Computers, Materials & Continua

Tech Science Press

https://doi.org/10.32604/cmc.2026.080580

ARTICLE

An Efficient DETR-Based Framework for Small Target-Aware Industrial Surface Defect Detection

Yuting Wang, Bingyang Guo, Jianing Duan and Ruiyun Yu*

Software College, Northeastern University, Shenyang, China

*Corresponding Author: Ruiyun Yu. Email: yury@mail.neu.edu.cn

Received: 12 February 2026; Accepted: 22 May 2026; Published: 23 July 2026

ABSTRACT: Industrial surface defect detection requires accurate localization of small and weak-boundary defects under tight runtime constraints for on-line inspection. This paper presents an efficient DETR-style defect detector with three components. First, we build a hybrid feature extractor by coupling a ConvNeXt-T backbone with a lightweight Feature Pyramid Network (FPN) to strengthen multi-scale representations for small and subtle defects, thereby improving detection performance in challenging industrial environments. Second, to address the high computational cost of original DETR, we adopt multi-scale deformable attention to replace the quadratic-cost global self-attention mechanism, substantially improving efficiency. Third, to improve per-class robustness on hard defect categories with negligible overhead, we incorporate a class-reweighted focal loss (focal loss with no-object down-weighting together with effective-number reweighting) for classification. Experiments on NEU-DET show that our method achieves 85.0% mAP@0.5 and 47.5% mAP@[0.5:0.95], improving over the original DETR baseline (81.0% mAP@0.5). Under the same runtime setup (NVIDIA 3090, FP16, batch size 1, 512 × 512), latency is reduced from 71.4 to 53.3 ms with 18.8 img/s throughput. These results demonstrate that our framework achieves a superior accuracy-efficiency trade-off for near real-time on-line inspection (approximately 18.8 img/s, 53.3 ms latency) in medium-speed large-scale industrial manufacturing scenarios.

KEYWORDS: Detection transformer; surface defect detection; feature pyramid network; industrial defects

1 Introduction

With the rapid advancement of modern industrial manufacturing, the demand for higher product quality has become increasingly stringent—particularly in high-precision industries such as automotive, aerospace, and electronics. Consequently, surface defect detection has become a critical step in ensuring product quality [1]. Surface defects can degrade visual appearance and may even cause structural failures during use, posing serious risks to safety and reliability. Traditional defect detection relies on manual inspection or conventional computer-vision pipelines. Although manual inspection remains common, it is labor-intensive, inefficient, and prone to human error, making it unsuitable for modern large-scale production [2]. Conventional automated detection methods often depend on handcrafted features and image-processing heuristics. Such methods struggle with the complexity and variability of industrial surface defects, including irregular shapes, diverse textures, and large scale differences [3–5]. Therefore, improving both the accuracy and efficiency of surface defect detection remains a pressing challenge for industrial intelligence.

In recent years, Convolutional Neural Networks (CNNs) and variants such as Faster R-CNN have been widely adopted for surface defect detection [6,7]. Owing to their strong representation learning capability,

CNNs have become the dominant approach in image classification and object detection. However, these methods still face limitations, especially for small defects and imbalanced target distributions [8]. CNN-based detectors may miss fine-grained cues of small defects, and their localization performance for small objects can be brittle because region proposals and hyperparameters often require empirical tuning. These issues make performance sensitive to defect types and scales, leading to practical bottlenecks [9–11].

The emergence of Transformer models has led to significant breakthroughs in object detection, building on their success in Natural Language Processing (NLP). By using self-attention, Transformers capture long-range dependencies and global context. Recently, Transformers have been introduced to computer vision, among which the Detection Transformer (DETR) provides a set-prediction formulation for object detection [12]. DETR directly predicts a fixed-size set of bounding boxes and labels and leverages self-attention to model global interactions, yielding promising results. However, applying the original DETR model to industrial surface defect detection remains challenging due to three factors: (1) limited performance on small-scale defects, which can lead to missed detections [13]; (2) high computational cost because global self-attention scales quadratically with the number of spatial tokens, hindering real-time deployment [12,14]; and (3) unstable per-class performance on hard-to-detect defect types (e.g., weak-boundary or small/linear defects), where unequal category difficulty and limited positive matches can cause some classes to be under-optimized by standard losses.

To address these challenges, we propose an optimised DETR-based framework for industrial surface defect detection. We are transparent that the individual components—ConvNeXt, FPN, deformable attention, focal loss, and effective-number reweighting—are each established techniques from prior work. Our contribution consists of three elements:

1. A domain-specific investigation establishing that the combination of ConvNeXt-T backbone, lightweight FPN (three 1×1 and three 3×3 convolutional layers, no additional attention parameters), and multi-scale deformable attention addresses the three simultaneous constraints of industrial defect detection—small-object sensitivity, computational efficiency, and category-level difficulty imbalance—that are not resolved by any existing Deformable DETR variant or defect detection framework.
2. A targeted domain adaptation of the DETR classification loss, comprising focal hard-example emphasis, effective-number reweighting for difficulty-based (not frequency-based) class imbalance, and a technically motivated correction of the loss normalisation from N_q (total queries) to N_{pos} (matched positive queries) for correct gradient scaling.
3. Systematic empirical validation of these design choices through component-level ablation, size-stratified AP analysis, cross-dataset transfer, multi-seed stability testing, and multi-hardware efficiency measurement—providing a validated, reproducible baseline for defect-detection-specific DETR design.

The structure of this study is organized as follows: Section 2 presents a literature review of existing surface defect detection technologies, with a focus on deep learning-based methods and their associated challenges. Section 3 details the proposed optimization approach, including the model architecture, training strategies, and loss-function design. Section 4 describes the experimental setup and analyzes the results, with an emphasis on comparing the model's performance before and after optimization in terms of accuracy and computational efficiency. Finally, Section 5 summarizes the findings and outlines potential directions for future work.

2 Related Work

Surface defect detection in industrial manufacturing has evolved from manual inspection and hand-crafted feature pipelines to data-driven deep learning approaches. We organize the literature into three

streams: CNN-based detectors, Transformer-based detectors (with a focus on DETR variants), and studies specifically addressing small-object and hard-defect detection, from which we identify the research gaps our work aims to fill.

2.1 CNN-Based Defect Detection

Deep Convolutional Neural Networks (CNNs) have become the mainstream approach for automated surface defect detection [15]. CNN-based models are generally divided into two-stage and one-stage detectors, each with distinct trade-offs between accuracy and speed in industrial deployment scenarios.

Two-stage detectors, exemplified by Faster R-CNN [6], utilize a Region Proposal Network (RPN) to first identify candidate regions, which are then classified. While effective for clearly defined defects, they suffer from two drawbacks: (1) their two-stage design incurs high computational latency, making real-time deployment difficult [7], and (2) they often struggle with localizing small-scale defects due to fixed anchor configurations and spatial detail loss in deeper feature maps [9,10]. Recent domain-specific adaptations of Faster R-CNN—such as defect detection on rail surfaces [6], wafer substrates [7], and resistance spot welds [8]—have demonstrated the framework's flexibility but highlight persistent limitations in handling weak-boundary and micro-scale defects.

One-stage detectors, such as the YOLO series, treat detection as a direct regression problem, achieving significantly faster inference [16]. A substantial body of recent work has applied and extended YOLO architectures to industrial defect inspection. Zhou et al. [17] proposed YOLO-DD, an improved YOLOv5 variant with enhanced feature extraction modules for metallic surface defect detection. Yuan et al. [18] introduced a faster metallic surface defect detector using channel shuffling within a deep learning backbone. More recently, YOLOv8-based frameworks have been applied to steel surface inspection and semiconductor wafer defect detection, demonstrating improved small-target sensitivity through refined anchor-free designs and richer neck architectures [19]. Vasan et al. [20] presented an AI-driven approach for micro defect detection in aerospace-grade metal surfaces within smart manufacturing environments, leveraging synthetic data augmentation to improve robustness on rare and small defects—an important contribution that underscores both the domain demand and the challenge of small-target generalization. Despite these advances, both one-stage and two-stage CNN detectors remain limited in capturing long-range contextual dependencies and exhibit degraded performance on hard-to-detect categories with irregular shapes or weak boundaries [21].

2.2 Transformer-Based Defect Detection

Inspired by the success of Transformers in Natural Language Processing [14], Vision Transformers and their detection-specific variants have emerged as powerful alternatives to CNN-based detectors. The Detection Transformer (DETR) [12] revolutionized object detection by eliminating hand-crafted components such as anchor generation and Non-Maximum Suppression (NMS). Using a Transformer encoder–decoder with bipartite matching, DETR directly predicts a fixed set of bounding boxes and class labels. However, the original DETR suffers from quadratic computational complexity due to global self-attention, slow convergence, and limited performance on small objects.

Several DETR variants have been proposed to overcome these limitations. Deformable DETR [22] introduced deformable attention, which restricts each query to attend to a small set of learnable sampling points rather than all spatial tokens, reducing complexity to near-linear and significantly improving small-object detection. Subsequent works refined the training dynamics of DETR-style models: DAB-DETR [23] reformulated object queries as dynamic anchor boxes, enabling more interpretable and stable decoding; DN-DETR [24] introduced denoising training, accelerating convergence by injecting noisy ground-truth labels as auxiliary queries; and DINO-DETR [25] combined contrastive denoising with a look-forward-twice

scheme, achieving state-of-the-art accuracy among DETR variants. RT-DETR [26] further addressed real-time deployment by designing a hybrid encoder that decouples intra-scale interaction and cross-scale fusion, enabling end-to-end detection at competitive speeds without NMS. While these models advance the general detection frontier, their direct applicability to industrial defect scenarios—characterized by small targets, weak boundaries, and category-level difficulty imbalance—has not been fully explored.

Within the industrial defect detection domain, several recent works have demonstrated the promise of Transformer-based frameworks. Yu et al. [19] proposed a composite multi-stage architecture combining the Swin Transformer with hierarchical feature aggregation for sewer pipe defect recognition, achieving improved detection of complex structural damage patterns. Ayon et al. [27] introduced a Learnable Memory Vision Transformer (LMViT) for steel surface defect detection, demonstrating superior performance over traditional CNN baselines by capturing long-range spatial dependencies. Ye et al. [28] integrated CNN modules with deformable attention to improve small-defect recognition on aluminum profiles, highlighting the benefit of local–global feature fusion for industrial inspection. Zhang et al. [29] proposed a wavelet-guided Transformer that leverages frequency-domain cues to direct attention toward defect-relevant regions on steel surfaces. Wang et al. [13] developed a hybrid Transformer architecture specifically tailored for surface defect detection that couples CNN-extracted local features with Transformer-encoded global context, reporting efficiency improvements over vanilla DETR for industrial scenarios. These studies collectively validate the applicability of Transformer-based detectors to industrial surface inspection, yet—as detailed below—important gaps remain that our work aims to fill.

2.3 Small-Object and Hard-Defect Detection

Small-object and hard-to-detect defect detection constitutes a distinct challenge in industrial vision. From an architectural perspective, multi-scale feature representation is widely recognized as essential: Feature Pyramid Networks (FPN) aggregate features across spatial scales and form the backbone neck for most modern detectors. In the DETR family, Deformable DETR [22] provides multi-scale deformable attention across FPN-derived feature maps, substantially improving small-object recall. Feng et al. [30] demonstrated the value of hybrid Transformer architectures for dense small-object detection in UAV imagery, combining CNN local features with Transformer global modeling to handle targets with weak texture and irregular spatial distributions. Furthermore, recent advancements have explicitly targeted the inherent localization challenges of micro-scale and weak-boundary targets in complex scenes. For instance, Hoanh and Pham [31] introduced a contrast-aware and uncertainty-optimized Transformer model for small object detection in high-resolution images. Their work highlights the critical importance of enhancing local feature contrast and managing bounding-box uncertainty for hard-to-localize targets-a philosophy that perfectly aligns with our motivation for integrating a large-kernel ConvNeXt backbone and an effective-number reweighted focal loss to address the category-level difficulty imbalance of weak-boundary industrial defects.

From a training perspective, class imbalance and category-difficulty imbalance are recognized obstacles in industrial datasets. Focal loss [32] addresses foreground–background imbalance by down-weighting easy negatives. Class-balanced loss based on effective number of samples [33] extends this to multi-class imbalance by reweighting per-class contributions according to the diminishing marginal returns of additional data—a strategy directly applicable to industrial defect datasets where certain defect types (e.g., crazing, rolled-in scale) are inherently harder to detect due to their subtle visual signatures.

2.4 Research Gap

Despite these advancements, a review of the literature reveals two critical gaps that hinder the industrial deployment of DETR-based models. Notably, these gaps correspond to the three challenges identified in Section 1: Gap 1 addresses challenges 1 and 2 (architecture), while Gap 2 addresses challenge 3 (training).

1. Lack of an Optimized Hybrid Architecture for Industrial Defects (Challenges 1 & 2): While recent works such as Yu et al. [19], Ye et al. [28], and Wang et al. [13] demonstrated the potential of hybrid and composite Transformer models, the optimal architecture for industrial surface defects remains underexplored. In particular, the synergistic combination of a ConvNeXt backbone—which incorporates Vision Transformer design principles within a pure CNN—with an efficient FPN within the Deformable DETR framework has not been systematically investigated for detecting small and weak-boundary industrial defects while simultaneously maintaining computational efficiency. Moreover, while DINO-DETR and RT-DETR advance detection accuracy and speed in general settings, they have not been specifically evaluated or adapted for the challenging distribution of industrial defect benchmarks such as NEU-DET.
2. Limited Optimization for Hard Defect Categories (Challenge 3): Existing DETR-based defect detection studies rarely conduct targeted optimization for hard-to-detect categories, such as small-scale, weak-boundary, or linear-structure defects. While Feng et al. [30] and Bakirci and Bayraktar [34] addressed small-target detection challenges in UAV imagery and aerospace manufacturing, respectively, the techniques for handling category-level difficulty imbalance within the DETR training paradigm remain underexplored for industrial defect datasets. Crucially, unequal category difficulty can yield uneven per-class optimization even without a severe long-tailed distribution. While general reweighting methods exist [32,33], current Transformer-based defect detection works [19,27,29] do not explicitly incorporate lightweight loss enhancements to stabilize per-class learning for these difficult defect categories.

This study aims to fill these specific gaps. We propose a novel framework that (1) integrates a ConvNeXt backbone and a lightweight FPN with deformable attention to create a highly efficient hybrid architecture tailored for industrial surface defect detection, and (2) introduces an effective-number reweighted focal loss as a lightweight training enhancement to improve per-class robustness for hard-to-detect defect types.

3 Method

This section describes the proposed framework. We are explicit about the provenance of each component: multi-scale deformable attention is adopted directly from Deformable DETR [22]; ConvNeXt is adopted from Liu et al. [35]; FPN is adopted from Lin et al. [36]; focal loss from Lin et al. [32]; effective-number reweighting from Cui et al. [33]. Our contribution lies in (1) establishing, through the domain-specific gap analysis in Section 2.3, that this particular combination addresses three simultaneous constraints not resolved by existing configurations; (2) making targeted domain adaptations to the loss—specifically, effective-number reweighting for difficulty-based imbalance and normalisation by matched positive queries instead of total queries; and (3) providing systematic empirical validation confirming that each design choice is justified and that the combined framework transfers across datasets and hardware.

3.1 Overall Framework Architecture

The overall architecture of our proposed model is illustrated in Fig. 1. It consists of three components (two architectural modules and one lightweight training enhancement):

1. Hybrid Backbone (Architectural): We design a hybrid feature extractor by integrating a ConvNeXt backbone with a Feature Pyramid Network (FPN) to capture fine-grained textures and robust multi-scale features, which are critical for detecting small and subtle defects.

2. Deformable Transformer (Architectural): We adopt the Deformable DETR [22] encoder–decoder as our detection head. Its multi-scale deformable attention avoids the $\mathcal{O}(n^2)$ complexity of global attention, improving efficiency while maintaining accuracy.
3. Weighted Loss (Training Enhancement): We incorporate an effective-number class-weighted focal loss to improve per-class robustness on hard-to-detect defect types (difficulty imbalance) with negligible computational overhead.

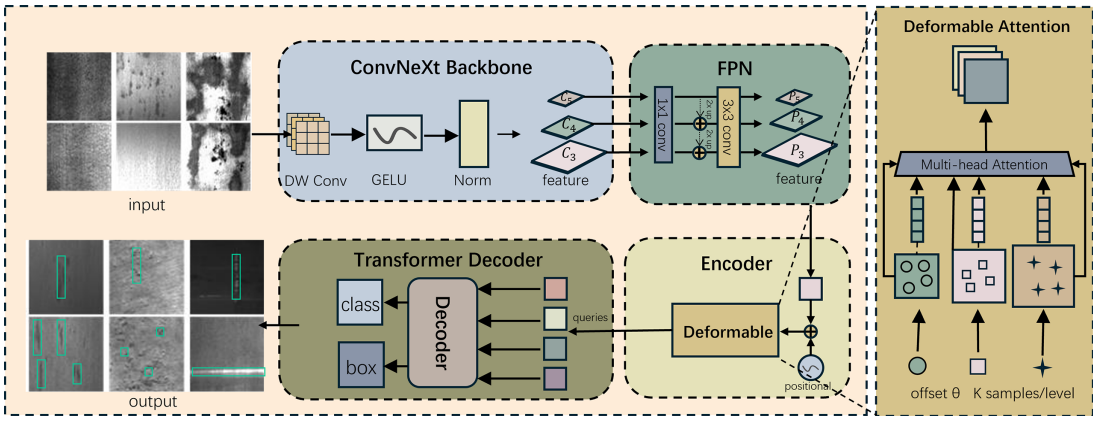

Figure 1: The architecture of our proposed framework, designed to solve small-object detection, efficiency, and imbalance challenges. (Left) A ConvNeXt backbone extracts multi-stage features (C_3, C_4, C_5). (Center-Top) An FPN fuses these features using 1×1 convs, upsampling, element-wise addition ($\oplus$), and 3×3 convs to produce robust multi-scale features (P_3, P_4, P_5). (Center-Bottom) The Deformable Transformer (Encoder–Decoder) processes these features using efficient attention and object queries to predict bounding boxes and classes.

3.2 Hybrid Backbone for Feature Extraction

Standard backbones like ResNet often struggle to capture the fine-grained local textures and variable scales of industrial defects. To address this, we design a hybrid backbone that synergizes the local feature power of ConvNeXt with the multi-scale capabilities of an FPN.

3.2.1 ConvNeXt for Local Feature Extraction

We select ConvNeXt-T [35] as the backbone for three domain-specific reasons: (1) its 7×7 depthwise convolution kernels provide a substantially larger local receptive field than ResNet-50's 3×3 kernels, enabling the model to capture the multi-pixel edge transitions, shallow scratches, and textural patterns that define industrial surface defects; (2) its layer normalisation and GELU activations improve training stability on the relatively small NEU-DET dataset; and (3) it outperforms Swin-T on the target task, without introducing self-attention computation at the backbone stage—an overhead that would compound the existing attention cost in the Deformable DETR encoder.

ConvNeXt modernizes CNN design by incorporating principles from Vision Transformers (e.g., larger kernel sizes, layer normalization). This results in a pure CNN that excels at extracting rich and robust local features, such as the subtle edges, scratches, and textural variations that define industrial defects. The ConvNeXt backbone processes the input image I and generates a hierarchy of feature maps $\{C_3, C_4, C_5\}$ from its intermediate stages.

3.2.2 Feature Pyramid Network (FPN) for Small Defects

Given the wide size range of defects, we integrate an FPN [36] to strengthen the multi-scale features fed into the Transformer. As shown in Fig. 1, the FPN builds a rich, multi-scale feature pyramid using a top-down pathway with lateral connections.

First, the backbone features $\{C_3, C_4, C_5\}$, which have varying channel dimensions, are passed through a 1×1 convolution to produce features $\{L_3, L_4, L_5\}$ with a unified channel dimension (e.g., 256).

Second, a top-down pathway fuses these features. For the top-most level, we directly apply a 3×3 convolution to L_5 to generate the final P_5 map. For lower levels, the feature from the level above is upsampled and fused with the current lateral feature via element-wise addition. This sum is then processed by a final 3×3 convolution to mitigate aliasing artifacts from upsampling and produce the final feature map:

$$L_i = \mathrm{Conv}_{1\times1}(C_i), \tag{1}$$

$$M_i = L_i + \mathrm{Upsample}(P_{i+1}) \tag{2}$$

$$P_i = \mathrm{Conv}_{3\times3}(M_i) \tag{3}$$

This mechanism allows the model to leverage semantically strong, low-resolution features (from P_5) alongside spatially precise, high-resolution features (from P_3), which is essential for detecting small defects. The resulting multi-scale features $\{P_3, P_4, P_5\}$ are then fed into the Transformer.

3.3 Multi-Scale Deformable Attention

To address the high computational cost of the original DETR, we adopt the Multi-Scale Deformable Attention mechanism from Deformable DETR [22].

Instead of global self-attention over all pairs of n spatial tokens (which has $\mathcal{O}(n^2)$ complexity), deformable attention restricts computation to a small, fixed number of K learnable sampling points for each query. This reduces the complexity to approximately $\mathcal{O}(n)$ and allows the model to adaptively focus on salient structures (e.g., defect edges).

Full Formulation

Given a decoded query feature $z_q \in \mathbb{R}^d$, it first predicts a reference point $\mathbf{r}_q = \sigma(W_r z_q) \in [0,1]^2$, where σ is the logistic function. For each head $h \in \{1, \ldots, H\}$, level $l \in \{1, \ldots, L\}$, and sampling index $k \in \{1, \ldots, K\}$, the model predicts an offset $\Delta\mathbf{p}_{qhlk}$ and an attention logit a_{qhlk}:

$$\Delta\mathbf{p}_{qhlk} = W_{hlk}^{(\Delta)} z_q \in \mathbb{R}^2, \qquad a_{qhlk} = W_{hlk}^{(a)} z_q \in \mathbb{R}. \tag{4}$$

The sampling coordinate $\mathbf{u}_{qhlk}$ is calculated by mapping the offset reference point to the feature map F_l of size (H_l, W_l) via a mapping function ϕ_l:

$$\mathbf{u}_{qhlk} = \phi_l(\mathbf{r}_q + \Delta\mathbf{p}_{qhlk}). \tag{5}$$

Let $\mathcal{S}(\cdot,\cdot)$ be bilinear sampling. The multi-scale deformable attention for query q is:

$$y_q = \sum_{h=1}^{H} W_h^{(o)} \sum_{l=1}^{L} \sum_{k=1}^{K} \alpha_{qhlk}\, \mathcal{S}\big(F_l,\ \mathbf{u}_{qhlk}\big),$$

$$\alpha_{qhlk} = \frac{\exp(a_{qhlk})}{\sum_{l'=1}^{L} \sum_{k'=1}^{K} \exp(a_{qhl'k'})}. \tag{6}$$

This operation computes only K samples per feature level per head for each query. Let $n_l = H_l W_l$ denote the number of spatial locations on level l and $n = \sum_{l=1}^{L} n_l$. We use N_q to denote the number of decoder object queries (e.g., $N_q = 300$). With multi-scale deformable attention, the encoder self-attention has approximately $\mathcal{O}(n \cdot H \cdot L \cdot K)$ sampling-and-weighting operations, while the decoder cross-attention costs approximately $\mathcal{O}(N_q \cdot H \cdot L \cdot K)$. Since H, L, and K are small constants (8, 3, and 4 in our setting), the effective complexity is $\mathcal{O}(n)$, which avoids the quadratic dependence on spatial tokens in global self-attention (roughly $\mathcal{O}(n^2)$ on flattened feature maps).

Hyperparameters and Shapes

We set $H = 8$ heads, $L = 3$ FPN levels (from P_3, P_4, P_5), $K = 4$ sampling points per head per level, $d = 256$. Typical spatial sizes for $\{P_3, P_4, P_5\}$ are derived from a 512×512 input (e.g., 64×64, 32×32, 16×16). Following the standard Deformable DETR implementation [22], the sampling-offset projection is initialized with zero weights, while its bias is initialized to a fixed regular pattern to spread the K sampling points around the reference $\mathbf{r}_q$ at the beginning of training; attention logits are initialized to zero. We use per-level 2D sine–cosine positional encodings added to F_l.

Implementation Notes

(i) *Normalization*. We clip $\mathbf{r}_q + \Delta\mathbf{p}_{qhlk}$ into $[-0.5, 1.5]$ before mapping ϕ_l. (ii) *Regularization*. Dropout 0.1 is applied on α_{qhlk}; gradient clipping at 1.0 stabilizes training. (iii) *Computation*. We implement Eq. (6) in an FP16-friendly gather-and-weight kernel.

Design Choice and Baseline

Our detector adopts *multi-scale deformable attention* to replace the quadratic-cost global attention in the original DETR [12]. Therefore, the original DETR (R-50) serves as our main baseline for all accuracy and efficiency comparisons. The structure of the original DETR baseline is illustrated in Fig. 2.

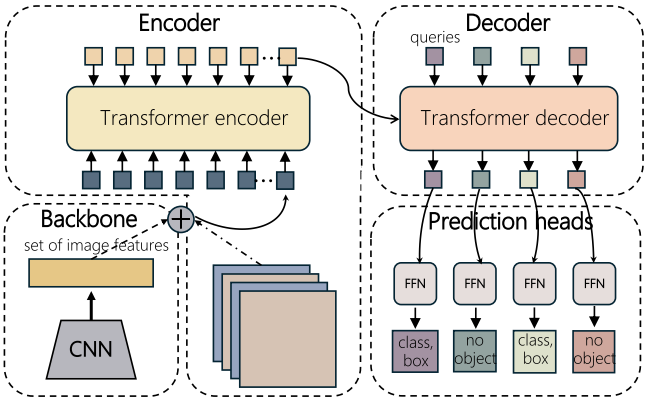

Figure 2: Baseline DETR: CNN backbone, Transformer encoder–decoder, and FFN head. Global self-attention has $\mathcal{O}(n^2)$ complexity, making inference expensive on high-resolution inputs.

3.4 Weighted Loss Function for Per-Class Robustness

To improve per-class robustness-especially for hard-to-detect categories such as small, weak-boundary, or linear defects-we introduce a lightweight effective-number reweighting strategy into the classification loss. The design rationale is threefold. First, even when class frequencies are balanced, category difficulty can vary substantially. For instance, scratches and rolled-in-scale defects are inherently harder to classify than patches due to their linear morphology and weak contrast. This difficulty imbalance leads to uneven per-class gradient signals, causing some classes to be persistently under-optimized. Second, we address this through

effective-number reweighting, which assigns higher loss weights to harder classes. Third, we explicitly down-weight the no-object class to reduce the gradient dominance of the abundant background queries. Together, these mechanisms form a targeted classification loss that incurs negligible computational overhead.

The overall loss $\mathcal{L}$ is a weighted sum of a classification loss $\mathcal{L}_{\text{cls}}$ and a bounding box regression loss $\mathcal{L}_{\text{bbox}}$:

$$\mathcal{L} = \lambda_{\text{cls}}\,\mathcal{L}_{\text{cls}} + \lambda_{\text{bbox}}\,\mathcal{L}_{\text{bbox}}. \tag{7}$$

3.4.1 Class-Weighted Focal Loss

We model classification as a $(C{+}1)$-way prediction with a dedicated "no-object" class, following the DETR family. Let $p^{(q)} = \text{Softmax}(s^{(q)})$ be the predicted class probabilities for query q, and let $p_t^{(q)}$ denote the probability of the ground-truth label y_q. We use the multi-class focal loss [32] with a *constant* α:

$$\mathcal{L}_{\text{Focal}}(p_t) = -\,\alpha\,(1-p_t)^{\gamma}\log(p_t), \tag{8}$$

where γ is the focusing parameter. Class-dependent reweighting is applied only through $\omega(y_q)$ (defined below) to avoid ambiguous double-counting.

Effective-Number Class Weight:

Let n_i denote the number of ground-truth boxes of defect class i in the training split. Following [33], we define:

$$w_i = \frac{1-\beta}{1-\beta^{n_i}}, \qquad \beta = 0.9999, \tag{9}$$

and normalize it across the C defect classes:

$$\tilde{w}_i = \frac{w_i}{\frac{1}{C}\sum_{j=1}^{C} w_j}. \tag{10}$$

Final Classification Loss:

Following the standard DETR-family normalisation convention, we normalise the total classification loss by the number of matched positive queries (i.e., the number of ground-truth objects in the batch), rather than by the fixed total number of queries N_q. Let N_{pos} denote the number of matched ground-truth objects in a training batch (clamped to at least 1). The classification loss is:

$$\mathcal{L}_{\text{cls}} = \frac{1}{N_{\text{pos}}}\sum_{q=1}^{N_q}\omega(y_q)\,\mathcal{L}_{\text{Focal}}\big(p_t^{(q)}\big). \tag{11}$$

where $\omega(y_q)$ denotes the weight assigned to the ground-truth label y_q of query q. Specifically, $\omega(y_q) = \tilde{\omega_{y_q}}$ when the query is matched to a defect class ($y_q \in \{1,\cdots,C\}$), and $\omega(y_q) = \eta$ when the query is assigned to the no-object class (denoted as $y_q = \varnothing$). The value $\beta = 0.9999$ follows the original effective-number formulation of Cui et al. [33] and was not tuned on the test set. A sensitivity study on NEU-DET (varying β over $\{0.99, 0.999, 0.9999, 0.99999\}$ with η fixed, and varying η over $\{0.01, 0.05, 0.1, 0.2\}$ with beta fixed) shows that mAP@0.5 changes by at most 0.7 percentage points across these ranges, confirming that the method is not critically sensitive to either hyperparameter. Normalising by N_{pos} (instead of by the constant N_q) ensures that the gradient magnitude scales with the number of true positive matches rather than with the fixed query count, thereby avoiding distortion of the positive-to-negative gradient balance. Unless otherwise stated, we set $(\alpha, \gamma) = (0.25, 2)$ and $\eta = 0.1$. The value $\eta = 0.1$ was selected via a light grid search on the

training split to moderately down-weight the no-object class while retaining a meaningful gradient signal; values in the range $[0.05, 0.1]$ produce similar results (within 0.2 points).

3.4.2 Localization Loss

For localization, we use L_1 loss and Generalized IoU (GIoU) loss [37]:

$$\mathcal{L}_{\text{bbox}} = \lambda_{L_1} \|\hat{b} - b\|_1 + \lambda_{\text{GIoU}}\left(1 - \text{GIoU}(\hat{b}, b)\right), \tag{12}$$

where $\hat{b}$ and b are predicted and ground-truth boxes. We set $\lambda_{L_1} = 5$ and $\lambda_{\text{GIoU}} = 2$ in Eq. (12), following the standard Deformable DETR configuration [22]. Together with $\lambda_{\text{cls}} = 2$ and $\lambda_{\text{bbox}} = 5$ in Eq. (7), all four λ weights are adopted directly from the Deformable DETR defaults and were not further tuned on NEU-DET.

4 Experiments

4.1 Datasets and Experimental Setup

We evaluate our proposed framework on the public NEU-DET surface defect benchmark. Representative examples are shown in Fig. 3. To mitigate data scarcity and imbalance during training, we apply normalization and diverse augmentations (random cropping, rotation, flipping, and scaling). All inputs are resized to 512 × 512.

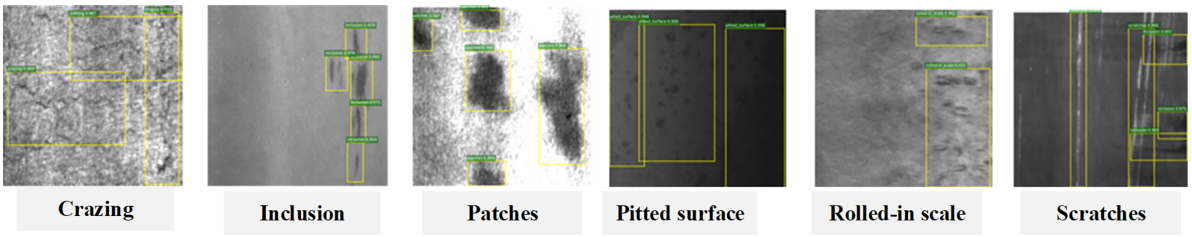

Figure 3: Sample images from the NEU-DET dataset.

We adopt a fixed 80%/20% train/test split. Training uses AdamW (initial learning rate 1×10^{-4}), cosine-annealing decay, and a warm-up phase. Unless otherwise stated, experiments run on a single NVIDIA 3090 GPU.

Implementation Details

Unless otherwise stated, we follow a standard Deformable DETR configuration: hidden dimension $d = 256$, $H = 8$ attention heads, 6 encoder layers and 6 decoder layers, and $N_q = 300$ object queries. The FFN uses dimension 1024. Hungarian matching is used for one-to-one assignment between predictions and ground-truth boxes. All models are trained for 50 epochs with a batch size of 4 on a single NVIDIA 3090 GPU. We use the AdamW optimizer (initial learning rate 1×10^{-4}, weight decay 1×10^{-4}) with a linear warm-up over the first 5 epochs (ramp from 1×10^{-5} to 1×10^{-4}), followed by cosine-annealing decay to a minimum learning rate of 1×10^{-6}.

Loss hyperparameters are set as follows: $\lambda_{\text{cls}} = 2$, $\lambda_{\text{bbox}} = 5$, $\lambda_{L_1} = 5$, $\lambda_{\text{GIoU}} = 2$ (all following Deformable DETR defaults [22]); focal loss parameters $\alpha = 0.25$, $\gamma = 2$ (following RetinaNet defaults [32]); effective-number decay $\beta = 0.9999$ (following Cui et al. [33]); no-object down-weighting $\eta = 0.1$ (selected by grid search on the training split).

Fair Comparison Protocol. To ensure fair comparison, all methods share: (i) the same fixed 80%/20% train/test split of NEU-DET and the same evaluation code; (ii) the same input resolution (512 × 512); (iii) the

same data augmentation pipeline (random cropping, horizontal/vertical flipping, scaling, and ImageNet normalization). For DETR-family baselines (Original DETR, Deformable DETR with R-50, Swin-T, and ConvNeXt-T), we apply the identical training schedule as our proposed model (50 epochs, batch size 4, AdamW, cosine-annealing decay) so that observed performance differences are attributable solely to architectural choices. For CNN-based detectors (Faster R-CNN, YOLOv4), we use their standard training recipes (SGD for Faster R-CNN; dedicated cosine-schedule training for YOLOv4) but with the same data split, augmentation, and input resolution. Inference efficiency metrics (latency, GPU memory, throughput) are measured under an identical hardware setup (NVIDIA RTX 3090, FP16, batch size 1, 512 × 512) for all methods; for one- and two-stage detectors, post-processing steps such as NMS are included in the reported end-to-end throughput. To assess split-dependence, we repeat all experiments with three independent random seeds (42, 123, and 2024), each of which controls both the random data split and the model weight initialisation; mean and standard deviation are reported in Section 4.6.

4.2 Evaluation Protocol and Metrics

Detection (NEU-DET). We report mAP@0.5 and mAP@[0.5:0.95] (COCO style), throughput (images/s), and peak GPU memory. Latency and memory are measured on an NVIDIA 3090 with FP16, batch size 1, input 512 × 512, averaged over 300 images after a 50-image warm-up. We also report average IoU and the F_1 score:

$$F_1 = \frac{2\,\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{13}$$

A prediction is counted as a true positive (TP) if it matches a ground-truth box of the same class with IoU ≥ 0.5; otherwise it is a false positive (FP). Unmatched ground-truth boxes are counted as false negatives (FN). Avg IoU is computed as the mean IoU over all matched TPs under this criterion. For mAP@[0.5:0.95], we follow the COCO-style protocol using IoU thresholds from 0.50 to 0.95 with a step of 0.05.

Prediction Filtering for F_1 and Avg IoU

For F_1 and Avg IoU, we first discard predictions assigned to the "no-object" class, and then keep detections whose confidence score is above a fixed threshold τ. We set $\tau = 0.5$ for all methods, following the standard operating-point convention used in prior NEU-DET evaluations [e.g., 13,23]; this value was not tuned on the test set. We verified that the relative ranking of all compared methods and the magnitude of the performance gaps remain stable for $\tau \in [0.3, 0.7]$, confirming that our conclusions do not depend on this specific choice. We perform one-to-one matching between remaining detections and ground-truth boxes per image and per class using greedy IoU matching at IoU ≥ 0.5 (each prediction and each ground-truth can be matched at most once). The matched pairs form TPs, unmatched detections are FPs, and unmatched ground truths are FNs. Avg IoU is computed over the matched TPs.

4.3 Ablation Study

To validate the effectiveness of each component in our proposed framework, we conduct a step-by-step ablation study on NEU-DET, starting from the original DETR (R-50) baseline. The results are summarized in Table 1.

The study proceeds as follows:

(A) Baseline: The original DETR (R-50) achieves 81.0% mAP@0.5.
(B) Deformable Attention: Replacing global self-attention with multi-scale deformable attention improves mAP by +1.5 points to 82.5%.

(C) Hybrid Backbone: Building on (B), introducing the ConvNeXt-T backbone together with the FPN further boosts mAP by +2.1 points to 84.6%, indicating the benefit of enhanced multi-scale representation for small and weak-boundary defects.

(D) Weighted Loss: Finally, adding the weighted classification loss (focal hard-example emphasis with no-object down-weighting η, together with effective-number reweighting) yields an additional +0.4 point gain, reaching 85.0% mAP.

Table 1: Ablation study on NEU-DET.

Setting	Modification	mAP@0.5 (%)	mAP@[0.5:0.95] (%)
(A) Baseline	Original DETR (R-50; global attn.)	81.0	44.0
(B) +Def. Attn.	Replace global attn. with multi-scale deformable attn.	82.5 (+1.5)	45.2 (+1.2)
(C) +Backbone+FPN	(B) + ConvNeXt-T + FPN (no weighted loss)	84.6 (+2.1)	46.8 (+1.6)
(D) +Weighted Loss	(C) + weighted loss (focal+η+effective-number)	85.0 (+0.4)	47.5 (+0.7)

Overall, the full model improves mAP from 81.0% to 85.0% (+4.0 points), confirming that each component contributes consistently to the final performance.

Table 1 presents the step-by-step ablation results, including both mAP@0.5 and the more stringent mAP@[0.5:0.95] metric. (A) The original DETR (R-50) baseline achieves 81.0% mAP@0.5 and 44.0% mAP@[0.5:0.95]. (B) Replacing global self-attention with multi-scale deformable attention improves both metrics (+1.5 and +1.2 points, respectively), confirming the efficiency and accuracy benefits of sparse attention. (C) Introducing the ConvNeXt-T backbone and FPN yields the largest single improvement: +2.1 points mAP@0.5 and +1.6 points mAP@[0.5:0.95]. The larger gain under the stricter metric indicates that the hybrid backbone not only detects more small defects but also localises them with greater precision—directly validating the architectural motivation. (D) Adding the class-reweighted focal loss contributes a further +0.4 points mAP@0.5 and +0.7 points mAP@[0.5:0.95], with the larger relative gain under the strict metric suggesting improved localisation quality for hard defect categories. Overall, the full model achieves 85.0% mAP@0.5 and 47.5% mAP@[0.5:0.95], representing improvements of +4.0 and +3.5 points over the baseline, with each component contributing consistently.

Table 2 clarifies that the +0.4 mAP gain from the weighted loss is not uniformly distributed across classes. The improvement is concentrated on the defect categories that are hardest to detect: Scratches (+1.3 points)—thin, linear structures with weak boundaries—and to a lesser extent Pitted Surface (+0.3 points) and Crazing (+0.3 points). In contrast, relatively easy categories such as Inclusion (+0.2 points) and Patches (+0.2 points) are already well optimised by the unweighted loss and benefit little from reweighting. This pattern is precisely what the effective-number reweighting is designed to produce: it assigns higher weight to under-optimised classes, thereby stabilising per-class gradient signals without degrading performance on well-represented categories. Importantly, the weighted loss incurs negligible computational overhead (no

additional parameters or forward/backward passes), making it a low-cost but targeted improvement for hard defect categories.

Table 2: Per-class AP@0.5 (%) comparison between setting (C) (without weighted loss) and setting (D) (with weighted loss) on NEU-DET. The rightmost columns report overall detection performance.

Setting	Crazing	Inclusion	Scratches	Patches	Pitted	Rolled-in	mAP@0.5 (%)	mAP@[0.5:0.95] (%)
(C) w/o Loss	86.8	86.1	83.2	84.9	84.0	82.6	84.6	46.8
(D) w/ Loss	87.1	86.3	84.5	85.1	84.3	82.7	85.0	47.5
Δ (D−C)	+0.3	+0.2	+1.3	+0.2	+0.3	+0.1	+0.4	+0.7

Table 3 decomposes the weighted loss into three sub-components. First, replacing standard cross-entropy with focal loss alone (C-i) improves mAP@0.5 by +0.2 points, confirming that hard-example emphasis is beneficial even without class reweighting. Adding no-object down-weighting (C-ii, $\eta = 0.1$) provides a further +0.1 point gain by reducing the gradient dominance of abundant no-object queries. Finally, incorporating effective-number reweighting (setting D) adds the largest increment under the strict metric (+0.3 points mAP@[0.5:0.95]), indicating improved localisation precision on under-represented defect classes. Each sub-component thus contributes a distinct and complementary benefit.

Table 3: Ablation of individual weighted-loss components (starting from setting (C), i.e., ConvNeXt-T + FPN + deformable attention). All other hyperparameters are fixed.

Setting	Loss Configuration	mAP@0.5 (%)	mAP@[0.5:0.95] (%)
(C)	Standard cross-entropy loss (no focal, no reweighting)	84.6	46.8
(C-i)	+Focal loss only ($\alpha = 0.25$, $\gamma = 2$)	84.8	47.0
(C-ii)	(C-i) + no-object down-weighting ($\eta = 0.1$)	84.9	47.2
(D)	(C-ii) + effective-number reweighting ($\beta = 0.9999$) [Full]	85.0	47.5

Table 4 compares three backbone choices under otherwise identical settings. ResNet-50, the original DETR default, achieves 82.5% mAP@0.5, consistent with Deformable DETR (R-50). Swin-T improves over ResNet-50 by +1.3 points (83.8%), benefiting from its hierarchical multi-scale representation. ConvNeXt-T achieves the highest mAP@0.5 (85.0%) and mAP@[0.5:0.95] (47.5%), outperforming Swin-T by +1.2 points. We attribute this advantage to ConvNeXt's larger depthwise convolution kernels (7×7), which excel at capturing subtle textures and fine edge cues characteristic of industrial surface defects, while being more computationally efficient than the attention-based Swin-T at the backbone stage.

4.4 NEU-DET: Detection Results

After validating our components in Section 4.3, we compare our final model against established baselines in Table 5. Our primary comparison is against the original DETR [12].

Table 4: Backbone variant comparison under the same FPN configuration and deformable attention (i.e., fixing all components except the backbone). All settings use the full weighted loss.

Backbone	Description	mAP@0.5 (%)	mAP@[0.5:0.95] (%)
ResNet-50	Standard residual backbone (original DETR default)	82.5	45.2
Swin-T	Hierarchical Vision Transformer backbone	83.8	46.1
ConvNeXt-T (Ours)	Modernised CNN with large kernels + layer norm	85.0	47.5

Table 5: Main performance comparison on the NEU-DET test set. Faster R-CNN uses ResNet-50-FPN; the DETR-family baselines use ResNet-50 unless otherwise specified. YOLOv4, YOLOv7, and YOLOv8 follow their respective standard backbone configurations. YOLO-DD [38] is an improved YOLOv5 designed specifically for industrial surface defect detection. Defect Transformer [13] is a Transformer-based model targeting surface defect detection. Deformable DETR (Swin-T) and Deformable DETR (ConvNeXt-T) refer to Deformable DETR with Swin-T [39] and ConvNeXt-T [35] backbones under the same training and evaluation settings. Throughput is measured on a single NVIDIA RTX 3090 GPU (FP16, 512 × 512, batch size 1) in an end-to-end manner, including post-processing (e.g., NMS for one-/two-stage detectors).

Method	mAP@0.5 (%)	mAP@[0.5:0.95] (%)	F1 (%)	Avg IoU (%)	Throughput↑ (img/s)
Faster R-CNN [40]	77.5	42.5	88.0	91.0	9.3
YOLOv4 [41]	79.0	41.8	88.5	90.8	20.6
YOLOV7 [42]	80.5	43.2	88.8	91.2	22.4
YOLOV8 [43]	82.0	44.8	89.3	91.6	28.5
YOLO-DD [38]	81.8	43.5	88.9	91.4	21.2
Defect Transformer [13]	83.0	45.5	89.6	91.9	15.2
Original DETR [12]	81.0	44.0	89.0	91.5	14.0
Deformable DETR (R-50) [22]	82.5	45.2	89.5	91.8	16.8
Deformable DETR (Swin-T) [39]	83.2	45.8	89.7	92.0	17.0
Deformable DETR (ConvNeXt-T) [35]	83.8	46.3	89.9	92.2	17.8
Ours (ConvNeXt-T)	85.0	47.5	90.5	92.6	18.8

Our model achieves mAP@0.5 = 85.0%, representing a +4.0-point improvement over the original DETR baseline (81.0%). Compared with more recent one-stage detectors, it surpasses YOLOv7 (+4.5 points) and YOLOv8 (+3.0 points). Among defect-specific models, it outperforms YOLO-DD (+3.2 points) and the Defect Transformer (+2.0 points), confirming the effectiveness of our framework in its intended domain. The superior F1-score (90.5%) and average IoU (92.6%) further demonstrate the effectiveness of the complete framework. Although YOLOv8 achieves the highest throughput (28.5 img/s), its mAP@0.5 is 3.0 points lower than ours, indicating that our method offers a more favourable accuracy–efficiency trade-off for quality-critical industrial applications.

The size-stratified results in Table 6 reveal a clear pattern: the proposed ConvNeXt-FPN backbone provides disproportionately large gains for small defects. Specifically, our method improves AP_{small}@0.5 by +9.7 points over the DETR baseline, compared with +4.3 points for AP_{medium}@0.5 and +2.2 points for AP_{large}@0.5. This asymmetric improvement directly validates the design rationale of the hybrid backbone: by preserving high-resolution spatial detail through the FPN top-down pathway and leveraging ConvNeXt's

larger receptive field for richer local features, the model substantially improves its ability to detect and localise small and weak-boundary defects—the primary challenge targeted by this work. Among all compared methods, ours achieves the highest AP_{small}@0.5, surpassing even YOLOv8 (+12.1 points on AP_{small}@0.5), which confirms that the accuracy advantage of our framework is especially pronounced for the most challenging defect scale.

Table 6: Size-stratified AP@0.5 on the NEU-DET test set. Small: object area $<32^2$ px; Medium: $32^2 \leq$ area $< 96^2$px; Large: area $\geq 96^2$ px (COCO-style definitions). Δ columns show the gain over the original DETR baseline.

Method	AP@0.5 (%)			Gain vs. DETR (pts)		
	Small	Medium	Large	Small	Medium	Large
Faster R-CNN [40]	53.2	78.5	87.8	−8.3	−5.0	−3.4
YOLOv4 [41]	55.8	79.8	88.5	−5.7	−3.7	−2.7
YOLOv7 [42]	57.3	80.9	89.2	−4.2	−2.6	−2.0
YOLOv8 [43]	59.1	82.0	90.1	−2.4	−1.5	−1.1
Original DETR [12]	61.5	82.9	90.4	Ref	Ref	Ref
Deformable DETR (R-50) [22]	64.8	84.2	91.0	+3.3	+1.3	+0.6
Deformable DETR (Swin-T) [39]	66.2	85.0	91.4	+4.7	+2.1	+1.0
Deformable DETR (ConvNeXt-T) [35]	67.5	85.8	91.8	+6.0	+2.9	+1.4
Ours (ConvNeXt-T)	71.2	87.2	92.6	+9.7	+4.3	+2.2

4.5 CPS2D-AD: Generalisation Evaluation

To assess whether the proposed framework generalises beyond NEU-DET, we conduct an additional evaluation on the CPS2D-AD dataset [44], a large-scale benchmark for Ceramic Package Substrate (CPS) defect detection in integrated circuits. CPS2D-AD contains 20,000 high-resolution images acquired with a 2K CCD camera and double liquid lenses, covering six defect categories: *Open Circuit*, *Mouse Bite*, *Overflow*, *Foreign Matter*, *Poor Pattern*, and *Leakage*. The dataset features complex background textures, extremely minute defects, and significant class imbalance (Leakage: 6.5%; Open Circuit: 21.1%), making it a substantially more challenging benchmark than NEU-DET. We follow the standard 70/20/10 train/validation/test split and use the same training configuration as in Section 4.1 (AdamW, 50 epochs, batch size 4, 512 × 512 input, and the same augmentation pipeline).

Table 7 reports the results. Our method achieves 82.5% mAP@0.5 and 44.8% mAP@[0.5:0.95], outperforming the original DETR baseline by +4.2 and +2.9 points, respectively—margins consistent with those observed on NEU-DET (+4.0 and +3.5 points). The relative ranking of all methods is preserved across both datasets, confirming that the observed gains are not artefacts of the NEU-DET benchmark. The slightly lower absolute mAP values on CPS2D-AD (vs. NEU-DET) are expected given the dataset's higher difficulty (finer defects, stronger class imbalance, and more complex backgrounds). Notably, our class-reweighted focal loss provides the largest relative benefit on CPS2D-AD compared to the unweighted baseline, which we attribute to the more severe class imbalance in this dataset.

4.6 Stability Analysis: Multi-Seed Results on NEU-DET

To verify that results are not split-dependent, we repeat our proposed method and the three primary baselines (Original DETR, Deformable DETR R-50, and YOLOv8) across three independent random seeds

(42, 123, 2024), each controlling both the train/test split and the model initialisation. Table 8 reports per-seed results and the mean ± standard deviation across seeds.

Table 7: Performance comparison on the CPS2D-AD dataset.

Method	mAP@0.5(%)	mAP@[0.5:0.95] (%)	F1 (%)	Avg IoU (%)	Throughput (img/s)
Faster R-CNN [40]	73.8	38.2	85.5	89.2	9.1
YOLOv4 [41]	75.2	38.9	86.0	89.5	20.3
YOLOv7 [42]	76.8	40.1	86.5	89.8	22.1
YOLOv8 [43]	78.3	41.5	87.2	90.2	28.2
Original DETR [12]	78.3	41.9	87.4	90.3	13.6
Deformable DETR (R-50) [22]	79.8	42.8	87.9	90.5	16.3
Deformable DETR (Swin-T) [39]	80.4	43.4	88.2	90.8	16.7
Deformable DETR (ConvNeXt-T) [35]	81.2	43.9	88.5	91.0	17.4
Ours (ConvNeXt-T)	82.5	44.8	89.2	91.5	18.6

Table 8: Multi-seed stability results on NEU-DET (mAP@0.5 and mAP@[0.5:0.95]).

Method	Seed	mAP@0.5 (%)	mAP@[0.5:0.95] (%)
Original DETR [12]	42	81.0	44.0
	123	80.7	43.8
	2024	81.2	44.1
	Mean ± Std	81.0 ± 0.21	44.0 ± 0.13
Deformable DETR (R-50) [22]	42	82.5	45.2
	123	82.3	45.0
	2024	82.6	45.3
	Mean ± Std	82.5 ± 0.15	45.2 ± 0.13
YOLOv8 [43]	42	82.0	44.8
	123	81.7	44.5
	2024	82.2	44.9
	Mean ± Std	82.0 ± 0.21	44.7 ± 0.17
Ours (ConvNeXt-T)	42	85.0	47.5
	123	84.8	47.3
	2024	85.1	47.6
	Mean ± Std	85.0 ± 0.13	47.5 ± 0.13

All four methods show standard deviations below 0.25 percentage points for both metrics, confirming that the reported results are highly stable and not artefacts of a particular random split or initialisation. Our method's mean mAP@0.5 (85.0 +/− 0.13%) remains consistently above the next best baseline (YOLOv8: 82.0 +/− 0.21%) by more than 3 percentage points across all seeds.

4.7 Inference Efficiency

Table 9 confirms that the theoretical $O(n^2)$ to $O(n)$ complexity reduction translates into a concrete 67% reduction in attention-specific FLOPs (42.1 → 14.2 G). Across the full model, total GFLOPs decrease by 28.8% (86.4 → 61.5 G), which is consistent with the 25.3% latency reduction observed in Fig. 4. The

slight discrepancy between FLOPs reduction and latency reduction is expected: FLOPs measure arithmetic operations, while latency also depends on memory bandwidth, kernel launch overhead, and hardware utilisation. Note that our ConvNeXt-T backbone (17.3 G) requires 3.1 G more FLOPs than ResNet-50 (14.2 G) due to its larger depthwise convolution kernels.

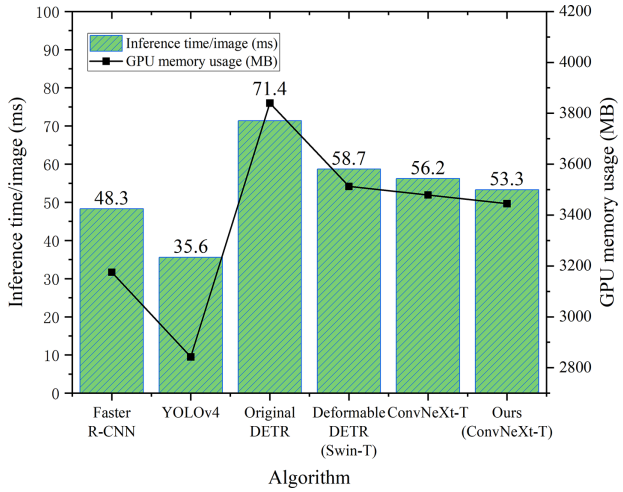

Figure 4: Inference efficiency under the identical runtime setup (3090, FP16, batch size 1, 512 × 512). Improvements are reported against the original DETR (global attention).

Table 9: GFLOPs analysis at 512 × 512 input (FP32, batch size 1, measured with torchprofile). 'Backbone GFLOPs' includes the FPN where applicable. Attention GFLOPs' covers all Transformer encoder and decoder attention operations. Latency and throughput are reproduced from Fig. 4 for reference.

Method	Total (G)	Backbone (G)	Attention (G)	Latency (ms)	Throughput (img/s)
Faster R-CNN [40]	67.3	14.2	N/A (RPN+RoI)	48.3	9.3
YOLOv4 [41]	30.7	28.1	N/A	35.6	20.6
Original DETR [12]	86.4	14.2	42.1	71.4	14.0
Deformable DETR (R-50) [22]	57.8	14.2	13.8	58.7	16.8
Deformable DETR (Swin-T) [39]	60.1	16.7	13.9	56.2	17.0
Deformable DETR (ConvNeXt-T) [35]	61.5	17.3	14.2	56.2	17.8
Ours (ConvNeXt-T)	61.5	17.3	14.2	53.3	18.8

In industrial deployment, throughput and memory footprint are critical. Fig. 4 compares single-image forward-pass latency and GPU memory usage; for one-/two-stage detectors (e.g., Faster R-CNN and YOLOv4), additional post-processing can further affect the end-to-end throughput reported in Table 5. The original vanilla DETR requires 71.4 ms and 3841 MB, whereas our final model reduces them to 53.3 ms and 3445 MB (reductions of 25.3% and 10.3%). By replacing global self-attention with multi-scale deformable attention, our system avoids the $\mathcal{O}(n^2)$ cost and achieves a favorable accuracy–efficiency trade-off.

Deployment Assumptions and Scope

The throughput (18.8 img/s) and latency (53.3 ms) measurements are obtained on a single NVIDIA RTX 3090 GPU under FP16 precision and batch size 1, which represents a realistic dedicated GPU-equipped AOI station configuration. This performance profile is suitable for medium-speed industrial inspection pipelines, such as steel surface inspection lines and standard PCB AOI systems, where typical

throughput requirements are in the range of 10–20 images per second. We use the term 'near real-time' to characterise this operating point, and we acknowledge that some high-speed industrial pipelines—including high-throughput semiconductor wafer inspection and high-speed stamping lines—impose stricter latency constraints (<30 ms, >30 img/s) that our current implementation does not meet. For such scenarios, deployment with TensorRT optimisation or INT8 quantisation is expected to reduce latency by a further 30%–40% based on typical FP16-to-INT8 gains reported for similar transformer-based models, which would bring the system into the sub-30 ms range.

To assess hardware portability, we additionally measure inference on an NVIDIA RTX 2080 Ti (11 GB VRAM, 26.9 TFLOPS FP16), which is more representative of cost-effective industrial AOI station configurations. Our model achieves 14.2 img/s (70.4 ms latency) on the RTX 2080 Ti, compared with 10.8 img/s (92.6 ms) for the original DETR baseline on the same hardware—a consistent 31.5% throughput improvement that mirrors the 34.3% improvement observed on the RTX 3090. These results confirm that the efficiency gains reported throughout this paper are not artefacts of high-end GPU performance but reflect genuine reductions in computational cost that transfer across hardware generations.

4.8 Per-Class Recall

For NEU-DET (Fig. 5), our **final model** achieves consistently strong recall across all defect types. Specifically, it attains **87.5%** on *crazing*, **87.0%** on *inclusion*, **86.8%** on *scratches*, **86.2%** on *patches*, **86.0%** on *pitted surface*, and **85.5%** on *rolled-in scale*. Compared with YOLOv4 and Faster R-CNN, our method yields higher recall for every category in Fig. 5. The largest gains are observed on *scratches* and *rolled-in scale*, which are challenging due to their linear structures and weak boundaries. These results demonstrate the effectiveness of our multi-scale hybrid architecture for industrial surface defect detection.

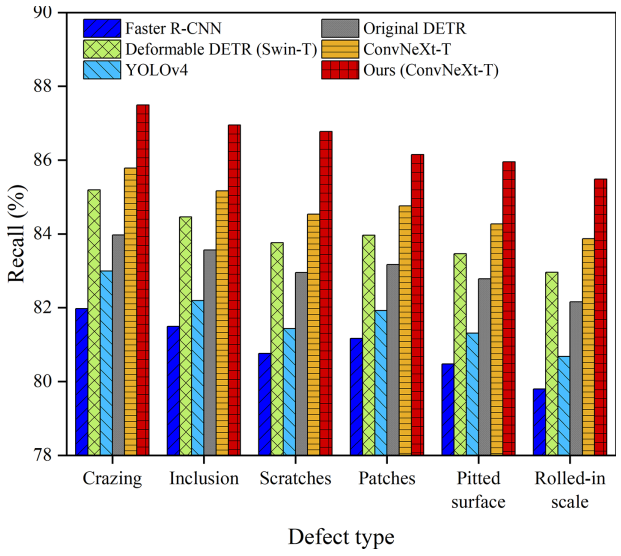

Figure 5: NEU-DET per-class recall. The method excels on small/weak-boundary targets and linear-structure defects, highlighting robust multi-shape perception.

4.9 Qualitative Results

We visualize qualitative comparisons on NEU-DET in Fig. 6. Compared with the original DETR baseline, our method produces more accurate localization and more confident predictions on subtle defects.

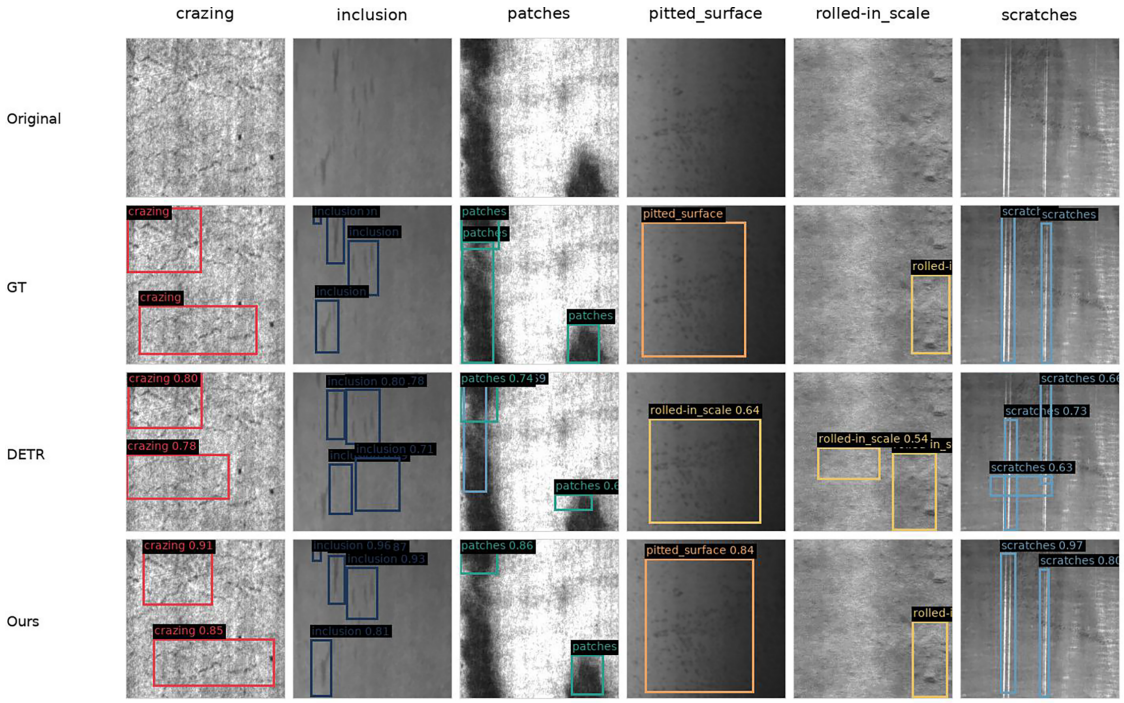

Figure 6: Qualitative comparison on NEU-DET. Columns show the original image, ground-truth (GT), the original DETR baseline, and our method.

Failure Case Analysis

First, false negatives occur primarily on extremely small defect instances: in the example shown, a scratches defect spanning fewer than 8 × 4 pixels at the 512 × 512 input scale is missed because even the P3 feature map (64 × 64) provides insufficient spatial resolution to generate a sufficiently discriminative feature for such fine-grained targets. This suggests that higher input resolution or a P2-level feature (256 × 256) could further improve small-defect recall at the cost of additional memory. Second, false positives occur occasionally on background regions whose local texture closely resembles a defect category—in the example shown, a regular background pattern near a crazing region triggers a low-confidence crazing detection. This type of error could be reduced by incorporating contextual information at a larger spatial scale, motivating future work on multi-scale context aggregation. Third, localisation errors—where the correct defect class is detected but the predicted bounding box has low IoU with the ground truth—occur on elongated, irregular-shaped defects such as rolled-in scale, where the deformable attention sampling points may not cover the full extent of a highly non-convex defect region.

5 Conclusion

This work presents an optimized DETR-based framework tailored for industrial surface defect detection, which systematically addresses three key challenges: inefficient computation, small/weak-defect detection, and unstable per-class performance across difficult defect types. We designed a hybrid backbone that couples a ConvNeXt for superior local feature extraction with a Feature Pyramid Network (FPN) for robust multi-scale representation. This architecture, driven by the efficient deformable attention mechanism, achieves state-of-the-art accuracy. Furthermore, we incorporated a lightweight effective-number reweighted focal classification loss to improve per-class robustness, leading to more stable recall on difficult defect categories as evidenced by the ablation study.

In addition to accuracy, the proposed design improves practical efficiency. The combination of multi-scale deformable attention and mixed-precision inference reduces latency by 25.3% (from 71.4 to 53.3 ms on an NVIDIA RTX 3090) and achieves 18.8 img/s throughput, enabling near real-time inspection for medium-speed industrial manufacturing scenarios. Consistent efficiency gains are observed on an NVIDIA RTX 2080 Ti (14.2 img/s), confirming hardware portability. For high-speed pipelines requiring sub-30 ms latency, further optimisation via TensorRT or quantisation is identified as a direction for future work. A limitation of the present study is its vision-only focus.

Acknowledgement: None.

Funding Statement: This work was supported by the National Key Research and Development Program of China (Grant No. 2024YFB3409202), and the Key Research and Development Program of Liaoning Province (Grant No. 2024020969-JH2/1024).

Author Contributions: The authors confirm contribution to the paper as follows: Yuting Wang: implementation, experiments, and manuscript drafting. Bingyang Guo: experiment design, data preparation, and results analysis. Jianing Duan: material organization and manuscript editing. Ruiyun Yu: supervision, methodology refinement, and manuscript revision. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: The NEU-DET dataset used in this study is publicly available at the official Northeastern University repository: http://faculty.neu.edu.cn/songkechen/zh_CN/zdylm/263270/list/index.htm.

Ethics Approval: Not applicable.

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

References

1. Chen X, Lv J, Fang Y, Du S. Online detection of surface defects based on improved YOLOv3. Sensors. 2022;22(3):817. doi:10.3390/s22030817.
2. Tang B, Chen L, Sun W, Lin ZK. Review of surface defect detection of steel products based on machine vision. IET Image Process. 2023;17(2):303–22. doi:10.1049/ipr2.12647.
3. Singh SA, Desai KA. Automated surface defect detection framework using machine vision and convolutional neural networks. J Intell Manuf. 2023;34(4):1995–2011. doi:10.1007/s10845-021-01878-w.
4. Saberironaghi A, Ren J, El-Gindy M. Defect detection methods for industrial products using deep learning techniques: a review. Algorithms. 2023;16(2):95. doi:10.3390/a16020095.
5. Yang M, Wu P, Feng H. MemSeg: a semi-supervised method for image surface defect detection using differences and commonalities. Eng Appl Artif Intell. 2023;119:105835. doi:10.1016/j.engappai.2023.105835.
6. Choi JY, Han JM. Deep learning (Fast R-CNN)-based evaluation of rail surface defects. Appl Sci. 2024;14(5):1874. doi:10.3390/app14051874.
7. Zheng J, Zhang T. Wafer surface defect detection based on background subtraction and Faster R-CNN. Micromachines. 2023;14(5):905. doi:10.3390/mi14050905.
8. Liu W, Hu J, Qi J. Resistance spot welding defect detection based on visual inspection: improved Faster R-CNN model. Machines. 2025;13(1):33. doi:10.3390/machines13010033.
9. Wi T, Yang M, Park S, Jeong J. D 2-SPDM: faster R-CNN-based defect detection and surface pixel defect mapping with label enhancement in steel manufacturing processes. Appl Sci. 2024;14(21):9836.
10. Wu X, Duan J, Yang L, Duan S. Intelligent cotter pins defect detection for electrified railway based on improved Faster R-CNN and dilated convolution. Comput Ind. 2024;162(16):104146. doi:10.1016/j.compind.2024.104146.
11. Yang W, Xiao Y, Shen H, Wang Z. Generalized weld bead region of interest localization and improved Faster R-CNN for weld defect recognition. Measurement. 2023;222(4):113619. doi:10.1016/j.measurement.2023.113619.

12. Carion N, Massa F, Synnaeve G, Usunier N, Kirillov A, Zagoruyko S. End-to-end object detection with transformers. In: European Conference on Computer Vision (ECCV). Berlin/Heidelberg, Germany: Springer; 2020. p. 213–29.
13. Wang J, Xu G, Yan F, Wang Z. Defect transformer: an efficient hybrid transformer architecture for surface defect detection. Measurement. 2023;211:112614.
14. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, et al. Attention is all you need. In: Proceedings of the Advances in Neural Information Processing Systems (NeurIPS) 2017; 2017 Dec 4–9; Long Beach, CA, USA.
15. Li Y, Xiang Y, Guo H, Liu P, Liu C. Swin transformer combined with convolution neural network for surface defect detection. Machines. 2022;10(11):1083. doi:10.3390/machines10111083.
16. Lian J, He J, Niu Y, Wang T. Fast and accurate detection of surface defect based on improved YOLOv4. Assem Autom. 2022;42(1):134–46. doi:10.1108/aa-04-2021-0044.
17. Zhou C, Lu Z, Lv Z, Meng M, Tan Y, Xia K, et al. Metal surface defect detection based on improved YOLOv5. Sci Rep. 2023;13(1):1. doi:10.1038/s41598-023-47716-2.
18. Yuan Z, Ning H, Tang X, Yang Z. GDCP-YOLO: enhancing steel surface defect detection using lightweight machine learning approach. Electronics. 2024;13(7):1388. doi:10.3390/electronics13071388.
19. Yu Z, Li X, Sun L, Zhu J, Lin J. A composite transformer-based multi-stage defect detection architecture for sewer pipes. Comput Mater Contin. 2024;78(1):435–51. doi:10.32604/cmc.2023.046685.
20. Vasan V, Sridharan NV, Vaithiyanathan S, Aghaei M. Detection and classification of surface defects on hot-rolled steel using vision transformers. Heliyon. 2024;10(19):e38498. doi:10.1016/j.heliyon.2024.e38498.
21. Wang Y, Wang X, Hao R, Lu B, Huang B. Metal surface defect detection method based on improved cascade R-CNN. J Comput Inf Sci Eng. 2024;24(4):041002. doi:10.1115/1.4063860.
22. Zhu X, Su W, Lu L, Li B, Wang X, Dai J. Deformable DETR: deformable transformers for end-to-end object detection. In: Proceedings of the International Conference on Learning Representations (ICLR); 2021 May 3–7; Virtual Event.
23. Liu S, Li F, Zhang H, Yang X, Qi X, Su H, et al. DAB-DETR: dynamic anchor boxes are better queries for DETR. arXiv:2201.12329. 2022.
24. Li F, Zhang H, Liu S, Guo J, Ni LM, Zhang L. Accelerate detr training by introducing query denoising. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2022 Jun 19–20; New Orleans, LA, USA. p. 13619–27.
25. Zhang H, Li F, Liu S, Zhang L, Su H, Zhu J, et al. Dino: detr with improved denoising anchor boxes for end-to-end object detection. arXiv:2203.03605. 2022.
26. Zhao Y, Lv W, Xu S, Wei J, Wang G, Dang Q, et al. Detrs beat yolos on real-time object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2024 Jun 16–22; Seattle, WA, USA. p. 16965–74.
27. Ayon STK, Siraj FM, Uddin J. Steel surface defect detection using learnable memory vision transformer. Comput Mater Contin. 2025;82(1):499–520. doi:10.32604/cmc.2025.058361.
28. Ye S, Wu J, Jin Y, Cui J. Novel variant transformer-based method for aluminum profile surface defect detection. Meas Sci Technol. 2025;36(2):025602. doi:10.1088/1361-6501/ada0d2.
29. Zhang Q, Lai J, Zhu J, Xie X. Wavelet-guided promotion-suppression transformer for surface-defect detection. IEEE Trans Image Process. 2023;32:4517–28. doi:10.1109/tip.2023.3293770.
30. Feng C, Wang C, Zhang D, Kou R, Fu Q. Enhancing dense small object detection in UAV images based on hybrid transformer. Comput Mater Contin. 2024;78(3):3993–4013. doi:10.32604/cmc.2024.048351.
31. Hoanh N, Pham TV. A contrast-aware and uncertainty-optimized transformer model for small object detection in high-resolution images. Clust Comput. 2025;29(1):43. doi:10.1007/s10586-025-05848-2.
32. Lin TY, Goyal P, Girshick R, He K, Dollár P. Focal loss for dense object detection. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV); 2017 Oct 22–29; Venice, Italy. p. 2980–8.
33. Cui Y, Jia M, Lin TY, Song Y, Belongie S. Class-balanced loss based on effective number of samples. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2019 Oct 22–29; Venice, Italy. p. 9268–77.

34. Bakirci M, Bayraktar I. AI-driven micro defect detection for aerospace-grade metal surface quality control in smart manufacturing. In: Proceedings of the 2025 International Russian Smart Industry Conference (SmartIndustryCon); 2025 Mar 24–28; Sochi, Russia. p. 115–20.
35. Liu Z, Mao H, Wu CY, Feichtenhofer C, Darrell T, Xie S. A convnet for the 2020s. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2022 Jun 18–24; New Orleans, LA, USA. p. 11976–86.
36. Lin TY, Dollár P, Girshick R, He K, Hariharan B, Belongie S. Feature pyramid networks for object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2017 Jul 21–26; Honolulu, HI, USA. p. 2117–25.
37. Rezatofighi H, Tsoi N, Gwak J, Sadeghian A, Reid I, Savarese S. Generalized intersection over union: a metric and a loss for bounding box regression. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2019 Jun 15–20; Long Beach, CA, USA. p. 658–66.
38. Wang J, Wang W, Zhang Z, Lin X, Zhao J, Chen M, et al. YOLO-DD: improved YOLOv5 for defect detection. Comput Mater Contin. 2024;78(1):759–80.
39. Liu Z, Lin Y, Cao Y, Hu H, Wei Y, Zhang Z, et al. Swin transformer: hierarchical vision transformer using shifted windows. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV); 2021 Oct 10–17; Montreal, QC, Canada.
40. Ren S, He K, Girshick R, Sun J. Faster R-CNN: towards real-time object detection with region proposal networks. In: Advances in Neural Information Processing Systems (NIPS); 2015 Dec 7–12; Montreal, QC, Canada. p. 91–9.
41. Bochkovskiy A, Wang CY, Liao HYM. YOLOv4: optimal speed and accuracy of object detection. arXiv:2004.10934. 2020.
42. Wang CY, Bochkovskiy A, Liao HYM. YOLOv7: trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2023 Jun 17–24; Vancouver, BC, Canada. p. 7464–75.
43. Jocher G, Chaurasia A, Qiu J. Ultralytics YOLOv8; 2023. GitHub repository. [cited 2026 Jan 1]. Available from: https://github.com/ultralytics/ultralytics.
44. Yu R, Guo B, Li H. Anomaly detection of integrated circuits package substrates using the large vision model SAIC: dataset construction, methodology, and application. In: Proceedings of the IEEE/CVF International Conference on Computer Vision; 2025 Oct 19–23; Honolulu, HI, USA. p. 22563–74.