electronics

MDPI

Article

Strengthening Small Object Detection in Adapted RT-DETR Through Robust Enhancements

Manav Madan * and Christoph Reich

Institute for Data Science, Cloud Computing and IT-Security (IDACUS), Hochschule Furtwangen University, 78120 Furtwangen im Schwarzwald, Germany; christoph.reich@hs-furtwangen.de
* Correspondence: manav.madan@hs-furtwangen.de

Abstract

RT-DETR (Real-Time DEtection TRansformer) has recently emerged as a promising model for object detection in images, yet its performance on small objects remains limited, particularly in terms of robustness. While various approaches have been explored, developing effective solutions for reliable small object detection remains a significant challenge. This paper introduces an adapted variant of RT-DETR, specifically designed to enhance robustness in small object detection. The model was first designed on one dataset and subsequently transferred to others to validate generalization. Key contributions include replacing components of the feed-forward neural network (FFNN) within a hybrid encoder with Hebbian, randomized, and Oja-inspired layers; introducing a modified loss function; and applying multi-scale feature fusion with fuzzy attention to refine encoder representations. The proposed model is evaluated on the Al-Cast Detection X-ray dataset, which contains small components from high-pressure die-casting machines, and the PCB quality inspection dataset, which features tiny hole anomalies. The results show that the optimized model achieves an mAP of 0.513 for small objects—an improvement from the 0.389 of the baseline RT-DETR model on the Al-Cast dataset—confirming its effectiveness. In addition, this paper contributes a mini-literature review of recent RT-DETR enhancements, situating our work within current research trends and providing context for future development.

Keywords: object detection; small object detection; transformer-based models; foundation models; RT-DETR

Academic Editor: Eva Cernadas

Received: 28 August 2025
Revised: 22 September 2025
Accepted: 24 September 2025
Published: 27 September 2025

Citation: Madan, M.; Reich, C. Strengthening Small Object Detection in Adapted RT-DETR Through Robust Enhancements. *Electronics* **2025**, *14*, 3830. https://doi.org/10.3390/electronics14193830

1. Introduction

Object detection has become a vital part of the manufacturing industry in recent years. It is directly integrated in visual inspection systems in manufacturing, which have undergone a dramatic transformation, evolving from reliance on traditional computer vision methods to sophisticated deep learning systems. Early approaches, such as edge detection, template matching, and feature descriptors like SIFT and HOG [1], provided a foundation but often struggled with the variability inherent in real-world production environments, such as changes in lighting, object pose, and subtle appearance differences. The automotive industry, an early adopter of machine vision and visual inspection systems, exemplifies its transformative impact on quality control (QC) [2] systems. Modern industrial QC relies heavily on machine vision systems to quickly and objectively identify defects, ranging from microscopic flaws to significant deviations [3]. The ability to detect rare but crucial defects, such as casting discontinuities, is enhanced by real-time object detection using machine learning and deep learning (ML/DL) models [4]. This combination of machine vision

and advanced learning techniques has fundamentally changed industrial quality control, enabling the rapid and precise detection needed to maintain high manufacturing standards. To meet these demands, the industry requires novel and robust object detection models.

Additionally, the evolution of newer object detection models is driving innovation for the manufacturing domain in several key areas. Firstly, the need for compact, efficient models suitable for edge devices is pushing the boundaries of ML and DL, particularly in the detection of subtle defects like micro-textures, where transformer-based object detection models show significant potential. Secondly, automating the model improvement pipeline is becoming increasingly important. Thirdly, foundation models [5] (including CLIP, DALL-E, and Vision Transformers), pre-trained on massive datasets, offer a novel approach, potentially solving data scarcity issues with their zero-shot capabilities. In conclusion, the future lies in a combination of miniaturized, edge-optimized models, potentially leveraging the strengths of transformer architectures in object detection and automated learning processes to achieve unparalleled accuracy and versatility across various industrial sectors.

Recent advances in object detection have shifted from anchor-based methods to anchor-free and transformer-based models like DETR (DEtection TRansformer) and RT-DETR (Real-Time DEtection TRansformer). This work focuses specifically on RT-DETR for object detection due to its inherent advantages in capturing global context and relationships within an image. Traditional models, while powerful, often excel at identifying local features. However, subtle defects, particularly small ones, may be missed if the model lacks a comprehensive understanding of the entire object and its surrounding environment. The term 'small objects' in the context of object detection typically refers to objects that occupy a relatively small number of pixels within an image or have limited spatial resolution compared with other objects. As defined by [6], objects can be categorized into small (area $< 32^2$), medium ($32^2 <$ area $< 96^2$), and large (area $< 96^2$) categories. We show how the simplest modifications to RT-DETR can improve detection of small objects. We deliberately refrain from assigning a new name to the optimized model. This decision reflects our intent to focus on meaningful improvements rather than contribute to the growing trend of renaming models for minor modifications. We also provide a comprehensive comparison of different variants which we refer to as the children of the RT-DETR model proposed in the literature. This bring the family of RT-DETR models together under a single umbrella. By doing so, we aim to highlight the various techniques, modifications, and strategies employed to enhance the detection of small objects, which can help other researchers to see gaps and opportunities for further research in improving the performance of RT-DETR. Furthermore, the modifications to RT-DETR proposed in this work are the result of experiments performed on the Al-Cast Detection X-ray dataset, which contains images from parts generated in high-pressure die-casting machines. The developed modified RT-DETR model is then tested on different datasets to prove the effectiveness of the proposed solution in improving small object detection. Concretely, in this work, we propose three targeted modifications to the RT-DETR architecture for small object detection:

- Randomized layer replacement in the encoder, designed to improve robustness under weak supervision.
- Multi-scale feature extraction and fusion of already extracted features from the encoder, enabling the model to retain fine spatial cues that are often lost during downsampling.
- Use of an adaptive focal loss function, which balances dense small object regions with sparse supervision to improve training stability.

While these ideas have been explored independently in prior works, our novelty lies in their deliberate integration within the RT-DETR framework. This integration addresses

a unique gap, as existing variants rarely combine such strategies in a coherent way, and it enables consistent improvements in industrial small object detection tasks.

2. Related Work

Modern manufacturing relies on automated quality control, where object detection helps identify defects. Advances in visual inspection enable deeper integration of deep learning (DL), yet small object detection remains a challenge. Although visual inspection systems have been significantly advanced, enabling a more effective deployment of DL for quality control, challenges persist, both on the software side and the system side. On the systems side a crucial area of improvement is high-resolution imaging and optics. Modern systems move beyond the limitations of earlier, lower-resolution cameras by incorporating high-resolution sensors, such as multi-megapixel CMOS and CCD sensors [7], paired with advanced optics like telecentric lenses. Telecentric lenses are particularly beneficial, as they minimize perspective distortion, ensuring consistent object size regardless of camera distance, which is vital to accurate measurements and defect detection [8]. Sophisticated lighting, including structured light and multi-spectral imaging, enhances contrast and highlights subtle features that might be invisible under standard lighting [9]. Despite these gains, capturing extremely small objects or defects necessitates pushing optical resolution limits, often requiring specialized microscopy techniques [10]. Another critical advancement lies in high-speed image acquisition and processing. To keep pace with high-speed production lines, real-time inspection demands high-frame-rate cameras and powerful processing units, such as GPUs and FPGAs [11]. Parallel processing, pipelined architectures, and hardware accelerators are used to minimize latency and handle the large data streams from high-resolution sensors. However, the computational demands of processing these images, particularly with complex deep learning models, remain a challenge, especially when detecting small objects that require fine-scale analysis across the entire image.

2.1. Small Object Detection in Industry

Object detection enhances accuracy and efficiency in visual inspection across diverse industries. It is used to identify defects on assembly lines, assist in medical diagnoses by finding abnormalities in scans, and analyze customer behavior in retail. By detecting minute flaws like cracks or scratches, this technology improves quality control, leading to higher customer satisfaction and less waste [12]. For such systems, particularly on high-speed manufacturing lines, detecting small objects with high speed in non-negotiable. Fast object detection models are essential to instantly identifying defects, ensuring that production throughput is maintained without compromising quality control. In industrial use cases, further importance is given to detecting small objects in addition to speed [13,14]. Even newer object detection models by design suffer in this regard [15]. Small objects inherently possess fewer discriminative features compared with their larger counterparts, a direct consequence of their limited pixel representation. Another significant challenge arises from the interplay between receptive fields and multi-scale representation. Deep neural networks often employ large receptive fields to capture broader contextual information, which is beneficial for understanding the overall scene [16]. However, a critical mismatch can occur when the receptive field for a low-resolution feature map becomes larger than the small object itself. In the literature much effort has been put into improving models such as RT-DETR (Real-Time DEtection TRansformer) [17] for small object detection. Numerous studies have also applied RT-DETR in visual inspection systems [18,19]. One main reason for the widespread adoption of RT-DETR is the limitation on the number of bounding boxes predicted by YOLO detectors, which has driven development toward transformer-

based models. However, beyond RT-DETR, there still has been a surge in object detection methods specifically tailored for small object detection in recent years. For example, improved Sparse R-CNN variants have been proposed for applications such as traffic sign detection in autonomous driving, demonstrating that sparsity-based approaches can be effective for small targets in complex environments [20]. Similarly, single-stage detectors such as YOLO continue to evolve: TSD-YOLO, based on improvements to YOLOv8, is specifically designed for small traffic sign detection and shows strong results in real-time applications [21]. These represent some of the industry-driven advancements for specific object detection use cases.

Furthermore, extracting enough rich semantic information is also essential to improving real-time performance of object detection systems. An ideal solution is RT-DETR, which balances high accuracy with exceptional speed. RT-DETR optimizes the DEtection TRansformer (DETR) architecture for real-time performance, eliminating the need for Non-Maximum Suppression (NMS) and demonstrating that transformer-based detectors can be efficient and end-to-end-trainable, bridging the gap between high accuracy and practical speed. RT-DETR was selected for this study. However, to determine whether the standard RT-DETR configuration or a specific variant would be most suitable, a literature review was undertaken. The objective of this review was to identify published research on improved or optimized versions of RT-DETR. The search was conducted using Google Scholar with the keywords 'Improved RT-DETR' and 'Optimized RT-DETR'. The top forty results were considered, resulting in the selection of thirty-two articles and the rejection of eight. The publication date range for the reviewed articles was 1 January 2024 to 25 March 2025. The comparison of these models (children of RT-DETR) is presented in Table 1. The excluded papers primarily fell into two categories: those that focused on improving specific applications in one domain without contributing substantial changes to the underlying model architecture and those that proposed alternatives to the DETR framework rather than building upon or improving the RT-DETR architecture, which is the focus of this study.

2.2. RT-DETR

RT-DETR (Real-Time DEtection TRansformer) is an evolution of the DETR (DEtection TRansformer) [22] model which removed Non-Max Suppression (NMS) and employed an end-to-end transformer architecture for object detection. The RT-DETR architecture can be modularly divided into four key components, backbone, feature fusion, attention mechanism (hybrid encoder and decoder), and loss function, as depicted in Figure 1. The backbone is responsible for extracting multi-level features (shown as S3, S4, and S5 in Figure 1) from the input image, typically using lightweight yet powerful convolutional networks such as ResNet. There are various versions of the RT-DETR model depending on the backbone network. For example, RT-DETR-L utilizes HGNetv2 as the backbone, whereas RT-DETR-R18/R34/R50 use ResNet18/34/50. Next is the hybrid encoder, within which the feature fusion module integrates feature maps. It is called a hybrid encoder because it combines the attention-based feature interaction (AIFI) module and the CNN-based Cross-Scale Feature Fusion (CCFF) module, leveraging the strengths of both transformers and convolutional neural networks. The CCFF module at the neck of the hybrid encoder plays the role of preserving both spatial detail and semantic context. These fused features are then processed by a decoder, where the global relationships captured by the encoder are refined, and the decoder matches a fixed set of learned object queries to potential detection results. The attention mechanism in the hybrid encoder and decoder leverages a transformer-based architecture that employs self-attention and cross-attention to model spatial and contextual relationships between image regions and object queries. Finally, the

loss function combines Generalized Intersection over Union loss, classification loss, and bounding box regression loss to learn both object classification and localization.

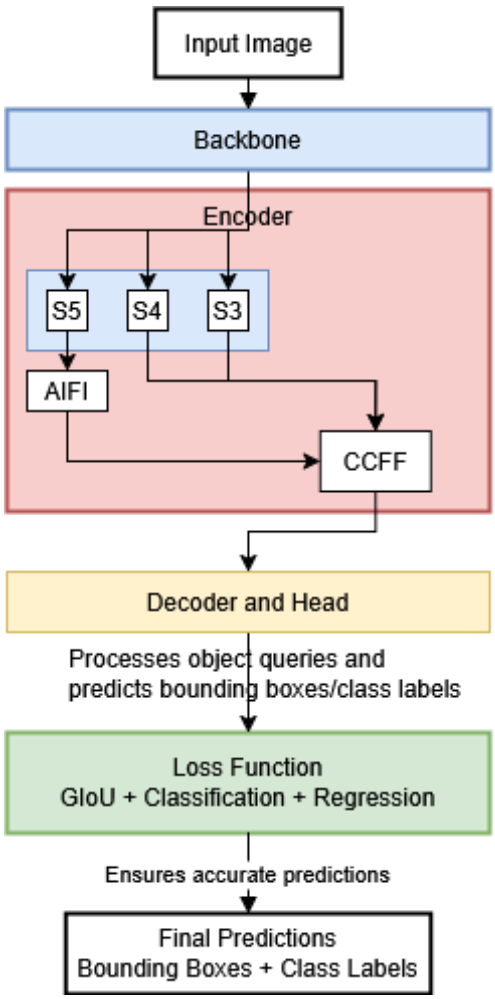

Figure 1. Simplified architecture of the RT-DETR model, comprising four main components: (1) a backbone network that extracts hierarchical visual features from the input image; (2) a Cross-Scale Feature Fusion (CCFF) module that combines multi-scale features to enhance representation; (3) the attention mechanism, which enables global context modeling and object query refinement; and (4) a set-based loss function that enables end-to-end training by directly matching predictions with ground-truth objects.

The model introduced several innovations to improve object detection performance while maintaining real-time inference speed [17]:

- Query selection module: It uses object queries which are learned embeddings that represent potential objects in the image.
- Attention mechanism: RT-DETR employs conventional multi-head self-attention within the encoder and cross-attention in the decoder that interacts with object queries and multi-scale image features. Certain variants also utilize deformable attention to enhance efficiency and accelerate inference.
- Hybrid encoder: Made from two modules, CNN-based Cross-Scale Feature Fusion (CCFF) and Attention-based Intra-Scale Feature Interaction (AIFI). CCFF and AIFI form the basis of feature fusion as in the neck of the hybrid encoder.

These innovations make RT-DETR a powerful alternative to traditional object detection models, particularly for scenarios requiring both high accuracy and real-time performance.

Table 1. Comparison of selected derivatives of RT-DETR from literature review.

Authors and Domain	Primary Challenges	Backbone Improvements	Feature Fusion	Attention Mechanism	Additional Innovations	Small Object Focus
1. Railway rutting defects [23]	Irregular defect distribution, varying sizes, and complex backgrounds	Faster CGLU module (combines PConv and gating mechanisms)	BiFPN with learnable weights (replaces CCFF)	H-AIFI with Hilo attention (high-/low-frequency paths)	Partial convolution	Yes
2. Free-range chicken detection [24]	Small target detection, multi-scale targets, and clustering occlusions	SDTM (Space-to-Depth Transformer Module)	Standard	BiFormer (sparse and fine-grained attention)	contextual token (CoT) module, Space-to-Depth conversion	Yes
3. Video object detection [25]	Motion blur, occlusions, and poor lighting in video frames	Standard	Decoupled Feature Aggregation Module	Separate self-attention	Two-step training strategy	Indirect
4. Detection in drone aerial images [26]	Small object sizes, indistinct features, and motion blur	CF-Block with CGLU (Convolutional Gated Linear Unit) and FasterNet Block with PConv	SOEP (Small Object Enhance Pyramid) with SPDConv + Omni-Kernel Module	Channel and spatial attention in HS-FPN	W-ShapeIoU loss function	Yes
5. Remote sensing object detection [27]	Infrared ship detection, small objects, and variable sizes	ResNet18	DRB-CFFM (Dilated Reparam Block-Based Cross-Scale Feature Fusion Module)	CGA-IFI (Intra-Scale Feature Interaction), which uses cascaded group attention	EIoU loss function	Yes
6. Drone object detection [28]	UAV aerial images, small objects, and diverse backgrounds	ESDNet backbone network with Fast-Residual and shallow feature enhancement module	Feature fusion using shallow SFEM layer	Enhanced Dual-Path Feature Fusion Attention Module	–	Yes
7. Drone object detection [29]	UAV aerial images, small objects, and diverse backgrounds	ResNet18	Re-calibration attention unit and re-parame-terized module	Deformable attention mechanism called DAttention	Focaler-IoU loss function	Yes
8. Pavement distress detection [30]	Complex road backgrounds, diverse shapes, and high computational resource requirements	Enhanced backbone made from ADown module and the uniquely crafted Layer Aggregation Diverse Branch Block	Proposed MEFF (Multi-enhanced feature fusion) with DySample and DBBFuse	Optimized Intra-Scale Feature Interaction with AICBAM (combining channel and spatial attention)	–	Indirect

Table 1. *Cont.*

Authors and Domain	Primary Challenges	Backbone Improvements	Feature Fusion	Attention Mechanism	Additional Innovations	Small Object Focus
9. Drone object detection [31]	Complex background, drastic scale changes, and dense small targets	GHSA (Gated Single-Head Attention) Block into ResNet-18	Proposed ESO-FPN, combining SPD Conv and LKDA-Fusion for fine-grained details	Enhanced AIFI with MMSA (Multi-Scale Multi-Head Self-Attention)	Introduced ESVF Loss, extending variFocal loss with EMA and slide weighting	Yes
10. Fire smoke detection [32]	Blurry smoke, high variability, and small objects	Enhanced with attention modules and 4D input	Redesigned with multi-scale fusion, 3D conv, and small object branch	Standard	New dataset and use of (EIoU) as the regression loss	Yes
11. Wheat ears detection [33]	Occluded wheat ears, complex background, and dense objects	Introduced Space-to-Depth (SPD-Conv) with non-stride convolutional layer	Context-Guided Blocks (CGBlocks) to form CGFM (Context-Guided Cross-Scale Feature Fusion Module)	Standard	Focal Loss to handle class imbalance and optimized weights for losses	No
12. Ground glass pulmonary nodule [34]	Small and ill-defined objects, and irregular shapes	Introduced FCGE (FasterNet + ConvGLU + ELA) blocks in ResNet18	HiLo-AIFI instead of AIFI and DGAK (Dynamic Grouped Alterable Kernel) blocks for fusion in CCFF	HiLo module to replace Multi-Head Self-Attention	–	Yes
13. Fruit ripeness detection [35]	High computation and multiple objects	PP-HGNet with Rep Block (with PConv) and Efficient Multi-Scale Attention (EMA) after the Stem block	Standard	Standard	Cross-space learning by reshaping channel dimensions into batch dimensions	No
14. Human detection [36]	Limited resolution, lack of detail, and poor contrast	ConvNeXt	Feature Pyramid Network (FPN) is added between the backbone and the encoder	Standard	GIoU loss is replaced with CIoU loss	No
15. Infrared electrical equip. detection [37]	Small targets and irregular target shapes	ResNet18 backbone is replaced with a custom-designed Multi-Path Aggregation Block (MAB)	The RepC3 module in the neck (CCFM) is replaced with the GConvC3 module	Multi-Scale Deformable Attention (MSDA)	BBox loss function is replaced with Focaler-EIoU	Yes

Table 1. *Cont.*

Authors and Domain	Primary Challenges	Backbone Improvements	Feature Fusion	Attention Mechanism	Additional Innovations	Small Object Focus
16. Tomato ripeness detection [38]	Computational cost and diverse shapes, sizes, and ripeness stages	ResNet-18 with a custom PConv Block	Neck network with a slimneck-SSFF architecture combining GSConv, VoVGSCSP, and the SSFF module	Deformable attention in AIFI (encoder)	Integrated Inner-IoU loss with EIoU into a new Inner-EIoU loss	Yes
17. Wafer surface defect detection [39]	Small particle defects and elongated linear scratches	ResNet50 with Dynamic Snake Convolution (DSConv)	CCFM module is replaced with a custom Residual Fusion Feature Pyramid Network (RFFPN)	AIFI module in the encoder is replaced with Deformable Attention Encoder (DAE)	–	Yes
18. Jet engine blade surface defect detection [18]	Small particles, scratches, and cracks	ResNet18 that integrates partial convolution (PConv) and FasterNet	CCFM is replaced with HS-FPN (with channel attention (CA) and Selective Feature Fusion (SFF) module)	Standard	Introduced IoU-aware query selection mechanism and Inner-GIoU loss fn	Yes
19. Rail defect detection [40]	Small target detection and noise interference	ResNet-50 with the Bottle2Neck module from Res2Net	Adds the RepBi-PAN structure to enhance CCFF	Standard	Uses loss WIoU and Hard Negative Sample Optimization Strategy	Yes
20. Infrared ship detection [41]	Small low-contrast objects, scale disparity, and complex marine environments	CPPA backbone (Cross-Stage Partial with Parallelized Patch-Aware Attention)	HS-FPN (High-Level Screening FPN) with channel attention	MDST (Multi-Layer Dynamic Shuffle Transformer)	Multi-branch feature extraction (local, global, and convolution)	Yes
21. Thermal infrared object detection [42]	Poor contrast and noise interference	Introduced partial convolution (PConv) and FasterStage to optimize ResNet18	Replaced RepC3 with FMACSP (Feature Map Attention Cross-Stage Partial)	Efficient Multi-Scale Attention (EMA) in FMACSP (attention in fusion block)	–	No
22. Remote sensing object detection [43]	Improving small object detection in remote sensing imagery	Standard	Made EBiFPN (enhanced BiFPN with DySample)	Cascaded group attention	Novel loss function called Focaler-GIoU	Yes
23. Coal gangue detection [44]	Computational complexity, small targets, indistinct features, and complex background	FasterNet network with EMA attention to improve FasterBlock module	In CCFM block, RepC3 is upgraded to Dilated Re-param Block (DRB)	Enhances the AIFI module with improved learnable position encoding	Data augmentation with Stable Diffusion + LoRA	Yes

Table 1. *Cont.*

Authors and Domain	Primary Challenges	Backbone Improvements	Feature Fusion	Attention Mechanism	Additional Innovations	Small Object Focus
24. UAV-based power line inspection [45]	Small object detection, low detection accuracy, and inefficient feature fusion	Replaced with GELAN (Generalized Efficient Layer Aggregation Network) for better feature extraction and efficiency	Instead of CCFF, handled within GELAN, improving multi-scale semantic fusion	Standard	New Reweighted L1 loss focusing on small objects	Yes
25. Drone object detection [31]	Small object detection, lightweight model design, and efficient feature fusion	Introduces GSHA block and MMSA in ResNet18	Proposes ESO-FPN, using large kernels + dual-domain attention to fuse features effectively for small objects	Standard	Introduces ESVF Loss (EMASlideVari-Focal Loss) to dynamically focus on hard samples	Yes
26. Tomato detection [46]	Small object detection, occlusion handling, low detection accuracy, and efficiency in complex environments	ResNet-50 replaced with Swin Transformer	Implicit fusion via Swin Transformer + BiFormer dual-level attention (no separate fusion module)	Standard	–	Yes
27. RT-DETRv3 (General) object detection [47]	Sparse supervision, weak decoder training, and slow convergence	Retains ResNet but adds CNN-based auxiliary branch for dense supervision during training	Better encoder learning from auxiliary supervision	Introduces self-attention perturbation and shared-weight decoder branch for dense positive supervision (training only)	VFL and distributed focus loss (DFL)	Indirect
28. RTS-DETR (general) object detection [48]	Small object detection, positional encoding limitations, and feature fusion inefficiency	Standard	Improves CCFM with Local Feature Fusion Module (LFFM) using spatial and channel attention, and multi-scale alignment	Core attention unchanged; improved via LPE and attentional fusion	Introduces new loss using Normalized Wasserstein Distance (NWD) + Shape-IoU for better small object regression accuracy	Yes
29. Aquarium object detection [49]	Small object real-time detection	HGNetv2 with ImageNet pre-training	Learnable weights for feature maps	Standard	Bottom-up paths preserving details	Yes

Table 1. *Cont.*

Authors and Domain	Primary Challenges	Backbone Improvements	Feature Fusion	Attention Mechanism	Additional Innovations	Small Object Focus
30. Traffic sign detection [50]	Small, distant, and poorly defined traffic signs	FasterNet Inverted Residual Block	Enhanced CCFM + S2 shallow layer integration	ASPPDAT (ASPP + Deformable Attention Transformer)	ASPPDAT, Inner-GIoU Loss, S2 fusion, and FasterNet in RT-DETR	Yes
31. Steel surface defect detection [51]	Resource constraints in industrial settings and need for edge deployment	MobileNetV3 (lightweight architecture)	DWConv and VoVGSCSP structure	Standard	MPDIoU loss function for improved bounding box prediction	No
32. Open-set object detection [52]	Novel class detection	Standard	RepC3 is replaced in CCFM (Manhattan Self-Attention)	MaSA (Manhattan Self-Attention)	MPDIoU Loss, a new bounding box regression loss	No

Learning from the Literature Review

The literature review of recent RT-DETR-based object detection models presented in Table 1 reveals consistent efforts to address domain-specific challenges through architectural and methodological innovations. For the comparison, we evaluated the selected works based on seven key features: application domain, primary challenges addressed, improvements made to the backbone, feature fusion methods, attention mechanisms used (primarily in the encoder and decoder), additional innovations, and whether the work specifically targeted small object detection.

The total thirty-two studies highlight the extensive adoption of the RT-DETR model across a variety of applications, such as drone-based imaging, remote sensing, industrial defect identification (e.g., railways, surfaces, and wafers), and agricultural monitoring (e.g., crops and livestock). A significant emphasis in these implementations lies in boosting performance by introducing innovative loss functions. Many researchers have proposed tailored bounding box regression losses, including EIoU, WShapeIoU, Focaler-IoU, WIoU, Inner-IoU variants (such as Inner-EIoU and Inner-GIoU), NWD+Shape-IoU, and MPDIoU, primarily to enhance localization precision, particularly for small or complex objects. Additionally, some efforts focus on refining classification losses, such as Focal Loss, ESVF Loss, and VFL/DFL, to address class imbalance or prioritize challenging samples. Apart from advancements in loss functions, notable innovations encompass architectural adjustments like contextual token modules (CoT), feature fusion methods, specialized training approaches (e.g., two-step processes and hard negative optimization), attention mechanisms, and cutting-edge data augmentation techniques leveraging generative models.

In terms of feature fusion, most approaches adapted the Cross-Scale Feature Fusion (CCFF) block, a CNN-based module originally designed to fuse features across different scales. Regarding attention mechanisms, the focus is mainly on modifications or adjustments applied to both the encoder and decoder. In the original RT-DETR architecture, the Attention-based Intra-Scale Feature Interaction (AIFI) block applied self-attention to the top-level features extracted by the backbone. Some works extended this by integrating attention directly into the backbone or at other stages of the network—for example, in the work by [35]. At least twenty-three of the reviewed works specifically focus on improving small object detection, which remains a persistent challenge across diverse domains,

such as railway inspection, poultry monitoring, and aquarium surveillance. Across the literature, there is no standard definition for a 'small object'. It is typically defined by quantitative criteria, such as absolute pixel count or relative size. For example, an object is often considered small if it is less than 32×32 pixels (as in the MS COCO dataset) or occupies less than 1% of the total image area. In specialized fields like aerial imagery, the threshold can be even smaller, such as 20×20 pixels. Out of the twenty-three models, many employ strategies like advanced feature fusion, attention mechanisms, and preservation of low-level features to enhance detection accuracy for small targets. Additionally, multiple works, including FHB-DETR [23] and MAFF-DETR [41], improve backbone efficiency by incorporating lightweight modules such as Faster CGLU or MobileNetV3, indicating a strong emphasis on reducing computational complexity for real-time or edge deployment scenarios. Moreover, training strategies and loss function innovations are highlighted in several models. The optimized RT-DETR with FAM in [25] introduces a two-stage training process that decouples localization from classification, improving convergence and accuracy. Similarly, RS-DETR [43] proposes a novel loss function (Focaler-GIoU) aimed at improving bounding box regression, especially for small and overlapping objects. These innovations underscore a broader trend of adapting training pipelines to better fit the detection objectives and dataset characteristics.

From Tables 1 and 2, several patterns emerge across RT-DETR backbone modifications. A large number of works retain the original ResNet backbone but introduce targeted enhancements such as partial convolutions, gated units, or aggregation blocks (e.g., Entries 5, 7, 9, 15–19, 21, 25, and 27). Another group of methods replaces the backbone with lightweight CNNs such as FasterNet, HGNet, GELAN, or MobileNetV3 (e.g., Entries 4, 6, 13, 23, 24, and 29–31), prioritizing efficiency for real-time industrial settings. Transformer-inspired alternatives, including ConvNeXt and Swin Transformer (Entries 14 and 26), emphasize global context modeling but increase computational costs. Attention-based modules (e.g., Space-to-Depth, EMA, GHSA, MMSA, and CPPA) are commonly integrated to enhance multi-scale representation, with particular focus on small object detection. Finally, several entries (3, 22, 28, and 32) retain the standard RT-DETR backbone to serve as baselines. Overall, the literature shows a clear trade-off: lightweight CNN-based improvements generally preserve real-time speed but may struggle with very small object detection, while attention-heavy or transformer-based backbones improve accuracy at the expense of efficiency. Our approach differs in that it does not redesign the backbone but instead introduces variations to increase the robustness of the encoder.

Furthermore, out of the derivates of RT-DETR that primarily adapted the original RT-DETR model by modifying specific components—such as replacing the backbone with a Swin Transformer in RT-DETR-Tomato [46]—RT-DETRv3 [47] represents a significant evolution of the RT-DETR framework. While other models focused on improving detection performance for certain use cases or object types (e.g., tomatoes or small objects) through component-level changes, RT-DETRv3 directly addresses a fundamental limitation of the RT-DETR architecture: the sparse supervision caused by one-to-one Hungarian matching during training. Rather than simply enhancing existing modules, RT-DETRv3 introduces a hierarchical dense positive supervision strategy that fundamentally rethinks how supervision is provided to both the encoder and decoder. Another model (not included in the comparison) into which RT-DETR has evolved is RF-DETR (an SOTA Real-Time Object Detection Model) [53]. RF-DETR improves upon the original RT-DETR by combining a lightweight transformer design with a powerful pre-trained DINOv2 backbone, allowing it to achieve higher accuracy (60+ mAP on COCO) while still running in real time. Unlike RT-DETR, which focuses mainly on speed and simplicity, RF-DETR is designed for better adaptability across domains.

In summary, the literature demonstrates a clear focus on improving small object detection, optimizing backbone architecture for efficiency, enhancing feature fusion, and refining attention mechanisms. While each model tackles different application domains, the recurring use of lightweight designs, multi-scale fusion, and custom attention modules illustrates a shared direction in improving the robustness and applicability of RT-DETR-based object detectors across real-world scenarios.

Table 2. Summary of design choices across surveyed works (numbers refer to papers 1–32 in Table 1). This grouping highlights recurring trends and research gaps.

Category	Representative Papers (IDs) and Observations
Backbone innovations	All except 4 (3, 22, 28, and 32) apply modifications to the backbone. Backbone modifications are the most common approach. Authors often replace ResNet with lighter (MobileNetV3 and FasterNet) or specialized designs (HGNetv2, GELAN, and ConvNeXt). This reflects the need for stronger low-level feature extraction for small objects. However, frequent reliance on ImageNet-pre-trained CNNs shows a trade-off: novelty vs. transferability.
Neck/feature fusion modules	All except 2 (2 and 13) apply modifications to the neck directly in the feature fusion part. Multi-scale feature aggregation is nearly universal, with BiFPN/EBiFPN, SOEP, slim-neck SSFF, and custom fusion blocks. These methods improve small object recall but also add computational overhead.
Training strategies and supervision	Not so common as seen in works by 3, 13, 19, 23, and 29. Auxiliary branches, two-step training, synthetic augmentation, and diffusion-based data expansion are less common but strategically important. This reflects recognition of data scarcity and weak supervision for small targets. However, limited adoption indicates challenges in reproducibility and computational costs.

3. Proposed New Child of RT-DETR

Transformer-based detectors leverage self-attention mechanisms to perform object localization across the entire image. However, as noted in [32], these models often emphasize larger target regions, which can lead to sub-optimal performance when detecting small objects. The modifications proposed should directly result in improving performance in small object detection. There are three modifications that we build incrementally into one final optimized configuration at the end.

3.1. Increasing Representational Capacity of Encoder

In the RT-DETR model architecture, the feed-forward neural network (FFN) within each encoder layer plays a critical role in enhancing the model's representational capacity. This FFN is situated after the multi-head self-attention mechanism and follows the subsequent steps of layer normalization and the GELU activation function. Within the FFN, the first linear transformation expands the dimensionality of the input embeddings, increasing the feature space to allow for more complex representations. This expansion is followed by another GELU activation, which introduces non-linearity into the processing pipeline. Afterward, a second linear transformation compresses the expanded representation back to the original dimensionality, ensuring compatibility with the rest of the encoder layer. Both linear layers form the block of fully connected layers in the architecture. We decided to experiment with replacing this block with the following layers:

- Hebbian Layer: Inspired by Hebbian theory, this layer updates weights based on the correlation between the input and output neurons. The learning rule strengthens connections where simultaneous activation occurs, but it lacks a mechanism to weaken connections, which can lead to unbounded growth of weights.

- Oja's Layer: An extension of Hebbian learning, Oja's rule modifies Hebbian learning to include a normalization factor that prevents the weights from growing indefinitely. It does this by introducing a forgetting factor, ensuring that the sum of the squares of the weights converges to a constant, thus stabilizing the learning process.
- Randomized Layers: The concept of randomized layers involves using network components where weights are randomly initialized and then frozen, exempting them from training. In our implementation, we replace specific feed-forward (nn.Linear) layers in the transformer encoder with a custom two-part module. The first part is a hidden layer with fixed random weights, initialized uniformly between -1 and 1. The second part is a standard, trainable output layer.

3.1.1. Detailed Discussion for Randomized Layers

As mentioned above, in our modification of the RT-DETR model with a Randomized Neural Network (RNN), we replaced the standard feed-forward layers (`fc1` and `fc2`) in the first transformer encoder block. In this configuration, the hidden layer weights are initialized randomly and remain fixed during training, while only the output layer weights are learned analytically using a closed-form solution. Mathematically, the transformation applied to the input $\mathbf{x} \in \mathbb{R}^{d_{\text{in}}}$ is as follows:

$$\mathbf{h} = \sigma(\mathbf{W}_{\text{rand}}\mathbf{x}), \quad \mathbf{y} = \mathbf{W}_{\text{out}}\mathbf{h}$$

where $\mathbf{W}_{\text{rand}} \in \mathbb{R}^{d_{\text{hidden}} \times d_{\text{in}}}$ is the fixed, randomly initialized weight matrix (drawn uniformly in $[-1, 1]$), $\sigma(\cdot)$ is a non-linear activation function (e.g., sigmoid), and $\mathbf{W}_{\text{out}} \in \mathbb{R}^{d_{\text{out}} \times d_{\text{hidden}}}$ is the output weight matrix.

Rather than training $\mathbf{W}_{\text{out}}$ via backpropagation, we compute it using the Moore–Penrose pseudoinverse of the hidden activations. Given a batch of input samples $\mathbf{X} \in \mathbb{R}^{N \times d_{\text{in}}}$ and target outputs $\mathbf{Y} \in \mathbb{R}^{N \times d_{\text{out}}}$, we first compute the hidden layer activations:

$$\mathbf{H} = \sigma(\mathbf{X}\mathbf{W}_{\text{rand}}^{\top})$$

Then, the optimal output weights minimizing the least squares error are given by

$$\mathbf{W}_{\text{out}} = \mathbf{Y}^{\top}\mathbf{H}^{+}$$

where $\mathbf{H}^{+}$ denotes the Moore–Penrose pseudoinverse of $\mathbf{H}$.

Practical Considerations

Unlike backpropagation, the pseudoinverse is not naturally compatible with mini-batch training: each batch yields a different $\mathbf{H}^{+}$ and hence a different local solution for $\mathbf{W}_{\text{out}}$. In our current implementation, we recompute $\mathbf{W}_{\text{out}}$ batch by batch, effectively treating each update as an approximation rather than a single consistent global solution. This means that $\mathbf{W}_{\text{out}}$ adapts locally to each mini-batch rather than being reconciled across the full dataset. In principle, reconciliation can be achieved in several ways, for example, (i) by computing $\mathbf{W}_{\text{out}}$ once using the entire training set (offline ELM-style training), (ii) by maintaining a running buffer of activations and recomputing the pseudoinverse periodically on a larger subset, or (iii) by adopting incremental least squares techniques such as Recursive Least Squares (RLS) that update a single consistent $\mathbf{W}_{\text{out}}$ across mini-batches.

Motivation

The rationale behind this modification is twofold: (i) reduce training complexity by avoiding gradient updates for certain layers and (ii) investigate whether randomized feature mappings enrich the representation space for downstream detection. Random-

ized projections have connections to kernel approximation and can, in some cases, improve generalization, which may be particularly useful for challenging tasks such as small object recognition.

Intuitive Explanation

The pseudoinverse can be understood as the operation that finds the 'best fit' weights in the least squares sense. Concretely, we want $\mathbf{W}_{\text{out}}$ such that the predictions $\mathbf{H}\mathbf{W}_{\text{out}}^{\top}$ are as close as possible to the targets $\mathbf{Y}$. Since an exact solution may not exist (or there may be infinitely many), the pseudoinverse chooses the solution that minimizes the squared error between predictions and targets. In other words, it plays the same role as ordinary linear regression: finding the line (or hyperplane) that best fits the data by minimizing the overall discrepancy.

3.2. Improving Extracted Features by Encoder

We argue that the encoded features produced by the original RT-DETR model are insufficient, as they fail to capture the fine-grained details necessary for small object detection. The following modifications, illustrated in Figure 2, were integrated into the RT-DETR model to enhance its encoder's feature processing, particularly for improving the detection of small objects and handling multi-scale features, while preserving the AIFI and CCFF stages of the original architecture. Figure 2 is drawn in such a way as to show from where the features were taken and which sections are then modified. This can be directly seen in comparison to Figure 1 of the original architecture. The arrow at the top in Figure 2 shows that the features passed inside the hybrid encoder are adapted and then further passed on to the decoder. This figure also helps clarify the implementation details, as the names shown in the figure correspond directly to the layer names in the implementation of RT-DETR in PyTorch and the HuggingFace Transformers library. The core components introduced are the Multi-Scale Feature Extractor module, the Fuzzy Attention module with positional encoding, and the Multi-Resolution Fusion module, which operate on the encoder's output after the hybrid encoder has processed the multi-scale feature maps (S3, S4, and S5) from the backbone. The Multi-Scale Feature Extractor employs convolutional layers with kernel sizes of 1×1, 3×3, 5×5, and 7×7 to extract features at multiple scales from the encoder's last hidden state, which already contains fused information from S3, S4, and S5 via CCFF. The larger 7×7 kernel captures broader contextual information, which is critical to detecting small objects that may lack sufficient local detail in higher-resolution feature maps. These multi-scale feature maps are concatenated and fused via a 1×1 convolution, creating a rich, unified feature representation that enhances the encoder's ability to represent small objects. The Fuzzy Attention module further refines high-resolution features by incorporating positional encoding and an attention mechanism, improving localization and emphasizing relevant regions, which is particularly beneficial for small objects that require precise spatial context. To leverage multi-resolution information, the Multi-Resolution Fusion module extracts features from different encoder layers, representing transformed versions of S3, S4, and S5 after AIFI, CCFF, and transformer processing; aligns their spatial dimensions through upsampling; processes them with convolutional layers; and fuses them into a single feature map. This fused feature map replaces the encoder's last hidden state, providing the decoder with a more comprehensive representation that combines multi-scale and multi-resolution information, thereby improving the detection of small objects by ensuring that both fine-grained details and global context are preserved.

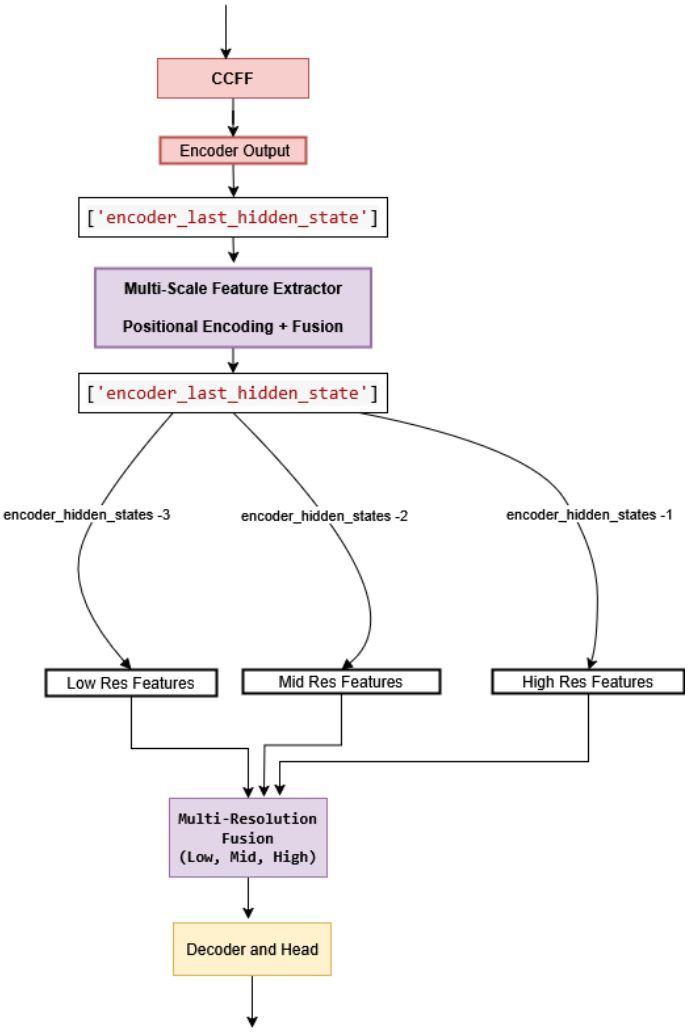

Figure 2. The diagram illustrates the adapted RT-DETR model, which processes the output of the hybrid encoder from the original model. The hybrid encoder combines backbone feature maps by using AIFI and CCFF. The Multi-Scale Feature Extractor and Fuzzy Attention module then refine this output, while the Multi-Resolution Fusion module combines features at different resolutions. Finally, the adapted output is passed to the decoder. The adapted architecture enhances small object detection and improves overall performance

3.3. Better Classification Loss

The original RT-DETR model employed a combination of three losses: classification loss, regression loss, and Generalized Intersection over Union (GIOU) loss. However, in this work, we have modified the loss function to improve the performance of the model in a similar fashion to [33]. Specifically, we have replaced the traditional classification loss with an adaptive focal loss (AFL) function. The AFL function is designed to adapt to different classes and balance the loss between them, which can help to improve the model's performance on classification tasks.

$$\text{Focal Loss} = -\alpha \cdot (1 - p_t)^{\gamma} \cdot \log(p_t), \tag{1}$$

where

- p_t represents the predicted probability of the true class;
- α is a balancing factor to address class imbalance;
- γ is a focusing parameter that reduces the loss contribution from easy-to-classify examples.

To compute the loss, we first encode the ground-truth labels into one-hot vectors and calculate the predicted probabilities by using the softmax function. The focal loss is then computed for each class and aggregated across all samples. The final adaptive focal loss is obtained by summing over the class dimensions and averaging across the batch:

$$\text{Adaptive Focal Loss} = \frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} -\alpha \cdot (1 - p_{t,i,c})^{\gamma} \cdot \log(p_{t,i,c}), \quad (2)$$

where

- N is the batch size;
- C is the number of classes;
- $p_{t,i,c}$ is the predicted probability of class c for sample i.

This formulation allows the model to dynamically adapt its focus to challenging examples, improving the classification accuracy for rare or difficult classes.

3.4. Final Configuration

In the final configuration of the proposed adapted RT-DETR model, the three previously described modifications were integrated to leverage their combined strengths and achieve optimal performance. Together, these modifications produced a synergistic effect, resulting in significantly improved outcomes compared with what could be achieved by applying any single modification in isolation. The proposed modifications differ from existing approaches in the literature in the following ways:

- The encoder's representational capacity was enhanced by replacing the original fully connected layer block with a more expressive structure.
- The features extracted by the hybrid encoder block were further refined using a combination of a Multi-Scale Feature Extractor (to capture features at varying spatial scales), Fuzzy Attention (to enhance multi-scale feature representation), and Multi-Resolution Fusion (to combine features from different resolutions—low, medium, and high—within the encoder).
- An adaptive focal loss was employed in place of the traditional classification loss. Unlike the Focaler-IoU proposed in [43] and the focal loss for classification in [33], this adapted loss function serves as a weighting mechanism in the classification part designed to better support the detection of smaller objects.

In the provided final configuration, the original model is extended by post processing the internal feature representations produced by the model's encoder. The modifications used for this are not inserted directly into the core architecture of the model, but instead act as auxiliary processing blocks that enhance the model's ability to capture and refine multi-scale and multi-resolution information. Although the base architecture remains intact, these additions increase the computational complexity of the model by introducing extra convolutional layers and operations, thereby increasing the number of parameters and processing time. At the same time, they enhance the effective complexity by improving the model's representational capacity, potentially leading to better performance on tasks such as small object detection and fine-grained feature recognition.

4. Description of Datasets Used for Model Evaluation

4.1. Al-Cast Detection

The Al-Cast dataset consists of 126 X-ray images of high-pressure die-cast aluminum parts, labeled for two types of internal defects: gas holes and shrinkage. The original images were captured in TIFF format with a resolution of 1000 × 1000 pixels and 8-bit grayscale. To facilitate their use in object detection algorithms, the images were later converted into

JPEG format without any additional preprocessing. Figure 3 shows one image from the dataset. The final dataset is divided into 2427 training images, 681 validation images, and 346 test images [54].

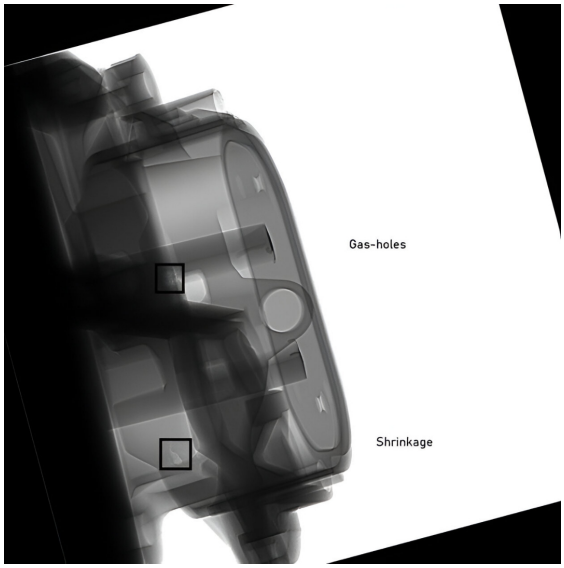

Figure 3. Sample from the Al-Cast dataset containing both defect classes: Gas Holes and Shrinkage.

4.2. Synthetic

To evaluate model performance under controlled conditions, a synthetic dataset consisting of 900 images (300 each for training, validation, and testing) designed to simulate various levels of occlusion and contrast was generated. The dataset includes two geometric object classes—circles and triangles—randomly placed on a white background with varying object sizes. The images in Figure 4 illustrate the challenges of partial occlusion under varying contrast conditions (low and medium), with the first image showing medium contrast and the second low contrast. In the dataset, each image contains a single object and belongs to one of two occlusion categories: no occlusion or partial occlusion. The images were originally created at a resolution of 480 × 480 pixels and saved in JPEG format. Occlusions were applied using randomly sized and positioned gray rectangular blocks to partially or heavily obscure the objects. For each occlusion and contrast combination, corresponding bounding box annotations in YOLO format were generated. This synthetic dataset is open-sourced, and it provides a controlled environment to assess object detection robustness against varying visual challenges such as occlusion and contrast changes.

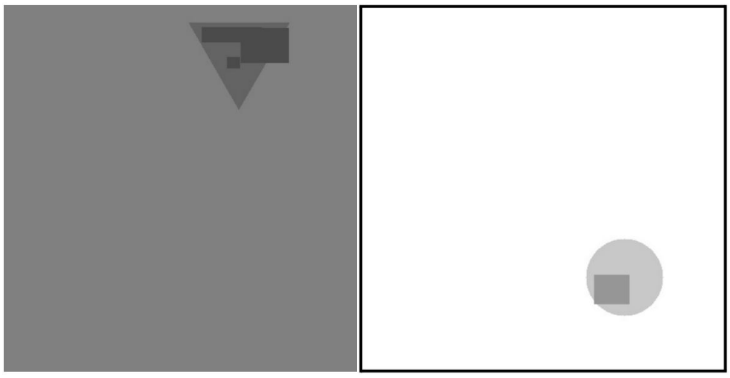

Figure 4. Images illustrating partial occlusion at different contrast levels from the synthetic dataset.

4.3. SerDes Receiver Pattern

To evaluate pattern recognition performance in a structured text-based setting, a synthetic dataset is formed with images generated from a single de Bruijn sequence. From the work [55], it is shown that de Bruijn sequences can be used for the above-compliance testing of SerDes receivers. The dataset simulates a grid-based textual environment by rendering an error sequence of PAM4-digits (4-level Pulse Amplitude Modulation) (0, 1, 2, and 3) into 224 × 224-pixel grayscale images, where each character is displayed in a 9 × 9 grid layout. Within this sequence, predefined digit patterns—such as '010', '030', and '333'—are found, and their corresponding bounding box annotations are generated in YOLO format. These patterns are also referred as problematic subsequences (PSSs). An example of such an image is shown in Figure 5. Learning from the START project [55], it can be deduced that some short patterns have a high possibility of causing design errors, and the aim is to find such small patterns. Each image encodes a contiguous 81-character block from a larger sequence, ensuring that the dataset captures both spatial and contextual variations in digit arrangement. The bounding boxes are precisely aligned to the grid cells encompassing each detected pattern, enabling accurate localization. A total of 1506 training images are used, and 205 images are used for validation.

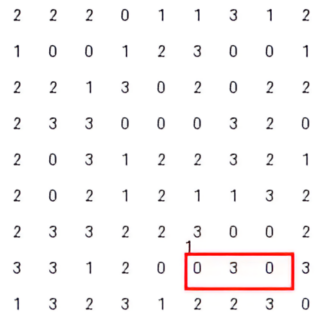

Figure 5. Segment of an error sequence as an image exhibiting the pattern 030 from the SerDes pattern dataset. The red box represents the bounding box drawn around the pattern 030.

4.4. PCB Dataset

The final quality inspection dataset used in this work is the PCB (printed circuit board) missing hole dataset. The dataset is freely available on the Roboflow universe [56]. It contains holes as defects in PCBs which are circular conductive areas with a small dark circle in the center, which represents a hole drilled through the board. One such example is shown in Figure 6. This type of feature is typically used for connecting component leads (through-hole pads) or for creating electrical connections between different layers of the PCB (vias). Overall, the dataset has 352 images divided as follows: 304 for training, 32 for validation, and 16 for testing.

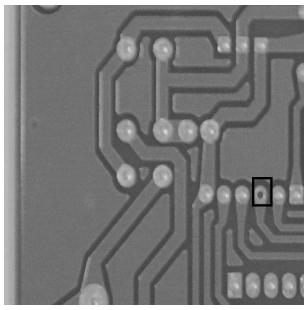

Figure 6. Image from the PCB quality inspection dataset, with the anomaly 'missing hole' marked by a bounding box.

4.5. Blood Cell Object Detection Dataset

To further evaluate the generalization of our RT-DETR modifications beyond industrial defect datasets, we performed experiments on a small-scale medical imaging dataset consisting of blood cell images. This dataset was originally open-sourced by `cosmicad` and `akshaylambda` and is publicly available via Roboflow [57,58]. It contains a total of 364 annotated images across three categories: white blood cells (WBCs), red blood cells (RBCs), and platelets. The dataset is split into 255 training images, 73 validation images, and 36 test images. The dataset provides a natural setting to assess small object detection capabilities, as platelet instances are significantly smaller in pixel area compared with RBCs and WBCs. A breakdown of object sizes per class is as follows:

- Platelets: 76 instances (51 small, 24 medium, and 1 large)
- RBCs: 819 instances (813 medium and 6 large)
- WBCs: 72 instances (10 medium and 62 large)

The dataset serves as a compact benchmark for testing medical imaging capabilities in object detection models. Detecting platelets is particularly challenging due to their small size, making this dataset well-suited for validating the small object detection improvements proposed in this study.

5. Experiments and Results

The experiments in this study are based primarily on the Al-Cast industrial dataset, selected for its relevance to the visual inspection domain. All optimization steps leading to the final configuration were conducted on this dataset to ensure consistency and practical applicability. The experiments were executed on Google Colab, utilizing a system with 12.7 GB of RAM and a GPU with 15.0 GB of memory, which provided sufficient computational resources for deep learning workflows. The implementation made use of various Python libraries, including `PyTorch` for model development, `Transformers` for leveraging pre-trained object detection models, `Torchmetrics` for evaluation using mean Average Precision (mAP), and `Albumentations` for data augmentation. Additional tools such as `Roboflow`, `Supervision`, and `PIL` were used to manage datasets and handle image processing tasks. The same hyperparameters were used for all experiments, with the seed value kept constant for reproducibility. The core implementation was adapted and extended from an open-source example provided in a publicly available blog [59], which served as a foundation for building and customizing the proposed approach. The model was trained for different epochs mentioned separately for each experiment with fixed seed, batch size of 16, learning rate of 5×10^{-5}, a maximum gradient norm of 0.1, and a warmup period of 300 steps to stabilize training. To evaluate the model's performance across a range of object sizes, we adopted a standard categorization scheme based on bounding box area. Objects were classified as small if their area was less than 32×32 pixels, medium if their area fell between 32×32 and 96×96 pixels (inclusive of the lower bound), and large if their area was 96×96 pixels or greater. These thresholds are commonly used in object detection benchmarks, such as the COCO dataset.

5.1. Metrics

In object detection, evaluation metrics such as mAP (mean Average Precision) provide a comprehensive measure of model performance. The most commonly used metric is mAP@[0.5:0.95], which averages precision across multiple IoU (Intersection over Union) thresholds ranging from 0.5 to 0.95 in steps of 0.05. Specific metrics like mAP@0.5 and mAP@0.75 correspond to precision at single IoU thresholds, representing lenient and strict localization criteria, respectively. Additionally, mAP is reported for different object sizes—small, medium, and large—based on the area of ground-truth boxes. These metrics

are calculated by matching predicted boxes to ground truths using IoU, then computing precision–recall curves and averaging the area under these curves to obtain the final AP values.

5.2. Al-Cast Dataset

5.2.1. Experiment 1: Comparison of Object Detection Models

In this experiment, we evaluate the performance of recent object detection models using the Al-Cast industrial dataset. The models are summarized below:

- YOLO-NAS: It is the item of progressed Neural Architecture Search incorporated in object detection, meticulously designed to address the confinements of past YOLO models [60]. Evaluated using pre-trained weights (COCO) and after fine-tuning for 20 epochs on the Al-Cast dataset.
- YOLO11vS: One of the most recent model in Ultralytics' YOLO series of real-time object detectors [61]. Yolo11Vs model version evaluated using pre-trained weights (COCO) and after fine-tuning for 20 epochs on the Al-Cast dataset.
- OWL-ViT: A foundation model based on a CLIP backbone [62], pre-trained on COCO and OpenImages (1.7M images). Fine-tuning for object detection is currently infeasible due to the lack of text descriptors for bounding boxes, as you need both for fine-tuning and only image modality with bounding boxes.
- RT-DETR: Trained on the COCO 2017 dataset (200k images) and evaluated using both pre-trained weights and fine-tuning for 20 epochs on the Al-Cast dataset.

Table 3 presents the results, highlighting performance differences across the models.

Table 3. Comparison of object detection models on the Al-Cast validation set, fine-tuned for 20 epochs.

Model	Direct Inference	mAP@0.50
YOLO-NAS	0	0.7507
YOLO11s	0	0.7000
OWL-ViT	0	–
RT-DETR	0	0.7560

Next we want to evaluate performance in small object detection. With the original model configuration of standard RT-DETR, the mAP scores across object sizes (based on COCO area definitions) after 100 epochs of training on validation set were as follows:

- mAP (small) (area $< 32^2$ pixels): 0.389;
- mAP (medium) ($32^2 <$ area $< 96^2$ pixels): 0.589;
- mAP (large) (area $> 96^2$ pixels): N/A.

5.2.2. Experiment 2: Comparison of Alternatives to FCNN

In this experiment, we explore the effect of Modification 1. This can be seen in Table 4, where after 20 epochs of training, the best results were achieved when the Randomized Neural Network replaces the feed-forward part.

To further explore the potential of the best-performing alternative, the Randomized Neural Network was fine-tuned for 100 epochs. The best result achieved was the following:

- Overall mAP@0.50: 0.619 (validation set).

Despite this improvement, challenges in detecting small objects remained evident. RT-DETR, even with the addition of a Randomized Neural Network and extended training, struggled to accurately detect small objects. This limitation was observed during evaluation

on the validation set. The mAP scores across object sizes (based on COCO area definitions) after 100 epochs on validation set were as follows:

- mAP (small) (area $< 32^2$ pixels): 0.23;
- mAP (medium) ($32^2 <$ area $< 96^2$ pixels): 0.83;
- mAP (large) (area $> 96^2$ pixels): N/A.

Table 4. Performance on the validation set of alternative layers replacing the feed-forward network (fine-tuned for 20 epochs).

Layer Type	Direct Inference	mAP@0.50:0.95	mAP@0.50	mAP@0.75
Linear (original)	0	0.39	0.75	0.34
HebbianLayer	0	0.43	0.82	0.40
Oja's Layer	0	0.33	0.70	0.24
Randomized Neural Network	0	0.45	0.88	0.40

Performance also varied by defect type, correlating with object size (after 100 epochs on the validation set), and the dataset had the following object size distributions:

- Training set: Small—2823, medium—1271, and large—0;
- Validation set: Small—788, medium—339, and large—0;
- Gas Holes (primarily small objects): mAP = 0.233;
- Shrinkage (larger objects): mAP = 0.739.

Even though the overall mAP@0.50 increased from 0.619 to 0.88, the performance in small object detection became worse, as evidenced by a drop in mAP (small) from 0.389 to 0.23. These results emphasize the difficulty of small object detection in industrial contexts, especially when using models like RT-DETR that may not be optimized for such use cases. Even after having more samples for small-sized defects, the performance remains poor.

5.2.3. Experiment 3: Randomized Layers with Enhanced Encoder Features

In this experiment, we explore the effect of Modification 2, that is, improving the extracted features from the last layer of the encoder with the Multi-Scale Feature Extractor module, the Fuzzy Attention module with positional encoding, and the Multi-Resolution Fusion module, together with the randomized layer part instead of the FCNN. Under this configuration, the randomized layers extract improved feature maps; then the final output of the encoder is adapted by fusing the different parts according to the resolution to leverage both fine-grained and high-level semantic information.

The results are displayed in the Table 5. These show the performance on the validation set after 100 epochs of training with two adaptations.

Table 5. Performance metrics for Modifications 2 + 1 on the validation set after 100 epochs.

Overall mAP@0.5	0.914
mAP@0.5 (Small Objects)	0.412

5.2.4. Experiment 4: Final Configuration with Combination of Adaptive Focal Loss and the Other Two Modifications

This experiment represents a comprehensive evaluation of the combined impact of three modifications on object detection performance on the Al-Cast detection dataset.

The results of this experiment is shown in Table 6. It focuses on using all the adaptations at once.

Table 6. Performance metrics for the final configuration (Modifications 1 + 2 + 3) on the validation set after 100 epochs.

Overall mAP@0.5	0.920
mAP@0.5 (Small Objects)	0.513

Final comparison on test set after the model was trained for 100 epochs. The results are shown in the Table 7. The experimental results showed a notable improvement in overall detection performance, with an mAP@0.5 of 0.900 on the test set, confirming the effectiveness of the proposed enhancements. For all the other datasets, we compared the results between the original model and the developed final configuration (Modifications 1 + 2 + 3), especially on small objects wherever possible. With the Al-Cast dataset, the optimized configuration demonstrated a significant improvement in detecting small objects, achieving an mAP@0.5 of 0.513 compared with 0.389 in the original model. This corresponds to a performance increase of approximately 31.9%. The results highlight the effectiveness of the optimization in enhancing small object detection capabilities. This can also be seen when the performance is compared with the latest release of YOLO detectors, which is the eleventh version. Using the model, performance in the mAP@0.5 metric was comparable to that of our adapted RT-DETR model. However, it still lagged behind in the mAP@0.50:0.95 metric. Furthermore, in order to have a better understanding of the effects of the modifications on the inference time, two experiments were conducted. The original RT-DETR model was directly compared with our adapted RT-DETR model firstly on the Google Colab platform and secondly on a local laptop with 12 GB RAM and a 2 GB NVIDIA GPU MX450. In the first experiment, an average inference time of 0.787 s per sample was seen with the original configuration versus 0.798 s. In the second experiment, an average inference time of 7.881 s with the original configuration was observed versus the 7.957 s with the optimized configuration. These results show that despite incorporating additional preprocessing steps, there was no significant increase in processing time.

Table 7. Comparison of test set results on the Al-Cast dataset using optimized RT-DETR (final configuration), original RT-DETR, and YOLOv11.

Final Config (Modifications 1 + 2 + 3)	mAP@0.5 = 0.900	mAP@0.50:0.95 = 0.750	mAP@0.75 = 0.630
Original RT-DETR	mAP@0.5 = 0.590	mAP@0.50:0.95 = 0.330	mAP@0.75 = 0.280
YOLO11vS	mAP@0.5 = 0.914	mAP@0.50:0.95 = 0.512	mAP@0.75 = N/A

5.2.5. Experiment 5: Ablation Study for All the Modifications

To individually study the effect of the adaptive focal loss, the Multi-Scale Feature Extractor, and the Fuzzy Attention module, an ablation study was performed. In each run everything else was kept, such as the hyperparameters, etc. What changed was that in the first run, the features from the encoder were directly passed only to the multi-resolution feature extractor with fusion; in the next run, this was removed, and the features were directly passed to the Fuzzy Attention module. However, in both cases, adaptive focal loss was still used, and in the last run, the original RT-DETR model was trained with only the adaptive focal loss and no Multi-Scale Feature Extractor or Fuzzy Attention module.

A comparison on the test set after the model was trained for 100 epochs can be seen below in Table 8. The results above show that the blocks individually have similar performance (mAP@0.50:0.95 = 0.480 vs. mAP@0.50:0.95 = 0.470 vs. mAP@0.50:0.95 = 0.469) on the test set after 100 epochs of training and that only when all of them are added together, the best results are achieved, with mAP@0.50:0.95 = 0.750.

Table 8. Comparison of test set results on the Al-Cast dataset with each modification applied separately.

Only Feature Extractor	mAP@0.5 = 0.810	mAP@0.50:0.95 = 0.480	mAP@0.75 = 0.540
Only Fuzzy Attention	mAP@0.5 = 0.850	mAP@0.50:0.95 = 0.470	mAP@0.75 =0.480
Only Adaptive Focal Loss	mAP@0.5 = 0.710	mAP@0.50:0.95 = 0.469	mAP@0.75 = 0.420

5.3. Synthetic Dataset

Table 9 presents the comparison of object detection performance (measured by mean Average Precision, mAP@0.5, for small objects on the validation set) between the original and final configurations. The model was trained for 20 epochs. In the No-Occ—Low Contrast scenario, both the original and optimized configurations achieved identical performance, with an mAP of 0.800, suggesting that the optimization did not yield further improvement under these specific conditions for small objects. However, for the No-Occ—Medium Contrast case, the optimized configuration slightly improved the mAP from 0.623 to 0.642, indicating better feature extraction or generalization under medium contrast without occlusion.

Table 9. Comparison of mAP@0.5 for small objects on the synthetic dataset validation set between the original and final configurations.

Condition	Original	Final Configuration
No-Occ—Low Contrast	0.800	0.800
No-Occ—Med Contrast	0.623	0.642
Partial-Occ—Low Contrast	0.785	0.812
Partial-Occ—Med Contrast	0.688	0.808

5.4. Pattern Dataset

Table 10 presents the performance results of the original and optimized configurations on the pattern dataset, evaluated using mean Average Precision (mAP@0.5) across three subsets: PSS1, PSS2, and PSS3. All the three patterns are designed to be small objects as defining objects. The performance shown is on the validation set after the model was trained for 20 epochs. The final configuration demonstrates improved performance on two out of the three subsets. For PSS1, the mAP increased from 0.903 to 0.932, and for PSS3, it rose from 0.881 to 0.918, indicating enhanced detection capability in these subsets. Although there is a slight decrease in PSS2 (from 0.943 to 0.936), the difference is minimal and does not significantly impact the overall trend.

Table 10. Comparison of mAP@0.5 for small objects on the pattern dataset validation set between the original and final configurations.

Subset	Original	Final Configuration
PSS1	0.903	0.932
PSS2	0.943	0.936
PSS3	0.881	0.918

5.5. PCB Dataset

The dataset is split into 304 training images, 32 validation images, and 16 test images, indicating a relatively small but structured dataset for model development and evaluation. Upon further evaluation, the validation set of the dataset reveals an uneven object size distribution within the 'Design' class, which represents holes, with a total of 72 instances.

The objects are categorized as medium-sized (62 instances), followed by a smaller portion of small-sized objects (10 instances), while no large-sized objects are present. The results obtained on the validation dataset with hyperparameters kept constant as indicated before are presented in Table 11.

Table 11. Comparison of metrics on the PCB dataset validation set between the original and final configurations after 100 epochs of training.

Metric	Original	Final Configuration
mAP@0.5	0.898	0.890
mAP@small	0.252	0.337
mAP@medium	0.431	0.459

Notably, the final configuration or newly adapted RT-DETR, despite incorporating additional preprocessing steps, did not contribute to any increase in processing time, achieving an average inference time of 0.477 s per sample—comparable to the original configuration (0.468 s) on the Google Colab platform.

5.6. Blood Cell Object Detection Dataset

We further validated our approach on the Blood Cell Detection Dataset, a small-scale medical imaging dataset containing 364 annotated images across three classes: red blood cells (RBCs), white blood cells (WBCs), and platelets. The dataset is divided into 255 training images, 73 validation images, and 36 test images [58]. A key challenge of this dataset lies in the highly imbalanced object size distribution. Platelets, in particular, are very small, with 51 small instances, 24 medium instances, and only 1 large instance across the dataset. In contrast, RBCs and WBCs are predominantly medium or large in size. This makes the dataset particularly relevant for evaluating small object detection performance in a medical imaging context. Table 12 summarizes the results on the validation set using the same hyperparameters as in earlier experiments. The proposed modifications yielded improvements in detecting small platelet instances. Specifically, mAP@0.5 for the platelets class improved from 0.449 to 0.547, confirming that the architectural modifications targeting small object detection generalized effectively also beyond industrial data to medical imagery.

Further per-class analysis reveals additional gains:

- mAP for RBC detection improved from 0.5126 to 0.5202.
- mAP for WBC detection saw the largest absolute increase, rising from 0.7770 to 0.870.

Table 12. Comparison of metrics on the BCCD dataset validation set between the original RT-DETR configuration and the final adapted model after 100 epochs of training.

Metric	Original	Final Configuration
mAP@0.5 (overall)	0.872	0.881
mAP@0.5 (platelets)	0.449	0.547

6. Discussion

In general, for all experiments conducted on Google Colab, the inference time was similar across both the optimized and original configurations. Notably, the optimized configuration, despite incorporating additional preprocessing steps, did not contribute to any increase in processing time, maintaining the same inference latency as the original setup. This preservation of processing efficiency, alongside the observed improvements in

mAP metrics, highlights a significant advantage for real-time visual inspection systems, where both rapid and accurate detection are paramount. Furthermore, the cumulative application of multi-scale feature extraction, fuzzy attention with positional encoding, and adaptive focal loss resulted in a more robust and accurate detection model.

This study investigated the modification of the RT-DETR object detection model for improved performance, particularly in the context of detecting small objects in industrial defect datasets. Our experiments focused on three key modifications: (1) replacing the feed-forward network (FFN) with alternative layers, (2) fusing feature maps from multiple encoder layers and refinement of extracted knowledge, and (3) incorporating adaptive focal loss instead of standard classification loss. We evaluated these modifications on the real-world aluminum casting defect dataset (Al-Cast). The best model configuration was then tested on three other datasets.

6.1. Experiment 2 (Modification 1: Randomized Layers)

Replacing the FFN with randomized layers yielded the best initial improvement in mAP on the Al-Cast validation set (Table 4). Further fine-tuning increased the overall mAP@0.5 to 0.619, while the performance on small objects remained a significant challenge (mAP (small) = 0.23). This highlights the inherent difficulty of detecting small objects, even with architectural modifications and extended training. The performance variation across defect types (e.g., 0.233 mAP for smaller objects versus 0.739 mAP for larger ones) further underscores the size-dependent performance limitations. This suggests that simply replacing a single component (the FFN) within a large, pre-trained model like RT-DETR is insufficient to overcome the challenges posed by small objects, which may require more fundamental changes to the model's architecture or training strategy.

6.2. Experiment 3 (Modifications 1 + 2: Randomized Layers with Enhanced Encoder Features)

Combining Modification 1 with feature map fusion (Modification 2) led to a substantial improvement in both overall mAP@0.5 (0.914) and mAP@0.5 for small objects (0.412) on the Al-Cast validation set (Table 5). This indicates that leveraging both fine-grained (from earlier encoder layers) and high-level semantic information (from later layers) is crucial to enhancing small object detection. The fusion process likely allows the model to better integrate local details with global context, improving its ability to distinguish small defects from the background.

6.3. Experiment 4 (Modifications 1 + 2 + 3: Final Configuration)

The final configuration, incorporating all three modifications, achieved the highest performance on the Al-Cast validation set (Table 6), with an overall mAP@0.5 of 0.920 and a small object mAP@0.5 of 0.513. This demonstrates the synergistic effect of the combined modifications. The improvement in small object detection, while still not perfect, represents a significant advancement compared with the baseline RT-DETR model and the individual modifications. The comparison on the Al-Cast test set (Table 7) confirms the substantial performance gain of the optimized configuration over the original RT-DETR.

6.4. Synthetic and Pattern Datasets

The results on the synthetic and pattern datasets (Tables 9 and 10) provide further insights into the strengths and limitations of the optimized configuration. On the synthetic dataset, the optimized configuration showed improvements in scenarios with partial occlusion and medium contrast, suggesting enhanced robustness to these challenging conditions. The minimal differences in the No-Occ—Low Contrast scenario indicate that the original model was already performing well under those easier conditions. More notable improvements were observed under the Partial Occlusion conditions. For Partial-Occ—Low

Contrast, the mAP increased from 0.785 to 0.812, and for Partial-Occ—Medium Contrast, the mAP showed a significant rise from 0.688 to 0.808. These improvements suggest that the optimized configuration is more robust to occlusion and varying contrast levels. On the pattern dataset, the optimized configuration consistently improved performance across most subsets (PSS1 and PSS3), demonstrating its ability to generalize to different types of small objects. The slight decrease on PSS2 is negligible and is very close to the optimized configuration.

6.5. PCB Dataset

The results on the PCB dataset also show a pattern similar to that of the Al-Cast detection dataset. The results are presented in Table 11. The results presented in the table compare the performance metrics of the original and optimized configurations on the PCB validation set, evaluated at the end of a 100-epoch training period. Across metrics for small- and medium-sized objects, the optimized configuration demonstrates superior performance compared with the original setup. For smaller objects, the mAP@small metric saw a substantial increase from 0.252 to 0.337, highlighting the optimized configuration's enhanced capability in detecting smaller features.

6.6. BCCD Dataset

The experimental results on the BCCD dataset validate the efficacy of the proposed architectural modifications in enhancing small object detection performance, as summarized in Table 12. Furthermore, the gains extend to other blood cell classes: the mAP@0.5 for red blood cells (RBCs) rose from 0.5126 to 0.5202, while that of white blood cells (WBCs) saw a leap from 0.7770 to 0.8702. It is worth noting that the platelet class remains the lowest-performing class. This suggests that there remains room for further optimization.

Overall, the newly adapted RT-DETR model demonstrates consistent or improved performance in all test scenarios. This study provides a strong foundation for developing more effective object detection systems for industrial applications, where the accurate detection of small defects is critical to quality control and process optimization.

7. Conclusions

To significantly improve the RT-DETR object detection model for detecting small objects in industrial defect datasets, it is essential to tailor the RT-DETR model specifically to the characteristics and challenges of industrial environments. To address this, having better loss function and integrating multi-scale feature aggregation techniques and improved transformer-based attention mechanisms can enhance the model's ability to capture low-level spatial cues across different scales. This is supported by our experiments and can also be seen in the literature review, where out of thirty-two works, only two use the standard original feature fusion module, while the others introduce some form of innovation in this aspect. This study demonstrated that the RT-DETR object detection model, while generally effective, can be significantly improved for detecting small objects in industrial defect datasets through the proposed targeted modifications. These modifications, especially when combined, resulted in substantial performance gains on a real-world aluminum casting defect dataset (Al-Cast), as evidenced by improvements in mAP for small objects, which increased from 0.389 to 0.513. The newly adapted RT-DETR model introduced in this work also showed improvements on the other datasets used in this work for validation. The results underscore the inherent challenges of small object detection in industrial settings, where limited visual features and variations in appearance make accurate detection difficult. Even with significant architectural modifications, perfect detection of small objects remains an open problem. However, the improvements achieved in this study

demonstrate the effectiveness of our approach and provide a strong foundation for further research. One area highlighted is that improving only the encoders can result in significant performance improvement. A strong image encoder should excel at converting images into rich, informative numerical representations. These representations enable higher accuracy on vision tasks and better discrimination between visual concepts, especially similar ones. Finally, while the individual modifications presented in this work may build upon existing techniques, their novel combination offers a significant contribution by streamlining the RT-DETR architecture. The key advantage of our approach lies in its compatibility with pre-trained RT-DETR models, as we refined the features extracted by the hybrid encoder, and this can also be performed for already-trained models.

Adapting the original RT-DETR model made detection more robust for small objects without sacrificing real-time speed. For automated visual inspection lines, this uplift directly translates into fewer missed micro-defects, earlier detection of process drifts, and measurable reductions in re-labeling. Because the performance boost remains stable on images with occlusion, low contrast, and other noise, integrators can expect greater robustness and less frequent camera or lighting re-calibration. The experiments also reinforce an emerging consensus in the literature that domain-specific tweaks to the neck, loss, and especially the encoder yield larger returns than simply deploying an off-the-shelf detector, encouraging manufacturers to collect pilot imagery, fine-tune backbones, and explore self-supervised pre-training to cut annotation costs. Furthermore, a key challenge in small object detection is extracting sufficient semantic information from deep learning models, as rich semantic content is crucial to learning discriminative features and achieving better detection performance. This is performed by our modifications working as extra preprocessing layers and further enhancing the features from the encoder, which results in improvements in small object detection. The takeaway message is that improving robustness of the encoder would become the cornerstone for improving small object detection. However, while our encoder-focused modifications partially improve feature preservation, they cannot fully overcome fundamental information bottlenecks. And this is seen with the absolute values in terms of improvement seen in the mAP values, which remain modest for almost all datasets.

Despite the improvements achieved in this work, several limitations remain. First, the proposed modifications increase the risk of overfitting, particularly when training on limited or imbalanced industrial datasets. This makes careful regularization and data augmentation strategies essential to stable performance. Second, although our approach improves detection of small objects, performance still needs improvement for extremely small targets that occupy only a few pixels. Third, our method was evaluated on limited defect datasets, so further validation on diverse real-world datasets with different imaging conditions (e.g., varying lighting, motion blur, or sensor noise) is necessary to establish broader generalizability. Addressing these limitations constitutes an important direction for future research. Concretely, future work will focus on leveraging deterministic methods, such as Centered Kernel Alignment (CKA), as additional loss functions to better regularize and align the encoder representations. We also aim to investigate semi-supervised learning approaches to improve the robustness of the encoder, particularly when annotated data are limited.

In conclusion, by improving the ability to detect minor defects, visual inspection systems can help improve product quality, minimize waste, and improve manufacturing efficiency. For this robust object detection, better models are required where our adapted RT-DETR can play a vital role. For further development, future efforts will involve exploring advanced data augmentation strategies, refining the encoder architecture of RT-DETR further, testing alternative loss functions and matching algorithms, evaluating ensemble

techniques, and employing Explainable AI (XAI) approaches to better understand the model's performance and pinpoint opportunities for further enhancement. Additionally, exploring self-supervised pre-training strategies on large-scale datasets could improve the encoder's capability of learning optimal feature representations, especially in scenarios with limited labeled data. The ongoing quest to improve small object detection remains a vital area of study in computer vision, with substantial potential to influence various industrial fields.

Author Contributions: Conceptualization, C.R. and M.M.; methodology, M.M.; formal analysis, M.M.; investigation, M.M.; resources, M.M.; data curation, M.M.; writing—original draft preparation, M.M.; writing—review and editing, C.R. and M.M.; supervision, C.R.; project administration, C.R.; funding acquisition, C.R. All authors have read and agreed to the published version of the manuscript.

Funding: This research study was supported by the BW-INVEST program (Praxissprint; grant number BW8_1395E), funded by the Ministry of Economic Affairs, Labor, and Tourism of Baden-Württemberg. The presented results are based on experiments conducted within the framework of the project "XAI 4 Human in the Loop Machine Learning".

Data Availability Statement: The datasets analyzed in this study can be found as follows: the NEU DET object detection repository at https://universe.roboflow.com/neudet-kroft/neu-det-0gjxj (accessed on 31 March 2025); the BCCD dataset at https://public.roboflow.com/object-detection/bccd (accessed on 31 March 2025); the synthetic object detection repository at https://drive.google.com/drive/folders/144KELwNssuNIb2AESKymg5PNX-pKqiud?usp=sharing (accessed on 31 March 2025); the PCB dataset at https://universe.roboflow.com/manav-madan/pcb-aibaz-gzujv (accessed on 31 March 2025).

Acknowledgments: We acknowledge the assistance of various AI chatbots, including Qwen, ChatGPT, Gemini, etc., in conducting a preliminary literature survey. Their support helped streamline the initial exploration of relevant research.

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

Abbreviations

The following abbreviations are used in this manuscript:

AIFI	Attention-based Intra-Scale Feature Interaction
API	Application Programming Interface
Al-Cast	Aluminum die-cast dataset
AutoML	Automated machine learning
CCFF	Cross-Scale Channel Feature Fusion
DL	Deep learning
IoT	Internet of Things
KPI	Key Performance Indicator
mAP	Mean Average Precision
ML	Machine learning
QA/QC	Quality assessment/quality control
RTDETR	Real-Time DEtection TRansformer Model

References

1. Ullah, Z.; Qi, L.; Pires, E.; Reis, A.; Nunes, R.R. A Systematic Review of Computer Vision Techniques for Quality Control in End-of-Line Visual Inspection of Antenna Parts. *Comput. Mater. Contin.* **2024**, *80*, 2387–2421. [CrossRef]
2. Bhanu Prasad, P.; Radhakrishnan, N.; Bharathi, S.S. Machine vision solutions in automotive industry. In *Soft Computing Techniques in Engineering Applications*; Springer: Cham, Switzerland, 2014; pp. 1–14.
3. Ren, Z.; Fang, F.; Yan, N.; Wu, Y. State of the art in defect detection based on machine vision. *Int. J. Precis. Eng. Manuf.-Green Technol.* **2022**, *9*, 661–691. [CrossRef]
4. Zou, Z.; Chen, K.; Shi, Z.; Guo, Y.; Ye, J. Object detection in 20 years: A survey. *Proc. IEEE* **2023**, *111*, 257–276. [CrossRef]

5. Bommasani, R.; Hudson, D.A.; Adeli, E.; Altman, R.; Arora, S.; von Arx, S.; Bernstein, M.S.; Bohg, J.; Bosselut, A.; Brunskill, E.; et al. On the opportunities and risks of foundation models. *arXiv* **2021**, arXiv:2108.07258. [CrossRef]
6. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft COCO: Common Objects in Context. In *Computer Vision—ECCV 2014*; Springer International Publishing: Cham, Switzerland, 2014; pp. 740–755. [CrossRef]
7. Janesick, J.R.; Elliott, T.; Collins, S.; Blouke, M.M.; Freeman, J. Scientific charge-coupled devices. *Opt. Eng.* **1987**, *26*, 692–714. [CrossRef]
8. Steger, C.; Ulrich, M.; Wiedemann, C. *Machine Vision Algorithms and Applications*; John Wiley & Sons: Hoboken, NJ, USA, 2018.
9. Chauhan, V.; Fernando, H.; Surgenor, B. Effect of illumination techniques on machine vision inspection for automated assembly machines. In Proceedings of the Canadian Society for Mechanical Engineering (CSME) International Congress, Toronto, ON, Canada, 1–4 June 2014; pp. 1–6.
10. Ashebir, D.A.; Hendlmeier, A.; Dunn, M.; Arablouei, R.; Lomov, S.V.; Di Pietro, A.; Nikzad, M. Detecting multi-scale defects in material extrusion additive manufacturing of fiber-reinforced thermoplastic composites: A review of challenges and advanced non-destructive testing techniques. *Polymers* **2024**, *16*, 2986. [CrossRef]
11. Chalich, Y.; Mallick, A.; Gupta, B.; Deen, M.J. Development of a low-cost, user-customizable, high-speed camera. *PLoS ONE* **2020**, *15*, e0232788. [CrossRef]
12. Czerwińska, K.; Pacana, A.; Ostasz, G. A Model for Sustainable Quality Control Improvement in the Foundry Industry Using Key Performance Indicators. *Sustainability* **2025**, *17*, 1418. [CrossRef]
13. Huang, S.H.; Pan, Y.C. Automated visual inspection in the semiconductor industry: A survey. *Comput. Ind.* **2015**, *66*, 1–10. [CrossRef]
14. Shao, R.; Zhou, M.; Li, M.; Han, D.; Li, G. TD-Net: Tiny defect detection network for industrial products. *Complex Intell. Syst.* **2024**, *10*, 3943–3954. [CrossRef]
15. Giri, K.J. SO-YOLOv8: A novel deep learning-based approach for small object detection with YOLO beyond COCO. *Expert Syst. Appl.* **2025**, *280*, 127447.
16. Dai, M.; Liu, T.; Lin, Y.; Wang, Z.; Lin, Y.; Yang, C.; Chen, R. GLN-LRF: Global learning network based on large receptive fields for hyperspectral image classification. *Front. Remote. Sens.* **2025**, *6*, 1545983. [CrossRef]
17. Zhao, Y.; Lv, W.; Xu, S.; Wei, J.; Wang, G.; Dang, Q.; Liu, Y.; Chen, J. DETRs Beat YOLOs on Real-time Object Detection. In Proceedings of the 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 16–22 June 2024; pp. 16965–16974. [CrossRef]
18. Wu, D.; Wu, R.; Wang, H.; Cheng, Z.; To, S. Real-time detection of blade surface defects based on the improved RT-DETR. *J. Intell. Manuf.* **2025**. [CrossRef]
19. Kim, B.; Shin, M.; Hwang, S. Design and Development of a Precision Defect Detection System Based on a Line Scan Camera Using Deep Learning. *Appl. Sci.* **2024**, *14*, 12054. [CrossRef]
20. Liang, T.; Bao, H.; Pan, W.; Pan, F. Traffic sign detection via improved sparse R-CNN for autonomous vehicles. *J. Adv. Transp.* **2022**, *2022*, 3825532. [CrossRef]
21. Du, S.; Pan, W.; Li, N.; Dai, S.; Xu, B.; Liu, H.; Xu, C.; Li, X. TSD-YOLO: Small traffic sign detection based on improved YOLO v8. *IET Image Process.* **2024**, *18*, 2884–2898. [CrossRef]
22. Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; Zagoruyko, S. End-to-end object detection with transformers. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; Springer: Berlin/Heidelberg, Germany, 2020; pp. 213–229. [CrossRef]
23. Yu, C.; Chen, X. Railway rutting defects detection based on improved RT-DETR. *J. Real-Time Image Process.* **2024**, *21*, 146. [CrossRef]
24. Li, X.; Cai, M.; Tan, X.; Yin, C.; Chen, W.; Liu, Z.; Wen, J.; Han, Y. An efficient transformer network for detecting multi-scale chicken in complex free-range farming environments via improved RT-DETR. *Comput. Electron. Agric.* **2024**, *224*, 109160. [CrossRef]
25. Chen, H.; Huang, W.; Zhang, T. Optimized RT-DETR for accurate and efficient video object detection via decoupled feature aggregation. *Int. J. Multimed. Inf. Retr.* **2025**, *14*, 5. [CrossRef]
26. Zheng, Z.; Jia, Y. An Improved RT-DETR Algorithm for Small Object Detection in Aerial Images. In Proceedings of the 2024 17th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI), Shanghai, China, 26–28 October 2024; pp. 1–6. [CrossRef]
27. Wang, A.; Xu, Y.; Wang, H.; Wu, Z.; Wei, Z. CDE-DETR: A Real-Time End-To-End High-Resolution Remote Sensing Object Detection Method Based on RT-DETR. In Proceedings of the IGARSS 2024—2024 IEEE International Geoscience and Remote Sensing Symposium, Athens, Greece, 7–12 July 2024; pp. 8090–8094. [CrossRef]
28. Kong, Y.; Shang, X.; Jia, S. Drone-DETR: Efficient Small Object Detection for Remote Sensing Image Using Enhanced RT-DETR Model. *Sensors* **2024**, *24*, 5496. [CrossRef] [PubMed]
29. Wei, X.; Yin, L.; Zhang, L.; Wu, F. DV-DETR: Improved UAV Aerial Small Target Detection Algorithm Based on RT-DETR. *Sensors* **2024**, *24*, 7376. [CrossRef] [PubMed]

30. Han, T.; Hou, S.; Gao, C.; Xu, S.; Pang, J.; Gu, H.; Huang, Y. EF-RT-DETR: A efficient focused real-time DETR model for pavement distress detection. *J. Real-Time Image Process.* **2025**, *22*, 63. [CrossRef]
31. Liu, Y.; He, M.; Hui, B. ESO-DETR: An Improved Real-Time Detection Transformer Model for Enhanced Small Object Detection in UAV Imagery. *Drones* **2025**, *9*, 143. [CrossRef]
32. ZHENG, Y.; HUANG, Z.; Binbin, C.; Chao, W.; ZHANG, Y. Improved Real-Time Smoke Detection Model Based on RT-DETR. *Int. J. Adv. Comput. Sci. Appl.* **2024**, *15*, 403. [CrossRef]
33. Pan, J.; Song, S.; Guan, Y.; Jia, W. Improved Wheat Detection Based on RT-DETR Model. *IAENG Int. J. Comput. Sci.* **2024**, *52*, 705.
34. Tang, S.; Bao, Q.; Ji, Q.; Wang, T.; Wang, N.; Yang, M.; Gu, Y.; Zhao, J.; Qu, Y.; Wang, S. Improvement of RT-DETR model for ground glass pulmonary nodule detection. *PLoS ONE* **2025**, *20*, e0317114. [CrossRef]
35. Wu, M.; Qiu, Y.; Wang, W.; Su, X.; Cao, Y.; Bai, Y. Improved RT-DETR and its application to fruit ripeness detection. *Front. Plant Sci.* **2025**, *16*, 1423682. [CrossRef]
36. Xie, L.; Ren, M. Infrared Image Human Detection Based on Improved RT-DETR Algorithm. In Proceedings of the 2024 4th International Conference on Electronic Information Engineering and Computer (EIECT), Shenzhen, China, 15–17 November 2024; pp. 310–314. [CrossRef]
37. Li, S.; Long, L.; Fan, Q.; Zhu, T. Infrared Image Object Detection of Substation Electrical Equipment Based on Enhanced RT-DETR. In Proceedings of the 2024 4th International Conference on Intelligent Power and Systems (ICIPS), Yichang, China, 6–8 December 2024; pp. 321–329. [CrossRef]
38. Wang, S.; Jiang, H.; Yang, J.; Ma, X.; Chen, J.; Li, Z.; Tang, X. Lightweight tomato ripeness detection algorithm based on the improved RT-DETR. *Front. Plant Sci.* **2024**, *15*, 1415297. [CrossRef]
39. Xu, A.; Li, Y.; Xie, H.; Yang, R.; Li, J.; Wang, J. Optimization and Validation of Wafer Surface Defect Detection Algorithm Based on RT-DETR. *IEEE Access* **2025**, *13*, 39727–39737. [CrossRef]
40. Liu, Y.; Cao, Y.; Sun, Y. Research on Rail Defect Recognition Method Based on Improved RT-DETR Model. In Proceedings of the 2024 5th International Conference on Computer Engineering and Application (ICCEA), Hangzhou, China, 12–14 April 2024; pp. 1464–1468. [CrossRef]
41. Liu, C.; Zhang, Y.; Shen, J.; Liu, F. Improved RT-DETR for Infrared Ship Detection Based on Multi-Attention and Feature Fusion. *J. Mar. Sci. Eng.* **2024**, *12*, 2130. [CrossRef]
42. Du, X.; Zhang, X.; Tan, P. RT-DETR based Lightweight Design and Optimization of Thermal Infrared Object Detection for Resource-Constrained Environments. In Proceedings of the 2024 43rd Chinese Control Conference (CCC), Kunming, China, 28–31 July 2024; pp. 7917–7922. [CrossRef]
43. Zhang, H.; Ma, Z.; Li, X. RS-DETR: An Improved Remote Sensing Object Detection Model Based on RT-DETR. *Appl. Sci.* **2024**, *14*, 10331. [CrossRef]
44. Yan, P.; Wen, Z.; Wu, Z.; Li, G.; Zhao, Y.; Wang, J.; Wang, W. Intelligent detection of coal gangue in mining Operations using multispectral imaging and enhanced RT-DETR algorithm for efficient sorting. *Microchem. J.* **2024**, *207*, 111789. [CrossRef]
45. Ding, H.; Zhou, C.; Lian, C. An Improved RT-DETR Model for UAV-Based Power Line Inspection Under Various Weather Conditions. In Proceedings of the 2024 China Automation Congress (CAC), Qingdao, China, 1–3 November 2024; pp. 941–945.
46. Zhao, Z.; Chen, S.; Ge, Y.; Yang, P.; Wang, Y.; Song, Y. RT-DETR-Tomato: Tomato Target Detection Algorithm Based on Improved RT-DETR for Agricultural Safety Production. *Appl. Sci.* **2024**, *14*, 6287. [CrossRef]
47. Wang, S.; Xia, C.; Lv, F.; Shi, Y. RT-DETRv3: Real-Time End-to-End Object Detection with Hierarchical Dense Positive Supervision. In Proceedings of the 2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), Tucson, AZ, USA, 26 February–6 March 2025; pp. 1628–1636. [CrossRef]
48. Li, W.; Li, A.; Li, Z.; Kong, X.; Zhang, Y. RTS-DETR: Efficient Real-Time DETR for Small Object Detection. In Proceedings of the 2024 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Kuching, Malaysia, 6–10 October 2024; pp. 1211–1216. [CrossRef]
49. Huang, J.; Li, T. Small Object Detection by DETR via Information Augmentation and Adaptive Feature Fusion. In Proceedings of the 2024 ACM ICMR Workshop on Multimodal Video Retrieval (ICMR '24), Phuket, Thailand, 10–14 June 2024; pp. 39–44. [CrossRef]
50. Liang, N.; Liu, W. Small Target Detection Algorithm for Traffic Signs Based on Improved RT-DETR. *Eng. Lett.* **2025**, *33*, 140.
51. Mao, H.; Gong, Y. Steel surface defect detection based on the lightweight improved RT-DETR algorithm. *J. Real-Time Image Process.* **2025**, *22*, 28. [CrossRef]
52. An, G.; Huang, Q.; Xiong, G.; Zhang, Y. VLP Based Open-set Object Detection with Improved RT-DETR. In Proceedings of the 2024 4th International Conference on Artificial Intelligence, Big Data and Algorithms (CAIBDA 2024), Zhengzhou, China, 21–23 June 2024; pp. 101–106. [CrossRef]
53. Robinson, I.; Robicheaux, P.; Popov, M. RF-DETR. SOTA Real-Time Object Detection Model. 2025. Available online: https://github.com/roboflow/rf-detr (accessed on 1 September 2025).

54. Parlak, I.E.; Emel, E. Deep learning-based detection of aluminum casting defects and their types. *Eng. Appl. Artif. Intell.* **2023**, *118*, 105636. [CrossRef]
55. Unakafov, A.; Unakafova, V.; Bouse, D.; Parthasarathy, R.; Schmitt, A.; Madan, M.; Reich, C. START: Self-Adapting Tool for Automated Receiver Testing—using receiver-specific stress symbol sequences for above-compliance testing. In Proceedings of the DesignCon, Santa Clara, CA, USA, 28–30 January 2025; pp. 876–900. Available online: https://www.designcon.com/ (accessed on 24 September 2025).
56. Roboflow. PCB Computer Vision Project. 2025. Available online: https://universe.roboflow.com/manav-madan/pcb-aibaz-gzujv (accessed on 18 June 2025).
57. cosmicad; akshaylambda. BCCD Dataset: Blood Cell Detection Dataset. 2018. Available online: https://public.roboflow.com/object-detection/bccd(accessed on 1 September 2025).
58. Ninja, D. Visualization Tools for BCCD Dataset. 2025. Available online: https://datasetninja.com/bccd (accessed on 16 September 2025).
59. Skalski, P. How to Train RT-DETR on a Custom Dataset with Transformers. 2024. Available online: https://blog.roboflow.com/train-rt-detr-custom-dataset-transformers/ (accessed on 11 April 2025).
60. Aharon, S.; Dupont, L.; oferbaratz; Masad, O.; Yurkova, K.; Fridman, L.; Lkdci; Khvedchenya, E.; Rubin, R.; Bagrov, N.; et al. Super-Gradients. GitHub repository, 2021. Available online: https://zenodo.org/records/7789328 (accessed on 24 September 2025).
61. Jocher, G.; Qiu, J. Ultralytics YOLO11. Available online: https://docs.ultralytics.com/models/yolo11/ (accessed on 24 September 2025).
62. Minderer, M.; Gritsenko, A.; Stone, A.; Neumann, M.; Weissenborn, D.; Dosovitskiy, A.; Mahendran, A.; Arnab, A.; Dehghani, M.; Shen, Z.; et al. Simple Open-Vocabulary Object Detection. In *Computer Vision—ECCV 2022*; Springer Nature:Cham, Switzerland, 2022; pp. 728–755. [CrossRef]