

GLA-YOLO: A Lightweight YOLO-Based Network for Industrial Surface Defect Detection

Jiayi Li

University of Malaya

Norrima Binti Mokhtar

norrimamokhtar@um.edu.my

University of Malaya

Wenjunliang Zhang

University of Malaya

Muhammad Amirul Aiman Asri

University of Malaya

Qiyuan Qin

University of Malaya

Mahidzal Bin Dahari

University of Malaya

Yixuan Dai

University of Malaya

Bingxing Liu

Chinese Academy of Sciences

Article

Keywords: Industrial surface defect detection, Lightweight object detection, YOLO, Linear attention, Deep learning

Posted Date: October 30th, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-11179045/v1>

License:   This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: No competing interests reported.

GLA-YOLO: A Lightweight YOLO-Based Network for Industrial Surface Defect Detection

Jiayi Li¹, Norrima Binti Mokhtar^{1*}, Wenjunliang Zhang¹,
Muhammad Amirul Aiman Asri¹, Qiyuan Qin¹,
Mahidzal Bin Dahari¹, Yixuan Dai², Bingxing Liu^{3,4,5}

¹Faculty of Engineering, Universiti Malaya, Kuala Lumpur, Malaysia.

²Institute of Advanced Studies (IAS), Universiti Malaya, Kuala Lumpur, Malaysia.

³Institute of Software, Chinese Academy of Sciences, Beijing, China.

⁴University of Chinese Academy of Sciences, Beijing, China.

⁵DP Technology, Beijing, China.

*Corresponding author(s). E-mail(s): norrimamokhtar@um.edu.my;

Abstract

Surface defect detection is essential to quality assurance in modern manufacturing, affecting product reliability, production efficiency, and operational safety across diverse industrial sectors. Despite significant advances in deep learning-based object detection, industrial defect inspection remains uniquely challenging because defects exhibit extremely low contrast against textured backgrounds, highly irregular morphologies, severe scale variations, and marked class imbalance in which critical defect types may appear in fewer than twenty training samples. Deployment on resource-constrained edge devices in factory environments additionally demands real-time inference that many high-accuracy architectures cannot deliver. This paper introduces Gated Local Attention YOLO (GLA-YOLO), a lightweight architecture built upon YOLO11, featuring three innovations: Lightweight Fusion Convolution (LFConv) for efficient shallow feature extraction with channel attention, a Dynamic Local Attention Fusion Block (DLABlock) integrating local depthwise convolution with Performer-inspired linear attention, and DLAC3k2/C2DLA composite modules that embed DLABlock into the YOLO CSP framework. On a self-collected industrial defect dataset (4 categories, 1,367 instances) and the GC10-DET benchmark (10 categories), GLA-YOLO achieves mAP@50 of 90.21% and 80.59%, improving upon YOLO11n by 3.15 and 4.40 percentage points respectively, with only 2.70M parameters and 6.2 GFLOPs, while delivering 168.57 FPS inference and establishing state-of-the-art performance among nano-scale detectors.

Keywords: Industrial surface defect detection, Lightweight object detection, YOLO, Linear attention, Deep learning

1 Introduction

Surface defect detection is a critical component of quality assurance in modern manufacturing, directly influencing product reliability, production efficiency, and operational cost [1–3]. Across diverse industrial sectors including automotive manufacturing, aerospace engineering, electronics fabrication, and steel production, surface anomalies such as cracks, corrosion spalling, scuffing, and inclusions can compromise structural integrity, precipitate premature component failure, and incur substantial economic losses. Conventional defect inspection predominantly relies on manual visual examination conducted by trained operators, a methodology that is inherently subjective, labor-intensive, and susceptible to fatigue-induced inconsistencies in detection accuracy. Consequently, the development of automated visual inspection systems grounded in computer vision and deep learning has emerged as an imperative research direction, offering the prospect of consistent, high-throughput, and objective quality assessment [4, 5].

Despite the remarkable progress achieved by general-purpose object detectors on natural image benchmarks, industrial surface defect detection confronts a constellation of distinct challenges that remain inadequately addressed by existing methodologies. First, industrial defects frequently show very low contrast against textured or non-uniform background surfaces, rendering precise detection and localization intrinsically difficult. Second, defect morphologies show marked irregularity and unpredictability: a hairline crack may span merely a few pixels, whereas corrosion spalling can cover extensive contiguous regions, posing extreme scale diversity. Third, severe class imbalance is common in industrial datasets, where certain defect categories appear only sporadically during production, yielding severely limited training exemplars, a scenario commonly designated as few-shot or long-tailed defect detection. Fourth, many detection architectures that achieve high accuracy rely on computationally intensive backbone networks or quadratic-complexity attention mechanisms, rendering them unsuitable for deployment on resource-constrained edge devices ubiquitously situated on factory floors, where real-time inference throughput is an operational requirement [6].

The YOLO family of single-stage detectors [6–12] has become the predominant approach for real-time object detection through iterative architectural refinements, including multi-scale prediction, CSPNet-based backbones, decoupled detection heads, and anchor-free paradigms. The contemporary YOLO11 [12] incorporates C3k2 blocks and C2PSA position-sensitive attention. Its nano-scale variant (YOLO11n) delivers a mean average precision at an intersection-over-union (IoU) threshold of 0.5 (mAP@50) of 76.19% on GC10-DET, with 2.62M parameters and 6.6 GFLOPs (giga floating-point operations), constituting a strong baseline for lightweight industrial defect detection.

Notwithstanding these advances, a critical examination of the YOLO11 architecture reveals several salient opportunities for structural refinement. First, the standard convolutional layers populating the shallow stem process all channels through dense 3×3 kernels, incurring considerable computational overhead at early stages where feature map resolutions remain high (e.g., 320×320 and 160×160). Second, the C2PSA module deployed at the deepest backbone stage employs conventional multi-head self-attention with quadratic complexity $O(N^2)$ relative to spatial resolution, a design that

may prove over-parameterized for the abstract, semantically enriched features at this hierarchical level. Third, the standard C3k2 blocks predominantly engage local convolution operations and lack explicit architectural provisions for capturing long-range spatial dependencies, a capability that is essential for detecting expansive defects and modeling contextual interrelationships among spatially distributed defect regions.

Motivated by these observations, this paper introduces Gated Local Attention YOLO (GLA-YOLO), a lightweight and effective object detection architecture purposefully engineered for industrial surface defect detection. Anchored upon the YOLO11 framework, GLA-YOLO advances three principal architectural innovations:

(1) Lightweight Fusion Convolution (LFConv): replaces standard convolutions in the backbone stem and neck with a dual-branch design fusing standard and depthwise separable convolutions, augmented by channel attention [13], enriching shallow feature representations at reduced computational cost.

(2) Dynamic Local Attention Fusion Block (DLABlock): integrates local depthwise convolution, Performer-inspired linear attention [14] with $O(N)$ complexity, a learnable gated fusion mechanism, and a compact MLP for adaptive local-global feature learning.

(3) DLAC3k2 and C2DLA: composite modules embedding DLABlock into the YOLO Cross Stage Partial (CSP) [15] framework. DLAC3k2 augments C3k2 at the deepest backbone stage, while C2DLA replaces C2PSA post-SPPF with stacked DLABlocks.

Experiments on two datasets, namely a self-collected industrial defect dataset comprising 480 images of mechanical fitting parts and casting blanks (4 categories, 1,367 instances) and the public GC10-DET benchmark [16] (10 steel surface defect categories), demonstrate the following:

(a) GLA-YOLO attains mAP@50 of 90.21% and 80.59% on the two datasets, surpassing the YOLO11n baseline by 3.15 and 4.40 percentage points, respectively.

(b) Each proposed component contributes positively to detection accuracy, and their combination yields the best performance.

(c) GLA-YOLO delivers 168.57 frames per second (FPS) inference with a compact parameter budget of 2.70M and reduced 6.2 GFLOPs, which is faster than YOLO11n (160.79 FPS).

(d) Class Activation Map (CAM) [17] visualizations confirm that GLA-YOLO learns spatially precise, defect-focused feature representations with reduced background attention.

The remainder of this paper is organized as follows. Section 2 surveys related work in object detection and industrial surface defect detection. Section 3 describes the proposed GLA-YOLO architecture in detail, covering dataset construction, the LFConv module, the DLABlock and its composite variants, and the overall model configuration. Section 4 presents exhaustive experimental results, including comparisons with state-of-the-art methods, ablation studies, efficiency analyses, and qualitative visualizations. Section 5 discusses the interpretability, practical implications, and limitations of the proposed method. Section 6 concludes the paper and outlines prospective directions for future investigation.

2 Related Work

2.1 Object Detection

Object detection is a foundational task in computer vision, involving the simultaneous localization and classification of objects of interest within monocular images. Over the preceding decade, deep learning-based detection methodologies have achieved transformative progress and fall into two principal categories: two-stage detectors and single-stage detectors.

Two-stage detectors, epitomized by the R-CNN lineage [1–3], initially generate region proposals and subsequently perform per-proposal classification and bounding box regression. Faster R-CNN [3] introduced the Region Proposal Network (RPN) for end-to-end training; nevertheless, the computational overhead of the two-stage paradigm constrains real-time industrial applicability.

Single-stage detectors execute classification and localization within a unified forward pass, avoiding an explicit region proposal phase and achieving a more favorable speed–accuracy trade-off. SSD [4] and RetinaNet [5] pioneered dense prediction methodologies, with RetinaNet introducing Focal Loss to redress the imbalance between foreground and background classes. The YOLO series [6–12] has become dominant through successive architectural innovations including multi-scale prediction, CSPNet-based backbone design, decoupled detection heads with anchor-free paradigms, and NMS-free training via consistent dual label assignments.

Concurrently, transformer-based detection architectures have garnered substantial research attention. DETR [18] and its descendants, including Deformable DETR [19] and RT-DETR [20], reformulate detection as a set prediction problem mediated by transformer encoder–decoder structures. Though achieving competitive accuracy, transformer-based detectors typically demand considerably larger parameter budgets and computational resources relative to CNN-based counterparts, constraining their viability for resource-limited industrial edge deployment.

Lightweight network design is an active research frontier for enabling efficient inference on computationally constrained platforms. MobileNet [21, 22] pioneered depthwise separable convolutions, factorizing standard convolution into spatial depthwise and channel-wise pointwise operations for dramatic parameter and FLOPs reduction. ShuffleNet [23] introduced channel shuffling for cross-group information exchange, while EfficientNet [24] proposed compound scaling to uniformly balance depth, width, and resolution. GhostNet [25] demonstrated that redundant feature maps can be economically generated through inexpensive linear transformations. These design principles have been widely adopted in contemporary object detectors to reconcile real-time throughput with competitive accuracy.

Attention mechanisms have evolved into indispensable constituents of modern CNN architectures. The Squeeze-and-Excitation (SE) network [13] introduced channel-wise attention for adaptive feature recalibration, while CBAM [26] extended attention to both channel and spatial dimensions. The vision transformer (ViT) [27] popularized self-attention for vision tasks; nevertheless, the quadratic computational complexity $O(N^2)$ of canonical self-attention with respect to input token count N poses formidable challenges for dense prediction applications. Efficient attention variants, most notably

the Performer architecture [14], mitigate this complexity to $O(N)$ through kernel-based approximations of the softmax attention operator, where the attention computation is reordered to avoid explicit construction of the $N \times N$ attention matrix. These theoretical advances directly inform the linear attention mechanism adopted within the proposed DLABlock.

2.2 Industrial Surface Defect Detection

Industrial surface defect detection endeavors to autonomously identify and spatially localize anomalies on manufactured product surfaces, fulfilling an indispensable function in quality control workflows across diverse manufacturing verticals. Relative to general object detection, the industrial defect detection task presents a distinctive ensemble of challenges: low-contrast defects against heterogeneous textural backgrounds, irregular and unpredictable morphological configurations, extreme scale variations spanning orders of magnitude, and severe long-tailed class distributions in which certain defect types appear only rarely [28–31].

Classical methodologies predominantly relied upon hand-engineered feature descriptors such as Local Binary Patterns (LBP) [32] and Histograms of Oriented Gradients (HOG) [33], coupled with traditional classifiers such as Support Vector Machines. These approaches, however, necessitate extensive domain-specific expertise and show limited cross-domain generalization. The advent of deep learning has largely supplanted these conventional pipelines.

A corpus of specialized deep architectures has been proposed for industrial defect detection. Liu et al. [28] developed a cascaded detection framework incorporating multi-scale feature fusion for steel surface anomalies. Wang et al. [29] introduced an attention-guided architecture integrating channel-wise and spatial attention for fine-grained defect discrimination. Dong et al. [30] engineered a pyramid feature fusion strategy for defects spanning disparate scales in printed circuit board inspection. Luo et al. [31] provided a comprehensive survey of deep learning approaches for flat steel surface inspection. More recently, lightweight defect detectors built upon improved YOLO variants have been reported for steel surface inspection, including an improved YOLOv7 with depthwise-separable modules [34], an improved YOLOv8 algorithm [35], and a YOLOv11-based small-defect framework with enhanced feature fusion [36].

Public benchmark datasets have been instrumental in catalyzing progress within this domain. The NEU-DET dataset [37] furnishes six categories of hot-rolled steel strip surface defects and has served as a canonical evaluation benchmark. GC10-DET [16] is a much more challenging and realistic dataset, covering ten categories of steel surface defects acquired from authentic production environments, showing marked variations in defect scale, textural properties, morphological configuration, and visual appearance. The heightened complexity of GC10-DET imposes elevated demands on the feature extraction and spatial localization proficiencies of detection models.

Despite these substantive advances, several gaps remain in the existing literature. First, numerous specialized defect detection architectures are tailored to specific defect typologies or product categories, lacking generalization capability for heterogeneous industrial deployment contexts. Second, approaches attaining high detection

accuracy frequently rely on computationally heavy backbone architectures or complex post-processing pipelines, rendering them impractical for real-time deployment on resource-constrained edge devices. Third, the challenge of detecting rare defect categories under few-shot conditions, where training exemplars may number fewer than twenty instances, remains inadequately addressed. GLA-YOLO is designed to address these limitations through the combination of lightweight architectural design with computationally efficient attention mechanisms, achieving elevated detection accuracy across heterogeneous defect typologies while preserving real-time inference throughput within a compact model footprint.

3 Proposed Method

3.1 Data

3.1.1 Self-Collected Industrial Defect Dataset

To rigorously evaluate the efficacy and generalization capacity of the proposed methodology under ecologically valid industrial conditions, a dedicated industrial defect dataset was constructed through direct acquisition from mechanical fitting components and casting blanks within an operational manufacturing environment. The dataset contains 480 high-resolution monocular images acquired at a native resolution of 3072×4096 pixels, collectively containing 1,367 manually annotated defect instances distributed across four defect categories: CORROSION SPALLING, CRACK, SCUFFING, and UNDERCUT. The detailed per-category statistics are catalogued in Table 1. Note that a single image may contain defects of several categories, so the per-category image counts do not sum to the total number of images.

Table 1 Detailed statistics of the self-collected industrial defect dataset.

Class	Instances	Images
CORROSION SPALLING	171	93
CRACK	19	11
SCUFFING	449	225
UNDERCUT	728	217
Total	1367	480

Notably, the CRACK category comprises merely 19 annotated instances across 11 images, forming a typical few-shot defect detection scenario that introduces severe class imbalance. Furthermore, some defects have blurred boundaries, reduced contrast against adjacent surface regions, and highly irregular morphological configurations, collectively amplifying the inherent difficulty of accurate defect localization and fine-grained classification.

All instances were manually annotated with tight bounding boxes by trained annotators to ensure inter-annotator consistency and labeling fidelity. To allow an objective assessment of the intrinsic feature extraction and defect detection capabilities of the

proposed architectural modifications, no auxiliary complex data augmentation strategies beyond the standard YOLO training augmentation pipeline were introduced during the training phase.

3.1.2 GC10-DET Dataset

In addition to the self-collected dataset, experiments were conducted on the publicly available GC10-DET benchmark [16] to further substantiate the generalization efficacy of the proposed methodology across diverse industrial domains. GC10-DET is a widely adopted industrial surface defect detection benchmark, containing a curated collection of steel surface defects acquired from authentic rolling mill production environments. The dataset shows marked heterogeneity in defect scale, textural composition, morphological characteristics, and photometric appearance. Relative to general-purpose object detection benchmarks, GC10-DET is more challenging owing to complex background textures, low-contrast defect boundaries, irregular morphologies, and substantial intra-class diversity, placing high demands on feature extraction and spatial localization of detection architectures. The key characteristics of GC10-DET are summarized in Table 2.

Table 2 Summary of the GC10-DET dataset.

Property	Description
Total images	2,294
Image resolution	2048 × 1000 pixels
Color space	Grayscale
Annotation format	Bounding boxes (YOLO format)
Defect categories	10 classes
Category names	crescent_gap crease silk_spot water_spot welding_line inclusion oil_spot rolled_pit punching_hole waist_folding

3.2 Overall Architecture

The architecture of GLA-YOLO is shown in Figure 1. GLA-YOLO is constructed upon the YOLO11 detection framework [12] and introduces three principal architectural innovations: (i) the LFConv module, which supplants standard convolutions in the backbone stem and neck for enriched multi-branch shallow feature extraction; (ii) DLABlock, which serves as the fundamental computational unit for feature learning augmented by attention; and (iii) the DLAC3k2 and C2DLA composite modules, which seamlessly integrate DLABlock into the YOLO CSP-style backbone-neck topology.

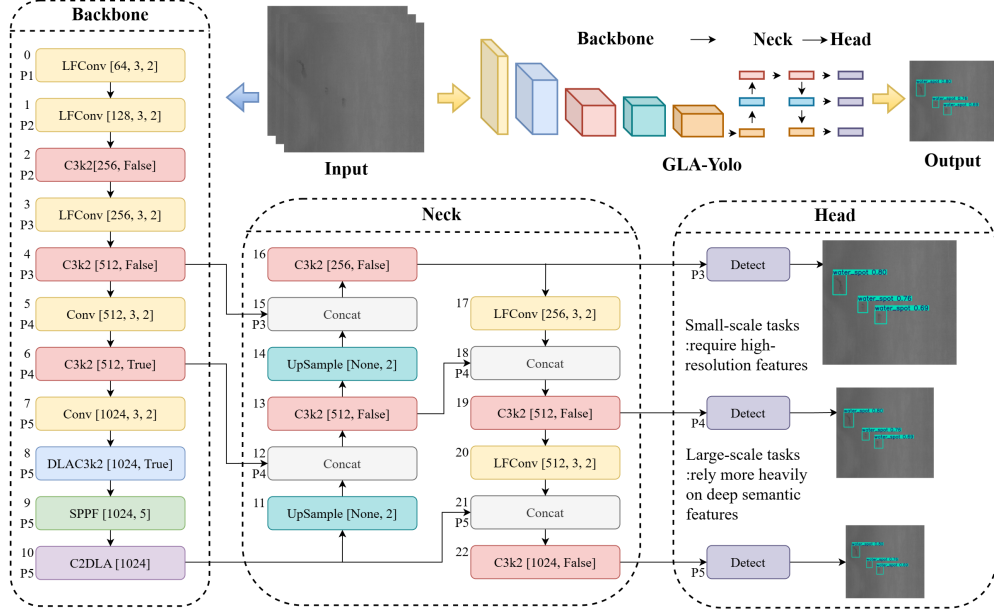


Fig. 1 Overall architecture of GLA-YOLO. The model integrates LFCnv for efficient shallow feature extraction, DLAC3k2 and C2DLA for local and global contextual interaction via lightweight linear attention and gated fusion, and multi-scale detection at P3, P4, and P5 output scales.

Given an input image $I \in \mathbb{R}^{3 \times H \times W}$, the backbone network performs hierarchical feature extraction through a cascade of spatial downsampling stages, generating multi-scale feature maps $\{P3, P4, P5\}$ at strides 8, 16, and 32 pixels, respectively. These feature maps are subsequently fused through a bidirectional Feature Pyramid Network (FPN) [38] neck comprising top-down and bottom-up pathways, yielding the enriched multi-scale representations $\{F3, F4, F5\}$. Finally, a decoupled detection head operates on each fused scale to predict bounding box regression parameters and per-category classification scores.

Backbone. The stem of the backbone replaces the first two standard convolutional layers (P1/2 and P2/4 stages) with LFCnv modules (Section 3.3). Each LFCnv employs a parallel dual-branch design that combines standard convolution with depth-wise separable convolution, enabling diverse feature extraction from high-resolution inputs while curtailing computational redundancy. At the deepest backbone stage (P5/32), the standard C3k2 block is supplanted by a DLAC3k2 module (Section 3.4.5), which allocates DLABlock-based attention-guided feature learning to one CSP branch, giving the network the ability to capture long-range spatial dependencies among high-level semantic features. The SPPF module is retained for efficient multi-scale receptive field aggregation and is succeeded by a C2DLA module (Section 3.4.5) that replaces the original C2PSA with stacked DLABlocks, using linear attention for position-sensitive feature refinement at reduced computational cost.

Neck. The neck implements a Feature Pyramid Network topology [38]. Top-down pathways upsample higher-level semantic features via nearest-neighbor interpolation

and fuse them with corresponding backbone features through channel-wise concatenation. Bottom-up pathways employ LFConv-based downsampling to propagate spatially precise localization signals to deeper feature maps. C3k2 blocks serve as lateral connection transformation modules.

Detection Head. A decoupled detection head operates on the three output scales (P3/8, P4/16, P5/32), with independent branches for bounding box regression and class probability estimation, facilitating improved training convergence through task-specific gradient pathways.

3.3 LFConv

LFConv is specifically engineered to supplant standard convolutional layers in the stem and neck of the YOLO11 baseline with a computationally economical yet representationally expressive alternative. Standard convolutions in early network stages process all input channels through dense 3×3 kernels, a design that introduces considerable parametric and computational redundancy at shallow layers where spatial dimensions remain large. As schematically illustrated in Figure 2, LFConv adopts a parallel dual-branch architecture that combines complementary feature extraction pathways, followed by channel-wise attention-based adaptive recalibration.

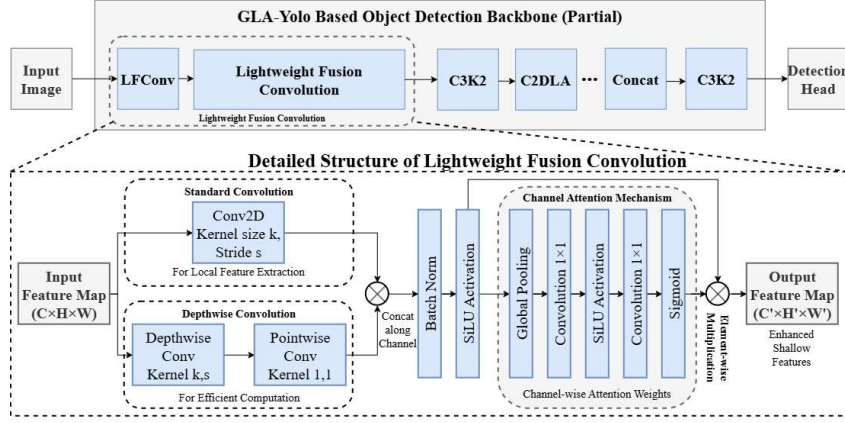


Fig. 2 Architecture of the proposed LFConv. Branch-1 performs standard 3×3 convolution; Branch-2 executes depthwise separable convolution (3×3 depthwise + 1×1 pointwise). Concatenated features undergo batch normalization, SiLU activation, and lightweight channel attention recalibration.

3.3.1 Dual-Branch Feature Extraction

The first branch (Branch-1) executes a standard 3×3 convolution with stride s , reducing the channel dimensionality from C_{in} to $C_{out}/2$ while preserving the receptive field characteristics of a conventional convolutional layer. The output of this branch is formally expressed as:

$$X_{std} = W_{std} * X, \quad W_{std} \in \mathbb{R}^{C_{out}/2 \times C_{in} \times 3 \times 3} \quad (1)$$

where $*$ denotes the standard convolution operator. This branch captures dense inter-channel interactions within a localized 3×3 spatial neighborhood, providing a rich foundation of feature representations.

The second branch (Branch-2) implements a depthwise separable convolution [21], which factorizes the standard convolution into two sequential operations: a 3×3 depthwise convolution operating independently on each input channel, followed by a 1×1 pointwise convolution that projects the resulting features to the target channel dimension:

$$X_{\text{dw}} = W_{\text{dw}} \otimes X, \quad W_{\text{dw}} \in \mathbb{R}^{C_{\text{in}} \times 1 \times 3 \times 3} \quad (2)$$

$$X_{\text{pw}} = W_{\text{pw}} * X_{\text{dw}}, \quad W_{\text{pw}} \in \mathbb{R}^{C_{\text{out}}/2 \times C_{\text{in}} \times 1 \times 1} \quad (3)$$

where $\otimes$ signifies the depthwise (channel-wise grouped) convolution operator. The depthwise separable factorization dramatically reduces the parameter count relative to an equivalent standard convolution. Specifically, for a convolution kernel of size k with input channels C_{in} and output channels $C_{\text{out}}/2$, the parameter complexity ratio is:

$$R_{\text{param}} = \frac{k^2 \cdot C_{\text{in}} + C_{\text{in}} \cdot C_{\text{out}}/2}{k^2 \cdot C_{\text{in}} \cdot C_{\text{out}}/2} = \frac{1}{C_{\text{out}}/2} + \frac{1}{k^2} \quad (4)$$

For the canonical configuration with $k = 3$ and $C_{\text{out}} = 64$ at the P1/2 stage, this yields $R_{\text{param}} \approx 0.14$, corresponding to an approximately 86% reduction in parameters for Branch-2 relative to a standard convolution of equivalent dimensionality.

The outputs of both branches are concatenated along the channel dimension and subsequently processed by batch normalization and SiLU activation:

$$F_{\text{fuse}} = \text{SiLU}(\text{BN}([X_{\text{std}}; X_{\text{pw}}])) \quad (5)$$

where $[\cdot; \cdot]$ denotes channel-wise concatenation, $\text{BN}(\cdot)$ represents batch normalization, and $\text{SiLU}(\cdot)$ is the SiLU (Sigmoid Linear Unit) activation function, $\text{SiLU}(x) = x \sigma(x)$.

3.3.2 Channel Attention Recalibration

To adaptively emphasize informative feature channels while suppressing less discriminative ones, a lightweight SE-style channel attention mechanism [13] is applied to the fused multi-branch features. The attention module operates through a bottleneck architecture:

$$s = \text{GAP}(F_{\text{fuse}}), \quad s \in \mathbb{R}^{C_{\text{out}}} \quad (6)$$

$$A = \sigma(W_2 \text{SiLU}(W_1 s)), \quad W_1 \in \mathbb{R}^{C_{\text{out}}/r \times C_{\text{out}}}, \quad W_2 \in \mathbb{R}^{C_{\text{out}} \times C_{\text{out}}/r} \quad (7)$$

where $\text{GAP}(\cdot)$ denotes global average pooling aggregating spatial information into a channel descriptor, $\sigma(\cdot)$ is the sigmoid gating function, $\text{SiLU}(\cdot)$ is the SiLU activation, and $r = 8$ is the channel reduction ratio. The final recalibrated output is obtained via channel-wise multiplication:

$$Y_{LF} = A \odot F_{\text{fuse}} \quad (8)$$

where $\odot$ denotes channel-wise broadcast multiplication. The overall parametric overhead introduced by the attention module is $\sum |W_1| + \sum |W_2| = (2/r) \cdot C_{\text{out}}^2$. For $r = 8$, this amounts to merely $0.25 \cdot C_{\text{out}}^2$ additional parameters, which is negligible relative to the convolutional parameters of the preceding branches.

3.4 DLABlock

DLABlock and its composite variants (DLAC3k2 and C2DLA), illustrated in Figure 3, form the core computational units of the proposed attention modules in GLA-YOLO. DLABlock integrates four complementary processing pathways to address specific limitations of conventional convolutional feature extraction for industrial defect detection: (i) a local depthwise convolution branch for fine-grained spatial detail preservation, (ii) a Performer-inspired linear attention mechanism (DWConvAttention) for long-range contextual modeling with linear complexity, (iii) a learnable gated fusion mechanism for adaptive integration of local and global features, and (iv) a CompactMLP for efficient non-linear feature transformation.

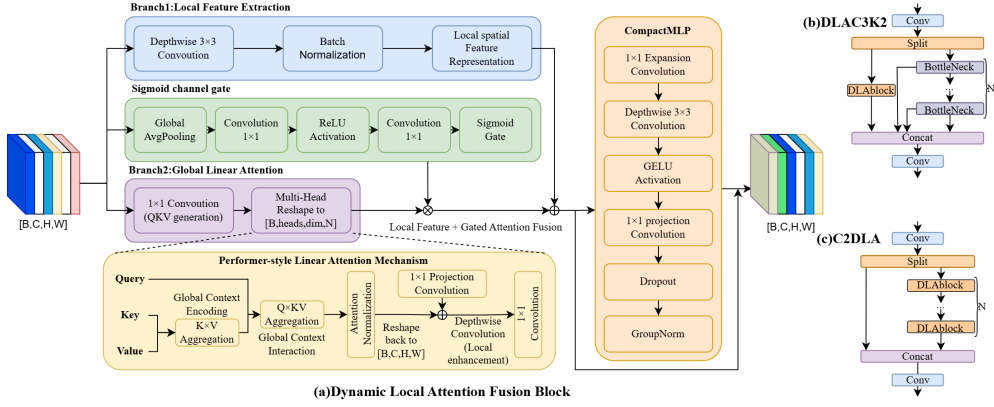


Fig. 3 Architecture of the proposed DLABlock and its composite modules DLAC3k2 and C2DLA. (a) DLABlock integrates local depthwise convolution, DWConvAttention with linear-complexity global context modeling, learnable gated fusion for adaptive feature integration, and CompactMLP for efficient post-fusion transformation. (b) DLAC3k2 embeds DLABlock into the C3k2 CSP framework at the P5/32 backbone stage. (c) C2DLA replaces C2PSA following SPPF with stacked DLABlocks for position-sensitive multi-scale feature refinement.

3.4.1 Local Convolution Branch

The local convolution branch applies a 3×3 depthwise convolution followed by batch normalization:

$$X_{\text{local}} = \text{BN}(W_{\text{local}} \otimes X), \quad W_{\text{local}} \in \mathbb{R}^{C \times 1 \times 3 \times 3} \quad (9)$$

This lightweight branch preserves edge discontinuities, textural variations, and fine boundary information that is critical for detecting small-scale defects (e.g., CRACK instances comprising merely 19 training exemplars) and low-contrast defect boundaries against complex industrial backgrounds.

3.4.2 DWConvAttention: Linear Attention for Spatial Context

To capture long-range spatial dependencies without incurring the quadratic computational complexity $O(N^2)$ of standard scaled dot-product self-attention [27], DLABlock incorporates DWConvAttention, a linear attention module that follows the linear-attention paradigm of fast autoregressive transformers [39] and is inspired by the Performer architecture [14]. Given an input feature map $X \in \mathbb{R}^{C \times H \times W}$, the query (Q), key (K), and value (V) projections are generated through a single 1×1 convolution:

$$[Q, K, V] = W_{\text{qkv}} * X, \quad W_{\text{qkv}} \in \mathbb{R}^{3C \times C \times 1 \times 1} \quad (10)$$

$$Q, K, V \in \mathbb{R}^{C \times H \times W} \longrightarrow \mathbb{R}^{N_h \times N \times d_h} \quad (11)$$

where each of Q , K , and V is reshaped into N_h attention heads, each comprising $N = H \times W$ spatial tokens of dimension $d_h = C/N_h$. A ReLU-based kernel feature map is then applied element-wise to the query and key tensors. This positive-definite feature map enables the attention mechanism to be decomposed via the associative property of matrix multiplication, reducing the computational complexity from $O(N^2)$ to $O(N)$:

$$\phi(x) = \max(0, x) + \varepsilon \quad (12)$$

where $\varepsilon = 1 \times 10^{-6}$ is a small constant that keeps the kernel values strictly positive.

$$\text{Attention}(Q, K, V) = \frac{\phi(Q) (\phi(K)^\top V)}{\phi(Q) (\phi(K)^\top \mathbf{1}) + \varepsilon} \quad (13)$$

where $\mathbf{1} \in \mathbb{R}^N$ denotes the all-ones vector. The numerator $\phi(Q)(\phi(K)^\top V)$ computes weighted value aggregation without explicitly materializing the $N \times N$ attention matrix, while the denominator normalizes the output via the inverse of the summed kernelized similarities. This formulation, derived from the Performer’s FAVOR+ mechanism [14], preserves the expressiveness of attention-based global context aggregation while maintaining linear asymptotic complexity.

A residual depthwise convolution branch operates in parallel on the original input features to inject local positional information, mitigating the restricted spatial inductive bias inherent in purely attention-based architectures. The final linear attention output is:

$$X_{\text{attn}} = \text{BN}(\text{DWConvAttention}(X) + W_{\text{res}} \otimes X) \quad (14)$$

3.4.3 Gated Adaptive Fusion

A learnable gating mechanism adaptively modulates the relative contributions from the local convolution and linear attention pathways. The gate is generated through a squeeze-and-excitation style bottleneck applied to the original input features:

$$g = \sigma(W_{g2} \text{ReLU}(W_{g1} \text{GAP}(X))), \quad W_{g1} \in \mathbb{R}^{C/r \times C}, \quad W_{g2} \in \mathbb{R}^{C \times C/r} \quad (15)$$

with reduction ratio $r = 4$. The gating vector $g \in \mathbb{R}^{C \times 1 \times 1}$ comprises channel-wise scalar coefficients in the interval $(0, 1)$. The fused feature representation is computed as:

$$F_{\text{gated}} = X_{\text{local}} + g \odot X_{\text{attn}} \quad (16)$$

This dynamic gating mechanism endows each channel with the capacity to autonomously determine the relative weighting of local versus global feature contributions contingent upon the input content. The gated features are subsequently projected through a 1×1 convolution:

$$F_{\text{proj}} = W_{\text{proj}} * F_{\text{gated}}, \quad W_{\text{proj}} \in \mathbb{R}^{C \times C \times 1 \times 1} \quad (17)$$

3.4.4 CompactMLP

The post-fusion features undergo further non-linear transformation via a CompactMLP module with an expansion ratio of 2, designed to provide additional representational capacity while maintaining parametric efficiency:

$$F_{\text{exp}} = W_{\text{fc1}} * F_{\text{proj}}, \quad W_{\text{fc1}} \in \mathbb{R}^{2C \times C \times 1 \times 1} \quad (18)$$

$$F_{\text{spatial}} = W_{\text{spatial}} \otimes F_{\text{exp}}, \quad W_{\text{spatial}} \in \mathbb{R}^{2C \times 1 \times 3 \times 3} \quad (19)$$

$$F_{\text{mlp}} = \text{Dropout}(W_{\text{fc2}} * \text{GELU}(F_{\text{spatial}}), p = 0.05), \quad W_{\text{fc2}} \in \mathbb{R}^{C \times 2C \times 1 \times 1} \quad (20)$$

$$Y_{\text{DLA}} = F_{\text{proj}} + \text{GroupNorm}(F_{\text{mlp}}) \quad (21)$$

The CompactMLP comprises a 1×1 expansion convolution (Eq. 18), a 3×3 depthwise convolution for spatial feature mixing (Eq. 19), GELU activation with a lightweight dropout regularization ($p = 0.05$, Eq. 20), and a residual connection with GroupNorm regularization (Eq. 21). Relative to conventional feed-forward networks employing expansion ratios of 4 [27], the CompactMLP with expansion ratio 2 reduces the parametric overhead by approximately 50% while maintaining adequate non-linear representational capacity for the feature transformations required after attention fusion.

3.4.5 DLAC3k2 and C2DLA Integration

DLABlock is incorporated into the YOLO detection architecture through two composite modules that conform to the CSP [15] design philosophy:

DLAC3k2. This module supplants the standard C3k2 block at the P5/32 backbone stage. Consistent with the CSP paradigm, input features with C_{in} channels are transformed by a 1×1 convolution to $2 \cdot C_{\text{hidden}}$ channels and partitioned into two equal branches of dimensionality C_{hidden} . The first branch is processed by a DLABlock for feature learning augmented by attention, while the second branch traverses a cascade of n standard Bottleneck blocks [12] for conventional convolutional feature extraction. The output of the DLABlock branch is concatenated with the second branch and with the outputs of all Bottleneck stages along the channel dimension, and the result is fused through a terminal 1×1 convolution. Formally:

$$Y_{\text{DLAC3k2}} = \text{Conv}_{1 \times 1}(\text{Concat}[\text{DLABlock}(F_1); F_2; \text{Bottleneck}(F_2); \dots; \text{Bottleneck}^n(F_2)]) \quad (22)$$

C2DLA. This module replaces the C2PSA module following the SPPF layer. Input features are partitioned through a 1×1 convolution into two equal channel groups: the first group is passed through unaltered (identity mapping), while the second group undergoes processing by a sequence of n stacked DLABlock modules. The outputs are concatenated and projected back to the original channel dimension:

$$Y_{\text{C2DLA}} = \text{Conv}_{1 \times 1}(\text{Concat}[F_1; \text{DLABlock}^n(F_2)]) \quad (23)$$

This design preserves the CSP gradient flow characteristics while substituting the standard multi-head self-attention in PSA with the computationally efficient linear attention of DLABlock, achieving a superior accuracy–efficiency equilibrium.

3.5 Model Configuration and Parameter Analysis

To ensure equitable comparison and to attribute performance improvements exclusively to architectural innovation rather than model scaling, GLA-YOLO adopts the identical compound scaling configuration as YOLO11n (nano-scale variant): a depth scaling factor $d = 0.50$, a width scaling factor $w = 0.25$, and a maximum channel constraint $C_{\text{max}} = 1024$. The input resolution is standardized at 640×640 pixels for both training and inference phases.

Table 3 catalogues the principal architectural divergences between the baseline YOLO11n and the proposed GLA-YOLO, accompanied by their corresponding model complexity metrics. The total parameter count P_{total} and floating-point operations G_{total} are computed as:

$$P_{\text{total}} = \sum_{l \in L} P_l, \quad P_l = k_l^2 \cdot C_{\text{in},l} \cdot C_{\text{out},l} \quad (24)$$

$$G_{\text{total}} = \sum_{l \in L} G_l, \quad G_l = 2 \cdot h_l \cdot w_l \cdot P_l \quad (25)$$

where L denotes the set of all convolutional layers, and h_l, w_l are the spatial dimensions of the output feature map at layer l . For depthwise separable convolutions, the parameter count is modified to $P_l = k_l^2 \cdot C_{\text{in},l} + C_{\text{in},l} \cdot C_{\text{out},l}$.

The structural modifications in GLA-YOLO are motivated by the following architectural considerations. First, the shallow stem layers (P1/2 and P2/4) operate on high-resolution feature maps (320×320 and 160×160 , respectively), where computational cost is most sensitive to channel dimensionality. By substituting standard convolutions with the dual-branch LFCnv, the model achieves diversified feature extraction at these performance-critical early stages without proportional escalation of the parameter budget. Second, the deepest backbone stage (P5/32) processes feature maps at the coarsest spatial resolution (20×20), providing an opportune location for introducing attention mechanisms: the computational overhead scales linearly with spatial extent, while the abstract semantic features at this depth benefit most from global contextual integration. The DLAC3k2 module at this juncture endows the network with the capacity to capture long-range dependencies between features. Third, the C2DLA module following SPPF further refines the aggregated multi-scale representations prior to neck propagation, ensuring that the feature representations passed to the detection head have both semantic richness and spatial precision.

Table 3 Comparison of architectural configuration and model complexity between YOLO11n and the proposed GLA-YOLO.

Component	YOLO11n	GLA-YOLO (Ours)
Stem (P1/2, P2/4)	Conv (3×3)	LFCnv (3×3 dual-branch)
Backbone P3/8 downsampling	Conv (3×3)	LFCnv (3×3 dual-branch)
Backbone P5/32	C3k2	DLAC3k2 (C3k2 + DLABlock)
Post-SPPF	C2PSA (multi-head self-attention)	C2DLA (stacked DLABlock)
Neck downsampling	Conv (3×3)	LFCnv (3×3 dual-branch)
Parameters (M)	2.62	2.70
GFLOPs	6.6	6.2

4 Experiments and Results

4.1 Experimental Setup

4.1.1 Datasets

Two datasets were used for evaluation: the self-collected industrial defect dataset (480 images, 4 categories, 1,367 instances) and the GC10-DET benchmark [16] (10 steel surface defect categories). Detailed dataset characterizations are provided in Section 3.1.

4.1.2 Evaluation Metrics

Model performance was assessed using the standard COCO evaluation protocol. The primary quantitative metrics were mAP@50 and mAP@50:95, the latter averaged over

IoU thresholds from 0.5 to 0.95 in steps of 0.05; Precision and Recall were additionally reported as further diagnostics. Computational efficiency was quantified through parameter count (millions, M), GFLOPs, and inference throughput (FPS), measured with PyTorch and ONNX Runtime on a single NVIDIA GeForce RTX 5090 GPU.

4.1.3 Implementation Details

All models were implemented in PyTorch [40] and trained on a single NVIDIA GeForce RTX 5090 GPU. Input images were resized to 640×640 pixels. Optimization employed Stochastic Gradient Descent (SGD) with Nesterov momentum $\beta = 0.937$, weight decay $\lambda = 5 \times 10^{-4}$, and an initial learning rate $\eta_0 = 1 \times 10^{-2}$ decaying via cosine annealing. The batch size was 16. The standard YOLO augmentation pipeline was applied, comprising Mosaic augmentation, random horizontal flipping, and HSV color space jittering. All models were trained for 300 epochs on the self-collected dataset and 200 epochs on GC10-DET from randomly initialized weights to preclude pretraining bias. Both datasets were partitioned at the image level into training and validation subsets in a 7:3 ratio with a fixed random seed, and all reported metrics were computed on the validation subsets. To account for training stochasticity, every detector compared in the main performance evaluations (Tables 4 and 5) was trained three times with different random seeds, and the reported accuracy metrics are the mean $\pm$ standard deviation over these runs; the precision–recall curves and qualitative visualizations were produced from the run closest to the mean performance. Ablation variants (Tables 6 and 7) were likewise trained three times. In the efficiency comparison (Table 8), the accuracy columns reproduce the mean $\pm$ standard deviation of Tables 4 and 5, whereas throughput, parameter count, and GFLOPs were obtained from a single representative run.

4.2 Results on Self-Collected Dataset

Table 4 presents the quantitative comparison between GLA-YOLO and state-of-the-art nano-scale detectors, including YOLOv5n [9], YOLOv8n [41], and YOLO26 [42], together with RT-DETR-18 as a substantially larger reference model, on the self-collected industrial defect dataset. GLA-YOLO attains the highest mAP@50 of 90.21% and mAP@50:95 of 62.75% among all compared detectors.

Relative to the YOLO11n baseline, GLA-YOLO improves mAP@50 by 3.15 percentage points (87.06% to 90.21%) and mAP@50:95 by 4.94 percentage points (57.81% to 62.75%), with a marginal parameter increment of 0.08M (+3.1%) and a GFLOPs reduction of 0.4 (−6.1%). Notably, GLA-YOLO surpasses the much larger RT-DETR-18 model [20] by 2.74 percentage points in mAP@50 despite RT-DETR-18 requiring $7.4\times$ more parameters (20.00M vs. 2.70M) and $9.7\times$ more GFLOPs (60.0 vs. 6.2). The per-category precision–recall curves are shown in Figure 4.

4.3 Results on GC10-DET Dataset

Table 5 summarizes the comparison results on the GC10-DET benchmark. GLA-YOLO achieves mAP@50 of 80.59% and mAP@50:95 of 47.99%, establishing the highest performance among all evaluated nano-scale detectors.

Table 4 Performance comparison on the self-collected industrial defect dataset (mean $\pm$ standard deviation over three training runs). Best accuracy is highlighted in bold.

Method	Recall (%)	Precision (%)	mAP@50 (%)	mAP@50:95 (%)	Params (M)	GFLOPs
YOLOv5n	80.47 $\pm$ 0.62	82.76 $\pm$ 0.55	83.39 $\pm$ 0.51	54.21 $\pm$ 0.57	2.65	7.8
YOLOv8n	81.58 $\pm$ 0.58	83.71 $\pm$ 0.49	83.54 $\pm$ 0.46	55.31 $\pm$ 0.52	3.16	8.9
YOLOv10n	83.41 $\pm$ 0.51	84.73 $\pm$ 0.47	85.67 $\pm$ 0.43	56.97 $\pm$ 0.48	2.78	8.7
YOLO11n	84.48 $\pm$ 0.45	86.74 $\pm$ 0.39	87.06 $\pm$ 0.37	57.81 $\pm$ 0.41	2.62	6.6
YOLO12n	84.22 $\pm$ 0.49	86.38 $\pm$ 0.43	86.41 $\pm$ 0.41	58.29 $\pm$ 0.39	2.60	6.7
RT-DETR-18	85.91 $\pm$ 0.64	87.18 $\pm$ 0.56	87.47 $\pm$ 0.52	60.26 $\pm$ 0.59	20.00	60.0
YOLO26	83.87 $\pm$ 0.55	86.29 $\pm$ 0.48	86.02 $\pm$ 0.45	59.08 $\pm$ 0.50	2.57	6.1
GLA-YOLO (Ours)	88.02 $\pm$ 0.39	87.14 $\pm$ 0.46	90.21 $\pm$ 0.36	62.75 $\pm$ 0.41	2.70	6.2

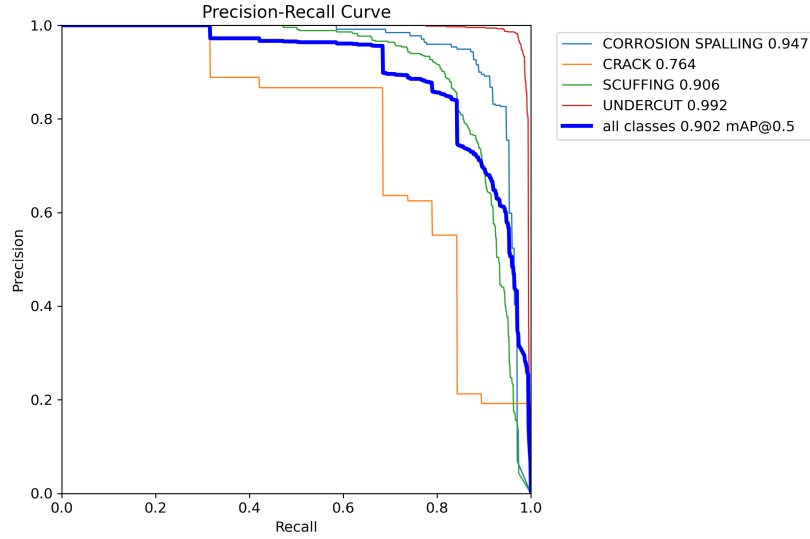


Fig. 4 Precision–Recall curves of GLA-YOLO on the self-collected industrial defect dataset for each defect category.

Compared to YOLO11n, GLA-YOLO improves mAP@50 by 4.40 percentage points (76.19% to 80.59%) and mAP@50:95 by 4.23 percentage points (43.76% to 47.99%). The recall improves from 74.03% to 78.37% (+4.34 percentage points), indicating enhanced defect coverage, while precision improves from 76.48% to 79.82% (+3.34 percentage points), reflecting reduced false positive incidence. The corresponding precision–recall curves are shown in Figure 5.

Table 5 Performance comparison on the GC10-DET dataset (mean $\pm$ standard deviation over three training runs). Best accuracy is highlighted in bold.

Method	Recall (%)	Precision (%)	mAP@50 (%)	mAP@50:95 (%)	Params (M)	GFLOPs
YOLOv5n	70.08 $\pm$ 0.84	72.61 $\pm$ 0.69	70.41 $\pm$ 0.73	38.11 $\pm$ 0.61	2.65	7.8
YOLOv8n	72.31 $\pm$ 0.62	75.26 $\pm$ 0.77	73.89 $\pm$ 0.56	40.32 $\pm$ 0.68	3.16	8.9
YOLOv10n	72.81 $\pm$ 0.79	74.93 $\pm$ 0.65	74.91 $\pm$ 0.63	40.94 $\pm$ 0.52	2.78	8.7
YOLO11n	74.03 $\pm$ 0.58	76.48 $\pm$ 0.71	76.19 $\pm$ 0.47	43.76 $\pm$ 0.56	2.62	6.6
YOLO12n	74.67 $\pm$ 0.67	76.09 $\pm$ 0.55	76.38 $\pm$ 0.62	44.27 $\pm$ 0.49	2.60	6.7
RT-DETR-18	76.71 $\pm$ 0.73	77.94 $\pm$ 0.61	77.68 $\pm$ 0.57	46.16 $\pm$ 0.64	20.00	60.0
YOLO26	74.18 $\pm$ 0.64	76.81 $\pm$ 0.59	76.54 $\pm$ 0.53	44.08 $\pm$ 0.58	2.57	6.1
GLA-YOLO (Ours)	78.37 $\pm$ 0.43	79.82 $\pm$ 0.41	80.59 $\pm$ 0.38	47.99 $\pm$ 0.42	2.70	6.2

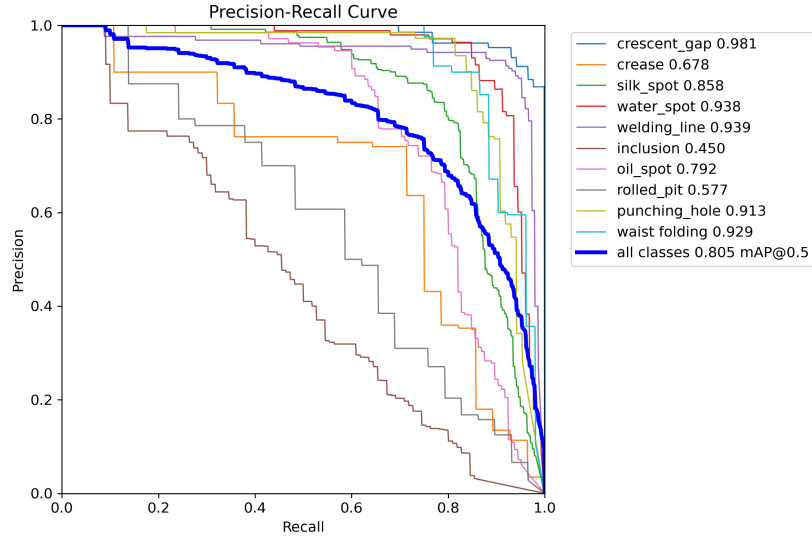


Fig. 5 Precision-Recall curves of GLA-YOLO on the GC10-DET dataset.

4.4 Ablation Studies

Systematic ablation experiments were conducted on both datasets to isolate the individual contribution of each proposed architectural component. For direct comparability, the baseline rows of Tables 6 and 7 reuse the same YOLO11n results obtained over three runs reported in Tables 4 and 5, respectively.

4.4.1 Ablation on Self-Collected Dataset

The ablation results on the self-collected dataset are reported in Table 6. Each component contributes positively, and the complete GLA-YOLO achieves the optimal configuration, improving over the baseline by +3.15% mAP@50 and +4.94%

Table 6 Ablation study on the self-collected industrial defect dataset (mean $\pm$ standard deviation over three training runs).

Variant	LFConv	DLAC3k2	C2DLA	mAP@50 (%)	mAP@50:95 (%)	Params (M)	GFLOPs
Baseline (YOLO11n)	×	×	×	87.06 $\pm$ 0.37	57.81 $\pm$ 0.41	2.62	6.6
+ LFConv	✓	×	×	88.51 $\pm$ 0.53	58.61 $\pm$ 0.48	2.53	6.2
+ DLAC3k2	×	✓	×	89.08 $\pm$ 0.57	60.41 $\pm$ 0.51	2.75	6.7
+ C2DLA	×	×	✓	88.34 $\pm$ 0.46	59.38 $\pm$ 0.54	2.66	6.6
GLA-YOLO (All)	✓	✓	✓	90.21 $\pm$ 0.36	62.75 $\pm$ 0.41	2.70	6.2

mAP@50:95. The elevated absolute mAP values reflect the higher native image resolution and domain-specific defect characteristics.

4.4.2 Ablation on GC10-DET

Table 7 Ablation study on the GC10-DET dataset (mean $\pm$ standard deviation over three training runs).

Variant	LFConv	DLAC3k2	C2DLA	mAP@50 (%)	mAP@50:95 (%)	Params (M)	GFLOPs
Baseline (YOLO11n)	×	×	×	76.19 $\pm$ 0.47	43.76 $\pm$ 0.56	2.62	6.6
+ LFConv	✓	×	×	77.76 $\pm$ 0.47	44.39 $\pm$ 0.55	2.53	6.2
+ DLAC3k2	×	✓	×	79.12 $\pm$ 0.61	46.38 $\pm$ 0.49	2.75	6.7
+ C2DLA	×	×	✓	78.54 $\pm$ 0.54	45.41 $\pm$ 0.52	2.66	6.6
GLA-YOLO (All)	✓	✓	✓	80.59 $\pm$ 0.38	47.99 $\pm$ 0.42	2.70	6.2

The ablation results on GC10-DET clarify the contribution of each component: (1) LFConv (+1.57% mAP@50): replacing standard convolutions with LFConv in the stem and neck raises detection accuracy while simultaneously reducing both parameters (2.62M to 2.53M) and GFLOPs (6.6 to 6.2), corroborating the representational efficiency of the dual-branch design with channel attention. (2) DLAC3k2 (+2.93% mAP@50): integrating DLABlock into the C3k2 block at the deepest backbone stage furnishes the largest individual gain, underscoring the importance of long-range dependency modeling for abstract semantic features at P5/32. The mAP@50:95 improvement of 2.62 percentage points is particularly salient, indicating enhanced localization precision across a spectrum of IoU thresholds. (3) C2DLA (+2.35% mAP@50): substituting C2PSA with C2DLA after SPPF yields consistent gains, validating that linear attention can effectively supplant standard multi-head self-attention with comparable or superior representational quality at reduced computational cost. (4) When all three components are synergistically combined, GLA-YOLO achieves a

combined improvement of +4.40% mAP@50 and +4.23% mAP@50:95 over the baseline, with the final model adding merely 0.08M additional parameters and 0.4 fewer GFLOPs. The same ordering of component contributions is observed on the self-collected dataset, substantiating the robustness and generalization capacity of the proposed design principles.

4.5 Efficiency Analysis

Table 8 Efficiency comparison of GLA-YOLO with existing nano-scale detectors. Accuracy columns reproduce the mean $\pm$ standard deviation of Tables 4 and 5; throughput, parameter count, and GFLOPs are measured from a single representative run. Best values within the listed detectors are highlighted in bold, namely the highest throughput and mAP@50 and the lowest parameter count and GFLOPs.

Method	Torch FPS	ONNX FPS	Self mAP@50 (%)	GC10-DET mAP@50 (%)	Params (M)	GFLOPs
YOLOv5n	181.43	252.18	83.39 $\pm$ 0.51	70.41 $\pm$ 0.73	2.65	7.8
YOLOv8n	202.77	289.96	83.54 $\pm$ 0.46	73.89 $\pm$ 0.56	3.16	8.9
YOLOv10n	118.62	159.13	85.67 $\pm$ 0.43	74.91 $\pm$ 0.63	2.78	8.7
YOLO11n	160.79	220.28	87.06 $\pm$ 0.37	76.19 $\pm$ 0.47	2.62	6.6
YOLO12n	110.15	153.65	86.41 $\pm$ 0.41	76.38 $\pm$ 0.62	2.60	6.7
GLA-YOLO (Ours)	168.57	235.32	90.21 $\pm$ 0.36	80.59 $\pm$ 0.38	2.70	6.2

Table 8 presents a comparison of efficiency and accuracy. GLA-YOLO delivers 168.57 FPS (PyTorch) and 235.32 FPS (ONNX), surpassing YOLO11n (160.79 / 220.28 FPS) while achieving higher accuracy on both datasets. This simultaneous improvement in both accuracy and inference speed is attributable to the architectural efficiency of the proposed modules: the dual-branch LFConv with depthwise separable factorization reduces the effective parameter count per convolutional layer, and the linear attention mechanism in DLBlock circumvents the quadratic complexity of canonical self-attention, maintaining computational tractability at full feature resolution. Relative to YOLOv10n [11], GLA-YOLO is 42.1% faster (168.57 vs. 118.62 Torch FPS) and achieves 5.68 percentage points higher mAP@50 on GC10-DET. Against YOLO12n [43], GLA-YOLO is 53.0% faster (168.57 vs. 110.15 Torch FPS) with a 4.21 percentage point advantage.

4.6 Qualitative Detection Comparison

To complement the quantitative evaluation, representative detection results are visualized to compare the predictive behavior of GLA-YOLO against baseline detectors under challenging industrial imaging conditions.

4.6.1 Detection Analysis on Self-Collected Dataset

Figure 6 presents a detection result comparison on representative samples from the self-collected industrial defect dataset, juxtaposing GLA-YOLO against RT-DETR-18 and YOLO11n. GLA-YOLO consistently produces higher confidence scores and demonstrates superior detection capability on small objects and overlapping targets. RT-DETR-18, despite its much larger parameter budget (20.00M), yields detection confidence scores lower than GLA-YOLO. YOLO11n misses small defect instances, particularly in cluttered and heterogeneous background regions. These qualitative observations corroborate the quantitative improvements reported in Table 4 and substantiate the efficacy of the proposed architectural innovations, specifically the gated fusion mechanism in DLABlock for fine-grained localization and the dual-branch LFCnv for robust shallow feature extraction.

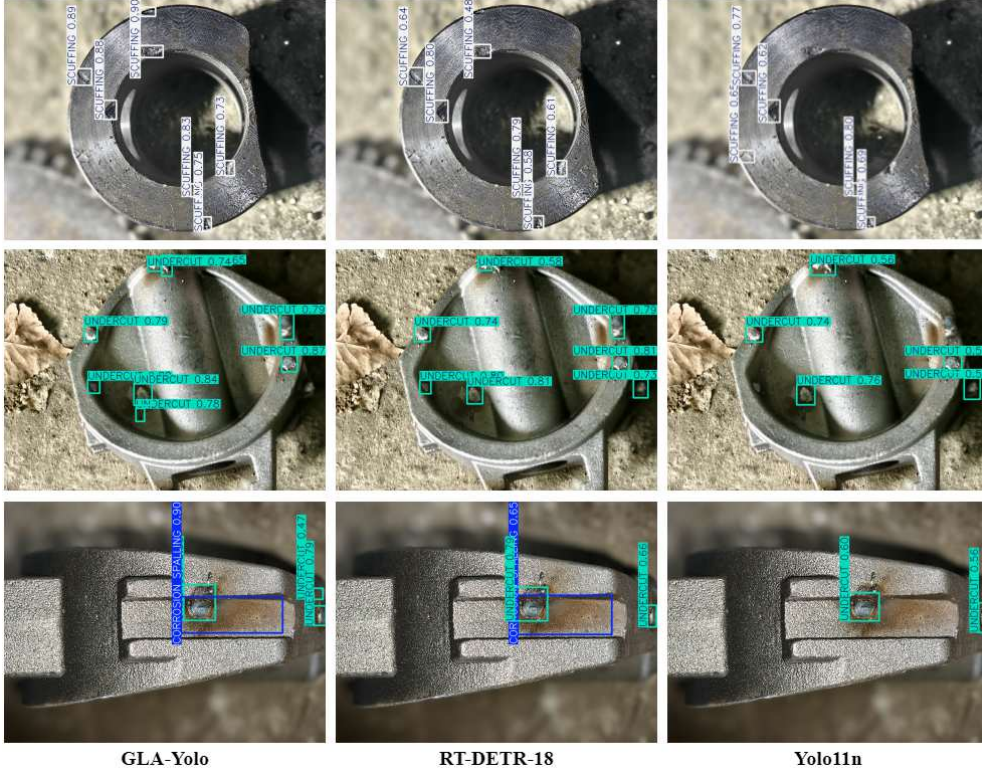


Fig. 6 Detection result comparison on representative samples from the self-collected industrial defect dataset, comparing GLA-YOLO (left column), RT-DETR-18 (middle column), and YOLO11n (right column).

4.6.2 Detection Analysis on GC10-DET Dataset

Figure 7 presents detection result comparisons on representative samples from the GC10-DET benchmark. GLA-YOLO detects all categories with elevated confidence scores across diverse defect categories. In contrast, both RT-DETR-18 and YOLO11n miss detections on the welding line category, a challenging defect type characterized by subtle morphological cues and low contrast against the surrounding steel surface. GLA-YOLO’s superior detection recall on welding line defects is attributable to the linear attention mechanism in DLABlock, which effectively captures long-range contextual dependencies spanning the elongated defect boundaries, enabling reliable discrimination from background texture patterns.

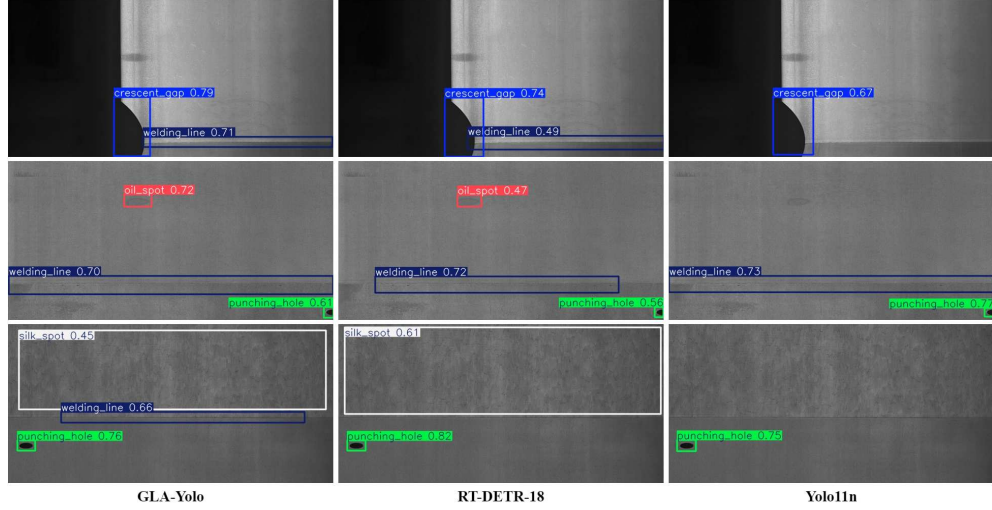


Fig. 7 Detection result comparison on representative samples from the GC10-DET dataset, comparing GLA-YOLO (left column), RT-DETR-18 (middle column), and YOLO11n (right column).

4.7 CAM Visualization

To furnish qualitative insight into the attentional mechanisms underpinning GLA-YOLO’s detection efficacy, CAM [17] visualizations were generated for representative defect exemplars and compared across models. CAM visualizations expose the spatial regions that exert the greatest influence on the model’s predictions, rendering the implicit decision process interpretable.

Figure 8 presents a CAM visualization covering four representative scenarios: GLA-YOLO on GC10-DET (first row), YOLO11n on GC10-DET (second row), GLA-YOLO on the self-collected dataset (third row), and YOLO11n on the self-collected dataset (fourth row). Each sample is accompanied by four visualization modalities: the original image, the YOLO detection output with bounding boxes and confidence scores, the grayscale CAM heatmap, and the CAM overlay on the original image. Across both datasets, GLA-YOLO consistently produces higher detection confidence

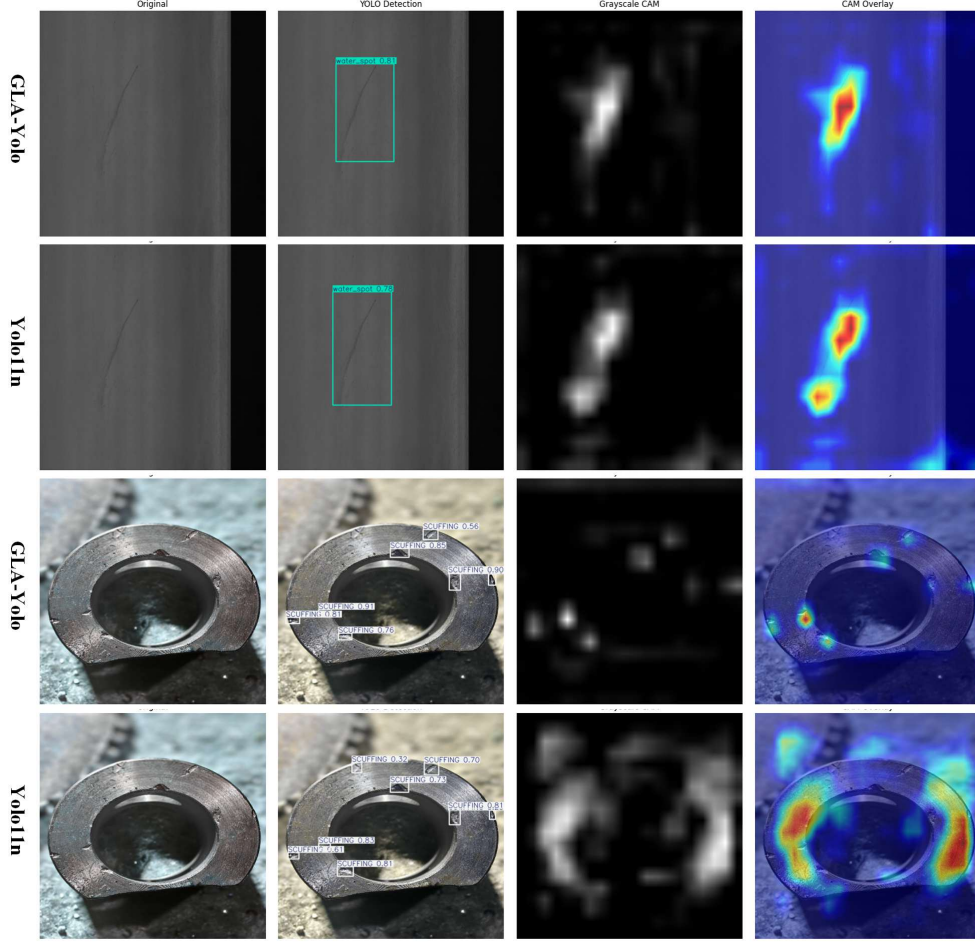


Fig. 8 CAM visualization comparison between GLA-YOLO and YOLO11n across both datasets. Four columns per sample: original image, YOLO detection output, grayscale CAM heatmap, and CAM overlay. Four rows: GLA-YOLO on GC10-DET (first row), YOLO11n on GC10-DET (second row), GLA-YOLO on self-collected dataset (third row), YOLO11n on self-collected dataset (fourth row).

scores and more spatially precise CAM activation regions compared to YOLO11n. On the self-collected dataset, for the CRACK category, the most challenging few-shot scenario with merely 19 training instances, GLA-YOLO generates markedly more compact and spatially accurate activation regions concentrated on the actual defect loci, whereas YOLO11n shows marked spatial diffusion of attention toward non-defect background regions. This enhanced localization precision is attributable to the gated fusion mechanism in DLABlock, which adaptively suppresses irrelevant textural features. On GC10-DET, GLA-YOLO’s CAM attention maps cover defect regions more completely and coherently, with reduced spurious background activation, confirming that the linear attention mechanism in DLABlock effectively captures long-range

contextual dependencies and distinguishes subtle defect features from complex background patterns. These qualitative findings provide mechanistic interpretability for the quantitative improvements reported across both datasets.

4.8 Summary of Experimental Findings

The experimental evaluation yields the following findings: (1) GLA-YOLO surpasses existing nano-scale detectors on both the self-collected industrial defect dataset (90.21% mAP@50 [+3.15%]) and GC10-DET (80.59% mAP@50 [+4.40%]), establishing new state-of-the-art performance. (2) Each proposed component (LFConv, DLAC3k2, and C2DLA) contributes positively, with DLAC3k2 providing the largest individual gain, confirming the criticality of long-range dependency modeling for industrial defect detection. (3) The integration of all three components attains the best accuracy; the individual gains partially overlap across components, so the combined improvement is lower than the naive sum of the gains of the individual components, while preserving a compact model profile (2.70M parameters, 6.2 GFLOPs). (4) GLA-YOLO achieves a Pareto-optimal trade-off between accuracy and efficiency, being simultaneously more accurate and faster than YOLO11n. (5) CAM visualizations confirm that GLA-YOLO learns more discriminative, spatially precise feature representations with reduced background attention, providing mechanistic interpretability for the quantitative improvements.

5 Discussion

The experimental evidence indicates that the three proposed components act on complementary aspects of industrial defect detection. DLAC3k2 yields the largest isolated gains on both datasets (+2.93% mAP@50 on GC10-DET), corroborating the premise that long-range spatial context is instrumental for discriminating extended, low-contrast defects such as welding lines from textured steel backgrounds. LFConv improves detection accuracy on GC10-DET while simultaneously reducing parameters (2.62M to 2.53M when applied alone) and GFLOPs, indicating that the dual-branch factorization principally removes redundant computation in the shallow, high-resolution stages rather than merely reallocating representational capacity. The C2DLA results demonstrate that Performer-inspired linear attention can substitute for conventional multi-head self-attention with comparable or superior accuracy at reduced cost. The CAM analyses provide convergent evidence: GLA-YOLO concentrates activation on the defect loci, including the few-shot CRACK category with only 19 training instances, while suppressing irrelevant background response, whereas YOLO11n shows more diffuse background attention.

One caveat tempers the interpretation of the efficiency results: all comparisons were conducted at the nano scale with 640×640 inputs, and whether the linear-attention components retain their advantage in accuracy and efficiency in larger backbones or at higher input resolutions remains untested.

Several limitations of this study should be acknowledged. The self-collected dataset comprises 480 images from a single manufacturing environment and only four defect categories, and the CRACK category contains merely 19 instances; the statistical

power for per-category conclusions is consequently limited, and the few-shot findings should be corroborated on larger defect corpora. The generalization of GLA-YOLO beyond metallic surface inspection, for example to textile, ceramic, or electronic component manufacturing, has not yet been evaluated. Finally, inference throughput was measured on a server-grade GPU rather than on resource-constrained embedded platforms, so the real-time claims of this study should be interpreted within this measurement context.

6 Conclusion

This paper has presented GLA-YOLO, a lightweight object detection architecture purposefully engineered to address the distinctive challenges characterizing industrial surface defect detection. Building upon the YOLO11 framework, GLA-YOLO introduces three complementary architectural innovations: LFConv for efficient dual-branch shallow feature extraction with channel attention recalibration (Equations 1–8); DLABlock, which synergistically integrates local depthwise convolution (Equation 9), Performer-inspired linear attention (Equations 10–14), learnable gated fusion (Equations 15–17), and a compact MLP (Equations 18–21); and the DLAC3k2 and C2DLA composite modules (Equations 22–23) that seamlessly embed DLABlock into the YOLO CSP-style backbone–neck–head detection pipeline.

Extensive empirical evaluation on a self-collected industrial defect dataset (mechanical fitting parts and casting blanks, 4 defect categories, 1,367 instances) and the publicly available GC10-DET benchmark (10 steel surface defect categories) demonstrates that GLA-YOLO consistently outperforms all compared nano-scale detectors. On the self-collected dataset, GLA-YOLO attains an mAP@50 of 90.21%, improving upon the YOLO11n baseline by 3.15 percentage points and surpassing the significantly larger RT-DETR-18 (20.00M parameters) by 2.74 percentage points. On GC10-DET, GLA-YOLO achieves an mAP@50 of 80.59%, a 4.40 percentage point improvement over YOLO11n. Ablation studies confirm that each proposed module contributes positively, with DLAC3k2 providing the largest individual gain (+2.93% on GC10-DET), and that the full model yields the best accuracy overall.

Importantly, GLA-YOLO achieves these accuracy improvements while operating within a compact parameter budget of 2.70M (+3.1% relative to baseline) and a reduced computational footprint of 6.2 GFLOPs (−6.1%). Moreover, GLA-YOLO delivers 168.57 FPS (PyTorch) inference throughput, surpassing YOLO11n (160.79 FPS) and establishing a Pareto-optimal balance of accuracy and efficiency. CAM visualizations provide mechanistic interpretability, confirming that GLA-YOLO learns more spatially focused and discriminative feature representations.

Several directions for future research are identified. First, extending GLA-YOLO to rotated bounding box detection would enable more natural parameterization of oriented defects prevalent in industrial surfaces. Second, the integration of semi-supervised and self-supervised pre-training strategies could mitigate the pervasive data scarcity challenge in industrial settings, particularly for rare defect categories with severely limited annotations. Third, exploring neural architecture search (NAS) and knowledge distillation techniques may yield further parametric and computational

efficiencies, enabling deployment on ultra-low-power embedded platforms. Fourth, extending the framework to pixel-level instance segmentation would furnish richer quality assessment information, including defect area, perimeter, and morphology descriptors, for downstream manufacturing process optimization and statistical process control. Fifth, evaluating GLA-YOLO’s cross-domain generalization on additional industrial benchmarks spanning diverse manufacturing verticals (e.g., textile inspection, ceramic surface analysis, and electronic component quality control) would further validate the universality of the proposed architectural design principles.

Data availability. The GC10-DET dataset used in this study is publicly available from its original repository (<https://github.com/lvxiaoming2019/GC10-DET-Metallic-Surface-Defect-Datasets>). The self-collected industrial defect dataset described in this study is not publicly available; it is available from the corresponding author on reasonable request.

Code availability. The source code and model configurations of GLA-YOLO will be made publicly available on GitHub upon acceptance of this manuscript.

Funding. This research received no specific grant from any funding agency in the public, commercial or not-for-profit sectors.

Competing interests. The authors declare no competing interests.

Author contributions. Jiayi Li conceived and designed the study, implemented the method, and conducted the experiments. Norrima Binti Mokhtar supervised the research and coordinated the project. Yixuan Dai and Bingxing Liu contributed to the experimental design and analysis. Qiyuan Qin, Wenjunliang Zhang, and Muhammad Amirul Aiman Asri supported the dataset preparation and validation. Mahidzal Bin Dahari contributed to the industrial application context. All authors reviewed and approved the final manuscript.

References

- [1] Girshick, R., Donahue, J., Darrell, T. & Malik, J. *Rich feature hierarchies for accurate object detection and semantic segmentation*. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 580–587 (IEEE, 2014).
- [2] Girshick, R. *Fast r-cnn*. Proceedings of the IEEE International Conference on Computer Vision (ICCV), 1440–1448 (IEEE, 2015).
- [3] Ren, S., He, K., Girshick, R. & Sun, J. *Faster r-cnn: Towards real-time object detection with region proposal networks*. Advances in Neural Information Processing Systems (NeurIPS), 91–99 (Curran Associates, Inc., 2015).
- [4] Liu, W. *et al. Ssd: Single shot multibox detector*. Proceedings of the European Conference on Computer Vision (ECCV), 21–37 (Springer, 2016).

- [5] Lin, T.-Y., Goyal, P., Girshick, R., He, K. & Dollár, P. *Focal loss for dense object detection*. Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2980–2988 (IEEE, 2017).
- [6] Redmon, J., Divvala, S., Girshick, R. & Farhadi, A. *You only look once: Unified, real-time object detection*. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 779–788 (IEEE, 2016).
- [7] Redmon, J. & Farhadi, A. Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767 (2018).
- [8] Bochkovskiy, A., Wang, C.-Y. & Liao, H.-Y. M. Yolov4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934 (2020).
- [9] Jocher, G. Ultralytics YOLOv5. GitHub: <https://github.com/ultralytics/yolov5> (2020).
- [10] Wang, C.-Y., Bochkovskiy, A. & Liao, H.-Y. M. *Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors*. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 7464–7475 (IEEE, 2023).
- [11] Wang, A. *et al.* *Yolov10: Real-time end-to-end object detection*. Advances in Neural Information Processing Systems (NeurIPS) (Curran Associates, Inc., 2024).
- [12] Jocher, G. & Qiu, J. Ultralytics YOLO11. GitHub: <https://github.com/ultralytics/ultralytics> (2024).
- [13] Hu, J., Shen, L. & Sun, G. *Squeeze-and-excitation networks*. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 7132–7141 (IEEE, 2018).
- [14] Choromanski, K. *et al.* *Rethinking attention with performers*. International Conference on Learning Representations (ICLR) (International Conference on Learning Representations, 2021).
- [15] Wang, C.-Y. *et al.* *CSPNet: A new backbone that can enhance learning capability of CNN*. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 390–391 (IEEE, 2020).
- [16] Lv, X., Duan, F., Jiang, J.-J., Fu, X. & Gan, L. Deep metallic surface defect detection: The new benchmark and detection network. *Sensors* **20**, 1562 (2020).
- [17] Zhou, B., Khosla, A., Lapedriza, A., Oliva, A. & Torralba, A. *Learning deep features for discriminative localization*. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2921–2929 (IEEE, 2016).

- [18] Carion, N. *et al.* *End-to-end object detection with transformers*. Proceedings of the European Conference on Computer Vision (ECCV), 213–229 (Springer, 2020).
- [19] Zhu, X. *et al.* *Deformable DETR: Deformable transformers for end-to-end object detection*. International Conference on Learning Representations (ICLR) (International Conference on Learning Representations, 2021).
- [20] Zhao, Y. *et al.* *DETRs beat YOLOs on real-time object detection*. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 16965–16974 (IEEE, 2024).
- [21] Howard, A. G. *et al.* MobileNets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861 (2017).
- [22] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A. & Chen, L.-C. *MobileNetV2: Inverted residuals and linear bottlenecks*. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 4510–4520 (IEEE, 2018).
- [23] Zhang, X., Zhou, X., Lin, M. & Sun, J. *ShuffleNet: An extremely efficient convolutional neural network for mobile devices*. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 6848–6856 (IEEE, 2018).
- [24] Tan, M. & Le, Q. V. *EfficientNet: Rethinking model scaling for convolutional neural networks*. Proceedings of the International Conference on Machine Learning (ICML), 6105–6114 (PMLR, 2019).
- [25] Han, K. *et al.* *GhostNet: More features from cheap operations*. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 1580–1589 (IEEE, 2020).
- [26] Woo, S., Park, J., Lee, J.-Y. & Kweon, I. S. *CBAM: Convolutional block attention module*. Proceedings of the European Conference on Computer Vision (ECCV), 3–19 (Springer, 2018).
- [27] Dosovitskiy, A. *et al.* *An image is worth 16x16 words: Transformers for image recognition at scale*. International Conference on Learning Representations (ICLR) (International Conference on Learning Representations, 2021).
- [28] Liu, Y., Xu, J. & Wu, Y. A cascaded learning framework for steel surface defect detection. *IEEE Transactions on Instrumentation and Measurement* **71**, 1–12 (2022).
- [29] Wang, J., Xu, P., Li, L. & Zhang, F. Assessing the attention of deep learning networks for surface defect recognition. *IEEE Transactions on Automation Science and Engineering* **20**, 1275–1286 (2023).
- [30] Dong, H. *et al.* PGA-Net: Pyramid feature fusion and global context attention network for automated surface defect detection. *IEEE Transactions on Industrial*

- Informatics* **16**, 7448–7458 (2020).
- [31] Luo, Q., Fang, X., Liu, L., Yang, C. & Sun, Y. Automated visual defect detection for flat steel surface: A survey. *IEEE Transactions on Instrumentation and Measurement* **69**, 626–644 (2020).
 - [32] Ojala, T., Pietikäinen, M. & Harwood, D. A comparative study of texture measures with classification based on featured distributions. *Pattern Recognition* **29**, 51–59 (1996).
 - [33] Dalal, N. & Triggs, B. *Histograms of oriented gradients for human detection*. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 886–893 (IEEE, 2005).
 - [34] Lu, J., Yu, M. & Liu, J. Lightweight strip steel defect detection algorithm based on improved YOLOv7. *Scientific Reports* **14**, 13267 (2024).
 - [35] Ma, S., Zhao, X., Wan, L., Zhang, Y. & Gao, H. A lightweight algorithm for steel surface defect detection using improved YOLOv8. *Scientific Reports* **15**, 8966 (2025).
 - [36] Wang, Y. *et al.* A lightweight YOLOv11-based framework for small steel defect detection with a newly enhanced feature fusion module. *Scientific Reports* **15**, 34322 (2025).
 - [37] Song, K. & Yan, Y. A noise robust method based on completed local binary patterns for hot-rolled steel strip surface defects. *Applied Surface Science* **285**, 858–864 (2013).
 - [38] Lin, T.-Y. *et al.* *Feature pyramid networks for object detection*. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2117–2125 (IEEE, 2017).
 - [39] Katharopoulos, A., Vyas, A., Pappas, N. & Fleuret, F. *Transformers are RNNs: Fast autoregressive transformers with linear attention*. Proceedings of the International Conference on Machine Learning (ICML), 5156–5165 (PMLR, 2020).
 - [40] Paszke, A. *et al.* *PyTorch: An imperative style, high-performance deep learning library*. Advances in Neural Information Processing Systems (NeurIPS), 8024–8035 (Curran Associates, Inc., 2019).
 - [41] Jocher, G. Ultralytics YOLOv8. GitHub: <https://github.com/ultralytics/ultralytics> (2023).
 - [42] Jocher, G. *et al.* Ultralytics YOLO26: Unified real-time end-to-end vision models. arXiv preprint arXiv:2606.03748 (2026).
 - [43] Ultralytics. YOLO12. GitHub: <https://github.com/ultralytics/ultralytics> (2025).