmax is applied across the three branch scores.

3.2.3 Agent-Based Defect Context Attention

At the highest backbone stage, the feature map becomes spatially compact and semantically stronger. Instead of using another full LSDM stage, we employ an Agent-based Defect Context Attention (ADCA) block to generate P_5 . The motivation is that, at this stage, global surface context is more important than explicit three-branch local-spectral-high-frequency decomposition. The spatial resolution of the feature map is already small, so attention can be applied with lower computational cost. Given the downsampled high-level feature F_5 , ADCA first applies normalization and Agent Attention:

$$\tilde{F}_5 = F_5 + AgentAttn(Norm(F_5)) \quad (25)$$

Then, a convolutional feed-forward network is used for local refinement:

$$P_5 = \tilde{F}_5 + ConvFFN(Norm(\tilde{F}_5)) \quad (26)$$

The $ConvFFN(.)$ module follows a point-wise expansion, depth-wise convolution, activation, and point-wise projection structure. The attention component captures long-range dependencies across the steel surface, while the $ConvFFN$ restores local spatial inductive bias. This is important because high-level context helps distinguish real defects from repetitive background patterns, but local convolutional refinement is still necessary to maintain defect boundary sensitivity. ADCA operates on the $512 \times 20 \times 20$ feature map. It uses eight attention heads and 49 agent tokens. Its convolutional feed-forward network has an expansion ratio of 4, resulting in an intermediate width of 2048 channels, and consists of a 1×1 expansion, 3×3 depth-wise convolution, SiLU activation, and 1×1 projection. Layer normalization is applied before Agent Attention and before the feed-forward network, and residual connections are used around both sublayers.

3.3 Multi-Scale Feature Fusion

After extracting the hierarchical backbone features, SteelSpec-RTDETR employs a defect-aware multi-scale fusion neck to generate a refined feature pyramid for subsequent UMQS-based query selection and RT-DETR decoding. Steel surface defects exhibit large variations in size, shape, contrast, and orientation. Some defects, such as pits and small inclusions, are highly localized and require fine spatial resolution, whereas scratches and rolled-in defects may extend over longer regions and require higher-level contextual reasoning. Therefore, a single feature level is insufficient for robust defect representation. The proposed neck consists of a lateral channel-alignment projection followed by two fusion components: Defect-Aware BiFPN (DA-BiFPN) and Intra-Level Defect Aggregation (ILDA). Given the backbone outputs $\{P_2, P_3, P_4, P_5\}$, the objective of the fusion neck is to produce a refined pyramid $\{N_2, N_3, N_4, N_5\}$, where N_2 preserves fine localization and surface texture, while N_3, N_4 , and N_5 provide progressively stronger semantic representations. In contrast to necks that discard the finest feature level, the proposed design keeps P_2 active throughout the fusion process. This is important because weak steel defects may remain visually meaningful only at shallow feature levels.

The backbone feature maps have different channel dimensions. Directly fusing them would require repeated channel conversion and would increase computational overhead. Therefore, we first apply a lateral 1×1 projection to align all pyramid levels to a unified channel width C_f :

$$\tilde{P}_i = Conv_{1 \times 1}(P_i) \quad (27)$$

here, $\tilde{P}_i$ denotes the channel-aligned feature at level i . In the default setting, C_f is set to 128 to balance representation capacity and inference efficiency. The projection layer also acts as a lightweight feature adapter between the heterogeneous backbone and the fusion neck.

3.3.1 Defect-Aware BiFPN

After channel alignment, the projected features are passed through the Defect-Aware BiFPN (DA-BiFPN). The purpose of DA-BiFPN is to propagate semantic information from high-level features to shallow levels while also allowing fine spatial information from lower levels to flow back to deeper representations. This bidirectional information exchange is useful for steel surface inspection because defect localization and defect discrimination are often distributed across different feature scales. Each DA-BiFPN repeat contains a top-down path followed by a bottom-up path. The top-down path transfers high-level semantic context to higher-resolution features:

$$T_4 = \mathcal{M}(\tilde{P}_4, Up(\tilde{P}_5)) \quad (28)$$

$$T_3 = \mathcal{M}(\tilde{P}_3, Up(T_4)) \quad (29)$$

$$T_2 = \mathcal{M}(\tilde{P}_2, Up(T_3)) \quad (30)$$

where $Up(\cdot)$ denotes a factor-2 upsampling operation and $\mathcal{M}(\cdot)$ denotes a weighted fusion node followed by convolutional refinement.

The bottom-up path then transfers the refined fine-grained information back to deeper levels:

$$O_3 = \mathcal{M}(T_3, Down(T_2)) \quad (31)$$

$$O_4 = \mathcal{M}(T_4, Down(O_3)) \quad (32)$$

$$O_5 = \mathcal{M}(\tilde{P}_5, Down(O_4)) \quad (33)$$

where $Down(\cdot)$ denotes a factor-2 downsampling operation. The output of one DA-BiFPN repeat is therefore:

$$\{O_2, O_3, O_4, O_5\} = \{T_2, O_3, O_4, O_5\} \quad (34)$$

In practice, two DA-BiFPN repeats are used in the default configuration. The first repeat performs initial cross-scale interaction, while the second repeat further refines defect-sensitive responses. More repeats may improve feature propagation in highly cluttered scenes, but they also increase latency. Therefore, two repeats provide a practical trade-off for real-time deployment.

For each fusion node, we use normalized learnable weights instead of direct summation or concatenation. Given a set of spatially aligned input features $\{U_j\}_{j=1}^m$, the normalized fusion weight is computed as:

$$\bar{w}_j = \frac{ReLU(w_j)}{\sum_{k=1}^m ReLU(w_k) + \tau} \quad (35)$$

where w_j is a learnable scalar and τ is a small constant for numerical stability. The fused feature is then computed as:

$$Y = DSCov_{3 \times 3} \left(\sum_{j=1}^m \bar{w}_j U_j \right) \quad (36)$$

where $DSCov_{3 \times 3}(\cdot)$ denotes a depth-wise separable convolution. This weighted fusion allows the network to adaptively select useful scale information. For example, fine scratches may rely more heavily on P_2 and P_3 , while larger patches or stains may benefit from P_4 and P_5 . Compared with concatenation-based fusion, weighted summation avoids channel growth and reduces both memory traffic and computational cost.

The defect-aware modification lies in preserving the finest pyramid level throughout all DA-BiFPN repeats. In many generic detection frameworks, very low-level features are either omitted or only weakly connected to the detection path. In contrast, SteelSpec-RTDETR keeps P_2 as an active participant in both top-down and bottom-up fusion. This design helps retain weak defect boundaries and small local anomalies that may disappear after repeated downsampling. All DA-BiFPN nodes operate at a fixed width of 128 channels. Nearest-neighbor interpolation is used in the top-down pathway, whereas bottom-up downsampling is implemented using a 3×3 depth-wise-separable convolution with stride 2. Every weighted fusion node is followed by a 3×3 depth-wise-separable convolution, batch normalization, and SiLU activation. The unnormalized fusion weights are constrained to be non-negative using a ReLU operation and are normalized using Eq. (35) with $\tau = 10^{-4}$. Two complete top-down/bottom-up repeats are used.

3.3.2 Intra-Level Defect Aggregation

Although repeated BiFPN fusion improves cross-scale interaction, relying only on the last fusion output may discard useful intermediate representations. Early fusion outputs may preserve stronger local texture, while later fusion outputs may contain more semantic and contextual information. To exploit these complementary representations, we introduce an Intra-Level Defect Aggregation (ILDA) module. For each pyramid level $l \in \{2, 3, 4, 5\}$, ILDA aggregates three same-level features: the lateral projection $\tilde{P}_l$, the output of the first DA-BiFPN repeat $F_l^{(1)}$, and the output of the second DA-BiFPN repeat $F_l^{(2)}$. Let

$$Z_l = \{\tilde{P}_l, F_l^{(1)}, F_l^{(2)}\} \quad (37)$$

For each candidate feature $z_l^{(r)} \in Z_l$, a global descriptor is first extracted by global average pooling:

$$s_l^{(r)} = \text{GAP}(z_l^{(r)}) \quad (38)$$

A lightweight scoring function then estimates the significance of each feature:

$$a_l^{(r)} = \text{MLP}(s_l^{(r)}) \quad (39)$$

The normalized aggregation coefficient is computed using a softmax function:

$$\alpha_l^{(r)} = \frac{\exp(a_l^{(r)})}{\sum_{q=0}^2 \exp(a_l^{(q)})} \quad (40)$$

The aggregated feature at level l is obtained by:

$$N_l = \text{DSConv}_{3 \times 3} \left(\sum_{r=0}^2 \alpha_l^{(r)} z_l^{(r)} \right) \quad (41)$$

This aggregation strategy allows the network to select information from different fusion depths in a content-adaptive manner. If a defect is dominated by local edge responses, the module may assign larger weights to shallow or early-fusion features. If the defect requires contextual discrimination from background texture, later BiFPN features may receive higher weights. Since all inputs to ILDA have the same spatial resolution and channel width, the aggregation can be implemented efficiently without expensive resizing or concatenation. The output of the multi-scale fusion neck is therefore:

$$\{N_2, N_3, N_4, N_5\} = \text{ILDA}(\{\tilde{P}_l\}, \{F_l^{(1)}\}, \{F_l^{(2)}\}) \quad (42)$$

3.4 Query Selection, Decoder, and Optimization Objective

After multi-scale feature fusion, the proposed SteelSpec-RTDETR follows the query selection and decoding strategy of RT-DETR [3]. Since the proposed fusion module has already performed cross-scale feature aggregation, it replaces the original RT-DETR hybrid encoder, including the AIFI and CCFF components. Therefore, no additional Transformer encoder is inserted after the proposed neck. Instead, the refined feature maps $\{N_2, N_3, N_4, N_5\}$ are directly used as the multi-scale memory for query selection and decoding. The four fused feature maps are first projected to the decoder hidden dimension and flattened:

$$\mathcal{M} = \text{Concat}(\text{Flatten}(\rho_2(N_2)), \text{Flatten}(\rho_3(N_3)), \text{Flatten}(\rho_4(N_4)), \text{Flatten}(\rho_5(N_5))) \quad (43)$$

where $\rho_i(\cdot)$ denotes a 1×1 projection for channel alignment. Positional and level embeddings are added following RT-DETR.

SteelSpec-RTDETR then uses the Uncertainty-Minimal Query Selection (UMQS) mechanism from RT-DETR to select high-quality memory tokens as initial object queries:

$$Q_0 = \text{UMQS}(\mathcal{M}, K) \quad (44)$$

where K is the number of selected queries. By using all four feature levels, UMQS can select query candidates from both fine-resolution and high-semantic representations. In particular, N_2 provides detailed localization cues for small defects, while N_3 , N_4 , and N_5 provide progressively stronger semantic and contextual information.

The selected queries and multi-scale memory are then forwarded to the RT-DETR decoder:

$$\hat{Y} = \text{Decoder}(Q_0, \mathcal{M}) \quad (45)$$

The decoder architecture, including query interaction, deformable cross-attention, iterative box refinement, and layer-wise prediction, follows RT-DETR and is not changed in this work. The final output is a fixed set of defect predictions:

$$\hat{Y} = \{(\hat{p}_i, \hat{b}_i)\}_{i=1}^K \quad (46)$$

where $\hat{p}_i$ and $\hat{b}_i$ denote the predicted class probability and bounding box of the i -th query, respectively. Since the model follows the RT-DETR end-to-end set prediction paradigm, no non-maximum suppression is required during inference.

Each of the four refined pyramid levels is projected from 128 to the decoder hidden dimension $d = 256$ using an independent 1×1 convolution. UMQS ranks all multi-scale memory tokens using the encoder classification score and selects the top $K = 300$ candidates. The decoder contains six layers, each using eight attention heads and multi-scale deformable cross-attention with four sampling points per head at each of the four feature levels. The feed-forward network dimension is 1024, and dropout is set to 0. The box reference points are iteratively refined after every decoder layer. Auxiliary classification and box-regression heads are attached to all six layers during training, while only the final-layer predictions are used during inference.

The optimization objective also follows RT-DETR. Hungarian matching is used to assign predicted queries to ground-truth defect instances. For each decoder layer, the detection loss is:

$$L_{det}^l = \lambda_{cls} L_{cls}^l + \lambda_{L1} L_{L1}^l + \lambda_{giou} L_{giou}^l \quad (47)$$

where the classification, L1 regression, and GIoU losses are implemented as in RT-DETR. Thus, the proposed method keeps the RT-DETR detection head and optimization pipeline unchanged, while replacing the original hybrid encoder with the proposed defect-oriented backbone and multi-scale feature fusion module. This design allows the model to exploit fine defect details from N_2 and contextual representations from N_3 – N_5 , while preserving the end-to-end and NMS-free advantages of RT-DETR.

4 Results and Discussion

4.1 Datasets and Evaluation Metrics

To evaluate the effectiveness and efficiency of the proposed SteelSpec-RTDETR, experiments are conducted on two widely used steel surface defect detection datasets, namely NEU-DET [41] and GC10-DET [42]. These two datasets cover different industrial surface conditions, defect categories, and annotation

characteristics, and therefore provide a suitable benchmark for assessing both detection accuracy and real-time deployment potential.

NEU-DET [41] is a steel surface defect detection dataset constructed from the Northeastern University surface defect database. It contains grayscale images collected from hot-rolled steel strip surfaces. The dataset includes six typical defect categories: crazing, inclusion, patches, pitted surface, rolled-in scale, and scratches. In total, NEU-DET contains 1800 grayscale images, with 300 images for each defect category. Compared with the original classification dataset, NEU-DET provides bounding-box annotations in XML format, where each defect instance is labeled with its category and bounding-box coordinates. The dataset contains nearly 5000 annotated defect boxes, making it suitable for evaluating both defect classification and localization performance. In our experiments, the dataset was divided into training, validation, and test subsets using a ratio of 7:2:1. Specifically, 1260 images were used for model training, 360 images were used for validation and checkpoint selection, and the remaining 180 images were retained as an independent test set. This partition preserves the commonly used 1260-image training subset while separating model selection from final performance evaluation.

GC10-DET [42] is a larger metallic surface defect detection dataset collected from real industrial production scenarios. It is designed to provide a more challenging benchmark for metallic surface inspection due to its larger number of images and richer defect categories. GC10-DET contains 3570 grayscale images of steel sheet surfaces and covers ten defect categories: punching, weld line, crescent gap, water spot, oil spot, silk spot, inclusion, rolled pit, crease, and waist folding. Compared with NEU-DET, GC10-DET contains more defect types and reflects more complex real-world surface conditions, such as low-contrast defects, production-related marks, local deformation, and surface contamination. Therefore, GC10-DET provides a complementary and more challenging in-domain benchmark for evaluating the robustness of SteelSpec-RTDETR across a broader range of metallic surface defects and background conditions. The complete GC10-DET dataset containing 3570 images was partitioned using the same 7:2:1 ratio, resulting in 2499 training images, 714 validation images, and 357 test images.

To comprehensively assess the proposed detector, both detection accuracy and computational efficiency are evaluated. Detection accuracy is measured using the COCO-style Average Precision (AP) protocol. Unless otherwise stated, mAP denotes the mean Average Precision averaged over all defect categories and ten IoU thresholds from 0.50 to 0.95 with an interval of 0.05, i.e., $\text{mAP}@ [0.50:0.05:0.95]$. In addition, mAP_{50} and mAP_{75} denote the mean Average Precision evaluated at single IoU thresholds of 0.50 and 0.75, respectively. All AP-based metrics are reported as percentages. For computational efficiency, Params, FLOPs, and Latency are used. Params measures the number of trainable parameters and reflects the model size, FLOPs indicates the computational complexity of a single forward pass, and Latency measures the average inference time per image. These metrics are used together to evaluate whether SteelSpec-RTDETR can achieve a favorable trade-off between detection accuracy and real-time deployment efficiency.

4.2 Implementation Details

All experiments were implemented using PyTorch and conducted on a workstation equipped with an NVIDIA GeForce RTX 5080 GPU. The proposed SteelSpec-RTDETR was trained and evaluated under the same experimental environment for both the NEU-DET and GC10-DET datasets to ensure a fair comparison. The backbone, defect-aware multi-scale fusion module, UMQS-based query selection module, and RT-DETR decoder were trained in an end-to-end manner. The detection head followed the set-prediction training strategy of RT-DETR, where classification and bounding-box regression objectives were jointly optimized during training. The model was optimized using the AdamW optimizer with an initial learning rate of 1×10^{-4} and a weight decay of 1×10^{-4} . The momentum coefficients of AdamW were set to $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The batch size was set to 8, and the model was trained for 100 epochs on each dataset.

Following the common RT-DETR configuration, the decoder used 300 object queries and six decoder layers to balance detection accuracy and inference efficiency. Regarding parameter initialization, SteelSpec-RTDETR was trained without ImageNet-pretrained weights because its defect-oriented backbone does not have a directly compatible pretrained checkpoint. The UMQS, decoder, prediction heads, reference-point initialization, and iterative box-refinement components followed the official RT-DETR initialization rules. For all reproduced comparison methods, the pretrained-weight usage and initialization strategy followed the corresponding official implementation, and no alternative initialization was introduced specifically for individual baselines. Because NEU-DET and GC10-DET contain relatively limited numbers of training images, on-the-fly data augmentation was applied only to the training sets to reduce memorization of dataset-specific surface textures and improve robustness to geometric and appearance variations. The augmentation pipeline included random horizontal flipping, scale jittering, IoU-constrained random cropping, and random image expansion or padding. Mild intensity and contrast perturbations were also applied to simulate variations in illumination and surface reflectance. Because both datasets consist of grayscale steel-surface images, aggressive hue and saturation transformations were not used. All geometric transformations were applied consistently to the corresponding bounding-box annotations, and boxes that became invalid after cropping were removed. During validation and testing, data augmentation was disabled, and the images were processed using deterministic resizing and normalization only. For every detector, model parameters were optimized using only the corresponding training subset. Performance was evaluated on the validation subset after each training epoch, and the checkpoint achieving the highest validation mAP was selected as the final model. When two checkpoints obtained the same validation mAP, the checkpoint with the higher validation mAP₇₅ was retained. The selected checkpoint was subsequently evaluated once on the held-out test subset to obtain the results. The test subsets were not used for early stopping, checkpoint selection, or hyperparameter adjustment.

4.3 Comparative Evaluation on the NEU-DET and GC10-DET Datasets

Tables 1 and 2 report the quantitative comparison between the proposed SteelSpec-RTDETR and representative CNN-based and Transformer-based object detectors on the NEU-DET and GC10-DET datasets, respectively. We used the official or publicly available implementations of YOLOv8, YOLOv9, YOLOv10, YOLO11, RT-DETR, and Deformable DETR. Every method was retrained using the fixed partitions and evaluated using the same input resolution of 640×640 pixels, preprocessing pipeline, augmentation policy, evaluation code, and hardware environment. Model-specific architectural settings and optimization procedures followed their official configurations unless otherwise stated. For every method, checkpoint selection was based exclusively on validation performance. Parameters, FLOPs, and inference latency were also measured under the same input dimensions and hardware conditions. As shown in Table 1, SteelSpec-RTDETR achieves the best overall detection performance on the NEU-DET dataset. The proposed model obtains 81.6% mAP, 88.5% mAP₅₀, and 74.2% mAP₇₅, outperforming all compared CNN-based and Transformer-based detectors. Among the CNN-based models, YOLO11-L achieves the highest mAP of 78.2%, while the proposed model improves the result by 3.4 percentage points. Compared with YOLO11-M, which provides a relatively lightweight CNN-based baseline, SteelSpec-RTDETR improves mAP from 77.0% to 81.6%. These results indicate that the proposed defect-oriented backbone and multi-scale fusion strategy provide stronger representation capability for steel surface defects than conventional CNN-based detectors. Compared with Transformer-based detectors, SteelSpec-RTDETR also shows clear advantages. RT-DETRv3-R50 achieves the strongest baseline performance among the compared Transformer-based models, with 79.5% mAP, 83.8% mAP₅₀, and 71.4% mAP₇₅. In contrast, the proposed model improves these values to 81.6%, 88.5%, and 74.2%, corresponding to gains of 2.1, 4.7, and 2.8 percentage points, respectively. The improvement in mAP₅₀ suggests that SteelSpec-RTDETR is more effective in identifying defect regions, while the improvement in mAP₇₅ indicates better localization accuracy under stricter IoU criteria. This is

particularly important for industrial defect inspection, where accurate localization is required for subsequent quality assessment and defect analysis.

Table 1: Quantitative comparison of the proposed SteelSpec-RTDETR and representative CNN-based and Transformer-based detectors on the NEU-DET dataset in terms of detection accuracy and computational efficiency.

Model	mAP (%)	mAP ₅₀ (%)	mAP ₇₅ (%)	Params (M)	FLOPs (G)	Latency (ms)
CNN-based models						
YOLOv8-M [14]	74.1	79.5	69.0	25.1	75.6	10.8
YOLOv8-L [14]	76.2	81.5	69.8	44.0	132.8	16.9
YOLOv9-M [15]	73.6	78.9	67.4	20.9	69.2	14.5
YOLOv9-C [15]	74.4	79.8	69.2	26.5	90.2	16.7
YOLOv10-M [16]	73.8	75.7	65.5	18.2	58.4	15.9
YOLOv10-L [16]	74.9	80.1	69.3	24.6	96.5	17.1
YOLO11-M [17]	77.0	82.8	70.1	20.1	56.0	14.3
YOLO11-L [17]	78.2	83.5	70.8	25.7	79.7	23.0
Transformer-based models						
RT-DETRv1-R18 [3]	75.0	80.0	69.1	18.9	44.7	21.1
RT-DETRv1-R50 [3]	78.1	83.2	70.5	44.0	120.2	31.3
RT-DETRv3-R18 [23]	77.2	82.4	70.0	20.6	53.8	19.5
RT-DETRv3-R50 [23]	79.5	83.8	71.4	44.7	124.0	26.4
Deformable DETR [2]	78.0	83.1	70.9	60.1	156.6	36.2
Our model						
SteelSpec-RTDETR	81.6	88.5	74.2	35.6	59.4	22.7

Table 2: Quantitative comparison of the proposed SteelSpec-RTDETR and representative CNN-based and Transformer-based detectors on the GC10-DET dataset.

Model	mAP (%)	mAP ₅₀ (%)	mAP ₇₅ (%)	Params (M)	FLOPs (G)	Latency (ms)
CNN-based models						
YOLOv8-M [14]	67.5	73.1	58.3	25.1	75.6	10.8
YOLOv8-L [14]	68.2	75.8	60.1	44.0	132.8	16.9
YOLOv9-M [15]	67.2	72.5	57.5	20.9	69.2	14.5
YOLOv9-C [15]	69.0	76.4	60.5	26.5	90.2	16.7
YOLOv10-M [16]	67.8	74.2	59.4	18.2	58.4	15.9
YOLOv10-L [16]	69.5	77.6	61.0	24.6	96.5	17.1
YOLO11-M [17]	69.9	79.1	62.5	20.1	56.0	14.3
YOLO11-L [17]	71.4	81.7	63.0	25.7	79.7	23.0
Transformer-based models						
RT-DETRv1-R18 [3]	69.5	77.5	59.8	18.9	44.7	21.1
RT-DETRv1-R50 [3]	71.7	80.2	62.8	44.0	120.2	31.3
RT-DETRv3-R18 [23]	70.5	79.5	62.7	20.6	53.8	19.5
RT-DETRv3-R50 [23]	73.1	83.2	65.6	44.7	124.0	26.4
Deformable DETR [2]	71.6	80.0	62.4	60.1	156.6	36.2
Our model						
SteelSpec-RTDETR	76.4	88.4	68.1	35.6	59.4	22.7

In terms of computational efficiency, SteelSpec-RTDETR uses 35.6 M parameters and 59.4 G FLOPs, with an inference latency of 22.7 ms. Although the proposed model is not the smallest or fastest detector among all compared methods, it achieves a favorable balance between accuracy and efficiency. In particular, compared with RT-DETRv1-R18, SteelSpec-RTDETR uses 88.4% more parameters, 32.9% more FLOPs, and 7.6% higher latency. This additional cost is accompanied by mAP improvements of 6.6 percentage points on NEU-DET and 6.9 percentage points on GC10-DET. Compared with RT-DETRv3-R18, the proposed model uses 72.8% more parameters, 10.4% more FLOPs, and 16.4% higher latency, while improving mAP by 4.4 and 5.9 percentage points on the two datasets. Compared with RT-DETRv3-R50, SteelSpec-RTDETR reduces the parameter count from 44.7 M to 35.6 M and the FLOPs from 124.0 G to 59.4 G, while improving mAP by 2.1 percentage points. This corresponds to approximately 20.4% fewer parameters and 52.1% fewer FLOPs than RT-DETRv3-R50. Compared with Deformable DETR, the proposed model is also substantially more efficient, requiring fewer parameters, fewer FLOPs, and lower latency while achieving higher detection accuracy. These results verify that the proposed model does not dominate every baseline on every efficiency measure. Instead, it occupies an intermediate operating point: it exchanges additional resources relative to compact detectors for substantially improved defect accuracy, while remaining considerably less computationally demanding than large Transformer-based alternatives.

Table 2 presents the comparison results on the GC10-DET dataset, which contains more defect categories and more complex industrial surface conditions. SteelSpec-RTDETR again achieves the best performance among all compared methods, with 76.4% mAP, 88.4% mAP₅₀, and 68.1% mAP₇₅. Compared with the strongest CNN-based detector, YOLO11-L, the proposed model improves mAP from 71.4% to 76.4%, yielding a gain of 5.0 percentage points. It also improves mAP₅₀ by 6.7 percentage points and mAP₇₅ by 5.1 percentage points. These improvements demonstrate that SteelSpec-RTDETR is more robust when handling diverse defect appearances, low-contrast regions, and complex surface backgrounds. Among Transformer-based baselines on GC10-DET, RT-DETRv3-R50 obtains the highest mAP of 73.1%. SteelSpec-RTDETR improves this result by 3.3 percentage points, while also increasing mAP₅₀ from 83.2% to 88.4% and mAP₇₅ from 65.6% to 68.1%. This consistent improvement over RT-DETR variants indicates that simply adopting a stronger Transformer detector is insufficient for steel defect detection. Instead, defect-specific feature extraction and defect-aware multi-scale fusion are necessary to capture weak local patterns, fine boundaries, and contextual cues in industrial surface images.

The accuracy-efficiency comparison in Fig. 3 further confirms the effectiveness of the proposed method. On both NEU-DET and GC10-DET, SteelSpec-RTDETR is positioned in the high-accuracy region while maintaining moderate model complexity and inference cost. Compared with large Transformer-based detectors such as RT-DETRv1-R50, RT-DETRv3-R50, and Deformable DETR, the proposed model achieves higher mAP with much lower FLOPs. Compared with CNN-based detectors, SteelSpec-RTDETR provides notably higher detection accuracy while keeping latency at a practical level. With an inference latency of 22.7 ms per image, the proposed model can process approximately 44 frames per second, indicating its potential for real-time industrial inspection scenarios.

4.4 Comparison with Recent Defect-Specific Detectors

In addition to the generic CNN- and Transformer-based baselines reproduced under our unified protocol, Table 3 provides a comparison with three recent detectors developed specifically for industrial surface defects. FC-DETR [43] is an end-to-end Transformer detector based on foreground supervision and cascade-refined hybrid matching. GC-Net [40] combines global attention with cascade feature fusion, while IMPACT-Net [44] targets low-contrast multi-scale defects and computation-efficient embedded inference.

These methods are more task-specific than the generic detectors considered in Tables 1 and 2 and therefore provide useful complementary references for positioning SteelSpec-RTDETR.

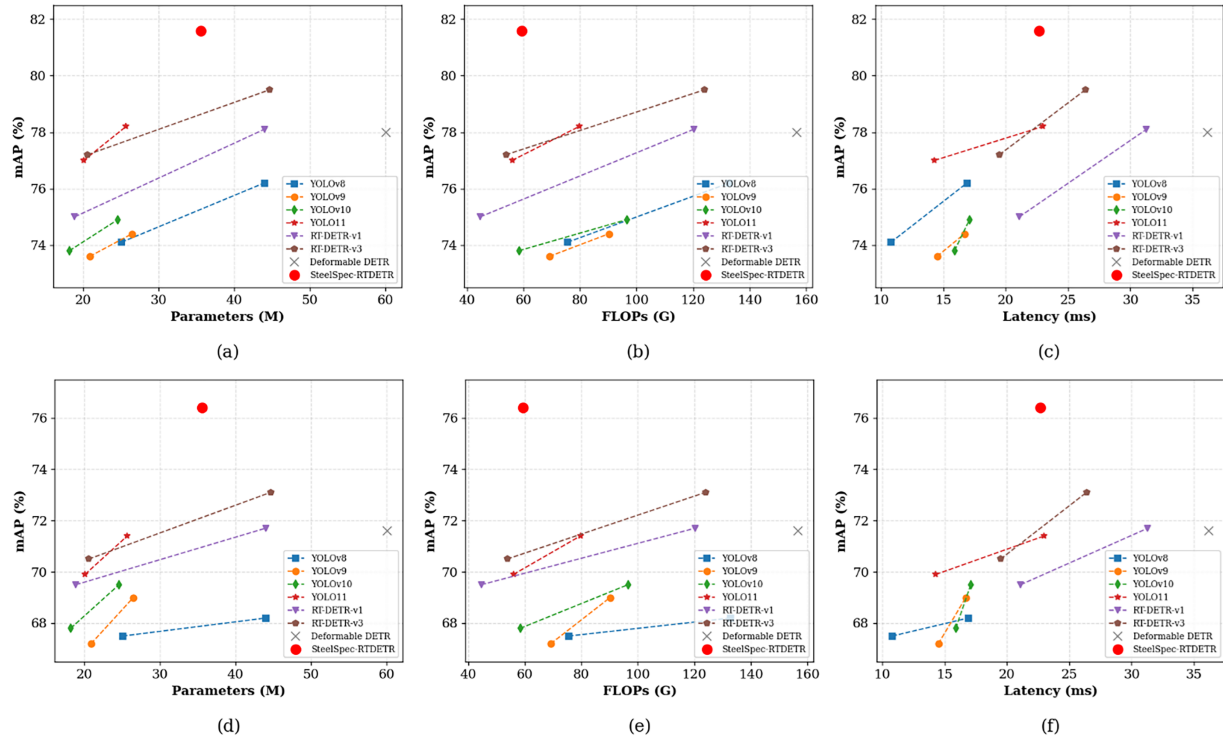


Figure 3: Accuracy-efficiency comparison of proposed model and competing detectors on the NEU-DET and GC10-DET datasets. (a) mAP vs. parameter count on NEU-DET, (b) mAP vs. FLOPs on NEU-DET, (c) mAP vs. latency on NEU-DET, (d) mAP vs. parameter count on GC10-DET, (e) mAP vs. FLOPs on GC10-DET, and (f) mAP vs. latency on GC10-DET.

Table 3: Comparison with recent defect-specific methods.

Method	Detection Framework and Principal Design	NEU-DET mAP ₅₀ (%)	GC10-DET mAP ₅₀ (%)
GC-Net [40]	Global attention, cascade fusion network, and Soft-NMS	77.1	63.5
FC-DETR [43]	End-to-end Transformer, foreground supervision, and cascade-refined hybrid matching	83.7	83.6
IMPACT-Net [44]	Precision-enhanced FPN, adaptive localization loss, and architecture-search-based optimization	78.9	71.4
SteelSpec-RTDETR	Defect-oriented heterogeneous backbone, finest-level-preserving DA-BiFPN, ILDA, and RT-DETR decoding	88.5	88.4

Based on the reported mAP_{50} values, SteelSpec-RTDETR obtains the highest numerical results on both datasets. On NEU-DET, its 88.5% mAP_{50} is 4.8 percentage points higher than FC-DETR's 83.7%, which is the strongest defect-specific result among the three selected methods. On GC10-DET, its 88.4% mAP_{50} is also 4.8 percentage points higher than FC-DETR's 83.6%. Nevertheless, these numerical differences should be interpreted cautiously. The literature methods were trained and evaluated using their respective dataset divisions, preprocessing pipelines, optimization settings, and hardware environments. Therefore, Table 3 provides contextual task-specific positioning rather than a strictly controlled ranking. In contrast, the generic baselines in Tables 1 and 2 were reproduced under the unified protocol described in Section 4.2. A fully controlled comparison with recent defect-specific detectors would require retraining their publicly available implementations using the same data partitions and preprocessing as SteelSpec-RTDETR.

4.5 Statistical Stability and Defect-Characteristic Analysis

4.5.1 Stability across Random Seeds

To assess whether the performance gains are robust to training stochasticity, SteelSpec-RTDETR and the two strongest reproduced comparison models, namely YOLO11-L and RT-DETRv3-R50, were trained independently using five random seeds: 42, 43, 44, 45, and 46. The training, validation, and test partitions were kept fixed across all runs so that the analysis isolated variations arising from parameter initialization, mini-batch ordering, and random data augmentation. For each run, the model checkpoint achieving the highest validation mAP was selected and subsequently evaluated on the fixed test subset. The input resolution, preprocessing pipeline, training schedule, and evaluation procedure remained unchanged. Corresponding runs of different methods used the same seed, and the results are reported as the arithmetic mean $\pm$ sample standard deviation over the five independent runs.

As shown in Table 4, SteelSpec-RTDETR maintains its performance advantage across the five independent runs. On NEU-DET, the proposed model achieves an average mAP of 81.5%, exceeding RT-DETRv3-R50 and YOLO11-L by 2.2 and 3.5 percentage points, respectively. On GC10-DET, it obtains an average mAP of 76.2%, corresponding to improvements of 3.3 percentage points over RT-DETRv3-R50 and 5.0 percentage points over YOLO11-L. Consistent gains are also observed at different localization thresholds. Compared with RT-DETRv3-R50, SteelSpec-RTDETR improves mAP_{50} by 4.7 and 5.2 percentage points and mAP_{75} by 2.8 and 2.5 percentage points on NEU-DET and GC10-DET, respectively. In addition to higher mean accuracy, SteelSpec-RTDETR exhibits relatively low run-to-run variation. Its mAP standard deviations are 0.24 on NEU-DET and 0.29 on GC10-DET, both substantially smaller than the corresponding average improvements over the competing detectors. The proposed model also produces lower standard deviations across all six reported accuracy measures. These results indicate that the observed gains are not attributable to a single favorable initialization and that the proposed defect-oriented architecture provides stable improvements under different training seeds.

Table 4: Stability comparison over five independent runs.

Model	NEU-DET mAP (%)	NEU-DET mAP_{50} (%)	NEU-DET mAP_{75} (%)	GC10-DET mAP (%)	GC10-DET mAP_{50} (%)	GC10-DET mAP_{75} (%)
YOLO11-L	78.0 $\pm$ 0.36	83.3 $\pm$ 0.41	70.6 $\pm$ 0.48	71.2 $\pm$ 0.44	81.5 $\pm$ 0.39	62.8 $\pm$ 0.53
RT-DETRv3-R50	79.3 $\pm$ 0.31	83.6 $\pm$ 0.35	71.2 $\pm$ 0.42	72.9 $\pm$ 0.38	83.0 $\pm$ 0.33	65.4 $\pm$ 0.46
SteelSpec-RTDETR	81.5 $\pm$ 0.24	88.3 $\pm$ 0.28	74.0 $\pm$ 0.31	76.2 $\pm$ 0.29	88.2 $\pm$ 0.26	67.9 $\pm$ 0.35

4.5.2 Performance across Defect Sizes

To examine the sensitivity to small defects more directly, the ground-truth instances in each test subset were grouped according to their bounding-box areas after deterministic resizing and letterbox padding to the 640×640 network input. Following the commonly used object-size definition, instances with an area below 32^2 pixels were categorized as small, those with an area from 32^2 to 96^2 pixels were categorized as medium, and those with an area of at least 96^2 pixels were categorized as large. The corresponding average precision measures are denoted as AP_S , AP_M , and AP_L , respectively. Unless otherwise specified, AP_S , AP_M , and AP_L are computed using the same IoU range of 0.50:0.05:0.95 as the overall mAP metric, but are restricted to the corresponding small-, medium-, and large-defect subsets. Each result in Table 5 represents the average over the same five independent runs used in the stability analysis.

Table 5: Detection performance grouped by defect size.

Dataset	Model	AP_S (%)	AP_M (%)	AP_L (%)
NEU-DET	YOLO11-L	69.0	78.5	81.2
	RT-DETRv3-R50	71.6	79.8	82.4
	SteelSpec-RTDETR	77.1	82.2	84.0
GC10-DET	YOLO11-L	60.8	71.9	75.8
	RT-DETRv3-R50	63.7	73.2	77.3
	SteelSpec-RTDETR	70.4	76.6	79.4

Table 5 shows that SteelSpec-RTDETR improves detection performance across all three defect-size groups, with the largest gains occurring for small defects. On NEU-DET, the proposed model achieves an AP_S of 77.1%, exceeding RT-DETRv3-R50 and YOLO11-L by 5.5 and 8.1 percentage points, respectively. On GC10-DET, its AP_S reaches 70.4%, providing gains of 6.7 percentage points over RT-DETRv3-R50 and 9.6 percentage points over YOLO11-L. The improvements for medium and large defects are positive but smaller. Compared with RT-DETRv3-R50, SteelSpec-RTDETR improves AP_M by 2.4 and 3.4 percentage points and AP_L by 1.6 and 2.1 percentage points on NEU-DET and GC10-DET, respectively. This pattern indicates that the proposed architecture does not improve aggregate accuracy solely through medium or large defect instances. Instead, retaining the finest pyramid level, strengthening local and high-frequency representations, and aggregating complementary features across fusion depths provide particularly strong benefits for small defects whose spatial information is more vulnerable to repeated downsampling. The simultaneous improvements in AP_M and AP_L further show that these design choices do not compromise detection at larger scales.

4.6 Visualization and Discussion

To further evaluate the detection behavior of the proposed SteelSpec-RTDETR, qualitative comparisons and feature response visualizations are provided in Figs. 4 and 5. While the quantitative results indicate a favorable accuracy-efficiency trade-off under the unified evaluation protocol, visual analysis provides additional insight into how the proposed model improves defect localization and suppresses background interference. Fig. 4 presents representative detection results produced by RT-DETR and the proposed SteelSpec-RTDETR. Compared with the baseline RT-DETR, the proposed model shows stronger capability in detecting small, weak, and texture-like steel surface defects. In several challenging cases, defects appear with low contrast, irregular shapes, or partial similarity to surrounding surface textures. These characteristics

often cause missed detections or inaccurate bounding boxes in generic detectors. In contrast, SteelSpec-RTDETR produces more complete and better-aligned bounding boxes around defective regions. This improvement indicates that the proposed defect-oriented backbone can preserve fine local cues such as scratch boundaries, pitted regions, and subtle surface discontinuities. The visual results also show that SteelSpec-RTDETR is more robust to complex background patterns. Steel surface images commonly contain rolling traces, illumination variations, repetitive textures, and non-defective high-frequency patterns. Such background responses may be incorrectly activated as defects, especially when the defect category has weak visual contrast. The proposed model reduces this problem by introducing the Surface Background Smoothing Block and Local-Spectral Defect Mixing blocks. The SBS module suppresses background-dominant responses at shallow stages, while the LSDM blocks enhance discriminative local, spectral-contextual, and high-frequency defect features. As a result, the proposed detector can better distinguish true defects from background noise and generates fewer false responses in non-defective regions.

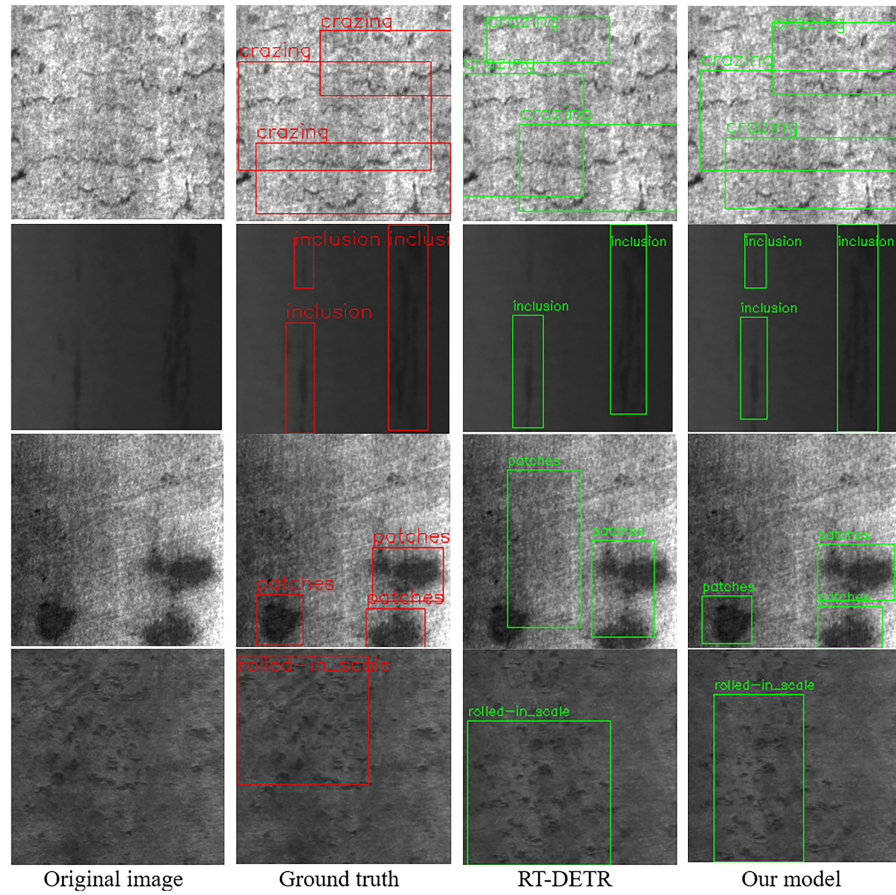


Figure 4: Qualitative comparison of detection results generated by RT-DETR and the proposed SteelSpec-RTDETR. Ground-truth boxes are displayed in red, and predicted boxes are displayed in green.

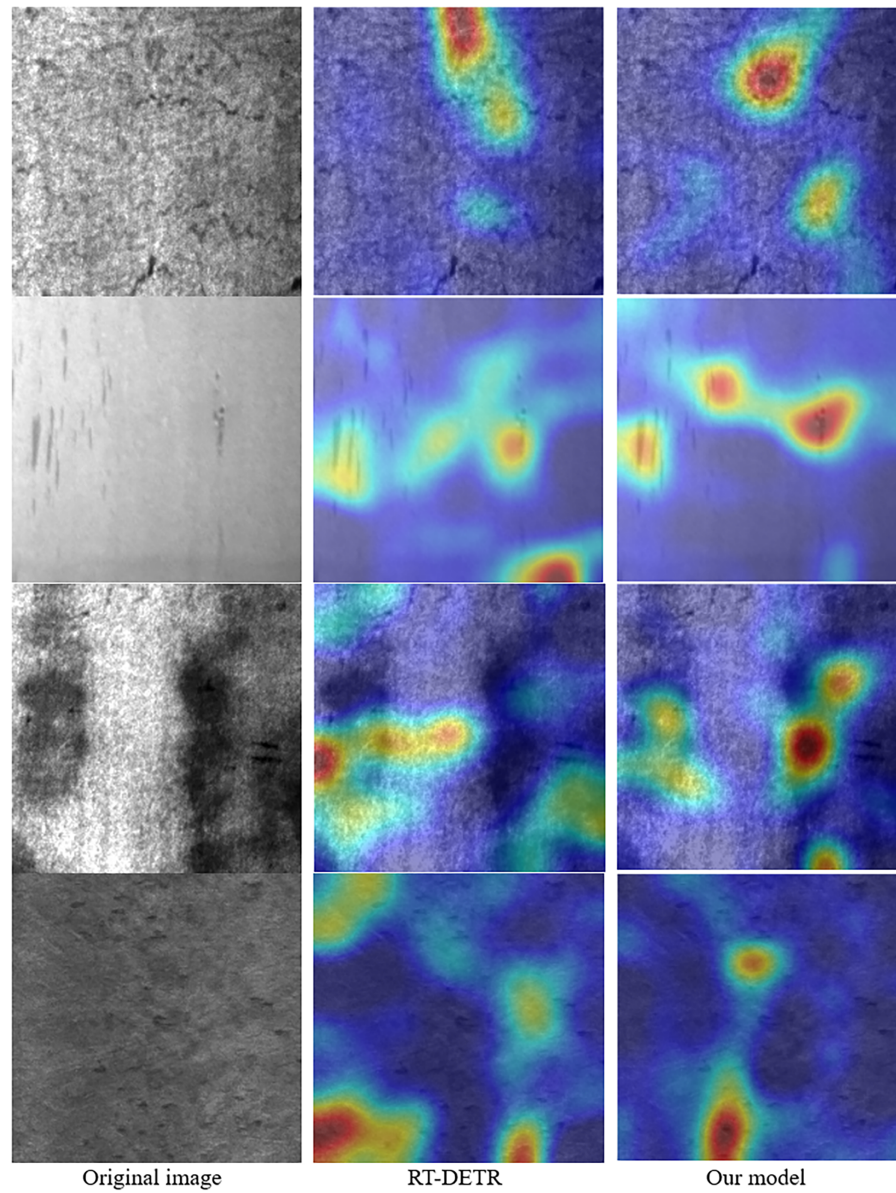


Figure 5: Class-agnostic pre-decoder feature-response maps for RT-DETR and SteelSpec-RTDETR. The visualized tensors are the corresponding stride-8 feature levels immediately before query selection.

Fig. 5 visualizes a common pre-decoder feature level for the two models. For RT-DETR, the visualized tensor is the highest-resolution output of its hybrid encoder at an effective stride of 8. For SteelSpec-RTDETR, the corresponding tensor is the stride-8 ILDA-refined feature N_3 , immediately before its projection to the decoder hidden dimension and UMQS-based query selection. Both tensors therefore have a spatial resolution of 80×80 for the 640×640 network input. The stride-8 level was selected because it is available in both architectures and provides a more controlled comparison than visualizing different pyramid resolutions. The response maps provide an intuitive explanation of why the proposed model achieves higher detection accuracy. For RT-DETR, the activated regions are relatively dispersed and may include irrelevant background textures. This suggests that the baseline model tends to distribute attention across both defective and non-defective surface regions, which weakens its ability to focus on subtle defect patterns. In contrast, the

response maps generated by SteelSpec-RTDETR are more concentrated around actual defect areas. The activated regions better correspond to the spatial locations of scratches, pits, inclusions, and other surface anomalies. This behavior can be attributed to the proposed multi-scale representation design. The defect-aware backbone extracts hierarchical features that preserve fine defect details while introducing compact contextual reasoning at deeper stages. Meanwhile, the DA-BiFPN module propagates semantic information from deep layers to shallow layers and transfers fine spatial details back to high-level representations. The ILDA module further aggregates same-level features from different fusion depths, allowing the model to adaptively select useful local and contextual information. Therefore, the final multi-scale memory used for query selection contains more defect-relevant representations than the original RT-DETR encoder features.

4.7 Comprehensive Ablation and Component Analysis

To systematically examine the contributions of the main architectural components in SteelSpec-RTDETR, the ablation analysis is organized into three complementary parts. First, SBS, LSDM, and ADCA are progressively introduced into a plain four-stage convolutional backbone to quantify their incremental contributions. Second, the feature-fusion experiments compare conventional FPN, bidirectional fusion, persistent finest-level preservation, and ILDA. Third, ADCA is compared with several established attention mechanisms to determine whether its benefit can be achieved using a conventional attention module. In all experiments, the dataset partitions, input resolution, backbone stage dimensions, training schedule, checkpoint-selection criterion, and evaluation procedure are kept unchanged. Only the component under investigation is modified.

4.7.1 Effectiveness of SBS, LSDM, and ADCA

Table 6 presents the ablation results of the proposed defect-oriented backbone components. The baseline model denotes SteelSpec-RTDETR with the proposed defect-oriented backbone replaced by a plain four-stage convolutional backbone. The baseline backbone preserves the same output feature levels, spatial strides, and channel dimensions as the proposed backbone, but uses conventional convolutional blocks instead of SBS, LSDM, and ADCA. Therefore, the contribution of each proposed backbone component can be evaluated by progressively replacing the corresponding plain stage with the proposed module.

Table 6: Ablation study of the proposed defect-oriented backbone components.

Model	SBS	LSDM	ADCA	mAP (%)		mAP ₅₀ (%)		mAP ₇₅ (%)	
				NEU-DET	GC10-DET	NEU-DET	GC10-DET	NEU-DET	GC10-DET
Baseline	×	×	×	73.2	66.7	78.6	75.4	67.0	55.6
Baseline + SBS	✓	×	×	74.8	68.4	80.1	78.6	68.6	56.9
Baseline + SBS + LSDM	✓	✓	×	80.0	74.5	86.4	85.9	72.9	66.2
Full model	✓	✓	✓	81.6	76.4	88.5	88.4	74.2	68.1

The baseline model achieves 73.2% mAP on NEU-DET and 66.7% mAP on GC10-DET. After introducing SBS, the mAP increases to 74.8% and 68.4%, respectively. This improvement indicates that suppressing background-dominant surface responses at the shallow stage is beneficial for steel defect detection, especially when defect regions are weak and easily confused with repetitive surface textures. A more significant improvement is observed after adding LSDM. With SBS and LSDM, the model achieves 80.0% mAP on

NEU-DET and 74.5% mAP on GC10-DET, corresponding to gains of 5.2 and 6.1 percentage points over the SBS-only setting. The mAP_{50} and mAP_{75} values also increase substantially, showing that LSDM improves both defect recognition and localization accuracy. This confirms the effectiveness of combining local spatial modeling, reduced spectral-context attention, and high-frequency enhancement for defect-sensitive representation learning.

Finally, adding ADCA further improves the performance to 81.6% mAP on NEU-DET and 76.4% mAP on GC10-DET. Compared with the plain backbone baseline, the full defect-oriented backbone improves mAP by 8.4 and 9.7 percentage points on the two datasets, respectively. These results demonstrate that SBS, LSDM, and ADCA are complementary: SBS reduces background interference, LSDM enhances mid-level defect features, and ADCA introduces compact contextual reasoning at the high-level stage.

4.7.2 Effectiveness of DA-BiFPN and ILDA

Table 7 reports the ablation results of the proposed multi-scale feature fusion module. In this study, the baseline fusion module is FPN, which performs conventional top-down feature fusion. The FPN baseline achieves 75.8% mAP on NEU-DET and 69.6% mAP on GC10-DET. Although FPN can combine semantic and spatial information to some extent, its one-directional fusion strategy is limited for steel surface defect detection, where both fine local texture and high-level contextual discrimination are important.

Table 7: Ablation study of the proposed multi-scale feature fusion module.

Fusion Setting	Finest Level Retained	DA-BiFPN	ILDA	mAP (%)		mAP ₅₀ (%)		mAP ₇₅ (%)	
				NEU-DET	GC10-DET	NEU-DET	GC10-DET	NEU-DET	GC10-DET
Baseline fusion	×	×	×	75.8	69.6	79.2	78.8	69.6	61.5
DA-BiFPN without finest level	×	✓	×	79.1	73.3	82.9	82.5	71.0	65.3
DA-BiFPN with finest level	✓	✓	×	79.5	73.9	84.1	83.6	71.7	66.2
DA-BiFPN + ILDA	✓	✓	✓	81.6	76.4	88.5	88.4	74.2	68.1

When FPN is replaced with DA-BiFPN without retaining the finest feature level, the mAP increases to 79.1% on NEU-DET and 73.3% on GC10-DET. This improvement demonstrates that bidirectional cross-scale fusion is more effective than conventional FPN because it enables both top-down semantic propagation and bottom-up spatial-detail refinement. The corresponding improvements in mAP_{50} and mAP_{75} further indicate better defect classification and localization. Retaining the finest feature level further improves the mAP to 79.5% on NEU-DET and 73.9% on GC10-DET. Although the gain is moderate, it is consistent across both datasets and all accuracy metrics. This confirms that the finest feature level contains useful shallow texture and boundary information, which is important for detecting small, weak, and low-contrast steel defects.

After introducing ILDA, the full fusion module achieves 81.6% mAP on NEU-DET and 76.4% mAP on GC10-DET. Compared with the FPN baseline, the proposed DA-BiFPN + ILDA improves mAP by 5.8 and 6.8 percentage points on the two datasets, respectively. These results show that ILDA effectively exploits complementary same-level features from different fusion depths. Overall, the proposed multi-scale fusion design provides stronger cross-scale interaction and more discriminative defect-aware representations than the conventional FPN baseline.

4.7.3 Comparison of ADCA with Alternative Attention Mechanisms

To determine whether the gain from ADCA can be obtained using a conventional attention module, we replaced ADCA at the highest backbone stage with several alternatives while keeping all other parts of the model unchanged. The comparison includes a convolution-only block without attention, squeeze-and-excitation attention, CBAM, standard multi-head self-attention, and the original Agent Attention operator. The complete ADCA configuration combines Agent Attention with convolutional feed-forward refinement at the spatially compact highest stage. Parameter counts and FLOPs are reported because attention alternatives may improve accuracy by introducing substantially different computational costs.

As shown in Table 8, conventional channel and spatial attention provide moderate improvements over the convolution-only configuration, but they remain less effective than contextual token interaction. Standard multi-head self-attention achieves higher accuracy than SE and CBAM but also introduces the highest computational cost among the evaluated alternatives. The original Agent Attention operator obtains 81.1% mAP on NEU-DET and 75.8% on GC10-DET, showing that efficient global contextual reasoning is beneficial at the highest stage. The complete ADCA configuration further improves these results by 0.5 and 0.6 percentage points. This additional gain indicates that Agent Attention alone does not account for the complete improvement. The following convolutional refinement and stage-specific configuration provide complementary local inductive bias. Relative to the convolution-only setting, ADCA improves mAP by 1.6 percentage points on NEU-DET and 1.9 percentage points on GC10-DET. Compared with CBAM, it improves mAP by 1.1 and 1.4 percentage points, respectively. ADCA also achieves higher accuracy than standard self-attention while requiring fewer FLOPs. These results support the use of compact Agent Attention followed by local convolutional refinement at the spatially reduced highest stage.

Table 8: Comparison of high-level attention configurations.

Highest-Stage Configuration	NEU-DET mAP (%)	GC10-DET mAP (%)	Params (M)	FLOPs (G)
Convolutional refinement without attention	80	74.5	34.8	58.2
SE attention	80.3	74.8	34.9	58.2
CBAM	80.5	75	35	58.4
Multi-head self-attention	80.7	75.3	36.5	62.1
Original Agent Attention	81.1	75.8	35.2	58.8
Proposed ADCA	81.6	76.4	35.6	59.4

4.8 Limitations and Generalization

Although SteelSpec-RTDETR achieves consistent improvements on NEU-DET and GC10-DET, the present evaluation has several limitations. NEU-DET contains 1800 images, whereas GC10-DET contains 3570 images, and both benchmarks consist primarily of grayscale metallic-surface images acquired under relatively constrained industrial imaging conditions. Consequently, training and test images within the same benchmark may share similar sensor characteristics, surface textures, illumination patterns, and production-related artifacts. The reported results should therefore be interpreted primarily as evidence of in-domain generalization rather than broad cross-domain industrial robustness. The proposed model contains a Transformer decoder and multiple adaptive feature-processing modules with sufficient representation capacity to learn dataset-specific appearance patterns. Although the adopted data augmentation and weight-decay regularization reduce this risk, overfitting cannot be completely excluded because of the limited

number and diversity of annotated samples. Moreover, the present experiments train and evaluate the model independently on NEU-DET and GC10-DET. They do not examine a stricter cross-dataset setting in which the model is trained on one dataset and directly evaluated on another without fine-tuning. Therefore, the current results do not establish robustness to unseen production lines, factories, sensors, steel grades, acquisition resolutions, illumination configurations, or defect taxonomies. Performance may decrease on larger and more heterogeneous industrial datasets containing severe class imbalance, rare or previously unseen defects, stronger background clutter, annotation noise, greater variations in defect scale and morphology, or substantial domain shifts. In particular, the Surface Background Smoothing, Local-Spectral Defect Mixing, and defect-aware fusion modules are designed according to the texture characteristics of steel surfaces. Their inductive biases may not transfer directly to substantially different materials, such as fabrics, ceramics, composite structures, or semiconductor wafers. In addition, retaining the finest pyramid level improves small defect representation but may increase memory consumption and latency when processing substantially higher-resolution industrial images. A more comprehensive assessment will require multi-site datasets and external validation protocols covering different production lines and acquisition systems. Future investigations should include cross-dataset transfer experiments, leave-one-domain-out evaluation, domain generalization and adaptation, semi-supervised or self-supervised pretraining, and uncertainty calibration under distribution shifts. These evaluations are necessary before broad industrial generalization or deployment robustness can be conclusively claimed.

5 Conclusions and Future Work

This paper proposed SteelSpec-RTDETR, an accuracy-efficiency-balanced multi-scale Transformer detector for steel surface defect inspection. Built on the end-to-end RT-DETR paradigm, the proposed framework redesigns feature extraction and multi-scale fusion to better represent small, low-contrast, irregular, and texture-like defects. The defect-oriented backbone integrates Surface Background Smoothing, Local-Spectral Defect Mixing, and Agent-based Defect Context Attention to suppress background interference while preserving local defect cues and contextual information. In addition, DA-BiFPN retains the finest feature level during bidirectional fusion, while ILDA aggregates complementary same-level representations from different fusion depths. The resulting multi-scale features are directly used for uncertainty-minimal query selection and RT-DETR decoding without non-maximum suppression. Experiments on NEU-DET and GC10-DET demonstrate a favorable balance between detection accuracy and computational efficiency among the evaluated CNN-based and Transformer-based detectors. Future work will focus on both generalization and practical deployment. Cross-domain robustness will be investigated through semi-supervised or self-supervised learning and domain adaptation across different production lines, imaging conditions, and steel materials. For edge deployment, hardware-aware optimization will be explored through FP16/INT8 quantization, structured pruning, and TensorRT-based acceleration. The optimized models will be evaluated on edge-GPU platforms at batch size one using end-to-end latency, throughput, peak memory consumption, and power consumption, including preprocessing and prediction decoding.

Acknowledgement: Not applicable.

Funding Statement: The authors received no specific funding for this study.

Author Contributions: Mien Ka Duong conducted the literature review, methodology, and experiments, analyzed the findings, and drafted the manuscript. Hoanh Nguyen contributed to modeling, interpretation, supervised the work, refined the review scope, and provided substantive revisions. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: The datasets used in this study, NEU-DET and GC10-DET, are publicly available benchmark datasets from the original sources cited in Refs. [41,42], respectively. No proprietary or privately collected data were used. The training, validation, and test partitions used in this study were generated from these publicly available datasets as described in [Section 4.1](#).

Ethics Approval: Not applicable.

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

References

1. Carion N, Massa F, Synnaeve G, Usunier N, Kirillov A, Zagoruyko S. End-to-end object detection with transformers. In: Proceedings of the Computer Vision–ECCV 2020; 2020 Aug 23–28; Glasgow, UK. doi:10.1007/978-3-030-58452-8_13.
2. Zhu X, Su W, Lu L, Li B, Wang X, Dai J. Deformable DETR: deformable transformers for end-to-end object detection. In: Proceedings of the ICLR 2021-9th International Conference on Learning Representations; 2021 May 3–7; Virtual.
3. Zhao Y, Lv W, Xu S, Wei J, Wang G, Dang Q, et al. DETRs beat YOLOs on real-time object detection. In: Proceedings of the 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2024 Jun 16–22; Seattle, WA, USA. doi:10.1109/cvpr52733.2024.01605.
4. Girshick R, Donahue J, Darrell T, Malik J. Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition; 2014 Jun 23–28; Columbus, OH, USA. doi:10.1109/cvpr.2014.81.
5. Girshick R. Fast R-CNN. In: Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV); 2015 Dec 7–13; Santiago, Chile. doi:10.1109/iccv.2015.169.
6. Ren S, He K, Girshick R, Sun J. Faster R-CNN: towards real-time object detection with region proposal networks. *IEEE Trans Pattern Anal Mach Intell.* 2017;39(6):1137–49. doi:10.1109/tpami.2016.2577031.
7. Lin TY, Dollar P, Girshick R, He K, Hariharan B, Belongie S. Feature pyramid networks for object detection. In: Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2017 Jul 21–26; Honolulu, HI, USA. doi:10.1109/cvpr.2017.106.
8. Guo C, Fan B, Zhang Q, Xiang S, Pan C. AugFPN: improving multi-scale feature learning for object detection. In: Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2020 Jun 13–19; Seattle, WA, USA. doi:10.1109/cvpr42600.2020.01261.
9. Ke J, He L, Han B, Li J, Gao X. ProFPN: progressive feature pyramid network with soft proposal assignment for object detection. *Knowl Based Syst.* 2024;299(6):112078. doi:10.1016/j.knosys.2024.112078.
10. Tan M, Pang R, Le QV. EfficientDet: scalable and efficient object detection. In: Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2020 Jun 13–19; Seattle, WA, USA. doi:10.1109/cvpr42600.2020.01079.
11. Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu CY, et al. SSD: single shot MultiBox detector. In: Proceedings of the Computer Vision–ECCV 2016; 2016 Oct 11–14; Amsterdam, The Netherlands. doi:10.1007/978-3-319-46448-0_2.
12. Lin TY, Goyal P, Girshick R, He K, Dollar P. Focal loss for dense object detection. *IEEE Trans Pattern Anal Mach Intell.* 2020;42(2):318–27. doi:10.1109/tpami.2018.2858826.
13. Wang CY, Bochkovskiy A, Liao HM. YOLOv7: trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In: Proceedings of the 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2023 Jun 17–24; Vancouver, BC, Canada. doi:10.1109/cvpr52729.2023.00721.
14. Varghese R, Sambath M. YOLOv8: a novel object detection algorithm with enhanced performance and robustness. In: Proceedings of the 2024 International Conference on Advances in Data Engineering and Intelligent Computing Systems (ADICS); 2024 Apr 18–19; Chennai, India. doi:10.1109/adics58448.2024.10533619.

15. Wang CY, Yeh IH, Mark Liao HY. YOLOv9: learning what you want to learn using programmable gradient information. In: Proceedings of the Computer Vision–ECCV 2024; 2024 Sep 29–Oct 4; Milan, Italy. doi:10.1007/978-3-031-72751-1_1.
16. Wang A, Chen H, Liu L, Chen K, Lin Z, Han J, et al. YOLOv10: real-time end-to-end object detection. In: Proceedings of the Neural Information Processing Systems 37; 2024 Dec 10–15; Vancouver, BC, Canada. doi:10.52202/079017-3429.
17. Khanam R, Hussain M. YOLOv11: an overview of the key architectural enhancements. arXiv:2410.17725. 2024.
18. Meng D, Chen X, Fan Z, Zeng G, Li H, Yuan Y, et al. Conditional DETR for fast training convergence. In: Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision (ICCV); 2021 Oct 10–17; Montreal, QC, Canada. doi:10.1109/iccv48922.2021.00363.
19. Liu S, Li F, Zhang H, Yang X, Qi X, Su H, et al. DAB-DETR: dynamic anchor boxes are better queries for DETR. In: Proceedings of the ICLR 2022–10th International Conference on Learning Representations; 2022 Apr 25–29; Online.
20. Li F, Zhang H, Liu S, Guo J, Ni LM, Zhang L. DN-DETR: accelerate DETR training by introducing query DeNoising. IEEE Trans Pattern Anal Mach Intell. 2024;46(4):2239–51. doi:10.1109/TPAMI.2023.3335410.
21. Zhang H, Li F, Liu S, Zhang L, Su H, Zhu J, et al. DINO: DETR with improved denoising anchor boxes for end-to-end object detection. In: Proceedings of the 11th International Conference on Learning Representations (ICLR); 2023 May 1–5; Kigali, Rwanda.
22. Lv W, Zhao Y, Chang Q, Huang K, Wang G, Liu Y. RT-DETRv2: improved baseline with bag-of-freebies for real-time detection transformer. arXiv:2407.17140. 2024.
23. Wang S, Xia C, Lv F, Shi Y. RT-DETRv3: real-time end-to-end object detection with hierarchical dense positive supervision. In: Proceedings of the 2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV); 2025 Feb 26–Mar 6; Tucson, AZ, USA. doi:10.1109/wacv61041.2025.00166.
24. Ma Y, Zhang Z. Position-guided hybrid convolutional neural network and transformer network for steel strip surface defect detection. Eng Appl Artif Intell. 2025;162(4):112741. doi:10.1016/j.engappai.2025.112741.
25. Chao C, Mu X, Guo Z, Sun Y, Tian X, Yong F. IAMF-YOLO: metal surface defect detection based on improved YOLOv8. IEEE Trans Instrum Meas. 2025;74(4):1–17. doi:10.1109/tim.2025.3548198.
26. Zhang M, Hu Y, Luo L, Xu B, Wang S, Wang Y. A lightweight metal surface defect detection network with hierarchical multi-branch feature extraction and group decision attention. Comput Ind. 2025;173(4):104390. doi:10.1016/j.compind.2025.104390.
27. Ma S, Li R, Hu H. Rail surface defect detection algorithm based on CSSI-YOLO. J Real Time Image Process. 2026;23(2):76. doi:10.1007/s11554-026-01876-2.
28. Du Y, Chen H, Fu Y, Zhu J, Zeng H. AFF-net: a strip steel surface defect detection network via adaptive focusing features. IEEE Trans Instrum Meas. 2024;73:1–14. doi:10.1109/tim.2024.3398131.
29. Chen CJ, Liu RS. A multi-scale attention network for steel surface defect recognition. Soft Comput. 2026;30(4):2843–59. doi:10.1007/s00500-025-11053-5.
30. Bai D, Yun J, Jiang D, Zhou D, Gao D, Li G, et al. HVA-YOLO: a real-time instance segmentation method under hybrid visual attention applied to steel plate surface defect detection. Mech Syst Signal Process. 2025;238(12):113191. doi:10.1016/j.ymssp.2025.113191.
31. Liu S, Liu C, Li J. LGP-DETR: an end-to-end object detection model for surface defect detection in light guide plates. Eng Appl Artif Intell. 2025;162(8):112416. doi:10.1016/j.engappai.2025.112416.
32. Gao B, Tong J, Fu R, Zhang Z, Yuan Y. CBH-YOLO: a steel surface defect detection algorithm based on cross-stage Mamba enhancement and hierarchical semantic graph fusion. Neurocomputing. 2025;656(7):131467. doi:10.1016/j.neucom.2025.131467.
33. Yan H, Zhang Z, Peng T, Hu X, Zhang J, Qi S. Enhancing feature interaction for improved generalization in few-shot metal surface defect segmentation. Knowl Based Syst. 2025;330(7):114606. doi:10.1016/j.knosys.2025.114606.
34. Liu X, Zhang Y, Wang Q, Li Z, Ke Xu C. Attention-guided few-shot learning for metal surface defect classification. Multimed Syst. 2025;31(6):452. doi:10.1007/s00530-025-02030-x.

35. Liu G, Chu M, Gong R. Semi-SSDDet: an adaptive semi-supervised steel surface defect detection. *Signal Image Video Process.* 2025;19(10):798. doi:10.1007/s11760-025-04416-w.
36. Wang D, Su S, Lu X. Progressive feature injection and global–local discrimination-based image generation network for improving rail surface defect inspection. *Pattern Recognit Lett.* 2025;197(2):250–6. doi:10.1016/j.patrec.2025.08.004.
37. Cao Y, He S, Wang F, Su S, Sun Y. A large-model-enhanced method for rail surface defect detection in heavy-haul railway. *IEEE Trans Intell Transp Syst.* 2025;26(12):23328–41. doi:10.1109/tits.2025.3611411.
38. Lee M, Choi J. ID-RAG: industrial defect retrieval-augmented generation for industrial surface defect detection. *Mach Vis Appl.* 2026;37(2):36. doi:10.1007/s00138-026-01797-x.
39. Han D, Ye T, Han Y, Xia Z, Pan S, Wan P, et al. Agent attention: on the integration of softmax and linear attention. In: *Proceedings of the Computer Vision–ECCV 2024; 2024 Sep 29–Oct 4; Milan, Italy.* doi:10.1007/978-3-031-72973-7_8.
40. Liu G, Chu M, Gong R, Zheng Z. Global attention module and cascade fusion network for steel surface defect detection. *Pattern Recognit.* 2025;158(1):110979. doi:10.1016/j.patcog.2024.110979.
41. He Y, Song K, Meng Q, Yan Y. An end-to-end steel surface defect detection approach via fusing multiple hierarchical features. *IEEE Trans Instrum Meas.* 2020;69(4):1493–504. doi:10.1109/tim.2019.2915404.
42. Lv X, Duan F, Jiang JJ, Fu X, Gan L. Deep metallic surface defect detection: the new benchmark and detection network. *Sensors.* 2020;20(6):1562. doi:10.3390/s20061562.
43. Xia Z, Zhao Y, Gu J, Wang W, Zhang W, Huang Z. FC-DETR: high-precision end-to-end surface defect detector based on foreground supervision and cascade refined hybrid matching. *Expert Syst Appl.* 2025;266(2):126142. doi:10.1016/j.eswa.2024.126142.
44. Wu R, Zhang Y, Lan R, Zhou L. IMPACT-Net: an integrated multi-scale and computation-efficient timely network for surface defect detection in industrial embedded systems. *Expert Syst Appl.* 2026;298(9):129867. doi:10.1016/j.eswa.2025.129867.