2025 International Conference on Intelligent Computing
July 26-29, Ningbo, China
https://www.ic-icc.cn/2025/index.php

Steel Surface Defect Detection Based on YOLOv9 with Denoising Diffusion Implicit Models

Haozhe Zhang[1] and Yujie Li[1] (✉)

[1] School of Artificial Intelligence, Guilin University of Electronic Technology, China
*Corresponding author:yujieli@guet.edu.cn

Abstract. Due to the complexity of steel processing environments, surface defects inevitably occur during production. Detecting these defects is critical for ensuring product quality and industrial safety. Traditional manual inspection methods suffer from inefficiency and subjectivity, while existing algorithms struggle with feature extraction in complex scenarios. We propose a novel steel surface defect detection model YOLODDIM-DWConv-C3 based on YOLOv9, which enhances feature extraction capabilities while significantly reducing computational complexity.To address the scarcity of original data, we employ the Denoising Diffusion Implicit Model (DDIM) for data augmentation. The proposedYOLO based defect detection model minimizes computational demands, enabling seamless deployment on edge devices for real-time defect monitoring. Experimental results on the NEU-DET dataset demonstrate that YOLO-DDC outperforms existing methods in both detection accuracy and computational efficiency. We have published the complete project at https://github.com/zhzhzsword/YOLO-DDC.

Keywords: Steel surface defect detection, DDIM, YOLO-DDC.

1 Introduction

Steel, as a foundational material, is widely used in construction, automotive, aerospace, and other industries. Its surface quality directly impacts product performance, safety, and longevity[1]. Surface defects such as cracks not only degrade material strength but also pose safety risks. Therefore, efficient and precise defect detection is vital for improving quality control and ensuring industrial safety. Furthermore, the performance of downstream industries (e.g., automotive and rail transport) heavily depends on steel quality. High-quality steel reduces cost overruns caused by material defects while enhancing product reliability and production efficiency. In sectors like automotive and rail transport, surface quality requirements have reached micron-level precision. For instance, cracks on high-speed train axles may induce stress concentration[2], leading to catastrophic failures during operation. Thus, establishing a comprehensive quality management system is a core objective in the digital transformation of steel enterprises.

The complexity of steel production environments and high - quality steel's sensitivity to external conditions cause defects like cracks, inclusions, etc. [3]. These flaws harm

product performance and safety. Traditional manual inspection is inefficient, subjective, and error - prone, especially for minor defects. It raises labor costs, may miss crucial flaws, and close - range inspection endangers workers.

With advancements in Convolutional Neural Networks (CNNs)[4], deep learning-based approaches have gained prominence in object detection tasks. CNNs automatically learn image features through convolutional and pooling operations, capturing local spatial and contextual dependencies. Among single-stage detectors, YOLO offers a simple yet efficient solution, while methods like CenterNet[5], SSD[6], and RetinaNet[7] provide additional innovations. Two-stage detectors, such as Fast R-CNN[8] and Mask R-CNN[9], further enhance accuracy. However, CNNs often fail to model global features and long-range dependencies, leading to homogenization of defects and limited sensitivity to subtle variations.

The Transformer architecture[10] addresses CNN limitations in global feature extraction. Vision Transformer (ViT)[11] has revolutionized computer vision tasks, including object detection and segmentation, through its global modeling capabilities. DETR[12] introduced end-to-end detection by eliminating handcrafted anchor boxes. However, Transformers struggle with local texture and fine-grained structure modeling while incurring high computational costs, hindering real-time deployment on edge devices.

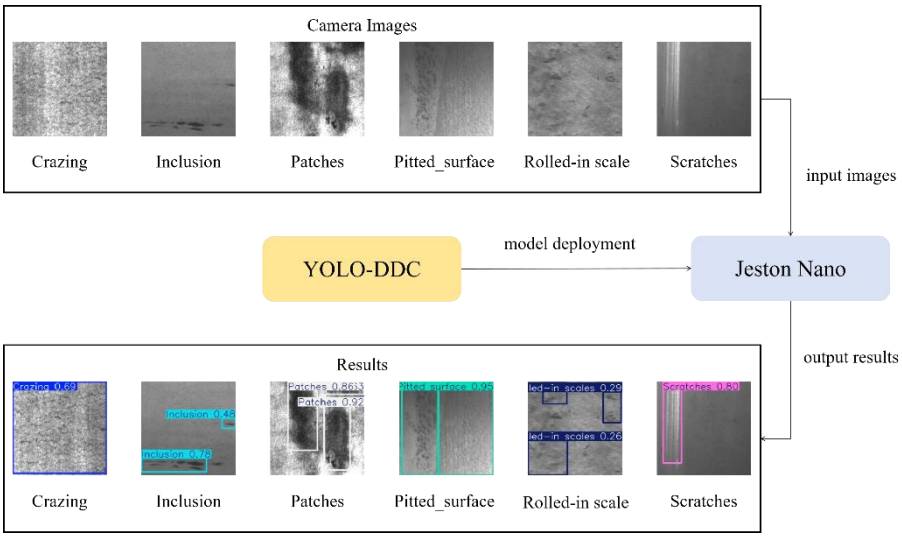

Fig. 1. We deploymet YOLO-DDC on Jeston Nano, use camera to obtain images. Finally output results to real-time data visualization.

To enhance defect detection accuracy and reduce computational overhead, carry out real-time detection of steel surface defects, we apply the above methods to carry out real-time detection (Fig. 1). We utilizes DDIM[13] for data augmentation. Generated samples undergo brightness thresholding and random sampling to form an enriched dataset. Besides, we propose the YOLO-DDC models, built upon YOLOv9s[14], which

integrates Depthwise Separable Convolution (DWConv)[15] for local feature extraction and the C3 module [16] to reduce computational complexity.Evaluations on the NEU-DET dataset[17] confirm the model's superiority. Our contributions are as follows:

1. **YOLO-DDC:** We propose an innovative real-time steel surface defect detection framework. By leveraging the DWConv and C3 modules, this framework enhances the model's detection accuracy while reducing its computational burden, thereby achieving an optimized balance between model performance and computational cost.
2. **DWConv in Backbone:** We replace standard convolutions with DWConv to reduce computations while improving local feature extraction, focusing on critical information and suppressing noise.
3. **C3 Module:** In the YOLO - DDC network, we introduced the C3 module to perform shunt processing, which reduces the computational complexity of the model. As a result, the model can extract semantic features at different levels, facilitating its deployment on edge computing devices for real - time object detection tasks.
4. **Efficient Deployment:** Our design effectively extracts feature information from the NEU-DET dataset, significantly reducing computational load during model training while retaining robust feature capture capabilities. It achieves excellent performance in both mAP@0.5 and GFLOPs metrics and has been deployed on a Jetson Nano for real-time object detection.

2 Related Work

2.1 CNN-based Object Detection

Due to the superior local feature extraction capabilities of convolutional neural networks (CNNs), CNN-based object detection methods have demonstrated significant potential in steel surface defect detection. Kou et al.[18] proposed Faster R-CNN+ with multi-task learning, where a multi-branch network simultaneously outputs defect locations, categories, and severity levels, significantly improving steel detection efficiency. Yang et al.[19] enhanced multi-scale detection by integrating ResNet[20] with attention mechanisms (channel and spatial attention modules) and feature pyramid networks (FPNs), substantially boosting the defect detection performance of the original model.

The YOLO method has provided a more concise and accurate detection framework. Its efficient computational approach and modular adjustability offer valuable improvement opportunities for object detection. Li et al.[21] modified the architecture of YOLOv7[22] to enhance its small object detection performance. YOLOv10[23] achieves non-maximum suppression (NMS)-free training through a dual-label assignment strategy, optimizing the architecture with lightweight classification heads, spatial-channel decoupled downsampling, and sorting-guided block design. This preserves efficient inference while significantly reducing parameters and latency. YOLOv11[24] improves detection accuracy by incorporating a Transformer backbone for long-range

dependency modeling, dynamic head design for adaptive resource allocation, and NMS-free training to simplify inference pipelines.

2.2 Transformer-based Object Detection

Simple CNN-based detection methods struggle to capture global features in defect images. The Transformer architecture offers a novel solution for improved global feature modeling in object detection. DETR[12] first introduced Transformers into object detection, leveraging an encoder-decoder structure to directly predict object categories and locations. This achieves end-to-end defect detection while reducing reliance on manually designed anchor boxes. Chen et al.[25] addressed small-object missing detection by integrating global and local features, thereby enhancing local texture representation.

TinyViT[26] employs semi-dynamic weight pruning and knowledge distillation to reduce model parameters by 90%. RT-DETR[27] designs an Efficient Hybrid Encoder (EHE) to optimize multi-scale feature processing through decoupled intra-scale interactions and cross-scale fusion, improving detection performance. ERF-NAS[28] uses zero-shot neural architecture search based on efficient receptive fields to automate lightweight model construction, also enhancing detection accuracy.

3 Methods

3.1 Overview

The proposed steel surface defect detection involves deploying the model on edge devices for real-time defect detection. To address the challenges of low model accuracy and high computational complexity, we designed the YOLO-DDC model. The network architecture of YOLO-DDC consists of three components: the Backbone, Neck, and Detection Head, as illustrated in Fig. 2.

Prior to training, we employed DDIM for data augmentation. By applying forward diffusion to gradually add noise to images and reverse diffusion to learn the inversion process for restoring original images, we generated synthetic images to enrich the original dataset.

In YOLO-DDC, input images are first processed by the Backbone for feature extraction. Through downsampling and feature extraction, the Backbone outputs feature maps at multiple scales. These features are then fed into the Neck for fusion, which enhances the model's sensitivity to defects of varying sizes and improves its semantic understanding of images. Finally, the Detection Head uses pre-defined anchor boxes to traverse multi-scale feature maps, selecting regions with high confidence scores as potential defect areas.

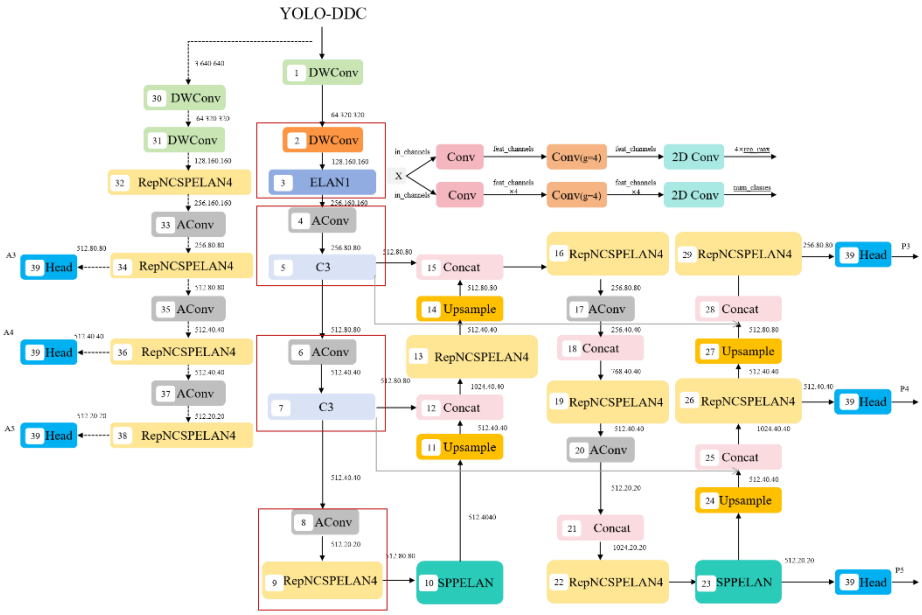

Fig. 2. Overall structure of the network model. This model introduce C3 into Backbone, using DWConv in Backbone to enhance the ability in extracting local spatial feature. the order of modules represented by numbers.

3.2 DDIM Augmentation

To address data scarcity in industrial steel defect detection, we employed DDIM for data augmentation. Steel defects exhibit significant scale variations, ranging from small-scale defects to large-scale ones. Traditional augmentation methods like flipping and additive noise introduce linear transformations that often fail to meet the precision requirements of defect detection. In contrast, DDIM-generated data preserves sharp defect features and enhances dataset quality.

DDIM operates by gradually adding Gaussian noise via forward diffusion and then learning the inversion process during reverse diffusion to denoise images and generate new defect samples. By controlling sampling parameters for original defect images, DDIM avoids introducing extraneous noise, thereby effectively augmenting the NEU-DET dataset.

3.3 Detailed modules of the proposed YOLO-DDC

We proposes YOLOv9-DDC, an efficient steel defect detection framework, comprising Backbone, Neck, and Head components. The Backbone utilizes DWConv with C3 module and ELAN structure for feature extraction, while the Neck employs multi-scale upsampling and cross-stage feature fusion. The Head predicts target categories and

bounding box coordinates. Through phased feature fusion and computational optimization, the framework balances accuracy and speed for industrial edge deployment.

We replaced the standard convolutions in the original Backbone with depthwise separable convolutions (DWConv). This design splits traditional convolution into two components: Depthwise Convolution (processing each input channel individually) and Pointwise Convolution (combining depthwise outputs). DWConv enhances feature extraction while significantly reducing computational complexity and parameter count, improving inference speed for edge device deployment in real-time object detection.

The C3 module splits input features into two streams: one undergoes complex feature extraction through stacked modules, while the other undergoes simple convolution. This architecture avoids redundant computations and reduces computational load. The C3 module enables hierarchical semantic feature extraction, leveraging lightweight design to minimize computational overhead for real-time tasks.

.The Backbone starts with DWConv, splitting standard convolution to reduce FLOPs during initial feature extraction. It then enters the ELAN1 module for efficient layer aggregation, enhancing feature propagation and fusion. AConv combines average pooling and convolution for defect feature capture. The C3 module, based on CSP, splits and fuses features to avoid redundancy and cut computation. Finally, the RepNCSPELAN4 module conducts cross - stage feature fusion to minimize computational redundancy.

Image features pass through the C3 module to cut computational redundancy and boost feature representation. After cross - stage fusion by RepNCSPELAN4, they enter the Neck network. Neck uses upsampling to enlarge high - level feature maps and cross - scale fusion via Concat. By integrating local and global defect features from C3 and RepNCSPELAN4, Neck creates multi - scale features, offering discriminative info to the Detection Head. This enhances detection accuracy and adaptability for different - sized objects.

The final stage is the Detection Head, which receives multi-scale feature maps from the Neck. Through internal convolutional operations (e.g., standard convolution, grouped convolution Conv ($g = 4$), Conv2D), the Head refines features to predict object categories, probabilities, and bounding box coordinates. The multi-scale adaptive design ensures effective detection of objects of varying sizes, outputting end-to-end results that include class labels, confidence scores, and precise locations.

4 Experiments

In this section, we evaluate the performance of YOLO-DDC for steel surface defect detection. We first introduce the benchmark dataset, implementation details, and evaluation metrics. We then compare our method with classical and state-of-the-art approaches and conduct ablation studies to validate the architectural innovations.

4.1 Dataset

Experiments were conducted using the NEU-DET dataset from Northeastern University. This dataset aims to address data scarcity in steel surface defect detection during production and advance deep learning-based automated inspection. It contains six typical steel defects, including **Crazing**, **Inclusion**, **Patches**, **Pitted Surface**, **Rolled-in Scales**, and **Scratches**, as shown in Fig. 3. The original dataset comprises 1,800 grayscale images (300 per defect, mostly 224×224 pixels). After DDIM (Denoising Diffusion Implicit Model) augmentation, we selected 1,800 synthetic images, resulting in a total of 3,600 grayscale images (600 per defect) to ensure comprehensive model training.

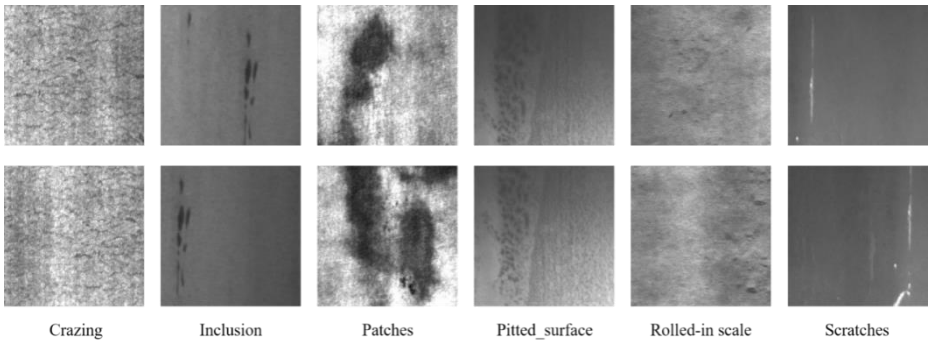

Fig. 3. Selected samples of the generated dataset after processing, from left to right, crazing, pitted surface, patches, rolled-in scales, inclusion, scratches.Two samples per defect category were randomly selected for analysis.

4.2 Implementation Details

YOLO-DDC was implemented using PyTorch and trained on an NVIDIA GeForce RTX 3070 Ti GPU with 8GB memory. The model was randomly initialized and trained from scratch without pre-trained weights. The dataset was processed with an input image size of 224×224 pixels, an initial learning rate of 0.01, a batch size of 32, and 300 training epochs.

4.3 Evaluation Metrics

To quantitatively analyze detection accuracy and computational complexity, we used mAP@0.5 (Mean Average Precision at IoU=0.5)[29] and GFLOPs (Giga Floating-point Operations per Second)[30]. The mAP@0.5 is calculated as:

$$mAP@0.5 = \frac{1}{N}\sum_{i=1}^{N}\left(\int_0^1 P_i\left(R\right)\mathrm{d}R\right) \tag{1}$$

where $P_i(R)$ denotes the precision-recall curve for the i -th class, integrated under the condition IoU..0.5 .

Similarly, the GFLOPs calculation formula is:

$$GFlops = \frac{Total\ Flops}{ExecutionTime(seconds)} \times 10^{-9} \tag{2}$$

$$Flops_{conv} = C_{in} \times C_{out} \times K^2 \times H \times W \times 2 \tag{3}$$

Here, C_m denotes the number of input channels, C_{out} represents the number of output channels, K is the convolution kernel size, and $H \times W$ indicates the feature map size. These formulas enable quantitative evaluation of model computational efficiency and guide lightweight design.

A higher mAP@0.5 indicates higher prediction accuracy and better detection performance. Conversely, lower GFLOPs signify reduced computational complexity, improving real-time detection feasibility.

4.4 Comparison with Other Detection Models

Table 1. Different method experimental results on the NEU-DET dataset.

Methods	Crazing	Inclusion	Patches	Pitted surface	Rolled-in scales	Scratches	mAP@0.5	GFlops
YOLOv5[31]	0.582	0.845	0.902	0.938	0.687	0.867	0.803	24.1
YOLOv8[32]	0.611	0.849	0.898	0.944	0.722	0.867	0.815	28.7
YOLOv9	0.627	0.864	0.898	0.946	0.696	0.866	0.816	32.2
YOLOv10	0.623	0.842	0.892	0.931	0.695	0.865	0.808	28.8
YOLOv11	0.607	0.849	0.903	0.945	0.679	0.863	0.808	28.4
RT-DETR	0.562	0.864	0.901	0.907	0.68	0.874	0.795	108
DE_RetinaNet[33]	0.558	0.819	0.947	0.892	0.702	0.777	0.783	—
Ours	0.611	**0. 872**	0.916	0.927	0.721	**0. 881**	**0. 823**	28.4

We compared YOLO-DDC against classical and state-of-the-art detection methods. Due to the stochastic nature of YOLO-based models, each experiment was repeated multiple times, and results were averaged. To ensure validity, experiments were conducted on the DDIM-augmented NEU-DET dataset, yielding consistent outcomes.

Table 1 illustrates the superior performance of YOLO-DDC in steel surface defect detection, demonstrating its effectiveness over contemporary networks. On the DDIM-augmented NEU-DET dataset, YOLO-DDC achieved a mAP@0.5 of 82.3%, outperforming all state-of-the-art counterparts. Notably, its computational complexity of 28.4 GFLOPs represents the lowest among models with comparable mAP@0.5 metrics.

Compared with the transformer-based detection model RT-DETR, YOLO-DDC achieved a mAP@0.5 improvement from 79.5% to 82.3%, while reducing GFLOPs from 108 to 28.4. This enhancement can be attributed to the complex Transformer architecture in RT-DETR, which increases computational complexity. YOLO-DDC's streamlined architecture, combined with the addition of DWConv and C3 modules, improves detection accuracy while reducing computational costs. Additionally, compared to YOLOv11, our model improved mAP@0.5 from 80.8% to 82.3% while maintaining the same GFLOPs of 28.4. This performance gain is attributed to the robust feature extraction capabilities of DWConv, which helps the model better capture defect features and enhance detection precision.

4.5 Ablation Studies

Table 2. Experimental results of different methods.

Methods	DDIM	DWConv	C3	Crazing	Inclusion	Patches	Pitted surface	Rolled-in scales	Scratches	Map 0.5	GFlops
no DDIM		√	√	0.435	0.809	0.887	0.82	0.584	0.828	0.727	35.1
no C3	√	√		0.635	0.863	0.909	0.951	0.723	0.873	0.826	31.2
no DWConv	√		√	0.624	0.864	0.897	0.956	0.685	0.87	0.816	36.8
Ours	√	√	√	0.611	**0.872**	**0.916**	0.927	0.721	**0.881**	0.823	**28.4**

This section investigates the contributions of DDIM data augmentation, DWConv, and C3 modules to detection performance, validating YOLO-DDC's advantages. By comparing models with configurations no DDIM, no DWConv, no C3, and the full YOLO-DDC, we quantitatively evaluate the impact of each component using mAP@0.5 and GFLOPs on the NEU-DET dataset.

As shown in Table 2, after DDIM augmentation, the mAP@0.5 increased from 72.7% to 82.3%, while GFLOPs decreased from 35.1 to 28.4. This improvement is

attributed to DDIM's generation of high-quality synthetic samples via noise inversion, significantly enhancing defect feature clarity.

On the DDIM-augmented dataset, removing the C3 module slightly reduced mAP@0.5 to 82.6% but maintained GFLOPs at 31.2, demonstrating that the C3 module minimizes computational load with minimal accuracy loss. Conversely, removing DWConv increased GFLOPs to 36.8, validating the critical role of depthwise separable convolutions in reducing parameter count and computational redundancy.

The full YOLO-DDC model, combining DDIM augmentation, DWConv, and C3 modules, achieved the lowest GFLOPs (28.4) while retaining 82.3% mAP@0.5, proving the complementary effects of its components. Ablation results confirm that DDIM improves generalization for small-sample defects, while DWConv and C3 modules optimize lightweight design.

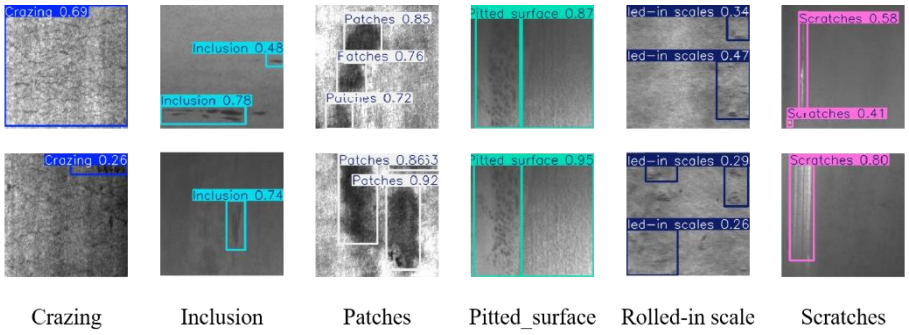

Fig. 4. Inspection results on Jetson Nano, from left to right, crazing, pitted surface, patches, rolled-in scales, inclusion, scratches.

5 Conclusion

This paper proposes YOLO-DDC, a new object detection network. It uses DWConv modules in the Backbone to better capture features of complex steel surface defects. With C3 modules, the model boosts complex feature extraction and cuts computational complexity. Additionally, DDIM is applied as a data augmentation technique to expand the NEU-DET dataset. Experiments on the DDIM-augmented dataset validate YOLO-DDC's superior performance in steel surface defect detection. Future work will optimize the network architecture to boost accuracy for similar defect features. We also plan to develop more accurate and efficient detection models, enhancing production efficiency and operational safety in related industries.

References

1. ASTM International. (2020). *Standard Practice for Evaluating Atmospheric Corrosion Resistance of Metals (G50-20)*. ASTM International.

2. Li, X., & Zhang, Y. (2023). Surface integrity control in high-speed milling of advanced steels. *Journal of Manufacturing Processes*, 89, 456–468.
3. He, Y., Song, K., Meng, Q., & Yan, Y. (2020). An end-to-end steel surface defect detection approach via fusing multiple hierarchical features. *IEEE Transactions on Instrumentation and Measurement*, 69(4), 1493–1504.
4. Salehi, A. W., Khan, S., Gupta, G., Alabduallah, B. I., Almjally, A., Alsolai, H., & Mellit, A. (2023). A study of CNN and transfer learning in medical imaging: Advantages, challenges, future scope. *Sustainability*, 15(7), 5930.
5. Zhou, X., Wang, D., & Krähenbühl, P. (2019). Objects as points. arXiv.
6. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016). SSD: Single shot multibox detector. In *Proceedings of the European Conference on Computer Vision* (pp. 21–37). Springer.
7. Lin, T. Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2017). Focal loss for dense object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 2980–2988). IEEE.
8. Girshick, R. (2015). Fast R-CNN. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 1440–1448). IEEE.
9. He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask R-CNN. arXiv.
10. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. arXiv.
11. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., E., Sutton, C., Duvenaud, D., Jojic, N., & Houlsby, N. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. arXiv.
12. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., & Zagoruyko, S. (2020). End-to-end object detection with transformers. In *Proceedings of the European Conference on Computer Vision* (LNCS, Vol. 12345, pp. 1–17). Springer.
13. Ho, J., Jain, A., & Abbeel, P. (2021). Denoising diffusion implicit models. arXiv.
14. Wang, C. Y., Yeh, I. H., & Liao, H. Y. M. (2024). YOLOv9: Learning what you want to learn using programmable gradient information. arXiv.
15. Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., & Adam, H. (2017). MobileNets: Efficient convolutional neural networks for mobile vision applications. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 1–9). IEEE.
16. Wang, C. Y., Liao, H. Y. M., Wu, Y. H., Chen, P. Y., Hsieh, J. W., & Yeh, I. H. (2020). CSPNet: A new backbone that can enhance learning capability of CNN. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 221–230). IEEE.
17. Zhang, Y., Wang, W., Li, Z., Shu, S., Lang, X., Zhang, T., & Dong, J. (2023). Development of a cross-scale weighted feature fusion network for hot-rolled steel surface defect detection. *Engineering Applications of Artificial Intelligence*, 117, 105628.
18. Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster R-CNN: Towards real-time object detection with region proposal networks. *Advances in Neural Information Processing Systems*, 28, 91–99.
19. Yang, Y., Wang, H., Xin, Z., et al. (2022). An automatic surface defect detection method with residual attention network. In *Artificial Intelligence: Revised Selected Papers of CICAI 2022* (LNCS, Vol. 13516, pp. 456–468). Springer.
20. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 770–778).

21. Li, Y. J., Wang, Y. F., Ma, Z. H., Wang, X. H., & Tang, Y. T. (2024). SOD-UAV: Small object detection for unmanned aerial vehicle images via improved YOLOv7. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*.
22. Wang, C. Y., Bochkovskiy, A., & Liao, H. Y. M. (2023). YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 7464–7475).
23. Wang, A., Chen, H., Liu, L., Chen, K., Lin, Z., Han, J., & Ding, G. (2024). YOLOv10: Real-time end-to-end object detection. arXiv.
24. Khanam, R., & Hussain, M. (240). YOLOv11: An overview of the key architectural enhancements. arXiv.
25. Chen, J. Y., Huang, H. T., & Li, Z. Y. (2024). Feature enhancement and metric optimization for defect detection on steel surface. *Laser & Optoelectronics Progress*, 61(24), 2412002.
26. Wu, K., Zhang, J., Peng, H., Liu, M. C., Xiao, B., Fu, J., & Yuan, L. (2022). TinyViT: Fast pretraining distillation for small vision transformers. In *Proceedings of the European Conference on Computer Vision* (LNCS, Vol. 13666, pp. 57–73). Springer.
27. Zhao, Y., Lv, W., Xu, S., Wei, J., Wang, G., Dang, Q., Liu, H., Wang, F., Jiang, Y., Deng, J., Zhang, Y., Li, Z., Yang, Y., Pan, X., Cheng, Y., Li, K., Zhang, Y., Zheng, S., Luo, P., Qiao, Y., & Loy, C. C. (2023). DETRs beat YOLOs on real-time object detection. arXiv.
28. Otmani, K. E., Mateo-Gabin, A., Rubio, G., & Ferrer, E. (2024). Accelerating high order discontinuous Galerkin solvers through a clustering-based viscous/turbulent-inviscid domain decomposition. arXiv.
29. Everingham, M., Eslami, S. M. A., Gool, L. V., Williams, C. K. I., Winn, J., & Zisserman, A. (2015). The PASCAL visual object classes challenge: A retrospective. *International Journal of Computer Vision*, 111(1), 98–136.
30. Sze, V., Chen, Y. H., Yang, T. J., & Emer, J. S. (2017). Efficient hardware for deep learning: A survey. *Proceedings of the IEEE*, 105(12), 2295–2329.
31. Khanam, R., & Hussain, M. (2024). What is YOLOv5: A deep look into the internal features of the popular object detector. arXiv.
32. Varghese, R., & S, M. (2024). YOLOv8: A novel object detection algorithm with enhanced performance and robustness. In *2024 International Conference on Advances in Data Engineering and Intelligent Computing Systems* (pp. 1–6). IEEE.
33. Cheng, X., & Yu, J. (2020). RetinaNet with difference channel attention and adaptively spatial feature fusion for steel surface defect detection. *IEEE Transactions on Instrumentation and Measurement*, 70, 1–11.