

Article

LDM-YOLO11: A Lightweight Steel Surface Defect Detection Method Based on LCAF-D and Hierarchical Mish Activation

Xinwei Wang, Jing Zhao *, Rui Zheng and Feng Wang

School of Electronic Information, Xijing University, Xi'an 710123, China; wangxinwei5974@163.com (X.W.)

* Correspondence: zhaojing.83@163.com

Abstract

Lightweight detectors for steel surface defects still struggle to balance robust feature representation with deployment efficiency when defects exhibit weak textures, fine details, and large variation in scale and shape. To narrow this gap, this study develops an incremental lightweight enhancement of YOLO11n rather than proposing a brand-new detection framework. The method combines two coordinated changes. First, an improved lightweight channel-aware fusion module with a dilated branch, termed LCAF-D, is inserted into the neck to strengthen context-aware multi-scale aggregation at low additional cost. Second, a hierarchical Mish activation strategy is introduced only in selected deep backbone layers and neck downsampling layers so that nonlinear modeling is strengthened without disturbing shallow low-level feature extraction. Experiments on NEU-DET indicate that the proposed design provides competitive lightweight detection performance. The benefit becomes more evident as the input resolution increases. Under the 640×640 setting, YOLO11n + LCAF-D + Mish reaches 77.1% mAP@0.5 and 44.3% mAP@0.5:0.95, exceeding the YOLO11n baseline. Five-seed repeated experiments further show slightly better mean accuracy with stable variation, and runtime benchmarking shows that the method keeps lightweight characteristics with only modest increases in parameters, GFLOPs, latency, and GPU memory. Additional evaluation on GC10-DET under the same 640×640 protocol also gives an overall improvement over the baseline, although the gains remain category-dependent rather than universal. Overall, the method is best understood as a competitive lightweight engineering refinement for steel surface defect detection, particularly when the input resolution is sufficient to preserve subtle defect details.

Keywords: steel surface defect detection; YOLO11n; lightweight detection; LCAF-D; hierarchical Mish activation; multi-scale feature fusion; industrial inspection



Academic Editor: Zhineng Chen

Received: 16 May 2026

Revised: 28 June 2026

Accepted: 16 July 2026

Published: 21 July 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

Artificial intelligence, industrial digitalization, and intelligent manufacturing have accelerated the adoption of vision-based quality inspection in modern production systems. Among the industrial materials, steel is used extensively in machinery, construction, transportation, and energy equipment, so its surface quality has a direct influence on downstream reliability and service safety. For this reason, accurate, efficient, and deployable steel surface defect detection has clear engineering importance in practical manufacturing environments.

During steel production, defects such as crazing, inclusion, patches, pitted surfaces, rolled-in scale, and scratches can arise from rolling, cooling, transportation, corrosion, extrusion, and other process fluctuations. These defects reduce appearance quality and

may also damage strength, corrosion resistance, and long-term service life. In severe cases, they can introduce downstream safety risks. As a result, fast and reliable automatic inspection is a critical component of steel quality control and an important research topic in industrial vision.

Earlier steel surface inspection systems mainly depended on human inspection, hand-crafted rules, and conventional machine-vision pipelines. Typical approaches relied on grayscale statistics, edge operators, texture descriptors, and morphological features, followed by classifiers such as support vector machines or decision trees. Although such methods can work in constrained environments, their robustness is often limited by illumination changes, background clutter, defect-scale variation, and complex textures. Manual inspection also suffers from low efficiency and unstable consistency, which makes it difficult to satisfy the real-time and automated requirements of modern production lines.

Deep learning has substantially improved industrial defect detection because convolutional detectors can learn hierarchical visual representations directly from data. Compared with traditional methods, these models adapt better to background complexity, scale variation, and diverse defect morphologies. Prior work in this area can be broadly grouped into conventional machine vision, CNN-based industrial detectors, YOLO-based lightweight detectors, attention- or fusion-enhanced models, activation-optimization strategies, and recent methods targeting tiny or fine-grained defects. Within these categories, YOLO-style detectors remain especially attractive because they offer an end-to-end pipeline with a favorable balance between speed and accuracy.

Many recent studies on steel and metal surface inspection have therefore focused on improving YOLO-like frameworks through lightweight backbone redesign, cross-scale fusion, attention enhancement, and defect-oriented refinement. The aim of the present work, however, is not to claim a fundamentally different detector architecture. Instead, it is to refine YOLO11n in a lightweight and targeted way for steel inspection. Under this positioning, the most relevant literature is the body of work that addresses multi-scale defects, weak-texture patterns, elongated targets with extreme aspect ratios, and context-aware recognition of small defects under limited computational budgets.

From a broader methodological viewpoint, the literature can be organized into several lines. Conventional and early CNN-based studies established benchmark settings for metallic defect detection and promoted the use of public datasets such as NEU-DET [1,2]. Subsequent studies moved toward lightweight YOLO optimization, including neck refinement, compact backbones, defect-specific enhancement modules, and deployment-aware design choices [3–10]. In parallel, the real-time detection community introduced strong efficiency-oriented baselines such as RT-DETR and YOLOv10, which are useful references when discussing the accuracy-efficiency trade-off beyond only YOLO internal variants [11,12].

Feature fusion and attention design form another closely related research direction. Lightweight modules such as CBAM and ECA-Net show that additional channel or spatial discrimination can be introduced with limited computational overhead [13,14]. Likewise, FPN and PANet demonstrate the importance of cross-scale aggregation and bidirectional information propagation when objects vary strongly in scale [15,16]. These observations support strengthening the neck fusion stage instead of depending only on shallow local enhancement. Nevertheless, many generic attention or fusion modules are not tailored to weak-texture, elongated, or fine-grained steel defects, where contextual perception, detail retention, and lightweight deployment must be balanced carefully.

Recent defect-detection studies have further emphasized sensitivity to small objects, context-aware modeling, and deployment-oriented evaluation. Representative examples include Faster Metallic Surface Defect Detection Using Deep Learning with Channel Shuffling,

HyperDefect-YOLO, GCM-DET, DANet, and several edge-oriented or attention-enhanced industrial detectors [17–24]. Taken together, these studies suggest that progress in industrial defect detection increasingly depends on three factors: effective multi-scale fusion, suppression of complex background interference, and preservation of inference efficiency on practical hardware. Accordingly, the present study is positioned as a lightweight engineering refinement of YOLO11n for steel surface defect detection rather than as a fundamentally new framework.

In summary, prior studies confirm that feature enhancement, improved fusion design, and lightweight modeling can all contribute to better industrial defect detection. Even so, several common weaknesses remain. Some methods gain accuracy at the cost of noticeably higher model complexity, which weakens their deployment advantage. Others emphasize overall mAP but provide limited insight into difficult categories such as tiny defects, weak-texture defects, or elongated defects with extreme aspect ratios. In addition, context-aware modeling for small targets is not always sufficiently addressed, which can reduce robustness when texture distributions change across datasets.

Motivated by these observations, this work proposes LDM-YOLO11, a steel surface defect detector built on lightweight channel-aware fusion and hierarchical Mish activation optimization. Starting from YOLO11n, the method first introduces LCAF and verifies through placement ablation that the neck is the more effective insertion position. It then upgrades LCAF to LCAF-D by adding a dilated depthwise separable branch to better handle fine cracks and complex textures. Finally, it applies a hierarchical Mish strategy only to selected deep layers. The primary contribution of the study therefore lies in a systematic lightweight optimization chain with validated module placement and activation design choices.

2. Network Architecture of YOLO11

YOLO11 is a new-generation one-stage object detection model released by Ultralytics. While inheriting the high-efficiency advantages of the YOLO series, it further improves feature extraction capability, parameter utilization efficiency, and multi-task adaptability. According to the official documentation, YOLO11 supports not only object detection, but also instance segmentation, image classification, pose estimation, and oriented object detection, demonstrating strong generality and extensibility [25,26].

From an overall structural perspective, as illustrated in Figure 1, YOLO11 still follows the typical three-part Backbone-Neck-Head framework. The backbone extracts hierarchical features from the input image; the neck fuses features at different scales; and the detection head outputs category predictions and bounding-box regression results based on the fused feature maps. This structure continues the end-to-end detection characteristic of the YOLO family and also provides a clear structural foundation for subsequent model optimization [25].

In the backbone, YOLO11 focuses on enhancing feature extraction. Compared with previous versions, the YOLO11 backbone is optimized to obtain stronger hierarchical feature representations. Official comparison materials show that YOLO11 replaces part of the traditional C2f modules with the more advanced C3k2 modules and introduces the C2PSA module to strengthen spatial feature modeling and attention to key regions. These modifications enable the model to represent targets more effectively in complex scenarios [27].

In the neck, YOLO11 further strengthens the multi-scale feature fusion mechanism. In object detection, shallow features contain rich edge and texture details, whereas deep features contain stronger semantic information. The purpose of the neck is to effectively integrate these features from different levels so as to balance small-object details and large-

object semantics. Official documents indicate that YOLO11 jointly optimizes both the backbone and the neck to improve detection ability in complex scenes [25].

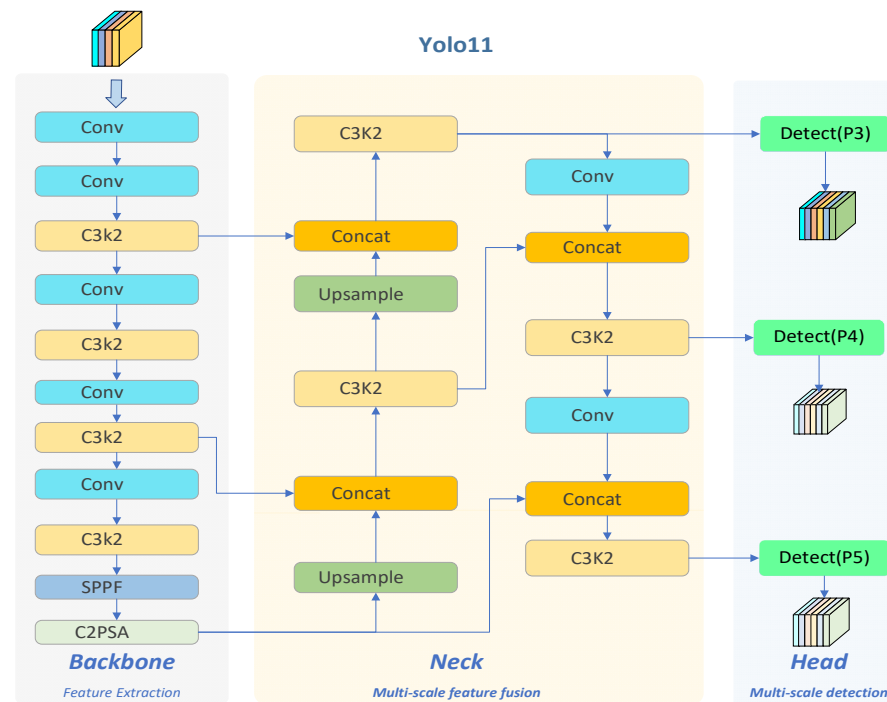


Figure 1. Architecture of YOLO11.

In the detection head, YOLO11 maintains the efficient prediction characteristics of one-stage detectors. The head uses the fused feature maps to complete object classification and bounding-box regression; therefore, its performance strongly depends on the feature quality provided by the backbone and neck. Compared with earlier YOLO models, the advantage of YOLO11 is not confined to any single component, but lies in the more coordinated overall structure formed by the backbone, neck, and head [25].

Compared with previous Ultralytics YOLO versions, the improvements of YOLO11 are mainly reflected in the following aspects. First, its feature extraction and feature fusion capabilities are stronger, enabling it to better adapt to complex backgrounds and multi-scale targets. Second, YOLO11 pays more attention to parameter utilization efficiency, improving detection performance per parameter through structural optimization. Third, YOLO11 achieves a better balance between accuracy and efficiency. Official comparisons show that under a 640-pixel input size, YOLO11n attains a COCO mAP@50-95 of 39.5, higher than 37.3 for YOLOv8n; YOLO11s reaches 47.0, higher than 44.9 for YOLOv8s; YOLO11m reaches 51.5, higher than 50.2 for YOLOv8m; YOLO11l reaches 53.4, higher than 52.9 for YOLOv8l; and YOLO11x reaches 54.7, higher than 53.9 for YOLOv8x [27].

From the perspective of lightweight models, YOLO11n achieves relatively high detection accuracy while maintaining low parameter count and computational cost. Official data indicate that YOLO11n has about 2.6M parameters and approximately 6.5B FLOPs, while YOLOv8n has about 3.2M parameters and 8.7B FLOPs. This suggests that YOLO11n can achieve higher accuracy with lower complexity, which is highly valuable for industrial vision tasks that must balance detection performance and deployment efficiency [27].

In addition, YOLO11 also shows certain advantages in deployment performance. Official documents indicate that it provides good inference speed in multiple runtime environments and is especially suitable for edge devices and real-time detection scenarios. Therefore, in terms of structural design, performance, and deployability, YOLO11 can serve as a solid base model for subsequent industrial defect detection research.

3. LDM-YOLO11 Object Detection Network

Ultralytics YOLO11 provides multiple model variants for object detection, including YOLO11n, YOLO11s, YOLO11m, YOLO11l, and YOLO11x. These models expand from lightweight to high performance and form different balances among parameter count, computational complexity, and detection accuracy, making them suitable for different computational conditions and application scenarios [25,26].

For steel surface defect detection, a model must not only achieve high accuracy, but also satisfy the requirements of real-time processing and deployment efficiency in industrial scenarios. Considering model complexity, inference cost, and lightweight deployment requirements, the proposed method selects YOLO11n as the base model. On the one hand, YOLO11n is the lightest detection variant in the YOLO11 series, with low parameter count and computational cost, making it more suitable for edge deployment in industrial scenarios. On the other hand, YOLO11n already exhibits a good balance between accuracy and efficiency, providing a solid basis for targeted improvements in steel surface defect detection [25,27].

On this basis, to address insufficient multi-scale feature fusion, inadequate representation of fine crack defects, and insufficient nonlinear representation of deep features in steel surface defect detection, the proposed method constructs the improved LDM-YOLO11 detection network, as illustrated in Figure 2. The network mainly includes three improvements: first, a lightweight channel-aware fusion module (LCAF) is introduced in the neck feature-fusion stage to enhance multi-scale feature interaction; second, an improved LCAF-D module is built upon the original LCAF to improve the representation of fine-grained textures and complex backgrounds; finally, a hierarchical Mish activation optimization strategy is introduced to strengthen the nonlinear modeling ability of the deep network for complex defect patterns. Through these improvements, LDM-YOLO11 further enhances steel surface defect detection while preserving lightweight characteristics.

3.1. Lightweight Channel-Aware Fusion Module (LCAF)

In steel surface defect detection, different defect categories show significant differences in scale, morphology, and texture. Fine crack defects are usually small and weak in texture, whereas patch-like and indentation-like defects may cover larger areas and appear with complex background features. In such cases, insufficient multi-scale feature fusion often leads to missed detection of small defects, false detections in complex backgrounds, and category confusion. To address this issue, the proposed method first introduces the lightweight channel-aware fusion module LCAF into the neck of YOLO11n, as shown in Figure 3.

The core idea of LCAF is to strengthen feature fusion under low computational overhead. Specifically, the module first uses a 1×1 convolution for channel alignment and transformation, then adopts parallel branches to extract local texture information and broader contextual information, and combines a lightweight channel-attention mechanism to reinforce key feature responses. Finally, residual connections are used to fuse the enhanced features with the input features. In this way, LCAF improves the representation of complex defect regions while maintaining lightweight characteristics.

From a structural perspective, LCAF has three main advantages. First, its dual-branch design jointly considers local texture and contextual semantic information. Second, the channel-attention mechanism helps highlight defect-related features and suppress background redundancy. Third, the entire module adopts a lightweight design that is more suitable for deployment in real-time industrial inspection frameworks. Therefore, LCAF provides effective support for subsequent multi-scale feature enhancement in steel surface defect detection.

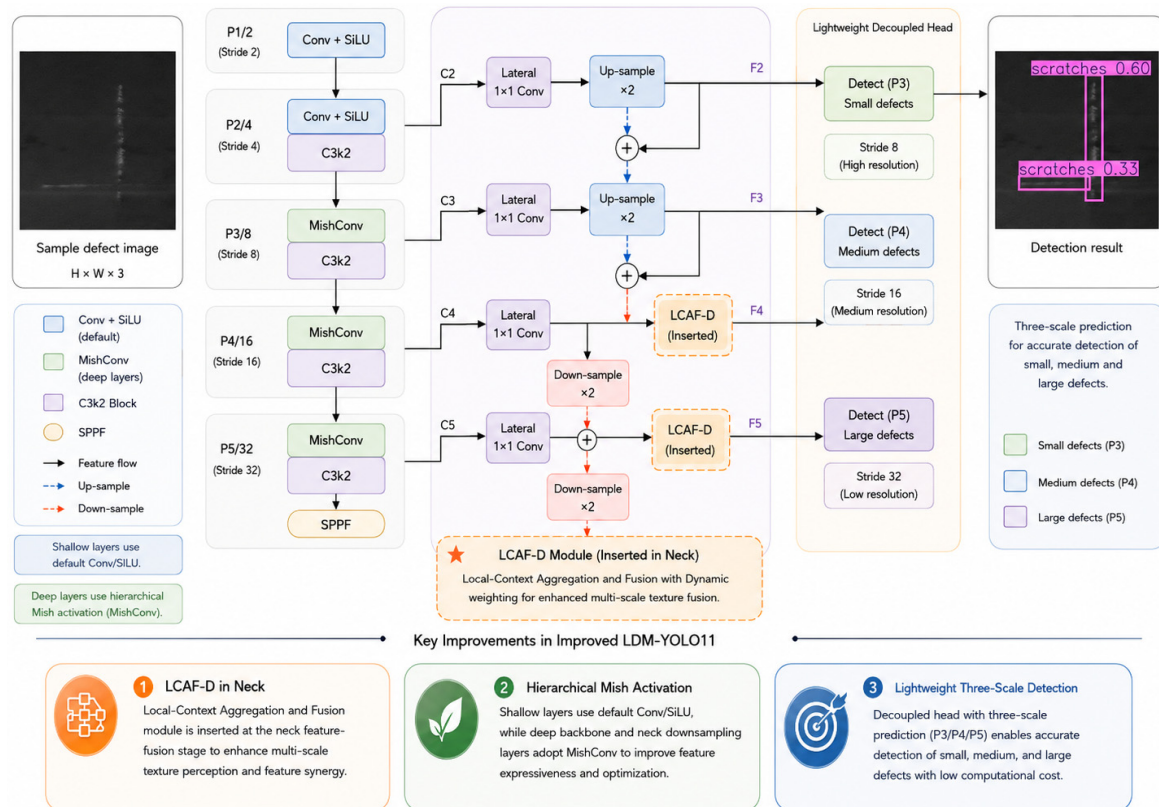


Figure 2. Overall architecture of the proposed LDM-YOLO11. The orange dashed blocks indicate the inserted LCAF-D modules, and the “+” symbol denotes element-wise feature fusion. Different colors are used to distinguish different functional components in the network.

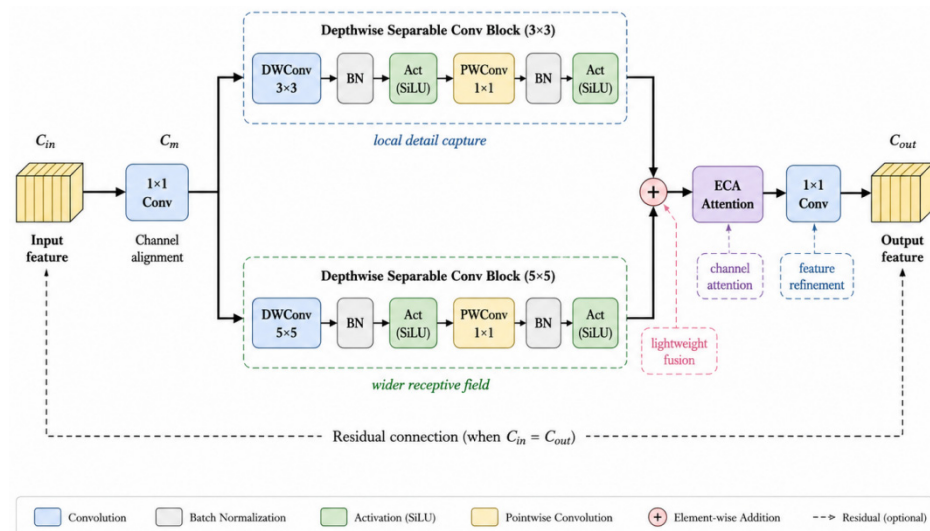


Figure 3. Structure of the lightweight channel-aware fusion module (LCAF).

3.2. Improved Dilated-Branch Fusion Module (LCAF-D)

Although LCAF enhances feature fusion to a certain extent, fine crack defects and complex texture defects are still difficult to detect in steel surface defect detection. The reason is that such defects often have blurred boundaries, weak textures, and high similarity to the background. If only conventional contextual fusion is used, fine-grained information may be weakened during feature transmission. To address this problem, the proposed method further proposes the improved LCAF-D module based on the original LCAF, as shown in Figure 4.

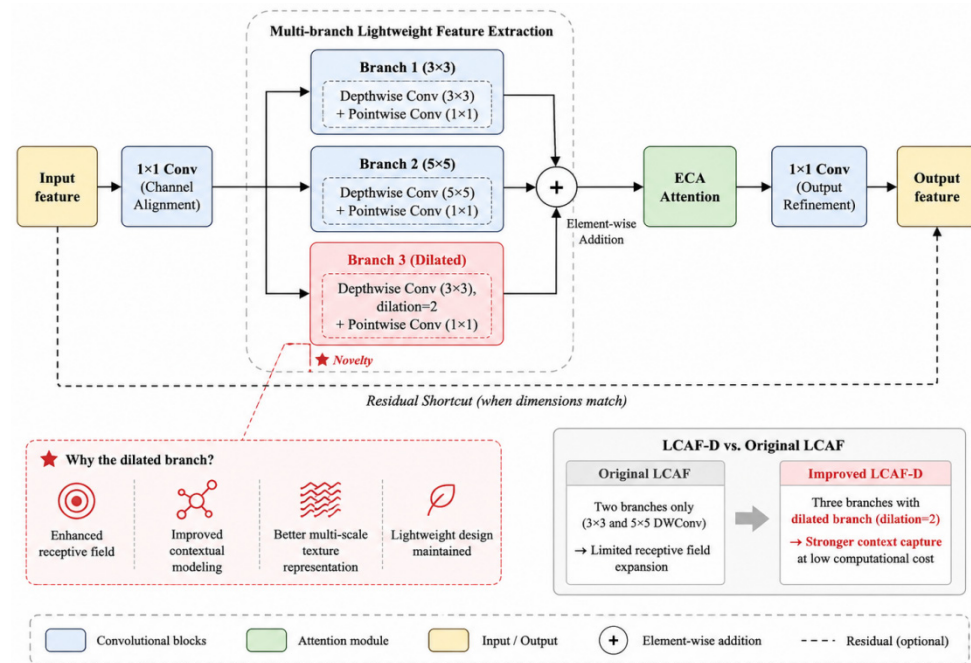


Figure 4. Structure of the improved dilated-branch fusion module (LCAF-D). The arrows indicate the feature propagation path. Blue blocks denote convolutional blocks, the green block denotes the attention module, and the yellow blocks denote the input/output features. The circled “+” symbol denotes element-wise addition, and the gray dashed path indicates the optional residual shortcut when the dimensions match. The red highlighted Branch 3 represents the newly introduced dilated branch, and the asterisk marks the novelty emphasized in the proposed design.

The main improvement of LCAF-D lies in the design of its feature-extraction branches. Instead of directly inheriting the original large-kernel contextual branch, LCAF-D introduces a dilated depthwise separable convolution branch to enlarge the receptive field with much lower extra complexity. In this way, the module gains stronger contextual perception while remaining lightweight. Compared with generic lightweight fusion or attention modules, the design emphasis of LCAF-D is more targeted: it seeks to jointly preserve fine-grained texture cues and enlarge contextual coverage, which is especially relevant for steel surface defects with weak contrast, elongated morphology, and irregular local texture.

In terms of module functionality, LCAF-D emphasizes the unification of contextual perception and detail preservation more strongly than the original LCAF. In steel surface defect detection, this improvement helps enhance the discrimination of weak targets and complex texture defects, and also provides a structural basis for subsequent improvements in overall network performance. Therefore, the contribution of LCAF-D should be understood as a lightweight, task-oriented refinement of YOLO11n feature fusion rather than a radically new architecture.

Regarding the specific placement of LCAF-D, the module is inserted in the neck after the upper-stage fusion block rather than in very shallow backbone layers or directly in the detection head. This placement is selected for three reasons. First, the neck is the stage where multi-scale semantic information from different feature levels is aggregated, so strengthening fusion at this point is more effective for defects with large scale variation than modifying an isolated shallow layer. Second, after the preceding upsampling and concatenation operations, the fused feature map already contains both higher-level contextual cues and mid-level structural details, which provides a more suitable input for the dilated branch to expand the receptive field while preserving fine-grained defect textures. Third, placing LCAF-D in this specific neck layer allows the module to enhance context-aware

representation before the subsequent prediction branches are formed, thereby benefiting small defects, elongated defects, and weak-texture defects without introducing unnecessary overhead to the entire network. Therefore, this location offers a practical balance between feature effectiveness and lightweight deployment.

3.3. Hierarchical Mish Activation Optimization Strategy

Besides the feature-fusion problem, steel surface defect detection faces another important challenge: complex texture defects and blurred-boundary defects impose higher requirements on the nonlinear representation of deep network features. If the deep network cannot model complex patterns adequately, classification may become unstable or localization may be inaccurate. To address this issue, the proposed method further proposes a hierarchical Mish activation optimization strategy, as illustrated in Figure 5.

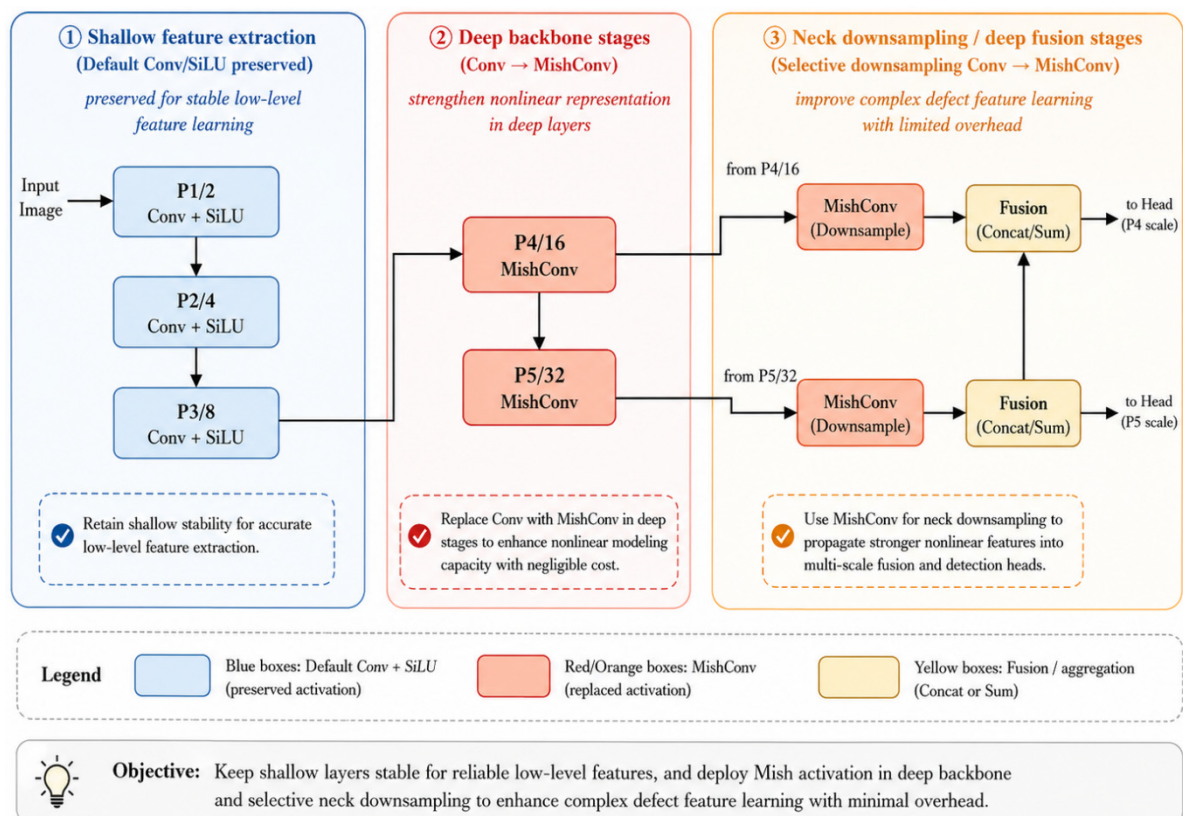


Figure 5. Hierarchical Mish activation deployment strategy in the proposed model.

This strategy does not replace all activation functions in the entire network uniformly. Instead, Mish is only used in the deeper backbone convolutions at Stage 4 and Stage 5 and in the key downsampling convolutions of the neck, while the shallow feature-extraction part and the detection head remain unchanged. The main rationale is that shallow features focus more on edges and basic textures, so preserving the original activation is beneficial to training stability; deep features, by contrast, undertake more complex semantic modeling and long-range contextual interaction, where Mish can better enhance nonlinear representation. In this sense, the hierarchical Mish design is also an incremental and position-sensitive optimization rather than a global activation replacement.

From the perspective of method characteristics, the hierarchical Mish activation optimization strategy avoids the instability that may be caused by globally replacing activations throughout the whole network, while still highlighting improvements in deep-feature representation. For common complex textures, blurred boundaries, and weak-texture defects

in steel surface defect detection, such a hierarchical optimization better serves the feature-modeling requirements of the object detection network.

4. Experimental Results and Analysis

4.1. Experimental Setup

All experiments were conducted on a workstation equipped with an NVIDIA GeForce RTX 4070 Laptop GPU (8 GB VRAM), using Python 3.11.15, PyTorch 2.7.1 + cu118, and Ultralytics 8.4.31. The NEU-DET dataset was used as the primary benchmark, and all models were trained under the same dataset split and optimization settings unless otherwise specified. During the early ablation stage, an input size of 224×224 was adopted to efficiently compare different lightweight module configurations, including LCAF, LCAF-D, activation strategies, and branch designs. However, considering that steel surface defects, especially crazing and rolled-in_scale, contain fine-grained textures, weak boundaries, and elongated structures, we further conducted input-size sensitivity analysis under 224×224 , 320×320 , and 640×640 . Based on the observed performance trend, the 640×640 setting was adopted in the repeated experiments, runtime benchmark, and cross-dataset verification. For all runs, the optimizer was SGD with an initial learning rate of 0.005, momentum of 0.937, weight decay of 0.0005, batch size 16, and 200 epochs. Repeated experiments were conducted using five random seeds (0, 11, 42, 123, and 3407) under the same split and protocol.

4.2. NEU-DET Dataset and Evaluation Metrics

The proposed method is mainly validated on the NEU-DET dataset. NEU-DET is a classic public dataset for steel surface defect detection and contains six representative defect categories: crazing, inclusion, patches, pitted_surface, rolled-in_scale, and scratches. These defects differ significantly in morphology, scale, and texture distribution. Some of them exhibit fine-crack or weak-texture characteristics, with blurred boundaries, small target scales, and a tendency to be confused with the background. Therefore, the dataset can well reflect the practical challenges of steel surface defect detection. For this study, NEU-DET can be used not only to evaluate overall detection performance, but also to observe category adaptability and feature representation across different defect types.

To visually illustrate the representative defect types in the NEU-DET dataset, original sample images of each defect category are provided in Figure 6. It can be seen that different defect types exhibit considerable differences in texture morphology, edge clarity, and spatial distribution. This further indicates that steel surface defect detection places high demands on multi-scale feature fusion, local detail preservation, and adaptation to complex backgrounds.

To comprehensively evaluate detection performance, the proposed method adopts commonly used metrics in object detection, including Precision, Recall, IoU (Intersection over Union), mAP@0.5, and mAP@0.5:0.95.

Precision measures how many of the samples predicted as positive by the model are truly positive, and is defined as:

$$P = TP / (TP + FP)$$

where TP denotes true positives, i.e., the number of targets correctly detected by the model, and FP denotes false positives, i.e., the number of targets incorrectly predicted as objects.

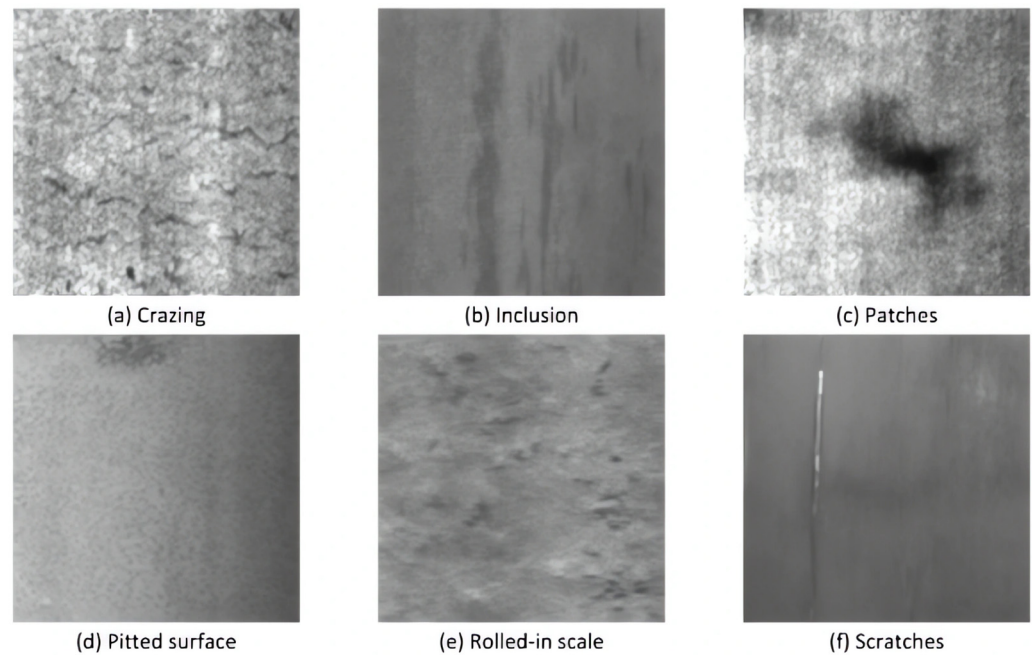


Figure 6. Original sample images of each steel surface defect category in the NEU-DET dataset.

Recall measures how many of all real targets are successfully detected by the model, and is defined as:

$$R = TP / (TP + FN)$$

where FN denotes false negatives, i.e., the number of targets that truly exist but are not detected by the model.

IoU is used to measure the overlap between the predicted box and the ground-truth box, and is defined as:

$$IoU = intersection(B_p, B_g) / union(B_p, B_g)$$

where B_p denotes the predicted bounding box, B_g denotes the ground-truth bounding box, the numerator indicates the intersection area of the predicted and ground-truth boxes, and the denominator indicates the union area of the two.

Based on the Precision–Recall curve, Average Precision (AP) can be further calculated to measure the overall detection performance of a category at different recall levels. It can be expressed as:

$$AP = \text{integral from } 0 \text{ to } 1 \text{ of } P(R) dR$$

where Precision–Recall denotes the functional relationship of precision with respect to recall on the PR curve.

For multi-class object detection, the mean Average Precision (mAP), i.e., the average of AP values over all categories, is typically used as the overall performance metric. The proposed method mainly uses the following two forms:

First, mAP@0.5, namely the mean AP value of all classes when the IoU threshold is set to 0.5, is defined as:

$$mAP@0.5 = (1/N) \times \text{sum from } i = 1 \text{ to } N \text{ of } AP_i^{0.5}$$

Second, mAP@0.5:0.95, namely the average AP computed over multiple IoU thresholds from 0.5 to 0.95 with a step size of 0.05, is defined as:

$$mAP@0.5:0.95 = (1/N) \times \text{sum from } i = 1 \text{ to } N \text{ of } ((1/10) \times \text{sum from } t = 0.50 \text{ to } 0.95 \text{ of } AP_i^t)$$

where N denotes the number of categories, $AP_i^{0.5}$ denotes the AP of the i -th category at $IoU = 0.5$, and AP_i^t denotes the AP of the i -th category at IoU threshold t .

Overall, Precision mainly reflects the correctness of model predictions, Recall reflects the ability to find real targets, IoU measures the overlap quality between predicted and ground-truth boxes, $mAP@0.5$ better reflects the overall detection ability under a relatively loose matching criterion, and $mAP@0.5:0.95$ is stricter and better reflects comprehensive performance under different localization-accuracy requirements. For steel surface defect detection, these metrics can comprehensively evaluate recognition and localization performance from different perspectives.

4.3. Ablation Results and Analysis

To systematically evaluate the effectiveness of the proposed improvements, this study conducts experiments from multiple perspectives, including module design, module placement, branch structure, and activation-function optimization. Since the final model is gradually evolved from LCAF, LCAF-D, and the hierarchical Mish activation strategy, this section analyzes the results of each stage according to the order of module improvement and verifies the rationality of each strategy through ablation and comparative experiments.

4.3.1. Introduction of LCAF and Comparison of Branch Designs

In steel surface defect detection, defect regions often exhibit large scale differences, complex local textures, and strong background interference. Relying only on the original lightweight detector's feature-fusion capability often makes it difficult to balance local details and contextual information. To address this problem, the proposed method first introduces the lightweight channel-aware fusion module LCAF to strengthen multi-scale feature interaction. Through the parallel design of a local branch and a larger-receptive-field branch, LCAF jointly models local details and contextual information in a lightweight structure. Its main advantages are twofold: it can strengthen the local representation of texture-like defects and supplement contextual semantics within a certain range, thereby improving defect recognition and localization.

On top of the original design, the proposed method further supplements branch-structure comparison experiments to analyze why the final design adopts a 3×3 dilated branch instead of smaller or larger alternatives. Table 1 reports the results of different alternative branch designs. The 2×2 branch obtains $mAP@0.5$ and $mAP@0.5:0.95$ of 0.729 and 0.408, respectively, while the 4×4 branch further declines to 0.713 and 0.390. In contrast, the finally adopted 3×3 dilated branch reaches 0.757 and 0.438 and achieves the best performance at similar complexity. This indicates that simply reducing or enlarging kernel size is insufficient to obtain the best detection effect, whereas the design of a compact 3×3 kernel combined with dilation achieves a better balance between contextual modeling and fine-grained texture preservation.

Table 1. Supplementary ablation experiment on LCAF-D branch design.

Model	Replacement Branch	Kernel Size	Dilation Rate	Params (M)	GFLOPs	P	R	$mAP@0.5$	$mAP@0.5:0.95$
YOLO11n + LCAF- 2×2	Even-sized small-kernel branch	2×2	1	3.266	10.4	0.678	0.680	0.729	0.408
YOLO11n + LCAF- 4×4	Even-sized larger-kernel branch	4×4	1	3.257	10.2	0.677	0.670	0.713	0.390
YOLO11n + LCAF-D	Compact dilated branch	3×3	2	3.256	10.2	0.753	0.696	0.757	0.438

As shown in Table 1, the finally adopted 3×3 dilated branch outperforms the 2×2 and 4×4 alternatives in both overall accuracy and strict localization metrics, thus providing experimental support for the final structure of LCAF-D.

4.3.2. Position Ablation Study of LCAF

After introducing the LCAF module, the proposed method further analyzes the effect of deploying it at different network positions. Since feature-enhancement modules may behave differently in backbone feature extraction and neck feature fusion, three settings are considered: backbone-only, neck-only, and backbone + neck, with results shown in Table 2. Here, backbone-only specifically means that LCAF is inserted into the deeper backbone stage corresponding to the P4/16 feature extraction block; neck-only means that LCAF is inserted into the neck fusion path after the concatenation of the upsampled P5 feature and the P4 feature; backbone + neck means that both of these positions are activated simultaneously.

Table 2. Position ablation results of LCAF.

Model	LCAF Position	Params (M)	GFLOPs	P	R	mAP@0.5	mAP@0.5:0.95
YOLO11n	None	2.583	6.3	0.733	0.662	0.730	0.405
YOLO11n + LCAF	Backbone only	2.857	6.5	0.716	0.705	0.734	0.407
YOLO11n + LCAF	Neck only	3.259	10.2	0.641	0.743	0.742	0.423
YOLO11n + LCAF	Backbone + Neck	2.928	6.8	0.726	0.646	0.735	0.426

The experimental results show that the backbone-only configuration brings only limited improvement over the baseline, with mAP@0.5:0.95 increasing from 0.405 to 0.407; the neck-only configuration, however, achieves 0.742 mAP@0.5 and 0.423 mAP@0.5:0.95, clearly performing better. This indicates that the main role of LCAF is more concentrated in the multi-scale feature-fusion stage rather than in isolated deep backbone enhancement. Although the backbone + neck configuration reaches 0.426 mAP@0.5:0.95, the neck-only setting more directly reflects the contribution of LCAF in the fusion stage while keeping the structural interpretation clearer. Therefore, all subsequent improvements are built upon the neck-only configuration, namely the fusion position after the P5-upsample and P4-feature concatenation.

4.3.3. Validation of the Improvement from LCAF to LCAF-D

The experiments with the original LCAF show that although overall performance improves, there is still room for further optimization on fine crack defects. To address this issue, LCAF-D is proposed by replacing the original large-kernel branch with a dilated depthwise separable convolution branch, thereby strengthening the preservation of fine-grained texture information in complex backgrounds. To verify whether this improvement is reasonable, LCAF-D is compared with the original LCAF as well as representative lightweight modules ECA and CBAM, and the results are reported in Table 3.

Table 3. Comparison results of representative lightweight modules.

Model	Params (M)	GFLOPs	P	R	mAP@0.5	mAP@0.5:0.95
YOLO11n baseline	2.583	6.3	0.733	0.662	0.730	0.405
YOLO11n + ECA (neck)	3.187	10.0	0.736	0.700	0.729	0.396
YOLO11n + CBAM (neck)	3.204	10.0	0.702	0.662	0.724	0.409
YOLO11n + LCAF (neck)	3.259	10.2	0.641	0.743	0.742	0.423
YOLO11n + LCAF-D (neck)	3.256	10.2	0.753	0.696	0.757	0.438

As shown in Table 3, ECA neck and CBAM neck achieve mAP@0.5 values of 0.729 and 0.724 and mAP@0.5:0.95 values of 0.396 and 0.409, respectively, both lower than the 0.742 and 0.423 obtained by the original LCAF neck. Furthermore, LCAF-D neck improves performance to 0.757 and 0.438, indicating that it can more effectively improve detection

accuracy and localization ability while maintaining similar complexity. It should be noted that an earlier uncorrected configuration produced larger complexity values for LCAF-D alone, whereas the revised manuscript reports the corrected and reverified configurations consistently; the final LCAF-D + Mish model in later tables is based on the clean corrected YAML and therefore cannot be directly compared with the earlier uncorrected complexity numbers. This result shows that LCAF-D is not an empirical module obtained by simply replacing a branch, but a better structure optimized from the original module for steel fine-crack and complex-texture defects.

4.3.4. Mish Activation Strategy and Joint Comparative Experiments

Besides feature-enhancement modules, the proposed method also considers the role of activation-function optimization in detecting complex defects. Since steel surface defects include many targets with complex textures, blurred boundaries, and weak local features, structural improvement alone may still be insufficient to fully enhance deep-feature representation. Therefore, a hierarchical Mish activation optimization strategy is further introduced. Experimental results show that the Mish-only model reaches 0.744 mAP@0.5 and 0.444 mAP@0.5:0.95 on NEU-DET, demonstrating a clear advantage under stricter localization metrics.

To further verify the applicability of Mish, the proposed method also compares LCAF-D + LeakyReLU, LCAF-D + Mish, and a joint scheme with an additional HeadRefine module. The results are listed in Table 4. LCAF-D + LeakyReLU achieves 0.718 and 0.405 on mAP@0.5 and mAP@0.5:0.95, respectively, clearly lower than the 0.767 and 0.436 of LCAF-D + Mish. Meanwhile, the additional HeadRefine attempt does not surpass the final scheme. This indicates that Mish is not only effective when used alone, but also shows better compatibility when combined with LCAF-D.

Table 4. Supplementary comparison results of activations and joint models.

Model	Params (M)	GFLOPs	P	R	mAP@0.5	mAP@0.5:0.95
YOLO11n + Mish	2.591	6.4	0.702	0.695	0.744	0.444
YOLO11n + LCAF-D + LeakyReLU	3.256	10.2	0.672	0.658	0.718	0.405
YOLO11n + LCAF-D + Mish	2.652	6.5	0.719	0.655	0.724	0.406
YOLO11n + LCAF-D + Mish + HeadRefine	2.854	7.1	0.699	0.684	0.741	0.431

4.3.5. Summary of Overall Ablation Experiments

To compare the contributions of all module improvements to model performance from an overall perspective, the proposed method further summarizes the experimental results under different configurations in Table 5. Starting from the baseline model, the table lists the performance changes brought by LCAF (neck), Mish, LCAF-D (neck), and the final model LCAF-D + Mish in turn. This table should be interpreted together with the later 640×640 sensitivity, repeated-run, and runtime analyses rather than as the sole basis for the final conclusion.

Table 5. Overall ablation results under the preliminary 224×224 setting.

Model	Params (M)	GFLOPs	P	R	mAP@0.5	mAP@0.5:0.95
YOLO11n	2.583	6.3	0.733	0.662	0.730	0.405
YOLO11n + LCAF (neck)	3.259	10.2	0.641	0.743	0.742	0.423
YOLO11n + Mish	2.591	6.4	0.702	0.695	0.744	0.444
YOLO11n + LCAF-D (neck)	2.652	6.5	0.659	0.693	0.713	0.422
YOLO11n + LCAF-D + Mish	2.652	6.5	0.719	0.655	0.724	0.406

As shown in Table 5, under the preliminary 224×224 setting used for early ablation, the baseline YOLO11n attains 0.730 and 0.405 in mAP@0.5 and mAP@0.5:0.95, respectively; adding LCAF (neck) improves them to 0.742 and 0.423; using Mish alone reaches 0.744 and 0.444; the corrected LCAF-D (neck) configuration yields 0.713 and 0.422; and the clean LCAF-D + Mish model achieves 0.724 and 0.406. These results indicate that the proposed design does not show a clear advantage under low-resolution settings, which motivates the additional higher-resolution analysis presented in the revised manuscript.

4.4. Analysis of PR Curves, Normalized Confusion Matrices, Class-Wise Results, and Failure Cases

To further analyze the detection-performance differences across categories, the proposed method compares the baseline model and the final model using Precision–Recall (PR) curves and normalized confusion matrices. Compared with overall metrics alone, PR curves can more intuitively reflect changes in precision at different recall levels, thereby revealing detection stability and comprehensive performance for different categories. The normalized confusion matrix can show the model’s classification ability, confusion patterns, and missed detections from the perspective of inter-class prediction distribution. Therefore, these two types of figures complement the effectiveness of the proposed improvements from the two aspects of overall performance change and class-level recognition behavior.

4.4.1. PR Curve Analysis

PR curves reflect how precision changes at different recall levels. If the PR curve of a class is closer to the upper-right corner, the model has better overall detection performance on that class; if the area under the overall PR curve is larger, the overall average precision is higher. Therefore, PR curves can reflect not only overall performance differences but also the improvement direction of each defect category.

Under the preliminary 224×224 setting, the baseline YOLO11n and the clean LCAF-D + Mish model show only close overall PR behavior, which is consistent with the quantitative ablation results. However, this low-resolution setting suppresses some of the benefits of the proposed design on fine-grained and elongated defects. Therefore, a higher-resolution sensitivity analysis is further introduced in Section 4.7 to more comprehensively evaluate the final model.

Further analysis shows that the final model improves the overall detection performance under the 640×640 setting. Figures 7 and 8 present the PR curves of YOLO11n and LDM-YOLO11, respectively, while Figure 9 further summarizes the final 640×640 class-wise AP@0.5 comparison. The proposed LDM-YOLO11 improves the overall mAP@0.5 from 0.730 to 0.771 and brings clear gains for challenging categories such as crazing, rolled-in_scale, and scratches. The class-wise changes are not uniform: inclusion and patches show slight decreases, indicating that the improvement is category-dependent rather than universal. This observation is consistent with the paper’s positioning of LDM-YOLO11 as a lightweight task-oriented refinement of YOLO11n.

4.4.2. Analysis of Normalized Confusion Matrices

The normalized confusion matrix reflects the prediction distribution among different categories. The normalized confusion matrix of the baseline YOLO11n is shown in Figure 10, while that of the proposed LDM-YOLO11 is shown in Figure 11. Its diagonal elements represent the proportion of correctly classified samples; larger values indicate more accurate recognition for that class. The off-diagonal elements reflect confusion among categories, while background-related areas reflect missed detections or false detections. Therefore, the normalized confusion matrix helps analyze which categories improve and whether the proposed method reduces inter-class confusion and background errors.

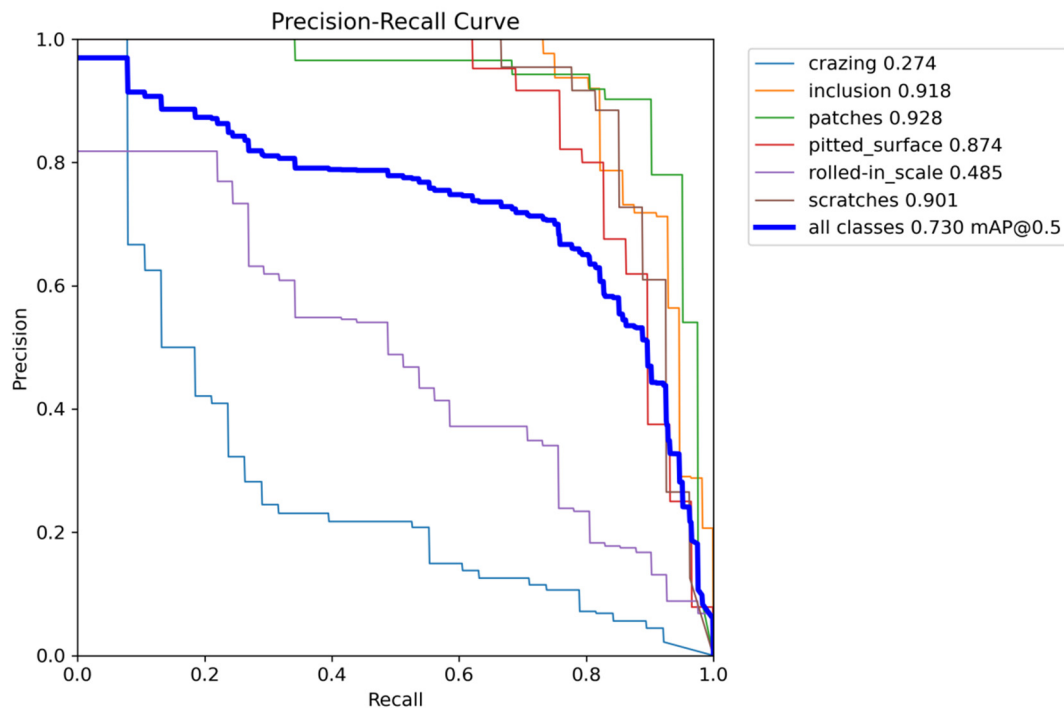


Figure 7. PR curve of YOLO11n.

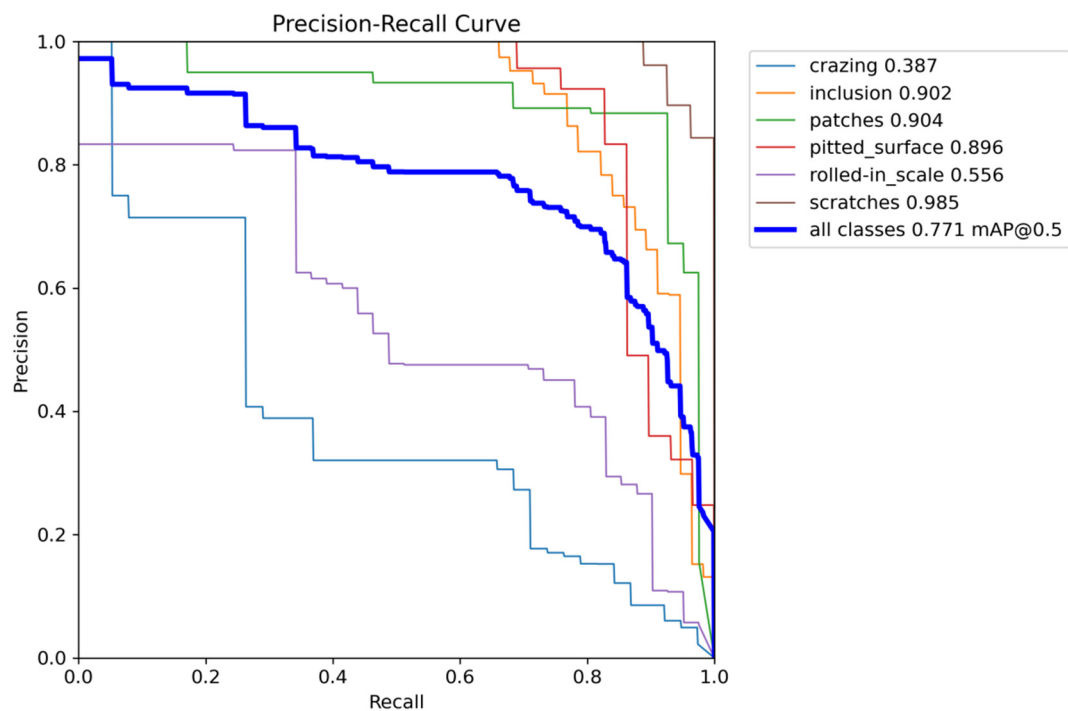


Figure 8. PR curve of LDM-YOLO11.

From the confusion-matrix comparison in the proposed method, the final model achieves better diagonal recognition results than the baseline on multiple categories. For example, the number of correctly recognized crazing instances increases from 9 to 10, and inclusion increases from 45 to 47, indicating that the improved model strengthens classification ability on these categories. Although patches decreases from 39 to 38, pitted_surface from 24 to 20, and rolled-in_scale from 22 to 20, the PR curves show that the overall AP of these categories is still higher in the final model. This suggests that the model improves

the precision–recall balance rather than only the number of correct classifications under a single threshold.

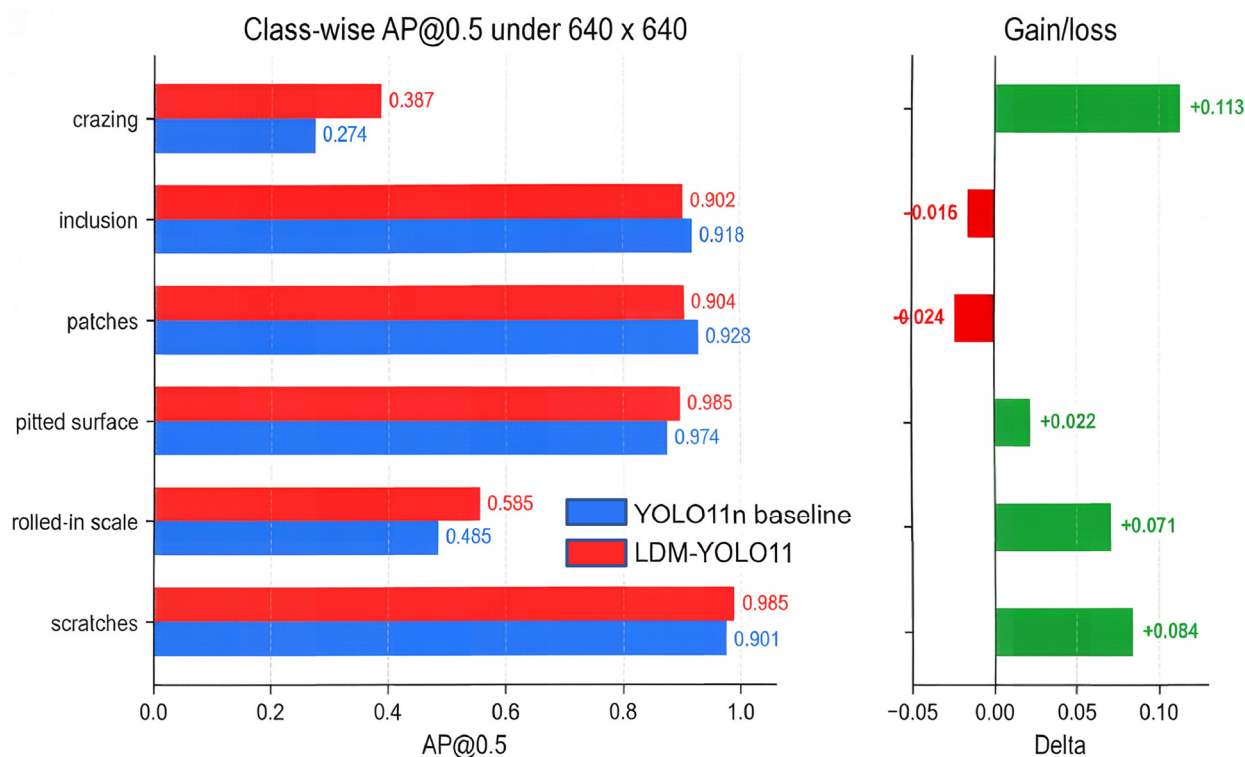


Figure 9. Class-wise AP@0.5 comparison between YOLO11n and LDM-YOLO11 on the NEU-DET dataset under the final 640×640 setting.

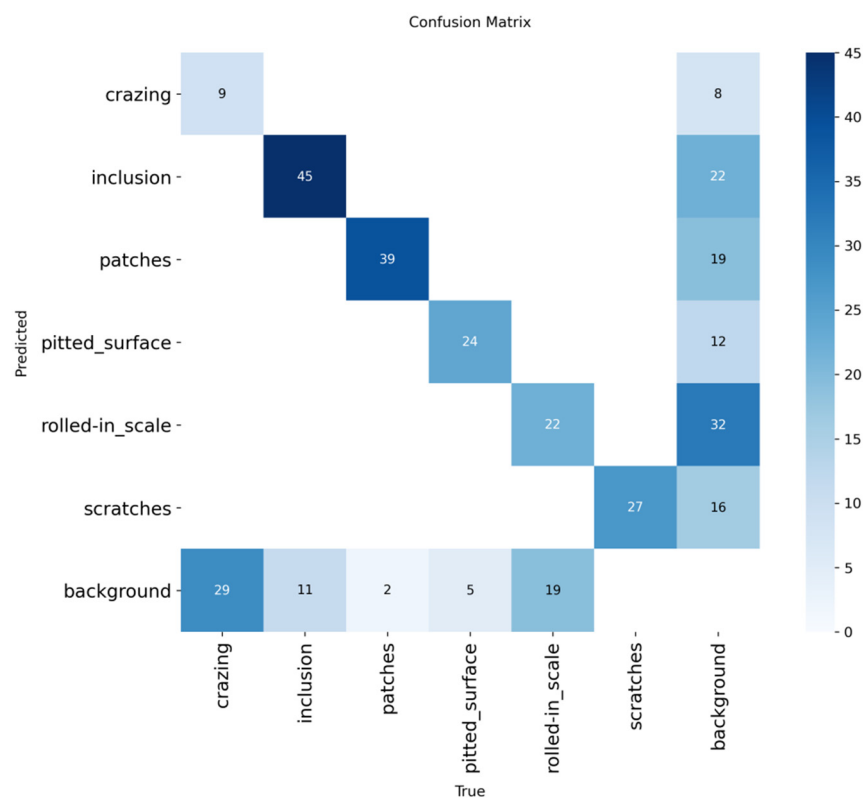


Figure 10. Normalized confusion matrix of YOLO11n.

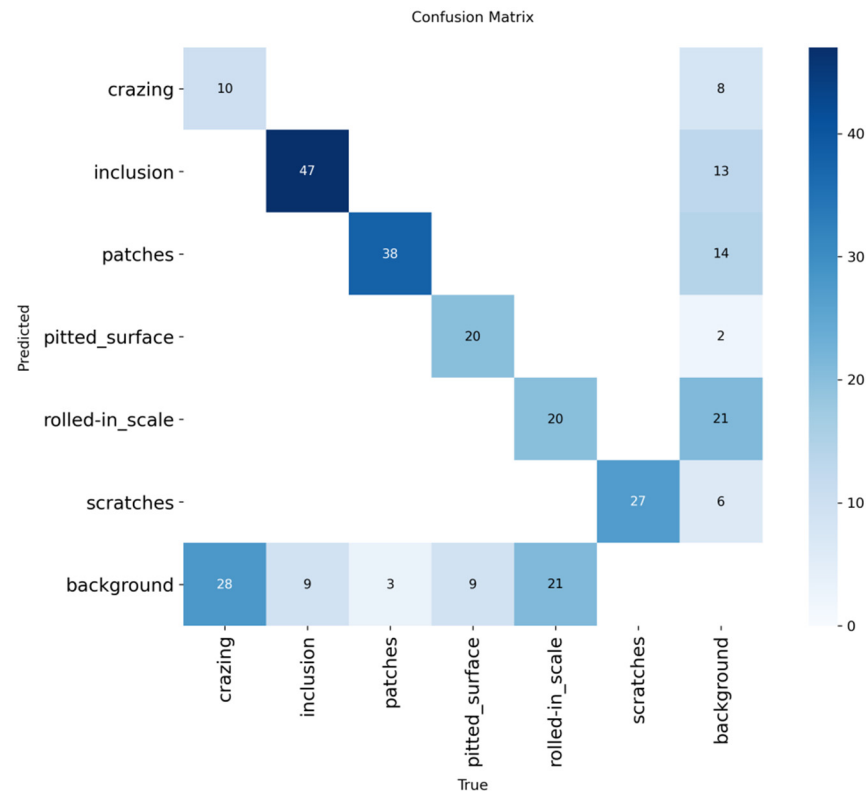


Figure 11. Normalized confusion matrix of LDM-YOLO11.

From the perspective of background errors and missed detections, the improved model also shows advantages. For the background column, the baseline confusion values of inclusion, patches, rolled-in_scale, and scratches are 22, 19, 32, and 16, respectively, whereas they decrease to 13, 14, 21, and 6 in the final model. This indicates that the improved model clearly reduces background errors or missed detections for these categories. In particular, the background confusion of scratches drops from 16 to 6, which is consistent with the increase in its AP from 0.944 to 0.989 in the PR curves, demonstrating substantially improved detection stability and classification accuracy for this category.

For difficult categories such as crazing and rolled-in_scale, although the final model still exhibits a certain amount of background confusion, the overall PR curves and confusion matrices both show more positive trends. This indicates that the proposed method enhances discrimination for difficult defect categories. However, because these categories inherently exhibit weak texture, blurred boundaries, and high similarity to the background, they remain key targets for future optimization.

Overall, the PR curves, normalized confusion matrices, and the final 640×640 class-wise AP comparison jointly verify the effectiveness and limitations of the proposed design. The results indicate that the gain of LDM-YOLO11 is concentrated on several difficult or fine-grained defect categories, while some categories remain sensitive to data distribution and visual ambiguity. For this reason, the revised manuscript further supplements input-size sensitivity analysis and repeated experiments under 640×640 .

4.4.3. Class-Wise Analysis on NEU-DET

Table 6 presents the class-wise comparison on the NEU-DET dataset under the 640×640 setting. The proposed YOLO11n + LCAF-D + Mish improves several challenging categories, especially crazing, rolled-in_scale, and scratches. For example, the AP@0.5 of crazing increases from 27.4% to 38.7%, while rolled-in_scale improves from 48.5% to 55.6%. The AP@0.5:0.95 of scratches also increases substantially from 39.9% to

56.4%. These results indicate that the proposed feature fusion and hierarchical activation strategy are particularly beneficial for defect categories with fine-grained texture variation and ambiguous boundaries. At the same time, the improvements are category-dependent rather than uniform across all categories.

Table 6. Class-wise comparison between YOLO11n and the proposed YOLO11n + LCAF-D + Mish on the NEU-DET dataset under the 640×640 setting.

Class	Baseline AP@0.5	Baseline AP@0.5:0.95	Proposed AP@0.5	Proposed AP@0.5:0.95
crazing	0.274	0.109	0.387	0.157
inclusion	0.918	0.510	0.902	0.532
patches	0.928	0.640	0.904	0.555
pitted_surface	0.874	0.548	0.896	0.574
rolled-in_scale	0.485	0.243	0.556	0.274
scratches	0.901	0.399	0.985	0.564

4.4.4. Failure Case Analysis

Figure 12 shows representative failure cases and challenging examples for the baseline YOLO11n and the proposed model. In Cases A1–A2, the proposed YOLO11n + LCAF-D + Mish improves the detection of difficult defect regions, especially those with weak texture contrast or elongated morphology. In Cases B1–B2, however, the baseline performs better, indicating that the proposed design does not consistently improve all samples. In Cases C1–C2, both models remain challenged by highly ambiguous boundaries, subtle defect traces, or severe background interference. These observations suggest that the proposed method improves overall detection performance under the 640×640 setting, but difficult categories and visually ambiguous cases still remain open challenges.

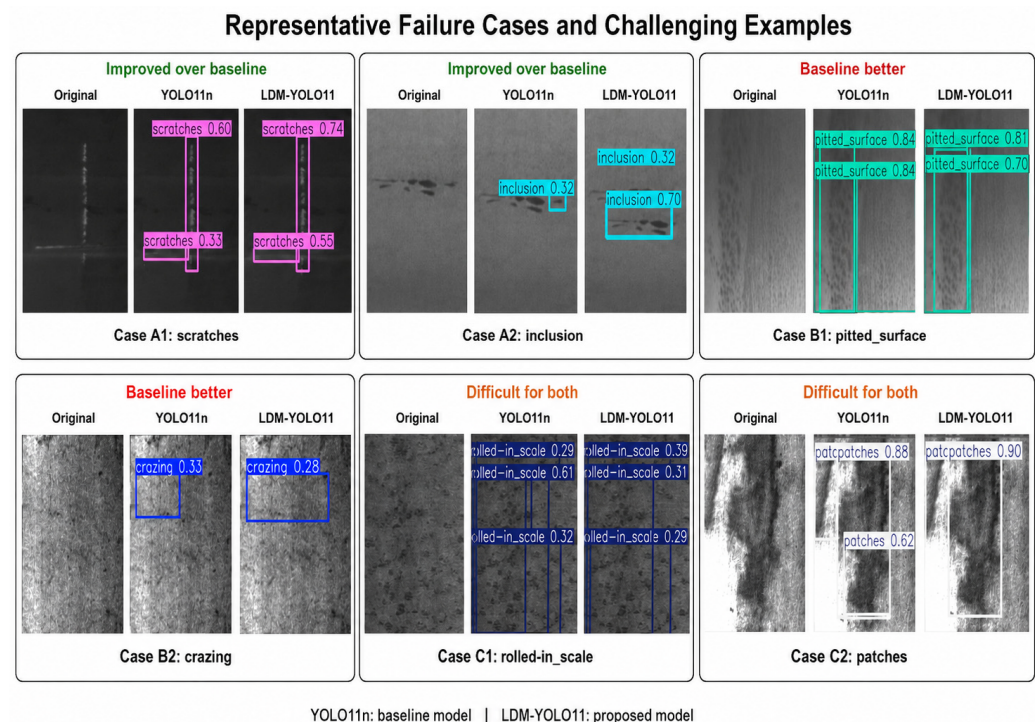


Figure 12. Representative failure cases and challenging examples for YOLO11n and LDM-YOLO11.

4.5. Comparison and Discussion with Related Methods

To further evaluate the practical position of the proposed method in steel surface defect detection, this study collects published results of several representative recent methods on

the NEU-DET dataset and compares them with the final model. It should be noted that different papers may vary in experimental settings, input size, data split, training schedule, augmentation policy, and metric-reporting format. Therefore, the purpose of this horizontal comparison is not to claim state-of-the-art superiority, but to analyze how the proposed method is positioned relative to recent lightweight and defect-oriented detectors under publicly reported conditions.

According to publicly reported results, existing methods have achieved strong performance on steel surface defect detection. Representative directions include feature-fusion-enhanced lightweight detectors, attention-guided lightweight variants, and task-specific improvement frameworks built on YOLO-style backbones. Some methods report higher mAP@0.5 values than the proposed model on NEU-DET, while others provide stronger trade-offs in complexity or speed. Therefore, the revised manuscript deliberately treats these published results as comparative references rather than a strict ranking.

As shown in Table 7, the existing studies have already achieved a high level of performance on the NEU-DET dataset, and different methods exhibit different advantages in accuracy, complexity, and inference efficiency. To improve comparability, the revised manuscript additionally reports the updated result of the proposed method under the 640×640 setting, where it reaches 77.1 mAP@0.5 and 44.3 mAP@0.5:0.95 while maintaining a relatively lightweight structure. Nevertheless, the manuscript no longer implies that the proposed model is uniformly superior to all published methods, and instead emphasizes its value as a competitive lightweight engineering optimization within the YOLO11n framework.

Table 7. Publicly reported results of different methods on the NEU-DET dataset.

Method	Baseline	mAP@0.5	mAP@0.5:0.95	Params (M)	GFLOPs	FPS
LDM-YOLO11 (Ours)	YOLO11n	77.1	44.3	2.652	6.5	70.7
YOLOv8n	—	76.3	46.1	3.0	8.9	158.3
YOLO11n	—	77.2	45.6	2.5	6.4	158.3
YOLO-LSDI	YOLO11n	83.0	48.0	2.7	6.1	162.1
FMV-YOLO	YOLO11n	80.2	—	2.6	5.7	99
Improved YOLOv7	YOLOv7	79.3	—	—	—	72
LCED-YOLO	—	79.8	—	2.1	5.0	151

Note: In the table, “—” indicates that the corresponding metric is not explicitly reported in the publicly available article or abstract, and thus is not inferred here. The results of the proposed model come from the experiments in this work, whereas the other methods are taken from the corresponding publications or their publicly available abstracts [3,6,7,9]. Since the compared methods do not necessarily share the same split, input resolution, or training protocol, Table 7 should be read as a loose reference for contextual positioning rather than as a strictly controlled benchmark.

Overall, the related methods all demonstrate good performance and research value in steel surface defect detection, while the proposed LDM-YOLO11 provides, within the lightweight YOLO11n framework, a competitive trade-off between accuracy and efficiency. At the same time, the manuscript now avoids claiming universal superiority, since published methods may still use different splits, input sizes, training policies, and reporting protocols.

4.6. System Performance Analysis

To further evaluate the deployment efficiency of the proposed method, we conducted a unified runtime benchmark on the same hardware platform (RTX 4070 Laptop GPU) using 100 validation images. The average preprocessing time, inference time, postprocessing time, total latency, approximate FPS, and peak GPU memory were recorded for different models and input resolutions. The results are summarized in Table 8. Since the proposed method

aims at lightweight industrial applicability, these runtime metrics are reported together with parameters and GFLOPs rather than being replaced by complexity numbers alone.

Table 8. System performance comparison of different models under different input resolutions on the NEU-DET dataset.

Model	Input Size	Params (M)	GFLOPs	Preprocess (ms)	Inference (ms)	Postprocess (ms)	Total Latency (ms)	FPS	Peak GPU Memory (MB)
YOLO11n baseline	224 × 224	2.583	6.3	0.426	10.580	1.435	12.442	80.37	53.68
YOLO11n baseline	640 × 640	2.583	6.3	2.567	9.380	1.292	13.239	75.54	69.91
YOLO11n + LCAF-D + Mish	224 × 224	2.652	6.5	0.348	9.491	1.119	10.958	91.26	59.46
YOLO11n + LCAF-D + Mish	640 × 640	2.652	6.5	2.417	10.412	1.316	14.145	70.70	72.20

As shown in Table 8, the proposed YOLO11n + LCAF-D + Mish introduces only a slight increase in model complexity compared with the baseline YOLO11n, with parameters rising from 2.583 M to 2.652 M and GFLOPs increasing from 6.3 to 6.5. In terms of runtime efficiency, the proposed model maintains competitive performance across different input resolutions. At 640×640 , although the total latency increases to 14.145 ms, the model still reaches 70.70 FPS while delivering the best detection accuracy.

Meanwhile, the peak GPU memory only increases marginally from 69.91 MB to 72.20 MB under the 640×640 setting. These results indicate that the proposed method preserves lightweight deployment characteristics while improving detection performance under higher-resolution inputs. Figure 13 provides representative qualitative examples that are consistent with this quantitative trend. In the present manuscript, the deployment evidence is limited to GPU-based benchmarking under a unified environment; broader CPU, Jetson, ONNX, or TensorRT deployment will be considered in future work rather than being overstated here.

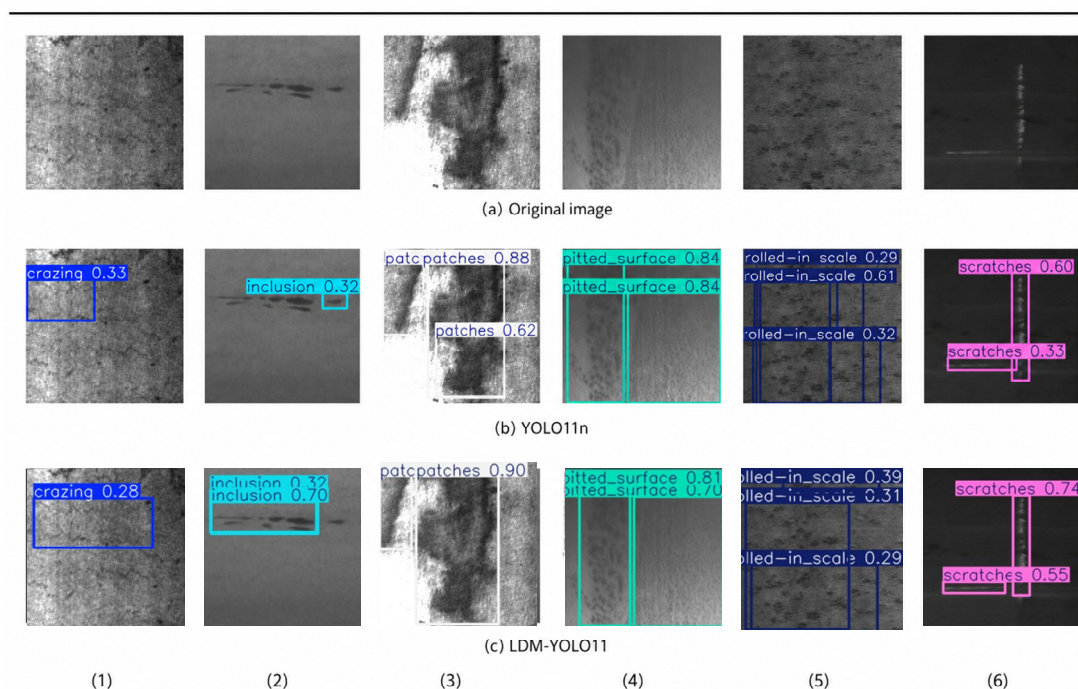


Figure 13. Visual comparison of detection results of YOLO11n and LDM-YOLO11 on representative steel surface defect samples. Different colored bounding boxes are used only to distinguish different defect categories for visual clarity.

Table 9 further reports the class-wise comparison on the GC10-DET dataset under the 640×640 setting. The proposed model achieves clear gains for several categories, especially 2_hanfeng, 3_yueyawan, and 10_yaozhed. For instance, the AP@0.5 of 2_hanfeng

increases from 25.3% to 85.1%, and 3_yueyawan improves from 92.5% to 95.3%. However, some difficult categories, such as 8_yahen, remain challenging for both models. These observations suggest that the proposed method improves the overall cross-dataset detection performance under the unified higher-resolution setting, while the category-wise difficulty remains uneven.

Table 9. Class-wise comparison between YOLO11n and the proposed YOLO11n + LCAF-D + Mish on the GC10-DET dataset under the 640×640 setting.

Class	Baseline AP@0.5	Baseline AP@0.5:0.95	Proposed AP@0.5	Proposed AP@0.5:0.95
1_chongkong	0.981	0.605	0.980	0.586
2_hanfeng	0.253	0.065	0.851	0.307
3_yueyawan	0.925	0.611	0.953	0.637
4_shuiban	0.796	0.468	0.791	0.471
5_youban	0.598	0.297	0.539	0.265
6_siban	0.520	0.244	0.529	0.240
7_yiwu	0.390	0.166	0.361	0.130
8_yahen	0.095	0.021	0.092	0.024
9_zhehen	0.415	0.223	0.430	0.259
10_yaozhed	0.774	0.343	0.837	0.335

4.7. Sensitivity, Reliability, and Cross-Dataset Analysis

To further examine the robustness of the proposed method and clarify the effect of input resolution on detection performance, we additionally conducted an input-size sensitivity analysis and repeated experiments with multiple random seeds. These supplementary experiments help provide a more reliable assessment of the proposed model beyond a single-run comparison.

Table 10 reports the sensitivity analysis results under three input settings, i.e., 224×224 , 320×320 , and 640×640 . For the baseline YOLO11n, the performance varies only moderately with the change in input size. In contrast, the proposed YOLO11n + LCAF-D + Mish shows a clearer performance gain when the input resolution is increased. Under the 224×224 setting, the proposed model only achieves performance comparable to the baseline. However, when the input size is increased to 640×640 , the proposed model reaches 77.1% mAP@0.5 and 44.3% mAP@0.5:0.95, outperforming the baseline YOLO11n, which obtains 73.0% and 40.8%, respectively.

Table 10. Sensitivity analysis of input image size on the NEU-DET dataset.

Model	Input Size	Precision	Recall	mAP@0.5	mAP@0.5:0.95	Params (M)	GFLOPs	Inference Time (ms)
YOLO11n baseline	224×224	0.784	0.639	0.729	0.403	2.583	6.3	2.5
YOLO11n baseline	320×320	0.745	0.716	0.754	0.390	2.583	6.3	0.7
YOLO11n baseline	640×640	0.700	0.665	0.730	0.408	2.583	6.3	1.9
YOLO11n + LCAF-D + Mish	224×224	0.719	0.655	0.724	0.406	2.652	6.5	1.1
YOLO11n + LCAF-D + Mish	320×320	0.760	0.664	0.714	0.410	2.652	6.5	0.9
YOLO11n + LCAF-D + Mish	640×640	0.768	0.699	0.771	0.443	2.652	6.5	1.6

To reduce the uncertainty caused by single-run evaluation on a relatively small dataset, repeated experiments were further conducted at 640×640 . Specifically, YOLO11n is treated as the 2025 official YOLO SOTA baseline, whereas YOLO26n is included as the 2026 official YOLO SOTA baseline. YOLO26n was trained and evaluated using the same five random seeds as the previous repeated experiments, namely 0, 11, 42, 123, and 3407, under the same NEU-DET split, input size, optimizer settings, training schedule, and hardware platform. The corresponding five-seed comparison results are summarized in Table 11, which reports the mean and standard deviation over the five runs together with Params,

GFLOPs, and FPS measured under the same runtime benchmark protocol. Table 12 reports the mean and standard deviation over the five runs, together with Params, GFLOPs, and FPS measured under the same runtime benchmark protocol. YOLO26n obtains the best mAP@0.5:0.95, indicating stronger localization under stricter IoU thresholds, while the proposed YOLO11n + LCAF-D + Mish achieves the highest mAP@0.5 among the compared models. These results show that the proposed method remains competitive as a task-oriented lightweight refinement of YOLO11n, although it is not claimed to universally outperform the newer YOLO26n family.

Table 11. Five-seed comparison with 2025 and 2026 official YOLO SOTA baselines at 640×640 on the NEU-DET dataset.

Model	Params (M)	GFLOPs	FPS	Precision	Recall	mAP@0.5	mAP@0.5:0.95
YOLO11n baseline	2.583	6.3	75.54	0.720 ± 0.023	0.699 ± 0.029	0.754 ± 0.016	0.425 ± 0.013
YOLO26n baseline	2.506	5.8	68.17	0.732 ± 0.047	0.721 ± 0.016	0.754 ± 0.016	0.469 ± 0.007
YOLO11n + LCAF-D + Mish	2.652	6.5	70.70	0.738 ± 0.031	0.694 ± 0.032	0.761 ± 0.008	0.431 ± 0.013

Table 12. Cross-dataset comparison between YOLO11n and the proposed YOLO11n + LCAF-D + Mish on the GC10-DET dataset under the 640×640 setting.

Model	Precision	Recall	mAP@0.5	mAP@0.5:0.95	Params (M)	GFLOPs	Inference Time (ms)
YOLO11n baseline	0.821	0.500	0.575	0.304	2.584	6.3	1.4
YOLO11n + LCAF-D + Mish	0.639	0.635	0.636	0.325	2.654	6.5	1.8

To further examine cross-dataset performance under a unified higher-resolution setting, we additionally evaluated the baseline YOLO11n and the proposed YOLO11n + LCAF-D + Mish on the GC10-DET dataset at 640×640 . As shown in Table 12, the proposed model achieved 63.6% mAP@0.5 and 32.5% mAP@0.5:0.95, outperforming the baseline YOLO11n, which obtained 57.5% and 30.4%, respectively. A class-wise comparison further shows that the proposed model brings clear improvements for several categories, especially 2_hanfeng and 3_yueyawan, while some difficult categories, such as 8_yahen, remain challenging for both models. Compared with NEU-DET, GC10-DET contains a wider variety of textures, more complex morphology, and a larger domain gap between categories, so the cross-dataset gains remain category-dependent rather than uniform. Therefore, the revised manuscript presents the proposed method more cautiously as a competitive lightweight improvement with encouraging cross-dataset performance under the higher-resolution setting, rather than claiming universal superiority across all categories or settings.

5. Conclusions and Future Work

This study presented a lightweight enhancement strategy for steel surface defect detection based on YOLO11n. The final design combines an improved channel-aware fusion mechanism with a hierarchical Mish activation strategy so that cross-scale interaction and deep nonlinear representation are strengthened while model complexity remains low. The contribution should be interpreted as an incremental lightweight engineering refinement of YOLO11n rather than as a fundamentally new detector architecture.

The revised experiments also show that the effect of the proposed design depends strongly on input resolution. Under lower-resolution settings, the advantage over the baseline is limited. Under the unified 640×640 setting, however, YOLO11n + LCAF-D + Mish achieves better overall detection accuracy than YOLO11n on both NEU-DET and GC10-DET. The five-seed experiments further indicate slightly better mean mAP values together with stable variation under the higher-resolution protocol.

From an application perspective, the method remains practically attractive. The unified runtime benchmark shows that the revised model keeps competitive inference speed while

introducing only small increases in latency and GPU memory, suggesting a reasonable trade-off between accuracy and deployability for lightweight industrial inspection.

Several limitations should nevertheless be acknowledged. Public horizontal comparisons are still influenced by differences in input size, split strategy, augmentation policy, and training configuration, so the proposed method should not be interpreted as universally superior to all recently reported approaches. In addition, difficult categories such as crazing, rolled-in_scale, and 8_yahen on GC10-DET remain challenging. Finally, although FPS, latency, and GPU memory are now reported under a unified GPU environment, CPU- and edge-device deployment results are still absent. The current work is therefore more appropriately positioned as a competitive lightweight engineering improvement with validated module-placement and activation-design choices.

Future work will focus on stronger modeling for fine cracks and blurred-boundary defects, more efficient high-resolution deployment strategies for real industrial environments, and more robust cross-dataset generalization with better category-balanced representation learning.

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

Funding: This work was supported by the Humanities and Social Sciences Foundation of Ministry of Education of China “Research on Negative Public Opinion, Emotional Communication Mechanisms and Collaborative Governance Strategies in Ethnic Minority Areas of Western China in the Era of Social Media” (No. 25XJA860001), the Key Program of National Social Science Foundation of China (NSSFC, 23AGL039), and the National Natural Science Foundation of China (No. 22578366).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The datasets used in this study are publicly available. The NEU-DET and GC10-DET datasets can be obtained from their corresponding public sources. Processed annotations and implementation details are available from the corresponding author upon reasonable request.

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

References

1. Fang, X.; Luo, Q.; Zhou, B.; Li, C.; Tian, L. Research Progress of Automated Visual Surface Defect Detection for Industrial Metal Planar Materials. *Sensors* **2020**, *20*, 5136. [\[CrossRef\]](#) [\[PubMed\]](#)
2. Lv, X.; Duan, F.; Jiang, J.-J.; Fu, X.; Gan, L. Deep Metallic Surface Defect Detection: The New Benchmark and Detection Network. *Sensors* **2020**, *20*, 1562. [\[CrossRef\]](#) [\[PubMed\]](#)
3. He, L.; Zheng, L.; Xiong, J. FMV-YOLO: A Steel Surface Defect Detection Algorithm for Real-World Scenarios. *Electronics* **2025**, *14*, 1143. [\[CrossRef\]](#)
4. Xu, H.; Zhang, Z.; Ye, H.; Song, J.; Chen, Y. Efficient Steel Surface Defect Detection via a Lightweight YOLO Framework with Task-Specific Knowledge-Guided Optimization. *Electronics* **2025**, *14*, 2029. [\[CrossRef\]](#)
5. Wang, Z.; Zhao, L.; Li, H.; Xue, X.; Liu, H. Research on a Metal Surface Defect Detection Algorithm Based on DSL-YOLO. *Sensors* **2024**, *24*, 6268. [\[CrossRef\]](#) [\[PubMed\]](#)
6. Wang, F.; Jiang, X.; Han, Y.; Wu, L. YOLO-LSDI: An Enhanced Algorithm for Steel Surface Defect Detection Using a YOLOv11 Network. *Electronics* **2025**, *14*, 2576. [\[CrossRef\]](#)
7. Lv, B.; Duan, B.; Zhang, Y.; Li, S.; Wei, F.; Gong, S.; Ma, Q.; Cai, M. Research on Surface Defect Detection of Strip Steel Based on Improved YOLOv7. *Sensors* **2024**, *24*, 2667. [\[CrossRef\]](#) [\[PubMed\]](#)
8. Huang, B.; Liu, J.; Liu, X.; Liu, K.; Liao, X.; Li, K.; Wang, J. Improved YOLOv5 Network for Steel Surface Defect Detection. *Metals* **2023**, *13*, 1439. [\[CrossRef\]](#)

9. Zheng, A.; Jiang, X.; Liu, W. An Efficient Lightweight Method for Steel Surface Defect Detection. *Sensors* **2025**, *25*, 7527. [[CrossRef](#)] [[PubMed](#)]
10. Wu, Z.; Yang, L. EIM-YOLO: A Defect Detection Method for Metal-Painted Surfaces on Electrical Sealing Covers. *Appl. Sci.* **2025**, *15*, 9380. [[CrossRef](#)]
11. Zhao, Y.; Lv, W.; Xu, S.; Wei, J.; Wang, G.; Dang, Q.; Liu, Y.; Chen, J. DETRs Beat YOLOs on Real-Time Object Detection. *arXiv* **2023**, arXiv:2304.08069. [[CrossRef](#)]
12. Wang, A.; Chen, H.; Liu, L.; Chen, K.; Lin, Z.; Han, J.; Ding, G. YOLOv10: Real-Time End-to-End Object Detection. *arXiv* **2024**, arXiv:2405.14458. [[CrossRef](#)]
13. Woo, S.; Park, J.; Lee, J.-Y.; Kweon, I.S. CBAM: Convolutional Block Attention Module. *arXiv* **2018**, arXiv:1807.06521. [[CrossRef](#)]
14. Wang, Q.; Wu, B.; Zhu, P.; Li, P.; Zuo, W.; Hu, Q. ECA-Net: Efficient Channel Attention for Deep Convolutional Neural Networks. *arXiv* **2019**, arXiv:1910.03151. [[CrossRef](#)]
15. Lin, T.-Y.; Dollar, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature Pyramid Networks for Object Detection. *arXiv* **2016**, arXiv:1612.03144. [[CrossRef](#)]
16. Liu, S.; Qi, L.; Qin, H.; Shi, J.; Jia, J. Path Aggregation Network for Instance Segmentation. *arXiv* **2018**, arXiv:1803.01534. [[CrossRef](#)]
17. Yasir, S.M.; Ahn, H. Faster Metallic Surface Defect Detection Using Deep Learning with Channel Shuffling. *arXiv* **2024**, arXiv:2406.14582. [[CrossRef](#)]
18. Zuo, Z.; Dong, J.; Gao, Y.; Wu, Z. HyperDefect-YOLO: Enhance YOLO with HyperGraph Computation for Industrial Defect Detection. *arXiv* **2024**, arXiv:2412.03969. [[CrossRef](#)]
19. Sun, S.; Deng, M.; Yu, X.; Xi, X.; Zhao, L. Self-Adaptive Gamma Context-Aware SSM-based Model for Metal Defect Detection. *arXiv* **2025**, arXiv:2503.01234. [[CrossRef](#)]
20. Li, S.; Chen, S.; Geng, W.; Xu, Z.; Liu, X.; Dong, C.; Tian, Z.; Chen, C. Semi-Supervised Defect Detection via Conditional Diffusion and CLIP-Guided Noise Filtering. *arXiv* **2025**, arXiv:2507.05588. [[CrossRef](#)]
21. Mia, M.S.; Voban, A.A.B.; Arnob, A.B.H.; Naim, A.; Ahmed, M.K.; Islam, M.S. DANet: Enhancing Small Object Detection through an Efficient Deformable Attention Network. *arXiv* **2023**, arXiv:2310.05768. [[CrossRef](#)]
22. Somtochukwu, E.E.; Rijal, N. Real-Time Industrial Defect Detection on Edge Hardware Using Fine-Tuned YOLOv8: A Systematic Benchmark on the NEU Surface Defect Database and MVTec AD with Automotive and Battery Manufacturing Extensions. *arXiv* **2026**, arXiv:2606.07659. [[CrossRef](#)]
23. Kang, J.; Cen, Y.; Cen, Y.; Wang, K.; Liu, Y. CFIS-YOLO: A Lightweight Multi-Scale Fusion Network for Edge-Deployable Wood Defect Detection. *arXiv* **2025**, arXiv:2504.11305. [[CrossRef](#)]
24. Ang, L.; Rahim, S.K.N.A.; Hamzah, R.; Aminuddin, R.; Yousheng, G. YOLO Algorithm with Hybrid Attention Feature Pyramid Network for Solder Joint Defect Detection. *arXiv* **2024**, arXiv:2401.01214. [[CrossRef](#)]
25. Ultralytics. YOLO11 Documentation [EB/OL]. Available online: <https://docs.ultralytics.com/models/yolo11/> (accessed on 15 May 2026).
26. Ultralytics. Models Supported by Ultralytics [EB/OL]. Available online: <https://docs.ultralytics.com/models/> (accessed on 15 May 2026).
27. Ultralytics. YOLO11 vs. YOLOv8: A Comprehensive Technical Comparison [EB/OL]. Available online: <https://docs.ultralytics.com/compare/yolo11-vs-yolov8/> (accessed on 15 May 2026).

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.