Scientific Reports

https://doi.org/10.1038/s41598-026-46074-z

Article in Press

Hierarchical depth aware YOLO for efficient metal surface defect detection

Received: 22 January 2026

Accepted: 24 March 2026

Published online: 04 April 2026

Cite this article as: Qin Q., Khairuddin A.S.M., Idros N. *et al.* Hierarchical depth aware YOLO for efficient metal surface defect detection. *Sci Rep* (2026). https://doi.org/10.1038/s41598-026-46074-z

Qiyuan Qin, Anis Salwa Mohd Khairuddin, Noorhayati Idros, Haichuan Liu, Liming Fan, Zuoming Yang, JIAYI LI & Chenjinhang Zhu

Hierarchical Depth Aware YOLO for Efficient Metal Surface Defect Detection

QIYUAN QIN[1], ANIS SALWA MOHD KHAIRUDDIN[1,*], NOORHAYATI IDROS[1,3], HAICHUAN LIU[1], LIMING FAN[1], ZUOMING YANG[1], JIAYI LI[1], and CHENJINHANG ZHU[2]

[1]Faculty of Engineering, Department of Electrical Engineering, Universiti Malaya, Kuala Lumpur, 50603, Malaysia
[2]Inner Mongolia Guangju New Materials Co., Ltd., Wuhai Low Carbon Industrial Park, Hainan District, Wuhai, 016000, China
[3]Centre of Printable Electronics, Institute of Advanced Studies, Universiti Malaya, 50603 Kuala Lumpur, Malaysia
[*]anissalwa@um.edu.my

ABSTRACT

Accurate and real-time metal surface defect detection under complex backgrounds and large appearance variations remains a critical challenge in intelligent manufacturing. Existing lightweight detectors often suffer from suboptimal performance due to uniformly applied feature refinement strategies across different network depths, which limits their ability to balance fine-grained representation and computational efficiency. To address this issue, we propose a hierarchical depth-aware refinement framework, termed HDR-YOLO, which explicitly aligns feature enhancement mechanisms with the distinct roles of shallow and deep representations. Specifically, a Query-Focused Convolution (QFC) block is introduced in shallow layers to enhance high-resolution texture and edge information, while a Query-Based Fusion (QBF) block is employed in deeper layers to improve global semantic modeling through adaptive feature interaction. The proposed design enables more effective detection of small-scale defects and irregular fine-grained patterns. Extensive experiments on the NEU-DET and GC10-DET datasets demonstrate that HDR-YOLO improves mAP@0.5 by 3.92% and 7.67%, respectively, over the baseline, while maintaining competitive inference efficiency. These results validate that depth-aware refinement is an effective strategy for enhancing lightweight defect detection under real-time industrial constraints.

Introduction

Casting is a fundamental and versatile manufacturing process for producing heavy-duty and high-precision components, playing a critical role in modern industry. It supports a wide range of applications, including automotive, aerospace, energy equipment, mechanical engineering, and rail transportation, where both structural integrity and dimensional accuracy are of paramount importance[1]. Although contemporary casting production has increasingly integrated automation and digitalized quality-control techniques into processes such as molding, pouring, and sorting, the absence of robust real-time monitoring and inspection technologies still hampers the realization of fully closed-loop manufacturing systems[2]. In practice, surface defects such as cracks, porosity, inclusions, scratches, and corrosion remain prevalent[3]. These defects often arise from process instabilities, non-uniform thermal field distributions, material stress concentrations, or frictional wear during transportation and assembly[4]. Such surface anomalies can significantly degrade structural integrity and fatigue life, and in severe cases may trigger sudden failure of critical components during service, leading to serious safety risks[5]. Consequently, the development of rapid, accurate, and cost-effective surface defect detection techniques is essential for improving manufacturing quality, reducing production costs, and ensuring operational safety.

Traditional surface defect inspection methods mainly include manual visual inspection, nondestructive testing (NDT), and classical image processing–based techniques[6]. Manual inspection relies heavily on operator experience and suffers from high labor intensity, strong subjectivity, low efficiency, and susceptibility to fatigue and inter-operator variability. NDT techniques, such as ultrasonic testing, magnetic particle inspection, and X-ray imaging, can achieve high detection accuracy; however, their application is often limited by expensive equipment, high inspection costs, and low throughput, making them unsuitable for high-speed industrial production lines[7]. Image processing–based methods attempt to reduce manual dependence by extracting handcrafted features such as texture, grayscale, and edges. However, their performance is highly sensitive to illumination variations, noise, and complex surface textures, which often leads to unstable detection results in real industrial environments. Recent studies have further demonstrated that variations in surface illumination and other operational factors in rolling production lines can significantly influence the reliability of defect detection systems, highlighting the importance of robust feature representation under changing industrial conditions[8]. Overall, these conventional approaches struggle to simultaneously satisfy the stringent industrial requirements of high accuracy, robustness, and real-time performance.

The rapid development of deep learning has brought significant advances to industrial visual inspection[9]. Convolutional neural networks (CNNs)[10], with their end-to-end representation learning capability, have largely replaced handcrafted feature engineering. By stacking convolutional, pooling, and nonlinear activation layers, CNNs can automatically learn hierarchical representations ranging from low-level edges and textures to high-level semantic abstractions. The success of AlexNet in the ImageNet challenge demonstrated the potential of deep learning for large-scale image recognition tasks[11]. Subsequently, VGG networks improved representational capacity through deeper architectures[12], while ResNet introduced residual connections to alleviate gradient degradation and enable the training of very deep networks[13]. These foundational architectures not only advanced natural image understanding but also established a solid technical basis for defect detection in industrial scenarios.

Building upon CNN-based feature learning, Girshick et al. proposed the R-CNN framework, marking an important milestone in deep learning–based object detection[14]. R-CNN employed selective search for region proposal generation and CNNs for feature extraction, achieving high detection accuracy but at the cost of substantial computational overhead. Fast R-CNN improved efficiency by sharing convolutional features via RoI pooling, while Faster R-CNN further introduced a region proposal network (RPN) to integrate proposal generation and feature extraction into a unified framework, significantly enhancing both speed and accuracy[15]. Mask R-CNN extended this paradigm by adding an instance segmentation branch, enabling pixel-level defect localization[16]. Despite their strong performance and precise localization capability, the multi-stage architectures and heavy computational demands of R-CNN–based methods limit their applicability in real-time industrial inspection.

To improve efficiency, Liu et al. proposed the Single Shot MultiBox Detector (SSD), which performs object localization and classification in a single forward pass. By exploiting multi-scale feature maps, SSD enhances detection performance across different object sizes while achieving substantially higher inference speed compared to two-stage detectors. This design makes SSD a promising solution for industrial applications. However, SSD still exhibits limitations when dealing with extremely small defects and complex background textures, as its robustness to fine-grained surface variations remains insufficient[17].

Another major line of research is represented by the YOLO (You Only Look Once) family of single-stage detectors. By formulating object detection as an end-to-end regression problem, YOLO simultaneously predicts bounding boxes and class labels in a single inference step, enabling real-time detection performance[18]. Subsequent versions have progressively improved both accuracy and efficiency: YOLOv3 incorporated multi-scale detection via feature pyramid networks; YOLOv4 introduced CSPDarknet and PANet-based feature fusion; and YOLOv5 further optimized modularity and lightweight design for deployment on resource-constrained devices. More recent developments, including YOLO-World[19], PP-YOLOE[20], and Gold-YOLO[21], have improved robustness and precision through advanced detection heads, dynamic label assignment, and enhanced multi-scale feature fusion. Collectively, these advances establish YOLO-based detectors as a dominant solution for real-time industrial inspection, with demonstrated effectiveness in detecting surface defects on metals, aluminum castings, ceramics, and steel products.

Despite this progress, several challenges remain unresolved in deep learning–based surface defect detection. First, defect appearances are often irregular, multi-scale, and tightly coupled with complex background textures, making robust feature extraction difficult. Second, tiny defects such as micropores and fine cracks are prone to being missed due to feature attenuation across deep network layers. Third, achieving an optimal balance between detection accuracy and real-time efficiency remains challenging: highly complex models tend to sacrifice speed, while excessively lightweight architectures often lack sufficient representational capacity. Furthermore, industrial defect datasets are typically limited in size and exhibit severe class imbalance, further restricting model generalization. These challenges collectively hinder the large-scale deployment of intelligent inspection systems in real-world industrial environments.

To address these issues, this work proposes HDR-YOLO, an enhanced YOLO11-based framework for casting surface defect detection. Unlike existing approaches that apply attention or feature refinement mechanisms uniformly across all network layers, HDR-YOLO is motivated by the observation that features at different depths serve fundamentally distinct representational roles. Shallow layers preserve high-resolution spatial information that is crucial for detecting small and low-contrast defects, where locality-aware and computationally constrained refinement is most effective. In contrast, deep layers encode high-level semantic representations with reduced spatial resolution, where global context modeling is essential for discriminating visually similar defect categories. Based on this hierarchical principle, HDR-YOLO introduces a Query-Focused Convolution (QFC-C3k2) module with windowed refinement in shallow layers to enhance local discriminability, while deploying Query-Based Fusion (QBF-C3k2) and C2QBF modules in deeper layers to strengthen global semantic integration. Here, HDR denotes a Hierarchical Depth-Aware Refinement strategy that explicitly tailors refinement mechanisms to the functional roles of different network depths. This design effectively alleviates the trade-off between detection accuracy and inference efficiency. Extensive experiments on the NEU-DET and GC10-DET benchmark datasets demonstrate that HDR-YOLO consistently improves detection performance while maintaining lightweight computational complexity, highlighting its potential for practical nondestructive inspection of industrial casting surfaces.

1 Related Works

Since the introduction of the YOLO framework, its end-to-end single-stage detection paradigm has attracted extensive attention in object detection research due to its favorable balance between accuracy and real-time performance[22]. YOLOv1 was the first to reformulate object detection as a unified regression problem, enabling real-time inference within a single forward pass. Subsequent versions, including YOLOv2 and YOLOv3, progressively improved detection robustness and generalization. YOLOv2 introduced Batch Normalization and multi-scale training to enhance stability, while YOLOv3 adopted multi-scale prediction to substantially improve the detection of small objects.

YOLOv4 further advanced the framework by integrating Mosaic data augmentation, Spatial Pyramid Pooling (SPP), and PANet-based feature fusion on top of the CSPDarknet backbone, achieving notable gains in both accuracy and efficiency. YOLOv5 continued this trajectory by emphasizing modular design and lightweight convolutional operations, which significantly improved deployment flexibility and made the YOLO family widely applicable in industrial inspection scenarios. Collectively, these developments established YOLO as a dominant paradigm for real-time surface defect detection.

More recently, attention mechanisms and Transformer-inspired designs have been increasingly incorporated into YOLO-style detectors to further enhance feature representation under complex backgrounds. YOLOv8 achieved a refined balance among detection accuracy, inference speed, and model compactness, demonstrating improved robustness in challenging environments[23]. Building upon this foundation, YOLO11 introduced architectural refinements in both the backbone and neck by incorporating the C3k2 and C2PSA modules to strengthen multi-scale feature extraction and semantic modeling. Specifically, C3k2 extends the C2f structure with a C3k-based bottleneck, enabling accelerated computation while preserving representational capacity. In deeper layers, YOLO11 integrates the C2PSA module, which combines convolutional feature scaling with PSABlock-based attention to enhance semantic discrimination without incurring excessive computational overhead[24].

Despite these advances, most existing YOLO-based lightweight detectors adopt uniform feature refinement or attention strategies across all network depths. Such designs implicitly assume that features at different hierarchical levels contribute similarly to detection performance, which may not hold in industrial defect inspection. In practice, shallow layers primarily encode high-resolution spatial details and fine-grained textures that are crucial for detecting small or low-contrast defects, whereas deep layers emphasize abstract semantic representations with broader receptive fields. Applying identical refinement mechanisms across all layers can therefore lead to suboptimal utilization of computational resources, either introducing unnecessary overhead in shallow layers or insufficient global modeling capacity in deeper layers.

Motivated by this observation, this work proposes HDR-YOLO, a hierarchical depth-aware refinement framework built upon YOLO11. The main contributions of this work are summarized as follows:

(1) A lightweight defect detection framework, termed HDR-YOLO, is proposed for accurate and real-time steel surface inspection, achieving a favorable balance between detection accuracy and computational efficiency.

(2) A hierarchical depth-aware refinement strategy is introduced, which explicitly considers the distinct representational roles of shallow and deep features in lightweight detection networks. Unlike conventional approaches that apply uniform attention or feature fusion across all layers, the proposed strategy performs differentiated refinement according to feature hierarchy, enabling more effective feature utilization.

(3) Two complementary refinement modules are designed. The QFC module enhances local texture representation in shallow layers through window-based query-focused refinement, while the QBF module enables global semantic interaction in deeper layers to strengthen defect shape modeling and contextual discrimination.

(4) Comprehensive experiments on the NEU-DET and GC10-DET datasets demonstrate that HDR-YOLO consistently outperforms several mainstream lightweight detectors in terms of accuracy–efficiency trade-off. Additional ablation studies further validate the effectiveness of key design components, including window size selection and KAN-based feature transformation.

(5) To improve model interpretability, class activation map (CAM) visualizations are incorporated to analyze the attention behavior of the proposed method. The results show that HDR-YOLO produces more focused and semantically consistent activation regions compared with baseline models, providing intuitive evidence for the effectiveness of the hierarchical depth-aware refinement strategy.

2 Proposed Methodology

Although various YOLO-based improvements have introduced attention mechanisms or feature fusion modules, most existing approaches apply identical refinement operations uniformly across all network layers. However, shallow and deep features in convolutional detectors serve fundamentally different roles in visual representation. Shallow layers primarily encode high-resolution spatial details and texture information, whereas deeper layers capture more abstract semantic representations with larger receptive fields.

Motivated by this observation, this study proposes the HDR-YOLO model, which introduces a hierarchical depth-aware refinement strategy that explicitly aligns refinement mechanisms with the functional roles of different feature levels. Instead

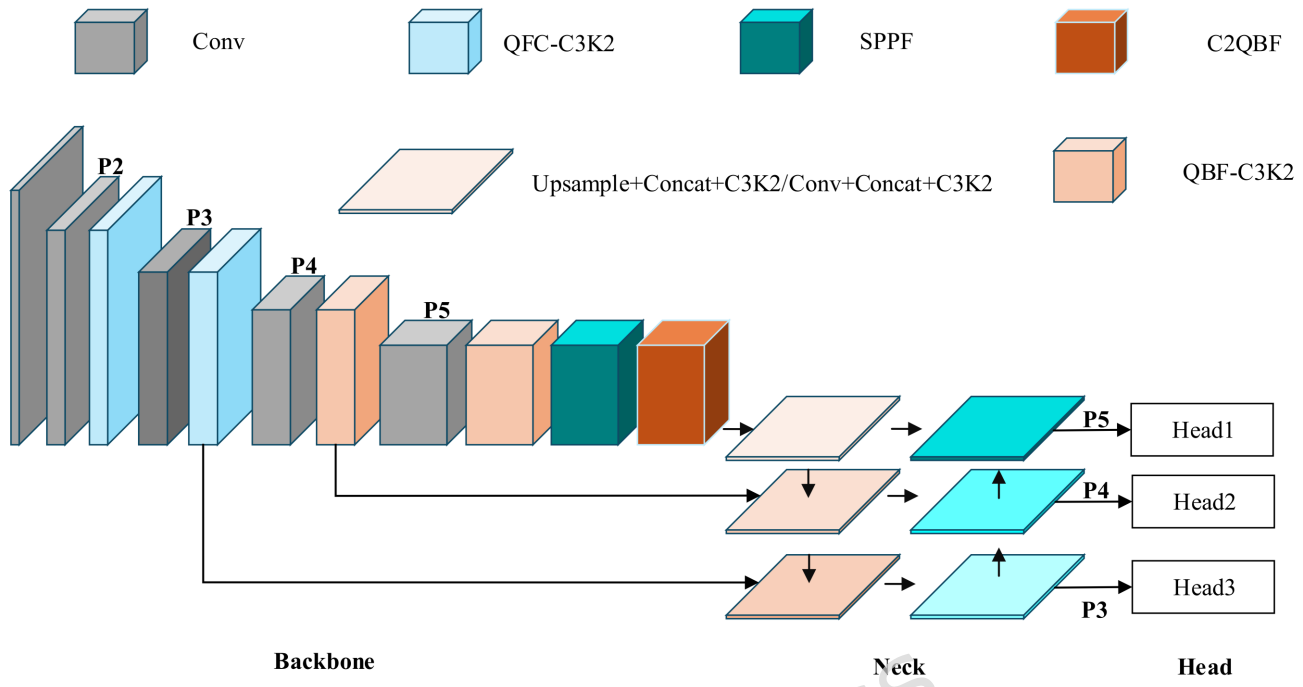

Figure 1. Structure of the proposed HDR-YOLO

of applying a single attention module across the entire network, HDR-YOLO performs differentiated feature refinement at different depths. This design enables local texture preservation in early layers while facilitating global semantic modeling in deeper layers, thereby improving detection accuracy without introducing excessive computational overhead.

The proposed framework remains well aligned with the hallmark characteristics of the YOLO family, namely ultra-fast inference speed and high detection accuracy. Owing to its compact architecture and reduced computational cost, HDR-YOLO is particularly suitable for deployment on industrial devices, thereby satisfying the practical requirements of intelligent manufacturing. In this work, feature fusion is defined not only as cross-scale aggregation, but also as a depth-aware refinement process that selectively enhances local and global information across feature hierarchies.

To overcome the limitations of the original YOLO11 architecture, the proposed framework redesigns its feature extraction and fusion stages by incorporating three novel modules—QFC-C3k2, QBF-C3k2, and C2QBF—which are explicitly tailored to the distinct representation characteristics of shallow and deep network layers. Each module introduces mechanisms such as Performer Attention, lightweight KAN, and a window partitioning strategy, which collectively strengthen semantic connectivity across different feature hierarchies while maintaining computational efficiency and reducing parameter overhead.

To improve the readability of the proposed method, we first provide an intuitive overview of the HDR-YOLO workflow before presenting the detailed mathematical formulations. As illustrated in Fig. 1, the overall framework consists of three main stages: feature extraction, hierarchical refinement, and defect prediction. The backbone network first extracts multi-scale feature maps from the input image. These features are then refined using the proposed hierarchical modules, including the QFC module for shallow-layer refinement and the QBF module for deeper semantic enhancement. Finally, the refined features are fused and passed to the detection head to generate bounding box predictions and classification scores.

2.1 QFC Block for Shallow Feature Refinement

2.1.1 Workflow

To improve the readability of the proposed module, we first provide an intuitive overview of the workflow of the QFC-Block before introducing the detailed mathematical formulations. The QFC-Block is designed to enhance local feature representation in shallow layers while maintaining computational efficiency. The overall workflow consists of five main steps:

1) Window partition: The input feature map is evenly divided into non-overlapping windows along the spatial dimensions to enable localized feature modeling while reducing the computational burden of global attention.

2) Feature flattening: Each window is flattened into a sequence representation to make it compatible with the self-attention mechanism.

3) Performer Attention: Efficient attention modeling is performed within each window using the Performer Attention mechanism, which approximates global self-attention with linear complexity.

4) Lightweight feature aggregation: The extracted features are further refined using the KAN module, which combines depthwise convolution and pointwise convolution to enhance nonlinear representation capability with low computational cost.

5) Feature fusion and reconstruction: Finally, local refined features and global contextual features are fused through convolutional operations, and the processed features are restored to their original spatial structure.

Through this hierarchical workflow, the QFC-Block effectively captures fine-grained local textures while preserving global contextual information, providing robust feature representations for subsequent detection tasks. This study proposes an efficent feature extraction framework with integrated global–local collaborative modeling capabilities, in which the core innovation lies in the improved QFC-Block structure. The module is composed of seven components that perform batch-wise partitioning and feature extraction on the input feature maps. When the extracted feature map passes through the first window-partition module, the input feature map x is evenly divided into a predefined number of windows along its height and width dimensions without altering the inherent feature content.The architecture of QFC-Block is illustrated in Figure 2.

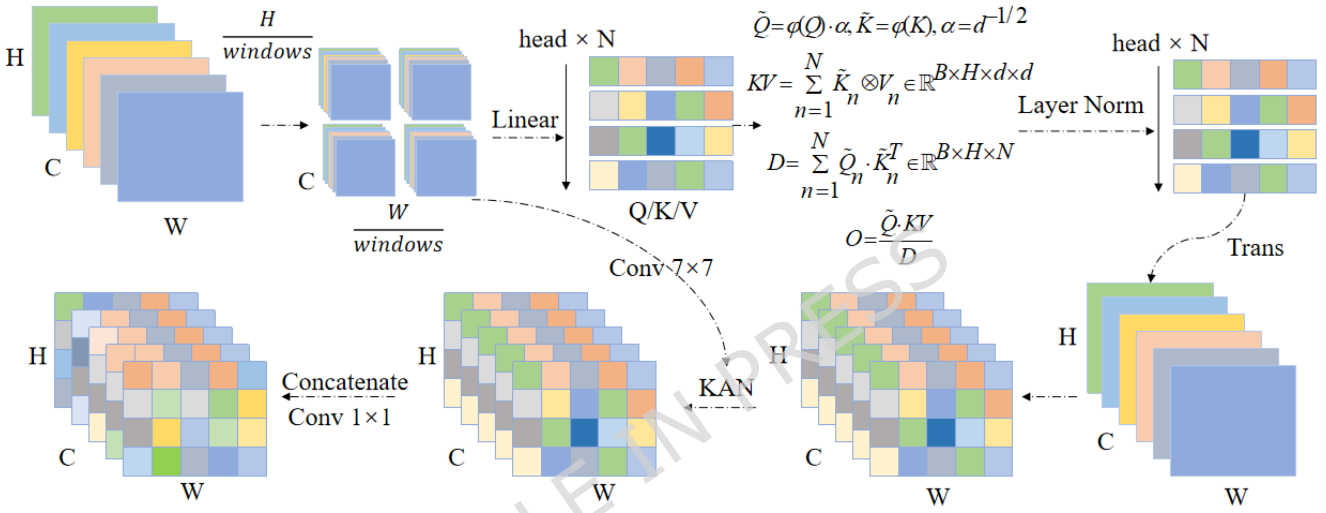

Figure 2. Structures of QFC-Block

2.1.2 Window partition

Since the shallow layers of YOLO11 produce large feature maps characterized by high computational complexity in global attention and abundant local texture and edge information, more refined local modeling is required. Therefore, the feature map is divided into $B\times\frac{H}{\text{window\_size}}\times\frac{W}{\text{window\_size}}$ non-overlapping windows for localized feature modeling, which effectively alleviates the computational burden associated with high-resolution feature maps while preserving critical local information.For an input feature map $X\in R^{HWC}$,the computational complexity of global self-attention is on the order of $O\left((\mathrm{HW})^2\mathrm{C}\right)$. By contrast, the window-based partitioning strategy reduces the complexity to $O\left(HWM^2C\right)$, where M denotes the window size. When $M\ll HW$, the complexity decreases by approximately a factor of $\frac{M^2}{HW}$, thereby substantially reducing redundant computations while maintaining robust local feature representation.

To further ensure that boundary information is preserved and that the spatial dimensions of the feature maps remain consistent after window partitioning, an automatic padding operation is applied. The padding function can be formally expressed in (1).

$$x'=\mathrm{Pad}(x) \tag{1}$$

After the automatic padding, the processed feature map is partitioned into non-overlapping windows to enable localized attention modeling. Formally, the window partitioning operation can be defined in (2).

$$W\left(X'\right)\in R^{N\times C\times M\times M}$$
$$M=\text{Window Size}$$
$$N=\frac{H'}{M}\cdot\frac{W'}{M}\cdot B \tag{2}$$

Here, B denotes the batch size, C the number of channels, and H, W the height and width of the feature map, respectively. Through this partitioning strategy, the two-dimensional spatial domain is divided into several non-overlapping windows. After partitioning, the convolutional features are organized into a four-dimensional tensor of shape. To make the features directly compatible with the sequential input format required by the self-attention mechanism, the spatial dimensions of each window are flattened. The flattening operation can be formally expressed in (3).

$$X_W = \text{Flatten}\left(W\left(x^{'}\right)\right), X_w \in R^{N\times M^2\times C} \tag{3}$$

2.1.3 Performer Attention

To mitigate the prohibitive time and space complexity of conventional self-attention, each flattened feature block is processed by Performer Attention while preserving both the spatial resolution and the channel dimensionality[25]. Specifically, let B denote the batch size, N the sequence length, and C the channel dimension. The input sequence is first linearly projected into the query (Q), key (K), and value (V) matrices. To further enhance representational diversity, the model employs a multi-head partitioning strategy, in which the dimensionality of each head is defined as $d = \frac{C}{H}$. Where H represents the number of attention heads. This design ensures that the Performer module can effectively approximate global self-attention with linear complexity while maintaining the original feature map resolution. A kernel-based approximation of the softmax function is introduced, and the formulation is given in (4).

$$\begin{aligned} &Q = xW_Q, \quad K = xW_K, \quad V = xW_V \\ &Q, K, V \in R^{B\times H\times N\times d} \\ &\text{softmax}\left(\frac{QK^\top}{\sqrt{d}}\right) \approx \phi(Q)\,\phi(K)^\top \end{aligned} \tag{4}$$

In this work, we adopt a kernel-based approximate attention mechanism to circumvent the quadratic complexity of conventional softmax attention when handling large-scale sequences. Within this framework, the query and key vectors are transformed through a positive-valued mapping function to ensure numerical stability during the computation of attention weights. Specifically, the Exponential Linear Unit (ELU) function provides a smooth exponential mapping in the negative domain while preserving linearity in the positive domain. By applying an additional one operation, the output values are strictly constrained to be non-negative, thereby preventing numerical oscillations that may arise from negative weights during kernel decomposition. The formulation is expressed as follows. By applying an additional $+1$ operation, the output values are strictly constrained to be non-negative, thereby preventing numerical oscillations that may arise from negative weights during kernel decomposition. The formulation is expressed in (5).

$$\phi(x) = \text{ELU}(x) + 1 = \begin{cases} x + 1, & x > 0, \\ \alpha\,(e^x - 1) + 1, & x \le 0, \end{cases} \tag{5}$$

In the attention mechanism, the dot-product between the query and key vectors tends to exhibit an increasing variance as the feature dimension d grows. This phenomenon leads to an excessively sharp softmax distribution, thereby diminishing the model's ability to discriminate among features at different positions. To ensure numerical stability and prevent the inner product values from becoming disproportionately large, the query vector is typically scaled. Specifically, the dot-product result is divided by the square root of the dimensionality$\sqrt{d}$, which can be expressed mathematically in (6).The scaling factor $\frac{1}{\sqrt{d}}$ effectively constrains the variance within a stable range, thereby preventing the gradients from becoming excessively large or vanishing during backpropagation. This operation enhances both the training stability of the model and its generalization capability.

$$\widetilde{Q} = \phi\,(Q)\cdot\alpha, \widetilde{K} = \phi\,(K), \alpha = d^{-\frac{1}{2}} \tag{6}$$

Subsequently, the obtained weight distribution is applied to the value vectors V, enabling a weighted fusion across different feature channels. This process yields a semantically enhanced representation by performing a weighted summation of the contextual information encoded in V, thereby incorporating global dependencies into the output feature representation. Essentially, this operation can be regarded as a form of key–value aggregation, which can be formally expressed in(7)

$$KV = \sum_{n=1}^{N} \widetilde{K_n} \otimes V_n \in R^{B\times H\times d\times d} \tag{7}$$

When the dimensionality of the query and key vectors is high, the unnormalized dot-product values grow rapidly, which excessively stretches the input distribution of the softmax function. This phenomenon drives the gradients toward zero and severely degrades the convergence performance of the model. To mitigate this issue, a normalization factor is introduced, formally defined in (8), the final computation result is presented in Equation (9).

$$D = \sum_{n=1}^{N} \widetilde{Q}_n \cdot \widetilde{K_n^T} \in R^{B\times H\times N} \tag{8}$$

$$O = \frac{\widetilde{Q}\cdot KV}{D} \tag{9}$$

Finally, the outputs of the multi-head mechanism are concatenated and subsequently projected through a linear mapping. Specifically, the results from all heads are concatenated along the feature dimension to form an expanded representation vector. A linear transformation is then introduced to prevent the dimensionality from growing linearly with the number of heads, thereby ensuring compatibility with the subsequent network layers. This operation preserves the diverse contextual patterns learned across different subspaces, enabling the model to capture dependencies at multiple levels—ranging from local to global and from sparse to dense. The formulation is given in (10).

$$PerformerAttention(x) = OW_o, W_o \in R^{C\times C} \tag{10}$$

2.1.4 KAN feature refinement

To achieve a lightweight architecture without compromising detection accuracy, this study introduces the KAN module after the Performer Attention layer for efficient feature extraction. The KAN module integrates depthwise convolution and pointwise convolution to form a depthwise separable convolution, which effectively decouples spatial and channel correlations, thereby improving computational efficiency and enhancing the model's nonlinear representation capacity, as illustrated in Figure 3. This design is theoretically supported by prior works such as MobileNet[26] and Xception[27] , which demonstrated that depthwise separable convolutions can reduce computational cost by nearly an order of magnitude while maintaining comparable accuracy to standard convolutions.

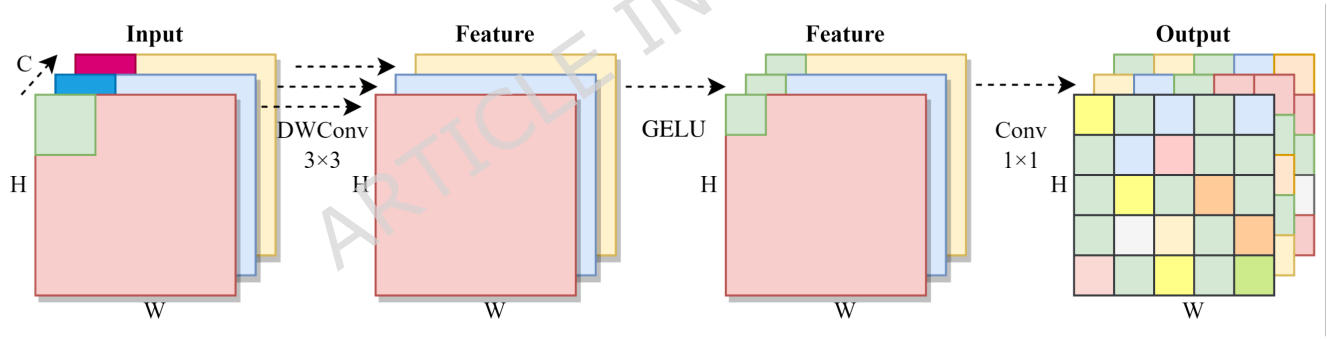

Figure 3. The KAN model integrating pointwise convolution with depthwise convolution

Specifically, the feature map is first processed by a depthwise convolution to extract spatial features, which primarily capture local structures such as edges and textures, while no information is exchanged across channels. To enable inter-channel interaction, a subsequent pointwise convolution is applied to aggregate information across different channels. This design allows flexible control of the channel dimensionality: channel expansion strengthens feature expressiveness, whereas channel reduction decreases computational complexity and parameter overhead. The final formulation of the KAN module is expressed in (11).

$$KAN(x) = Conv_{1\times1}\left(GELU\left(DWConv_{3\times3}(x)\right)\right) \in R^{B\times C\times H\times W} \tag{11}$$

As a lightweight submodule within the QFC-Block, this module achieves a substantial reduction in computational overhead while still maintaining strong feature representation capability. Let A denote the data after PerformerAttention processing, which is formulated in (12).

$$A = PerformerAttention(X_w) \tag{12}$$

To enhance training stability and facilitate feature learning, layer normalization (LN) is applied along the channel dimension to both data A and the X_w which is features flattened after window partitioning, ensuring that each feature vector has zero mean and unit variance. LN effectively mitigates internal covariate shift and promotes efficient gradient propagation. Within this framework, LN contributes to extracting robust features from both shallow and deep layers, thereby improving detection accuracy—particularly under small-batch training and high-resolution feature maps.The formulation is in (13).

$$A' = LayerNorm\left(A + X_w\right) \tag{13}$$

After that, the feature maps are processed by the KAN (Kolmogorov–Arnold-inspired Aggregation Network) module, which comprises pointwise convolution and depthwise separable convolution. In this work, the design of the KAN module is inspired by the concept of Kolmogorov–Arnold Networks, which emphasize enhanced nonlinear representation capability and efficient feature transformation. However, unlike the original Kolmogorov–Arnold Networks that are based on functional decomposition theory, the proposed KAN module is implemented as a lightweight convolutional aggregation block designed for efficient feature interaction in convolutional neural networks. Specifically, it employs pointwise convolution and depthwise separable convolution to enhance inter-channel information interaction and nonlinear representation capability while maintaining low computational overhead. The processed features are then restored to their original spatial structure through the inverse window operation, providing high-fidelity feature representations for subsequent attention mechanisms or convolutional operations. The formulation is in (14).

$$F_{loc} = W^{-1}\left(KAN\left(A'\right)\right) \tag{14}$$

2.1.5 Global feature modeling

To enhance the module's capability for modeling global dependencies, a 7×7 large-kernel depthwise convolution is employed as a global positional encoding branch, as formulated in (15). Compared with conventional 3×3 convolutions, the large-kernel convolution increases the receptive field while maintaining computational efficiency, thereby establishing an effective connection between local attention and global modeling. This design also significantly strengthens the model's spatial sensitivity and multi-scale feature integration capability. Subsequently, the processed data undergoes feature fusion, as described in (16).

$$F_{glob} = DWConv_{7\times7}\left(x'\right) \tag{15}$$

$$F = Conv_{1\times1}\left(\left[F_{loc}, F_{glob}\right]\right) \tag{16}$$

To ensure that the output dimensions remain consistent with the original input while retaining only valid data, the added padding may induce ineffective interactions in subsequent attention or convolution operations, thereby reducing the discriminative power of the representations. Therefore, padding is removed at the end of the module. Formulation is in (17).

$$\text{Output} = \text{Unpad}(F),\ \text{Output} \in R^{B\times C\times H\times W} \tag{17}$$

Distinct from traditional SSD that relies solely on fixed-scale sliding windows for local feature modeling, the windowing strategy of QFC-Block offers significant advantages. First, SSD achieves multi-scale detection by traversing the feature map point by point with convolutional kernels, but this introduces redundancy between adjacent windows and incurs high computational costs for high-resolution inputs. In contrast, QFC-Block partitions the feature map into non-overlapping uniform regions along spatial dimensions, effectively avoiding redundancy and substantially reducing the complexity of global self-attention.

Second, the windowing operation in SSD depends on the receptive field of convolutional kernels, which lacks the ability to model long-range dependencies. QFC-Block incorporates Performer Attention after window partitioning, enabling efficient local feature extraction while capturing larger-scale global dependencies, thereby reducing redundant computation and preserving global modeling capacity.

Finally, in post-window feature processing, the QFC-Block integrates Layer Normalization and the KAN module to alleviate numerical instability and gradient vanishing. The combination of depthwise and large-kernel convolutions balances local sensitivity and global receptive fields, enabling the model to accurately capture fine-grained textures in tasks such as small-object detection and surface defect recognition while preserving robust global representations for larger targets.

This design explicitly separates local texture enhancement from global semantic modeling, thereby simplifying the overall refinement process. Consequently, the hierarchical workflow improves both feature representation capability and computational efficiency.

2.2 QBF Block for Deep Feature Refinement

In addition, this study introduces the QBF-Block, which is integrated into the deeper layers of the YOLO network. In contrast to the QFC-Block, QBF-Block discards the window partitioning mechanism while preserving all other core components, as illustrated in Figure 4. This architectural modification is motivated by the intrinsic differences in feature characteristics between shallow and deep layers. In the shallow stages of YOLO11, feature maps exhibit high spatial resolution, leading to a quadratic increase in computational cost with respect to resolution. The windowing strategy in QFC-Block thus serves to effectively constrain the computational burden while maintaining fine-grained spatial detail, making it well-suited for high-resolution shallow representations.

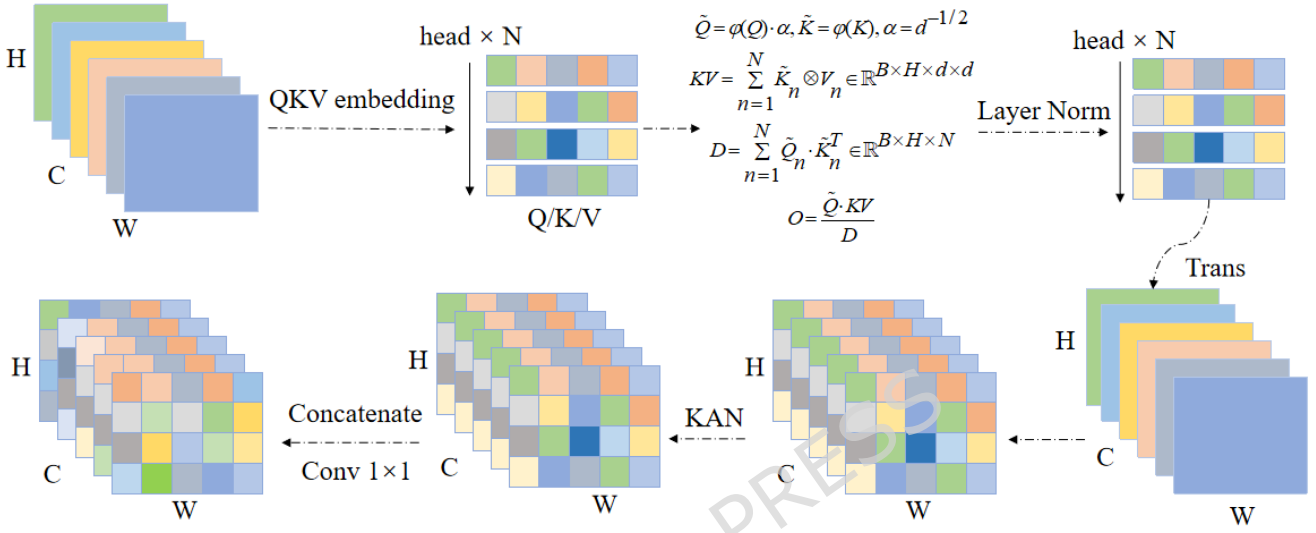

Figure 4. The structure of QBF-Block.

Conversely, in deeper layers, the spatial resolution of feature maps is substantially reduced. Under such conditions, window partitioning no longer provides a meaningful reduction in computational complexity and instead hinders the modeling of long-range dependencies, thereby constraining the discriminative capacity of deep representations. Therefore, the QBF-Block omits windowing to preserve global context modeling, while retaining other essential operations that remain advantageous in deep feature processing.

Specifically, the incorporation of Performer Attention within low-resolution deep features enables efficient global dependency modeling with linear computational complexity, substantially enhancing semantic coherence across the entire image. Furthermore, Layer Normalization alleviates inter-channel scale disparities, stabilizing feature distributions and improving optimization robustness. Finally, given that the linear representational forms of convolution and attention are often insufficient for capturing the complex nonlinear semantics of deep features, the KAN module introduces lightweight nonlinear enhancement, significantly strengthening the expressive and discriminative capability of the network's deep feature representations.

2.3 Design of Hierarchical Refinement Modules

In this study, the aforementioned designs are applied to improve the C3k2 and C2PSA structures, leading to the proposal of three novel modules: QFC-C3k2, QBF-C3k2, and C2QBF. These modules enhance feature extraction and representation at different levels, and their structural illustrations are presented in Fig 5.

This study introduces the QFC-C3k2 module, which replaces the C3k module within C3k2 with QFC-Block in HDR-YOLO to process shallow-layer features. In the YOLO11 backbone, the C3k2 module functions as an efficient CSP Bottleneck structure, primarily extracting features through stacked convolutions and residual connections. However, the traditional C3k module relies on 3×3 convolutions for local feature modeling, with receptive field growth dependent on the depth of convolutional stacking. This design exhibits low efficiency in modeling large-scale targets or long-range dependencies and struggles to capture global semantic relationships between distant pixels. Additionally, its nonlinearity is mainly provided by activation functions such as ReLU, whose representation capability is relatively limited. Consequently, in complex scenarios, it is difficult to simultaneously capture fine-grained local texture and global structural features. By replacing C3k2 with QFC-C3k2 in the high-resolution shallow layers, windowed processing not only reduces computational cost significantly but also enhances global semantic understanding.

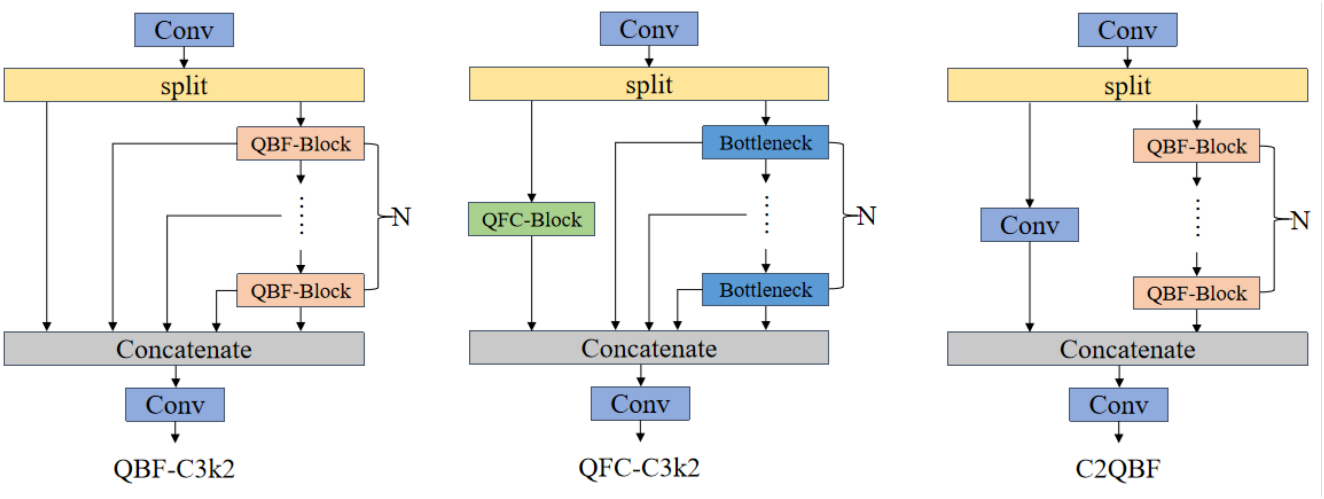

Figure 5. Structures of QFC-C3k2, QBF-C3k2, and C2QBF Modules

The study further proposes the QBF-C3k2 module, which replaces C3k with QBF-Block within C3k2 for processing deep-layer features in HDR-YOLO. This module addresses the limitations of C3k, which only extracts features from local neighborhoods. In deep layers, where spatial resolution is substantially reduced, the locality of convolutions is insufficient to express high-level semantics, limiting detection accuracy and generalization. Unlike QFC-Block, the QBF-C3k2 module removes the windowing mechanism to avoid compromising global dependency modeling while maintaining computational efficiency. At low resolutions in deep layers, computation is no longer the primary bottleneck. The introduction of Performer Attention enables the modeling of long-range dependencies across the entire feature map, effectively capturing the overall structure and global context of targets, enhancing the detector's understanding of complex scenes and reducing the risk of information loss. Moreover, the integration of the KAN module and large-kernel convolutions further adapts the network to multi-scale targets, improving generalization.

Finally, the C2QBF module is proposed. Compared with the traditional C2PSA, it retains the design of applying attention to only half of the channels while passing the other half through a shortcut path. This design significantly reduces both computational load and memory consumption relative to "full-channel attention blocks," striking an optimal balance aligned with YOLO's goal of fast detection. In this work, the PSA module within C2PSA is replaced with the self-developed QBF module. PSA essentially applies spatial attention to a portion of channels to enhance positional information. However, in deep layers of YOLO, where feature maps have low resolution, emphasizing positional relationships has limited effect because a single pixel covers a large spatial area, blurring global semantics. In contrast, QBF-Block employs a deterministic large-kernel (potentially dilated) convolution to achieve stable "near-global" aggregation at low resolution, offering deployment-friendly, noise-robust, and balanced feature representation. Its Performer Attention explicitly models long-range dependencies and cross-region interactions, compensating for the primarily local-to-medium receptive field of large-kernel convolutions. Therefore, replacing C2PSA with C2QBF preserves the original.

3 Simulation Results

3.1 Experimental Setup and Evaluation Protocol

Experiments were conducted on two public benchmark datasets for steel surface defect detection, namely NEU-DET and GC10-DET. NEU-DET consists of 1,800 grayscale images covering six representative defect categories, with balanced class distribution and substantial variations in defect morphology, orientation, and background complexity[28]. GC10-DET contains over 3,500 annotated images across ten defect types and was specifically designed for hot-rolled steel inspection, providing higher category diversity and greater sample complexity that closely reflect real industrial scenarios[29]. The complementary characteristics of these datasets enable a comprehensive evaluation of both accuracy and generalization capability.

The defect categories defined in these datasets follow commonly adopted metallurgical inspection practices for rolled steel surface quality. Typical defect types such as cracks, inclusions, patches, scratches, and pitted surfaces correspond to representative defect categories widely recognized in industrial steel surface inspection. The NEU-DET dataset was released by Northeastern University for steel surface defect detection research, while GC10-DET was constructed to simulate real hot-rolled steel inspection scenarios. During the construction of these datasets, the annotations were generated and verified

by domain experts according to clearly defined labeling guidelines. This expert-driven annotation process ensures consistent defect categorization and minimizes ambiguity between visually similar defect classes.

All images were resized to 640×640, normalized to $[0,1]$, and grayscale samples were expanded to three channels for network compatibility. Aspect ratios were preserved via automatic padding. To improve robustness and alleviate class imbalance, data augmentation techniques including random cropping, rotation, scaling, and mosaic augmentation were applied during training. Each dataset was split into training, validation, and test sets with a ratio of $7:1.5:1.5$.

Detection performance was evaluated using Precision, Recall, and mean Average Precision (mAP), including mAP@0.5 and mAP@0.5:0.95. Model efficiency was assessed in terms of parameter count and computational complexity measured by GFLOPs, providing an integrated evaluation of accuracy, compactness, and real-time inference capability.

All experiments were implemented using PyTorch (v1.10) with Python 3.9 under CUDA 11.8. Training was performed on a workstation equipped with an Intel i7-10875H CPU, 32 GB RAM, and a single NVIDIA RTX 5090 GPU (32 GB memory). The AdamW optimizer was adopted with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\varepsilon = 1 \times 10^{-8}$. The initial learning rate was set to 1×10^{-3} and decayed to 1×10^{-4} using a cosine annealing schedule, with weight decay fixed at 5×10^{-4}. All models were trained for 300 epochs with a batch size of 32.

For fair comparison, all baseline and competing detectors were trained and evaluated under identical experimental settings, including dataset splits, input resolution, data augmentation strategies, optimizer configuration, and training schedules.

3.2 Effectiveness of Hierarchical Depth Aware Refinement in HDR YOLO

To systematically evaluate the effectiveness of the proposed hierarchical depth-aware refinement strategy, a series of controlled ablation experiments are conducted on the NEU-DET and GC10-DET datasets, with YOLO11n adopted as the lightweight baseline. This baseline provides a suitable reference due to its balanced trade-off between computational efficiency and detection accuracy, enabling clear attribution of performance gains to specific architectural components.

Table 1. Ablation study of the proposed HDR-YOLO on the NEU-DET dataset. The impact of the proposed QFC and QBF modules is evaluated in terms of detection accuracy and computational efficiency.

Method	Recall	Precision	mAP@0.5	mAP@0.5:0.95	Params (M)	GFLOP
YOLO11n	75.49	71.91	77.85	48.65	2.62	**6.6**
+ QFC	76.80	74.20	79.16	50.20	**2.60**	7.1
+ QBF	76.50	73.80	78.92	49.70	2.71	**6.6**
HDR-YOLO	**76.94**	**75.63**	**81.77**	**52.59**	2.73	7.2

As reported in Table 1, introducing the QFC module into shallow layers leads to consistent improvements, increasing mAP@50 from 77.85% to 79.16% and mAP@50:95 from 48.65% to 50.20%. This gain can be attributed to the localized window-based refinement, which enhances fine-grained texture representation and edge sensitivity in early feature maps. In contrast, incorporating the QBF module in deeper layers improves global semantic modeling and long-range feature interaction, achieving mAP@50 and mAP@50:95 of 78.92% and 49.70%, respectively.

Although each module independently improves performance, their combination yields a more substantial gain. The full HDR-YOLO model achieves the best performance, reaching 81.77% mAP@50 and 52.59% mAP@50:95, with only a marginal increase in computational cost (2.73M parameters and 7.2 GFLOPs). This result indicates that shallow and deep refinement modules play complementary roles, and that explicitly aligning refinement mechanisms with feature hierarchy is more effective than uniform enhancement strategies.

Table 2. Ablation study of the proposed HDR-YOLO on the GC10-DET dataset. Results demonstrate the effectiveness of depth-aware feature refinement under complex industrial conditions.

Method	Recall	Precision	mAP@0.5	mAP@0.5:0.95	Params (M)	GFLOP
YOLO11n	70.02	72.22	73.92	42.05	2.62	**6.6**
+ QFC	75.81	77.62	77.39	46.38	**2.60**	7.1
+ QBF	76.14	77.90	78.27	47.25	2.71	**6.6**
HDR-YOLO	**79.58**	**78.24**	**81.59**	**50.79**	2.73	7.2

A similar trend is observed on the GC10-DET dataset, as shown in Table 2. The baseline performance of 73.92% mAP@50 is improved to 77.39% with QFC and further to 78.27% with QBF, while the complete HDR-YOLO model achieves the highest

accuracy of 81.59% mAP@50 and 50.79% mAP@50:95. These consistent improvements across datasets demonstrate the robustness and generalization capability of the proposed depth-aware refinement strategy under diverse defect distributions and industrial conditions.

Table 3. Sensitivity analysis of shallow window size.

Model Variant	Window Size	Params (M)	GFLOPs	NEU-DET mAP_{50} (%)	GC10-DET mAP_{50} (%)
Variant A	3×3	2.74	7.2	79.26	79.55
Variant B	5×5	2.73	7.2	80.59	80.29
Ours	7×7	2.73	7.2	**81.77**	**81.59**
Variant C	9×9	2.74	7.3	79.65	80.13

Table 4. Comparison of different structural configurations.

Variant	Structure	Params (M)	GFLOPs	NEU-DET mAP_{50} (%)	GC10-DET mAP_{50} (%)
Variant A	Performer + MLP	2.74	7.2	80.23	80.02
Variant B	Performer + Strip-MLP	2.72	7.1	80.52	80.36
Ours	Performer + KAN	2.73	7.2	**81.77**	**81.59**

To further justify the architectural design, additional ablation studies are conducted on key components, including shallow window size and structural configuration, as summarized in Tables 3 and 4.

From Table 3, the performance is sensitive to the choice of window size in shallow layers. A small window (3×3) limits contextual information, resulting in suboptimal performance, while an excessively large window (9×9) introduces redundant context and slightly degrades accuracy. The proposed 7×7 configuration achieves the best results on both datasets (81.77% on NEU-DET and 81.59% on GC10-DET), indicating that it provides an effective balance between local detail preservation and contextual modeling.

Table 4 further compares different structural configurations. Replacing the standard MLP with Strip-MLP yields moderate improvements, while the proposed Performer + KAN design achieves the best performance, improving mAP@50 to 81.77% and 81.59% on NEU-DET and GC10-DET, respectively. This result suggests that KAN enhances nonlinear feature transformation and representation flexibility, while Performer attention efficiently models long-range dependencies with reduced computational overhead.

Overall, these results consistently demonstrate that each component contributes positively to the final performance. More importantly, the combination of hierarchical refinement, adaptive window design, and enhanced feature transformation leads to a synergistic effect, enabling HDR-YOLO to achieve a superior accuracy–efficiency trade-off compared with conventional lightweight detectors. This validates the core design principle of aligning refinement strategies with the hierarchical roles of feature representations.

3.3 Comparison with State of the Art Methods

Table 5. Quantitative comparison with mainstream lightweight and Transformer-based detectors on the NEU-DET dataset. All methods are trained and evaluated under identical experimental settings.

Method	Recall	Precision	mAP@0.5	mAP@0.5:0.95	Params (M)	GFLOP
YOLOv5n	72.10	70.25	75.12	46.20	2.65	7.8
YOLOv8n	73.85	71.40	76.08	47.10	3.16	8.9
YOLOv10n	74.20	72.00	76.93	47.50	2.78	8.7
YOLO11n	75.49	71.91	77.85	48.65	2.62	**6.6**
YOLO12n	76.10	72.80	77.93	49.10	**2.60**	6.7
RT-DETR-18	**77.20**	73.50	78.26	49.90	20.00	60.0
HDR-YOLO	76.94	**75.63**	**81.77**	**52.59**	2.73	7.2

To comprehensively evaluate the effectiveness of the proposed HDR-YOLO framework, systematic comparisons were performed with representative lightweight and Transformer-based object detectors under fully unified training and evaluation protocols. Specifically, all competing models were trained using identical datasets, input resolution, data augmentation

Table 6. Quantitative comparison with mainstream detectors on the GC10-DET dataset. The proposed HDR-YOLO achieves a favorable accuracy–efficiency trade-off.

Method	Recall	Precision	mAP@0.5	mAP@0.5:0.95	Params (M)	GFLOP
YOLOv5n	68.90	70.12	70.31	38.45	2.65	7.8
YOLOv8n	70.35	73.28	72.22	40.12	3.16	8.9
YOLOv10n	69.82	72.91	72.19	41.23	2.78	8.7
YOLO11n	70.02	72.22	73.92	42.05	2.62	**6.6**
YOLO12n	71.56	72.33	74.13	43.27	**2.60**	6.7
RT-DETR-18	77.92	75.15	75.26	46.88	20.00	60.0
HDR-YOLO	**79.58**	**78.24**	**81.59**	**50.79**	2.73	7.2

schemes, optimization configurations, and training schedules, thereby ensuring fairness and reproducibility of the experimental results. The comparative study includes YOLOv5n[30], YOLOv8n[31], YOLOv10n[32], YOLO11n[24], YOLOv12n[33], and the Transformer-based RT-DETR-18[34].

Table 5 summarizes the quantitative results on the NEU-DET dataset. Among all compared methods, HDR-YOLO achieves the best overall performance, attaining an mAP@50 of 81.77% and an mAP@50:95 of 52.59%, which correspond to absolute improvements of 3.92% and 3.94% over the baseline YOLO11n. In addition, HDR-YOLO consistently improves both Precision (75.63%) and Recall (76.94%), indicating enhanced detection reliability and robustness across defect categories.

Notably, these performance gains are achieved while preserving a lightweight model structure. HDR-YOLO contains only 2.73M parameters and requires 7.2 GFLOPs, which is comparable to YOLO11n (2.62M parameters and 6.6 GFLOPs). In contrast, the Transformer-based RT-DETR-18 exhibits a substantially higher computational cost, requiring approximately 20M parameters and 60 GFLOPs, yet delivers inferior detection accuracy on NEU-DET. This comparison highlights that explicitly aligning refinement mechanisms with feature hierarchy can yield superior accuracy–efficiency trade-offs without resorting to heavy attention-based architectures.

The generalization capability of HDR-YOLO is further validated on the more challenging GC10-DET dataset, as reported in Table 6. The proposed model achieves an mAP@50 of 81.59% and an mAP@50:95 of 50.79%, outperforming YOLO11n by 7.67% and 8.74%, respectively. Meanwhile, Precision and Recall are improved to 78.24% and 79.58%, demonstrating enhanced sensitivity to diverse and complex defect patterns. Despite these substantial gains, HDR-YOLO maintains its lightweight design, preserving real-time inference capability required for industrial deployment.

Overall, the experimental results on both NEU-DET and GC10-DET consistently demonstrate the effectiveness of the proposed hierarchical depth-aware refinement strategy. Compared with existing lightweight YOLO-based detectors, HDR-YOLO significantly improves detection accuracy while maintaining comparable model complexity and computational cost. Although Transformer-based detectors such as RT-DETR introduce stronger global modeling capability, their substantially higher parameter counts and computational overhead limit their practicality for real-time industrial inspection scenarios. In contrast, HDR-YOLO achieves a more favorable balance between accuracy, efficiency, and deployment feasibility, making it particularly suitable for high-speed industrial surface defect detection tasks.

3.4 Classwise Performance Evaluation

Table 7. Class-wise detection performance of HDR-YOLO on GC10-DET dataset

Class	Box(P)	R	mAP50	mAP50-95
all	0.756	0.769	0.817	0.525
crazing	0.622	0.565	0.624	0.301
inclusion	0.758	0.800	0.869	0.559
patches	0.855	0.905	0.980	0.722
pitted_surface	0.775	0.759	0.824	0.566
rolled-in_scale	0.700	0.681	0.684	0.422
scratches	0.826	0.904	0.921	0.580

To further evaluate the performance of the proposed method, a class-wise analysis is conducted on both GC10-DET and NEU-DET datasets, as presented in Tables 7 and 8.

On the GC10-DET dataset, the proposed method achieves strong performance across most defect categories. In particular,

Table 8. Class-wise detection performance of HDR-YOLO on NEU-DET dataset

Class	Box(P)	R	mAP50	mAP50-95
all	0.782	0.795	0.815	0.507
crescent_gap	0.893	0.905	0.977	0.691
crease	0.682	0.699	0.641	0.362
silk_spot	0.841	0.860	0.917	0.575
water_spot	0.835	0.851	0.914	0.565
welding_line	0.831	0.843	0.913	0.558
inclusion	0.622	0.636	0.520	0.271
oil_spot	0.775	0.791	0.827	0.475
rolled_pit	0.700	0.712	0.590	0.320
punching_hole	0.853	0.871	0.933	0.610
waist_folding	0.790	0.782	0.916	0.645

patches and scratches obtain the highest detection accuracy, with mAP50 values of 0.980 and 0.921, respectively. This can be attributed to their relatively clear structures, high contrast, and distinct texture patterns, which are easier to capture by convolutional features and hierarchical refinement. In contrast, crazing and rolled-in scale exhibit relatively lower performance, with mAP50 values of 0.624 and 0.684. These defect types are characterized by irregular morphology, weak boundaries, and low contrast against the background, which increases the difficulty of accurate localization and feature discrimination.

Similarly, on the NEU-DET dataset, most defect categories achieve stable and competitive results. Crescent gap and punching hole achieve the best performance, with mAP50 values of 0.977 and 0.933, respectively, due to their relatively regular shapes and clear edge information. However, inclusion, rolled pit, and crease remain challenging, achieving lower mAP50 values of 0.520, 0.590, and 0.641. These defects typically involve small-scale structures, subtle texture variations, or complex background interference, making them more difficult to distinguish from normal surface patterns.

Overall, the results demonstrate that the proposed method performs robustly across diverse defect types. Defects with clear geometric structures and strong visual contrast are easier to detect, whereas those with low contrast, irregular shapes, or weak texture cues remain challenging. Notably, the consistent performance across both datasets indicates that the proposed hierarchical depth aware refinement effectively enhances both local texture representation and global semantic discrimination, contributing to improved robustness under varying defect characteristics. Nevertheless, further improvements are still needed for capturing extremely subtle and ambiguous defect patterns.

3.5 Computational Efficiency and Real Time Performance

Table 9. Efficiency and performance comparison of different lightweight detectors. Experiments were conducted on an NVIDIA RTX 5090 GPU (32GB) under Windows 11. The best results are highlighted in bold.

Method	Torch FPS	ONNX FPS	NEU-DET mAP_{50} (%)	GC10-DET mAP_{50} (%)	Params (M)	GFLOPs
YOLOv5n	181.43	252.18	75.12	70.31	2.65	7.8
YOLOv8n	**202.77**	**289.96**	76.08	72.22	3.16	8.9
YOLOv10n	118.62	159.13	76.93	72.19	2.78	8.7
YOLO11n	160.79	220.28	77.85	73.92	2.62	**6.6**
YOLO12n	110.15	153.65	77.93	74.13	**2.60**	6.7
HDR-YOLO	139.93	196.68	**81.77**	**81.59**	2.73	7.2

In addition to theoretical efficiency indicators such as parameter count and GFLOPs, practical industrial deployment also requires evaluating real-time inference performance. In industrial surface inspection systems, inference latency and throughput (FPS) directly determine whether the detection algorithm can operate reliably under high-speed production line conditions. Therefore, we further report the inference speed and corresponding latency of different lightweight detectors.

The inference latency is calculated according to the relationship $Latency = 1000/FPS$, which represents the average processing time per image in milliseconds. All experiments were conducted on a workstation equipped with an NVIDIA RTX 5090 GPU (32 GB) running the Windows 11 operating system. Two inference frameworks were evaluated, including the PyTorch runtime and the optimized ONNX inference engine, which is commonly adopted in industrial deployment pipelines due to its portability and efficient inference acceleration.

The efficiency comparison results are summarized in Table 9. The proposed HDR-YOLO achieves 139.93 FPS (7.15 ms

latency) under PyTorch inference and 196.68 FPS (5.08 ms latency using ONNX acceleration). Although the raw throughput is slightly lower than the fastest baseline detector, HDR-YOLO significantly improves detection accuracy on both benchmark datasets. Specifically, HDR-YOLO achieves 81.77% mAP_{50} on NEU-DET and 81.59% on GC10-DET, outperforming all compared lightweight YOLO variants while maintaining competitive computational efficiency.

These results indicate that the proposed hierarchical depth-aware refinement strategy improves feature representation capability with only marginal computational overhead. As a result, HDR-YOLO achieves a favorable balance between detection accuracy and inference efficiency. The obtained latency level satisfies the real-time requirements of industrial surface inspection systems operating at high production line speeds.

It should also be noted that transformer-based detectors such as RT-DETR were not included in the latency comparison. Although such architectures may achieve strong detection performance, their inference latency is generally significantly higher than lightweight CNN-based detectors under the same hardware configuration. Consequently, they are less suitable for real-time industrial inspection scenarios where strict latency constraints must be satisfied. For this reason, our efficiency comparison focuses on lightweight YOLO-based detectors that are commonly adopted in practical industrial vision systems.

3.6 Qualitative Visualization and Attention Analysis

While the quantitative results demonstrate the accuracy and efficiency advantages of the proposed method, qualitative visualizations are further provided to offer intuitive insights into both detection behavior and feature attention mechanisms. As illustrated in Figs. 6 and 7, representative lightweight detectors (YOLOv8n and YOLO11n) together with the Transformer-based RT-DETR-18 are selected for comparison.

On the NEU-DET dataset, HDR-YOLO exhibits more stable and precise localization for fine-grained defects such as scratches, inclusions, rolled-in scales, and crazing, particularly in scenarios where defect boundaries are weak or partially blended with background textures. Similarly, on the GC10-DET dataset, HDR-YOLO demonstrates stronger robustness in detecting complex defect types, including crescent gaps, welding lines, punching holes, oil spots, inclusions, and silk spots. To further improve visual interpretability, zoomed-in views of dense defects and hard cases are additionally provided. As shown in the figures, HDR-YOLO produces clearer boundary delineation, fewer false positives, and more consistent confidence predictions under challenging conditions.

To further analyze the internal decision-making process, class activation map (CAM) visualizations are introduced, as shown in Fig. 8–11. Each group is arranged from left to right as the original image, YOLO11 detection results, grayscale CAM, and CAM overlay, with the upper row corresponding to HDR-YOLO and the lower row to YOLO11.

On the GC10-DET dataset, for the inclusion defect (Fig. 8), HDR-YOLO achieves higher detection accuracy (0.82), while YOLO11 tends to misclassify it as water spot (0.52). The corresponding activation maps reveal that HDR-YOLO concentrates its responses on the true defect regions with sharper and more compact activation patterns, whereas YOLO11 exhibits dispersed and less discriminative responses, leading to semantic confusion. For the water spot category (Fig. 9), HDR-YOLO also improves detection performance (0.52 vs. 0.36) and generates more coherent activation regions, indicating enhanced sensitivity to low-contrast defect patterns.

On the NEU-DET dataset, similar observations can be made. For scratches (Fig. 11), HDR-YOLO achieves higher performance (0.76 / 0.49) compared with YOLO11 (0.68 / 0.27), with activation maps that align more consistently along elongated defect structures. In contrast, YOLO11 produces fragmented and discontinuous responses. For patches (Fig. 10), although both methods achieve comparable accuracy (0.91 vs. 0.90), HDR-YOLO yields more concentrated and noise-suppressed activation patterns, reflecting stronger feature discrimination capability.

Overall, the CAM visualizations demonstrate that HDR-YOLO produces more focused, compact, and semantically meaningful attention distributions, with reduced activation on irrelevant background regions. This behavior provides strong evidence that the proposed hierarchical depth-aware refinement strategy effectively enhances both local texture sensitivity in shallow layers and global semantic discrimination in deeper layers.

Compared with existing lightweight detectors and Transformer-based methods, HDR-YOLO shows more consistent activation responses and improved boundary awareness, particularly for small, low-contrast, and irregular-shaped defects. Nevertheless, certain challenging cases remain, including extremely small defects and visually ambiguous patterns such as inclusions and oil spots, where feature overlap may still lead to occasional misclassification. These observations highlight the intrinsic difficulty of fine-grained industrial defect detection and suggest directions for future improvement.

Overall, the qualitative and attention-based analyses consistently support that aligning refinement mechanisms with hierarchical feature roles leads to more reliable localization and stronger defect discrimination capability, further validating the effectiveness of HDR-YOLO in practical industrial inspection scenarios.

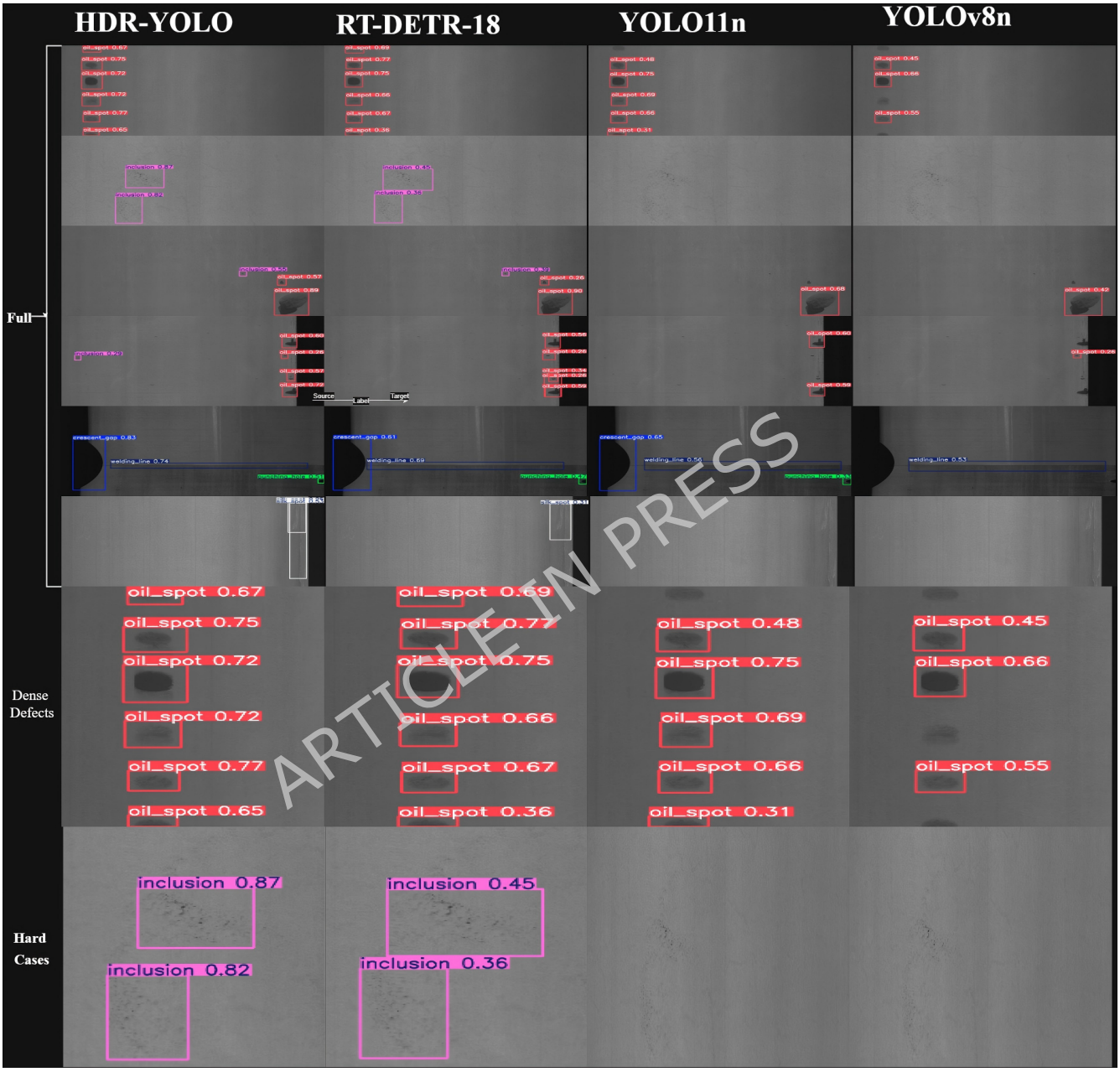

Figure 6. Visualization comparison of detection results of different algorithms on the GC10-DET dataset. The figure shows the performance of multiple methods in detecting various steel surface defects, including Crescent gap, Welding line, Punching hole, Oil spot, Inclusion, and Silk spot, along with dense defects and hard cases.

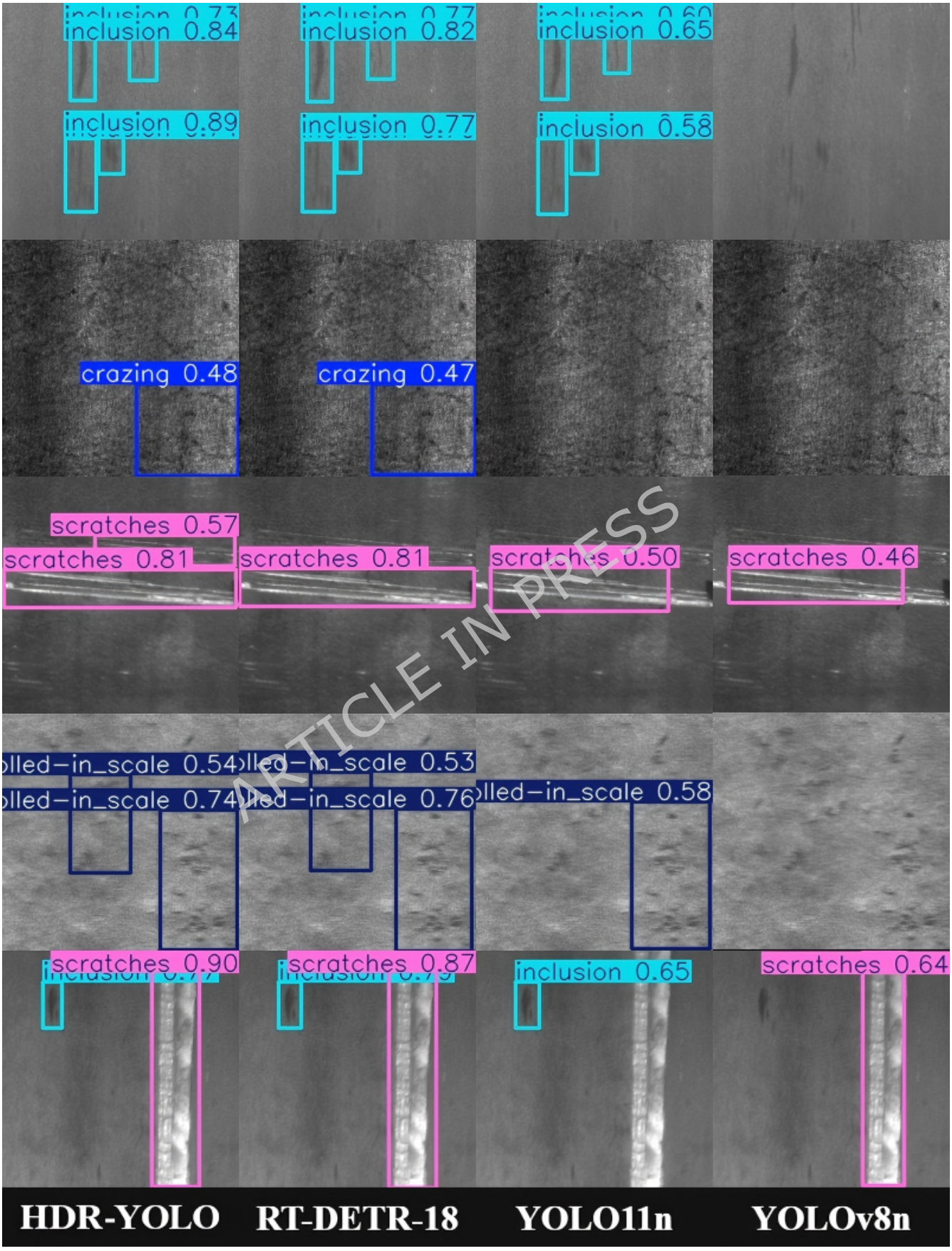

Figure 7. Visualization comparison of detection results of different algorithms on the NEU-DET dataset. The figure illustrates the detection performance of multiple methods on diverse steel surface defects, including Scratches, Inclusion, Rolled-in Scale, and Crazing.

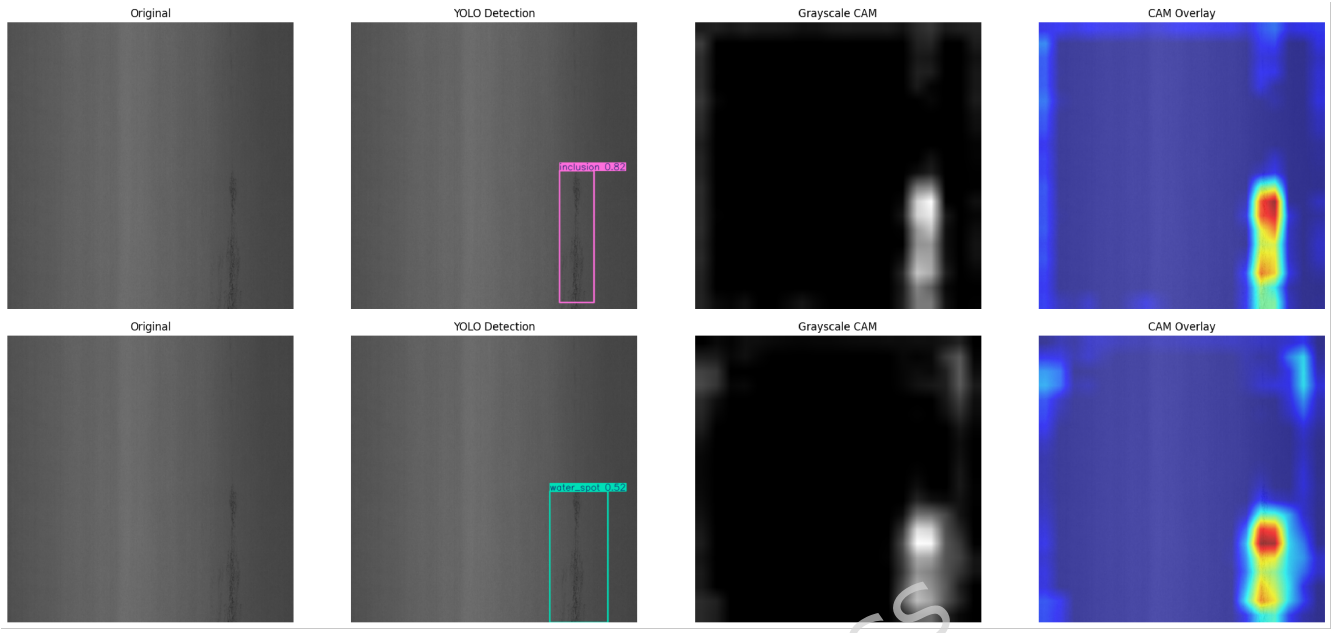

Figure 8. GC10-DET Inclusion Detection and Attention Visualization

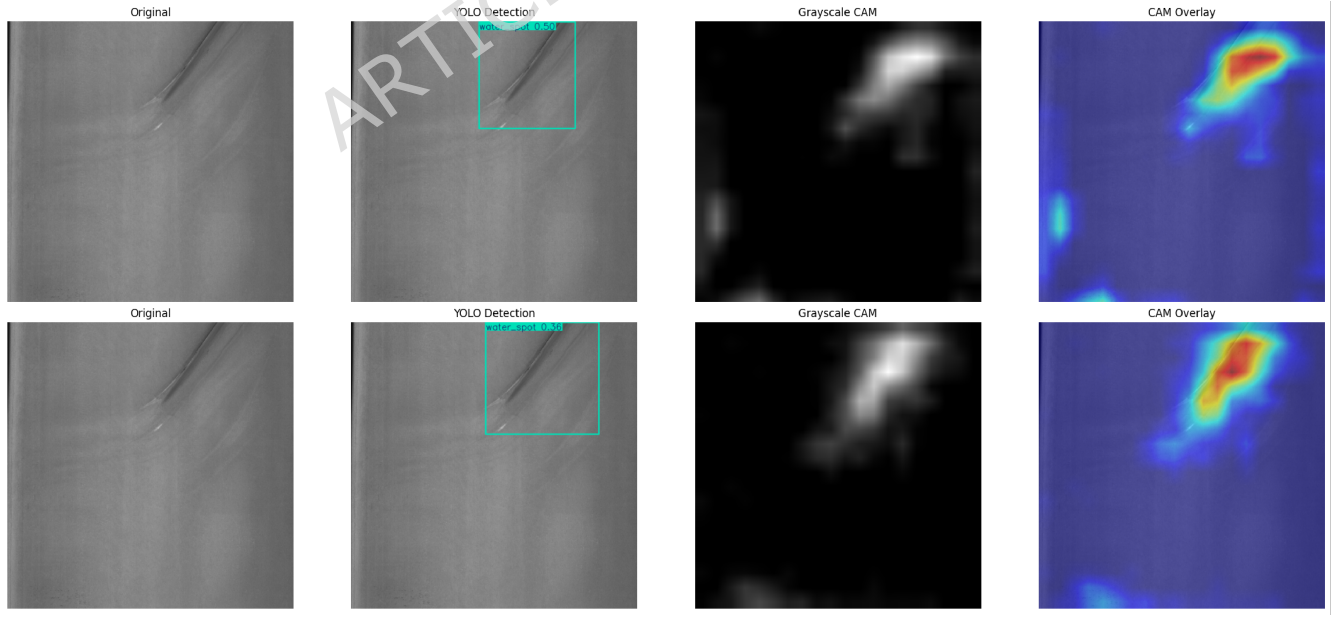

Figure 9. GC10-DET Water Spot Detection and Attention Visualization

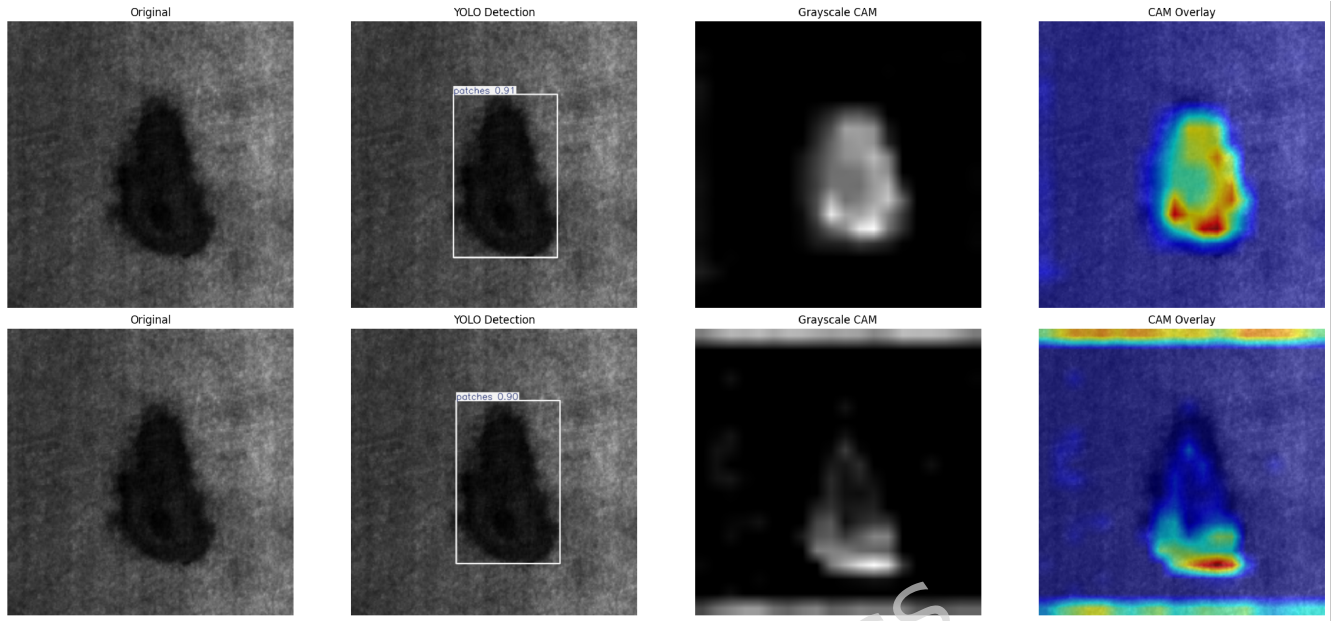

Figure 10. NEU-DET Patches Detection and Attention Visualization

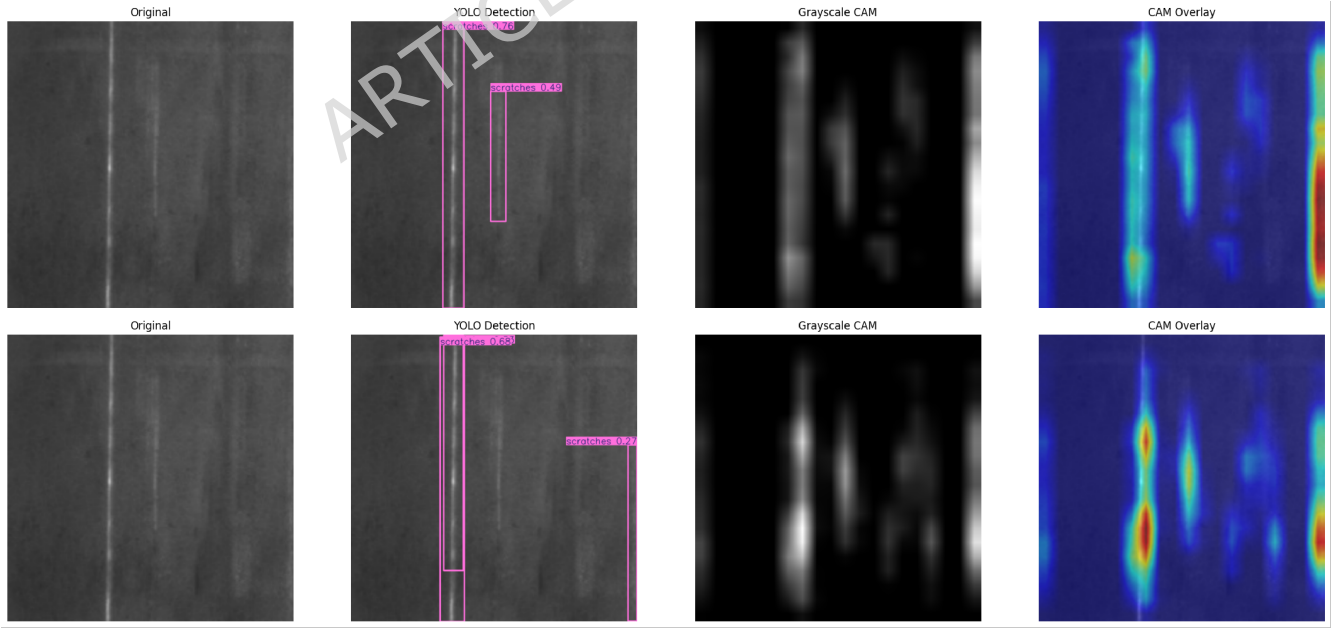

Figure 11. NEU-DET Scratches Detection and Attention Visualization

4 Discussion

This study proposes HDR-YOLO, a lightweight object detection framework designed for accurate and efficient steel surface defect inspection. The core idea of the proposed method is a hierarchical depth-aware refinement strategy that applies differentiated feature enhancement mechanisms to shallow and deep network layers. Since shallow layers mainly encode high-resolution spatial details and texture information, whereas deeper layers capture more abstract semantic representations, aligning the refinement process with these hierarchical roles enables the network to utilize feature representations more effectively. As a result, HDR-YOLO can better capture fine-grained defect textures while maintaining strong semantic discrimination capability.

Extensive experiments conducted on two widely used steel surface defect datasets, NEU-DET and GC10-DET, demonstrate that HDR-YOLO consistently outperforms several mainstream lightweight detectors. These datasets are well-recognized benchmarks containing diverse defect categories with varying scales, textures, and appearance characteristics, enabling comprehensive and fair comparisons with existing methods. The proposed framework achieves superior detection accuracy while maintaining a compact model size and competitive inference efficiency, indicating a favorable balance between accuracy and computational cost.

To further validate the effectiveness of the proposed design, both ablation studies and attention visualizations are conducted. The ablation results confirm that each component, including the shallow-layer QFC module, deep-layer QBF module, and the KAN-based feature transformation, contributes positively to performance improvement. In particular, the sensitivity analysis on window size demonstrates that a moderate receptive field (7×7) provides the best trade-off between local detail preservation and contextual modeling, while the integration of KAN enhances nonlinear representation capability without increasing computational complexity.

Moreover, class activation map (CAM) visualizations provide intuitive insights into the internal decision-making process of the model. Compared with baseline detectors, HDR-YOLO produces more concentrated and semantically aligned activation regions, with reduced responses on irrelevant background areas. This indicates that the proposed hierarchical refinement strategy effectively improves both attention focus and feature discrimination, which directly contributes to the observed performance gains.

From an industrial perspective, automated visual inspection systems must operate reliably under challenging production conditions. In real manufacturing environments, factors such as illumination variation, mechanical vibration, sensor noise, and surface reflectance changes may affect image quality and defect visibility. Previous studies have reported that illumination variation can significantly influence the recognition accuracy of surface defects in rolled metal products[8]. Compared with traditional handcrafted-feature approaches that are more sensitive to such disturbances, deep learning models generally provide stronger robustness due to their hierarchical feature learning capability. In HDR-YOLO, the proposed depth-aware refinement mechanism enhances both local texture preservation in shallow layers and semantic discrimination in deeper layers, enabling the model to maintain stable detection performance even under moderate appearance variations.

For practical deployment in industrial production environments, several prerequisites should be satisfied. First, the image acquisition system should provide stable imaging conditions and sufficient spatial resolution to capture fine defect textures. Second, relatively consistent illumination conditions are recommended to reduce excessive appearance variations on reflective metal surfaces. Third, the inspection platform should provide adequate computing resources capable of supporting real-time inference. Under these conditions, the lightweight architecture and high inference efficiency of HDR-YOLO enable reliable deployment in automated inspection systems operating on high-speed manufacturing lines.

Nevertheless, several limitations remain in the current study. Although NEU-DET and GC10-DET provide representative benchmarks, the experimental evaluation is limited to these two publicly available datasets. Further validation on additional datasets, different material surfaces, and real production-line data would provide a more comprehensive assessment of generalization capability and robustness. In addition, extremely small, low-contrast, or visually ambiguous defects remain challenging, indicating the need for further improvement in fine-grained feature modeling.

Overall, this study demonstrates that explicitly aligning feature refinement mechanisms with hierarchical feature roles can effectively improve the accuracy–efficiency trade-off in lightweight object detectors. The combination of hierarchical refinement, adaptive receptive field design, and enhanced feature transformation provides a principled and effective solution for robust defect detection. The proposed design paradigm may offer valuable insights for developing efficient and reliable visual inspection systems in intelligent manufacturing and other real-time industrial vision applications.

5 Conclusion

Achieving a favorable balance between detection accuracy and real-time efficiency remains a fundamental challenge in metal surface defect inspection, particularly for lightweight neural detectors deployed in industrial environments. To address this challenge, this work proposes HDR-YOLO, a hierarchical depth-aware refinement framework built upon YOLO11 that explicitly exploits the distinct representational roles of features across network depths.

Unlike conventional lightweight detectors that apply uniform refinement or attention mechanisms across all layers, HDR-YOLO introduces differentiated refinement strategies for shallow and deep feature maps. Window-based query-focused refinement is applied to shallow layers to preserve fine-grained texture and edge information under strict computational constraints, while deeper layers perform global semantic fusion to enhance contextual modeling and defect discrimination capability.

Extensive experiments conducted on the GC10-DET and NEU-DET benchmark datasets demonstrate that the proposed hierarchical refinement strategy leads to consistent performance improvements. HDR-YOLO achieves mAP@50 and mAP@50:95 scores of 81.59% and 50.79% on GC10-DET, and 81.77% and 52.59% on NEU-DET, outperforming the YOLO11n baseline while maintaining a lightweight architecture with only 2.73M parameters and 7.2 GFLOPs. Compared with several state-of-the-art lightweight detectors, HDR-YOLO provides improved localization accuracy and stronger discriminative capability across diverse defect categories, demonstrating its practical potential for real-time industrial inspection tasks.

Despite these advantages, detecting extremely small, low-contrast, or highly irregular defects remains challenging due to their weak visual characteristics. Future work will explore more adaptive depth-aware refinement mechanisms, including dynamic receptive-field attention, lightweight dynamic convolution, and hierarchy-aware feature interaction. In addition, integrating emerging neural architectures such as Transformer[35] and Mamba-based sequence modeling[36] may further enhance feature interaction capability and improve the robustness and generalization of lightweight industrial defect detectors.

Data Availability

Due to project-related constraints, the full source code is not publicly released at this stage. However, the implementation corresponding to the results reported in this study can be shared for academic purposes under reasonable request, ensuring reproducibility.

References

1. Maleki, E. *et al.* Surface post-treatments for metal additive manufacturing: Progress, challenges, and opportunities. *Addit. Manuf.* **37**, 101619 (2021).
2. Ragab, M. G. *et al.* A comprehensive systematic review of yolo for medical object detection (2018 to 2023). *IEEE Access* **12**, 57815–57836 (2024).
3. Wang, Q. *et al.* A casting surface dataset and benchmark for subtle and confusable defect detection in complex contexts. *IEEE Sensors J.* **24**, 16721–16733 (2024).
4. Mery, D. Aluminum casting inspection using deep object detection methods and simulated ellipsoidal defects. *Mach. Vis. Appl.* **32**, 72 (2021).
5. Karimi, N., Mishra, M. & Lourenço, P. B. Automated surface crack detection in historical constructions with various materials using deep learning-based yolo network. *Int. J. Archit. Herit.* **19**, 581–597 (2025).
6. Ren, Z., Fang, F., Yan, N. & Wu, Y. State of the art in defect detection based on machine vision. *Int. J. Precis. Eng. Manuf. Technol.* **9**, 661–691 (2022).
7. Tian, J. H. *et al.* An improved yolov5n algorithm for detecting surface defects in industrial components. *Sci. Reports* **15**, 9756 (2025).
8. Maruschak, P., Konovalenko, I. & Osadtsa, Y. Surface defects of rolled metal products recognised by a deep neural network under different illuminance levels and low-amplitude vibration. *The Int. J. Adv. Manuf. Technol.* **139**, 449–464 (2025).
9. Hütten, N. *et al.* Deep learning for automated visual inspection in manufacturing and maintenance: A survey of open-access papers. *Appl. Syst. Innov.* **7**, 11 (2024).
10. Alzubaidi, L. *et al.* Review of deep learning: concepts, cnn architectures, challenges, applications, future directions. *J. Big Data* **8**, 53 (2021).
11. Agarwal, A. & Patni, K. Lung cancer detection and classification based on alexnet cnn. In *Proceedings of the 6th International Conference on Communication and Electronics Systems (ICCES)* (2021).
12. Shah, S. R. *et al.* Comparing inception v3, vgg16, vgg19, cnn, and resnet50: A case study on early detection of a rice disease. *Agronomy* **13**, 1633 (2023).
13. Xu, W., Fu, Y.-L. & Zhu, D. Resnet and its application to medical image processing: research progress and challenges. *Comput. Methods Programs Biomed.* **240**, 107660 (2023).

14. Xie, X. *et al.* Oriented r-cnn for object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)* (2021).

15. Xu, X. *et al.* Crack detection and comparison study based on faster r-cnn and mask r-cnn. *Sensors* **22**, 1215 (2022).

16. Bi, X., Hu, J., Xiao, B., Li, W. & Gao, X. Iemask r-cnn: Information-enhanced mask r-cnn. *IEEE Transactions on Big Data* **9**, 688–700 (2022).

17. Arwidiyarti, D. Single shot multibox detector (ssd) in object detection: A review. *IJACI: Int. J. Adv. Comput. Informatics* **1**, 118–127 (2025).

18. Jiang, P., Ergu, D., Liu, F., Cai, Y. & Ma, B. A review of yolo algorithm developments. *Procedia Comput. Sci.* **199**, 1066–1073 (2022).

19. Cheng, T. *et al.* Yolo-world: Real-time open-vocabulary object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 16901–16911 (2024).

20. Qu, Y. *et al.* Optimization algorithm for steel surface defect detection based on pp-yoloe. *Electronics* **12**, 4161 (2023).

21. Wang, C. *et al.* Gold-yolo: Efficient object detector via gather-and-distribute mechanism. *Adv. Neural Inf. Process. Syst.* **36**, 51094–51112 (2023).

22. Terven, J., Córdova-Esparza, D. M. & Romero-González, J. A. A comprehensive review of yolo architectures in computer vision: From yolov1 to yolov8 and yolo-nas. *Mach. Learn. Knowl. Extr.* **5**, 1680–1716 (2023).

23. Hussain, M. Yolo-v1 to yolo-v8, the rise of yolo and its complementary nature toward digital manufacturing and industrial defect detection. *Machines* **11**, 677 (2023).

24. He, L. H., Zhou, Y. Z., Liu, L., Cao, W. & Ma, J. H. Research on object detection and recognition in remote sensing images based on yolov11. *Sci. Reports* **15**, 14032 (2025).

25. Choromanski, K., Likhosherstov, V., Dohan, D. *et al.* Rethinking attention with performers. In *Proceedings of the International Conference on Learning Representations (ICLR)* (2021).

26. Sinha, D. & El-Sharkawy, M. Thin mobilenet: An enhanced mobilenet architecture. In *Proceedings of the IEEE 10th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference (UEMCON)*, 280–285 (2019).

27. Rahimzadeh, M. & Attar, A. A modified deep convolutional neural network for detecting covid-19 and pneumonia from chest x-ray images based on the concatenation of xception and resnet50v2. *Informatics Medicine Unlocked* **19**, 100360 (2020).

28. Wang, Y., Wang, H. & Xin, Z. Efficient detection model of steel strip surface defects based on yolo-v7. *IEEE Access* **10**, 133936–133944 (2022).

29. Wang, K., Teng, Z. & Zou, T. Metal defect detection based on yolov5. In *Journal of Physics: Conference Series*, vol. 2218, 012050 (2022).

30. Jocher, G. *et al.* ultralytics/yolov5: v3.0 (2020). Zenodo.

31. Sohan, M., Thotakura, T. S. R. & Reddy, C. V. R. A review on yolov8 and its advancements. In *Proceedings of the International Conference on Data Intelligence and Cognitive Informatics* (2024).

32. Wang, A. *et al.* Yolov10: Real-time end-to-end object detection. *Adv. Neural Inf. Process. Syst.* **37**, 107984–108011 (2024).

33. Sun, Y., Yan, H., Shang, Z. & Yang, M. Mch-yolov12: Research on surface defect detection algorithm for aluminum profiles based on improved yolov12. *Sensors* **25**, 5389 (2025).

34. Xue, R., Hua, S. & Xu, H. Feci-rtdetr: A lightweight unmanned aerial vehicle infrared small target detector algorithm based on rt-detr. *IEEE Access* (2025).

35. Vaswani, A. *et al.* Attention is all you need. *Adv. Neural Inf. Process. Syst.* **30** (2017).

36. Gu, A. & Dao, T. Mamba: Linear-time sequence modeling with selective state spaces. In *First Conference on Language Modeling* (2024).

Acknowledgements (not compulsory)

This research work was supported by Universiti Malaya under Project Number UMREG025-2025.

Funding

This research work was supported by Universiti Malaya under Project Number UMREG025-2025.

Author Contributions Statement

Q.Q. conceived and designed the study, developed the methodology, conducted all experiments, analyzed the results, and wrote the manuscript. A.S.B.M.K. supervised the research and provided critical feedback on the manuscript. N.B.I., H.L., L.F., Z.Y., J.L., and C.Z. provided guidance and advice on the research design and data interpretation. All authors reviewed the manuscript.