Journal of Electronic Research and Application, 2026, Volume 10, Issue 6
http://ojs.bbwpublisher.com/index.php/JERA
Online ISSN: 2208-3510
Print ISSN: 2208-3502

YOLO11-SPMB: Application of an Improved Instance Segmentation Network for Tire Defect Detection

Haixu You, Jun Xing*

School of Information Science and Engineering, Dalian Polytechnic University, Dalian 116034, China

Author to whom correspondence should be addressed.

Abstract: Tire quality is vital for vehicle safety, but finding surface defects automatically is hard. Current systems often fail to spot tiny flaws hidden inside complex tread patterns. Also, because defects vary in size and blend into the dark rubber, standard models usually output broken or messy segmentation masks. To solve this, we built YOLO11-SPMB, a new instance segmentation network based on YOLO11. We designed it specifically for tire inspection. First, we added a Multi-Scale Adaptive Feature Fusion (MS-AFF) module to the network. It filters out the repeating noise from normal treads so tiny defects can stand out. Second, we created a Boundary Enhancement and Imbalance Calibration (BE-IC) strategy. It uses a boundary Dice loss to keep the mask edges smooth and whole. At the same time, it adjusts class weights to help the model catch rare but major damage, fixing the data imbalance problem. Tests on our custom dataset show the model works very well. YOLO11-SPMB achieves a mAP@0.5:0.95 of 0.702, a mIoU of 0.782, and a recall of 0.928. It also runs at 60 frames per second (FPS), making it fast and accurate enough for real factory production lines.

Keywords: Tire defect detection; Instance segmentation; YOLOv11

Online publication: Jul 23, 2026

1. Introduction

Tires are the vital parts that keep a vehicle on the road, so their build quality is key to safety. In the world of industrial checks, using deep learning to automate this task has become a common practice. Many have already used smart networks to find flaws. For instance, deep learning models helped detect defects in weld surfaces, while YOLO algorithms with extra attention features were used to find flaws in metal [1,2]. At the same time, finding defects in ultrasonic images using neural networks and PCB flaw detection models worked much better than the old manual ways [3,4]. As factories began to demand faster and more precise checks, researchers built decoupled YOLO structures to inspect bridge surfaces, improved Faster R-CNN layouts for 3D printed parts, and updated YOLOv3-Tiny to spot surface flaws more quickly [5–7].

Even with these gains, checking tires remains uniquely difficult due to their material and shape. Since tires are made of black rubber, they absorb most light, and their molded tread patterns create heavy background noise that often hides tiny cracks. Just drawing boxes around defects isn't enough to understand these odd-shaped problems; so, researchers started using pixel-level networks. Better instance segmentation tools were made to get clearer results [8]. To keep track of exact borders under tricky light, methods based on shape contours were built to check copper tubes, while multi-scale Faster R-CNN frameworks were used to better pin down micro-defects on steel [9,10]. These spatial tracking tools were also used in bigger areas, like finding flaws on aircraft using modern recognition techniques [11]. Big reviews of surface defect detection have confirmed that modern automated checks depend heavily on how well a network can pick out features on the fly [12]. With this in mind, smarter networks were born; for example, better YOLOv8 setups made it easier to spot small flaws on PCBs, and the YOLOv8-LSD algorithm cut down the math needed to check steel surfaces on fast lines [13,14].

Recently, the YOLO series has changed quickly to handle the quick pace and messy nature of factory work. YOLOv9 made feature extraction better by using programmable gradient information to stop data loss [15]. Soon after, YOLOv10 removed the delay from non-maximum suppression (NMS) and was used to spot small items in satellite photos, as well as helping drones find people in rescue missions [16,17]. In the newest generation, YOLOv11 has taken another step forward in seeing details. New studies have successfully used improved YOLOv11 models to find subtle defects in electronic items [18]. Also, lightweight versions of YOLOv11 that focus on small parts were made for insulator checks, and highly updated YOLOv11 algorithms were used to pull out anomalies from printed circuit boards [19,20].

Although today's YOLO models work well, they often figure out mask losses using simple binary cross-entropy, which lets the flat background wash out the thin boundary lines. This often leads to messy or broken edges when spotting cracks on dark tires. To solve this, this paper brings forward an improved single-stage instance segmentation framework called YOLO11-SPMB. By rebuilding the neck of the network with a multi-scale adaptive feature fusion module, the network can filter out tread noise on the fly.

We also added a Boundary Enhancement and Imbalance Calibration (BE-IC) step. This puts strict rules on the defect shapes. It stops the masks from splitting into pieces when the contrast is poor. At the same time, it adjusts the class weights to help find rare flaws.

2. Proposed method

2.1. Overall architecture of YOLO11-SPMB

Inspecting tire surfaces is hard. The background patterns are noisy, and the defect edges are difficult to see. We created YOLO11-SPMB to handle this. It is a single-stage instance segmentation model. We started with the standard YOLO11 and changed its structure. These updates boost accuracy but keep the model lightweight. **Figure 1** shows the new layout. It has a backbone for feature extraction, a custom neck, and an updated prediction head. We put the Multi-Scale Adaptive Feature Fusion (MS-AFF) module inside the neck. This module blocks tread noise before mixing the features. We also added a Boundary Enhancement and Imbalance Calibration (BE-IC) step at the head. BE-IC applies geometric rules and updates sample weights. This helps the network draw better shapes for irregular flaws while still running fast.

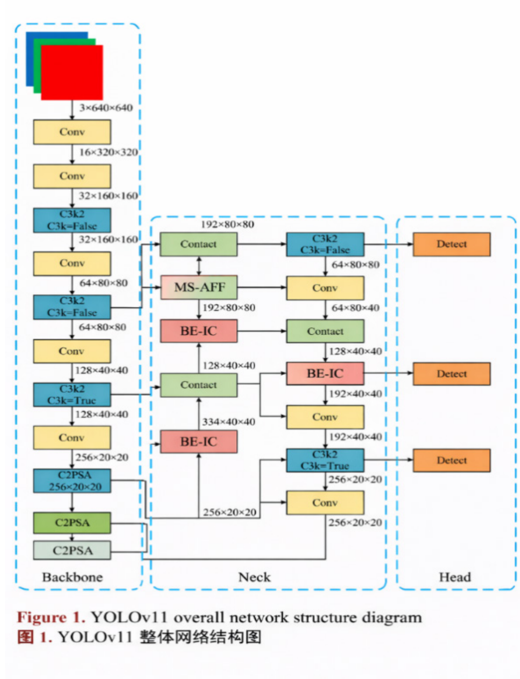

Figure 1. YOLOv11-SPMB network architecture diagram.

2.2. Multi-scale adaptive feature fusion (MS-AFF)

Tire tread patterns easily hide small cracks. Standard models do not filter this background noise. Instead, they pass it directly into deeper layers. We built the MS-AFF module to solve this problem. As **Figure 2** shows, it filters the feature maps using channel and spatial data.

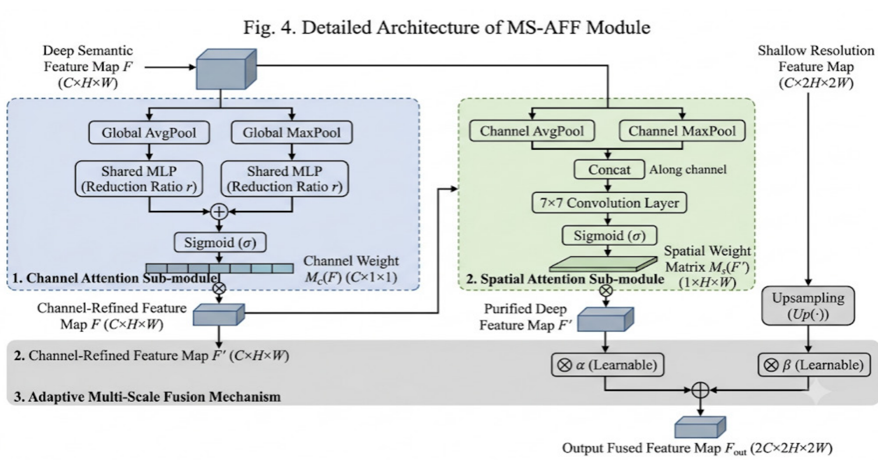

Figure 2. Schematic diagram of MS-AFF module structure.

The backbone extracts a feature map. $F \in R^{C \times H \times W}$. The channel attention module then calculates a 1D weight vector. This vector highlights the channels that actually contain defect features. This is achieved by utilizing both global average pooling and global max pooling, followed by a shared multi-layer perceptron (MLP):

$$M_c(F) = \sigma(MLP(AvgPool(F)) + MLP(MaxPool(F))) \tag{1}$$

where σ denotes the Sigmoid activation function, the channel-refined feature map is obtained by element-wise multiplication:

$$F^{'} = M_c(F) \otimes F \tag{2}$$

Subsequently, to precisely locate spatial anomalies and suppress interference from uniform tread patterns, a spatial attention mechanism is applied. The channel-refined feature map $\mathrm{F}^{'}$ is first aggregated along the channel dimension and then processed by a large-kernel convolution operation to capture broader spatial receptive fields:

The fully purified feature map is thus represented as

$$F^{'} = M_s(F^{'}) \otimes F \tag{3}$$

When merging the purified deep semantic features $F^{'}_{\mathrm{deep}}$ with the high-resolution shallow features $F^{'}_{\mathrm{shallow}}$, MS-AFF employs learnable scalar weights α and β to adaptively balance the fusion proportion:

$$F_{out} = a \times F^{'}_{deep} \oplus \beta \cdot \mathrm{Up}(F_{shallow}) \tag{4}$$

Here, $\oplus$ stands for concatenation, and $\mathrm{Up}(\cdot)$ refers to spatial upsampling. This adaptive method helps the network pay more attention to unusual pattern changes. By doing this, the model can better tell the defects apart from the busy tread background.

2.3. Boundary enhancement and imbalance calibration (BE-IC)

Tire defect detection is not just about background noise; it also struggles with the shapes of the defects. Since both the defects and the tire rubber are black, they blend (as shown in **Figure 3**). This makes it hard for standard loss functions to see the edges, often leading to broken or jagged mask lines. Also, serious damage is rare compared to small surface marks. This creates a data imbalance where the model sees fewer important defects than common ones. Our BE-IC strategy works at the prediction head to fix both of these problems.

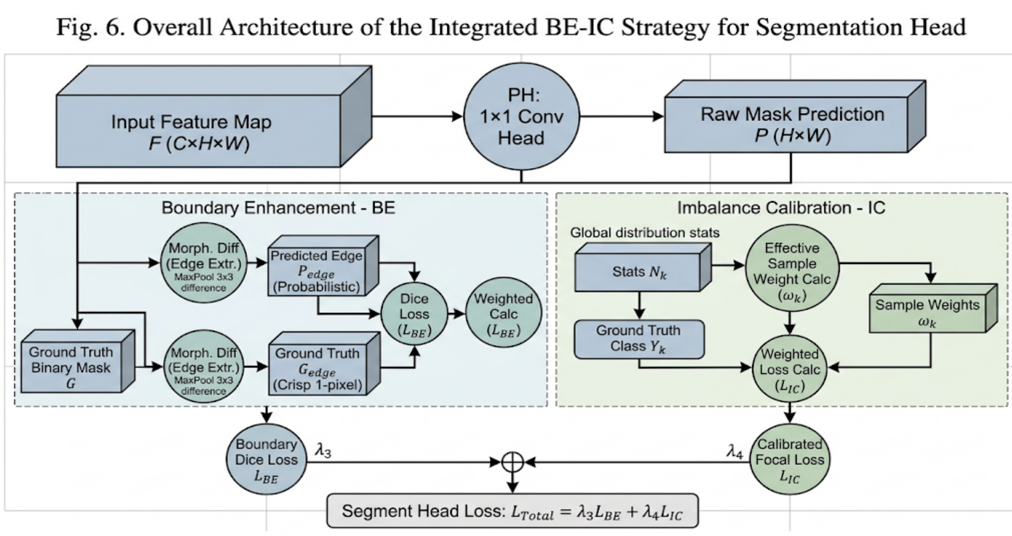

Figure 3. BE-IC module schematic diagram.

To make sure the defect shapes stay whole, the Boundary Enhancement (BE) part uses a specific geometric rule. We get the sharp, single-pixel ground-truth boundary. G_{edge} and the predicted boundary P_{edge} by using a max-pooling difference method:

$$G_{edge}\text{=MaxPool}_{3\times}\ (G)\text{-}G \tag{5}$$

where G denotes the binary ground-truth mask. To penalize boundary misalignment, a boundary-specific Dice loss L_{BE} is formulated as follows:

$$L_{BE}=1-\frac{2\sum_{i,j}\left(P_{edge}^{i,j}\cdot G_{edge}^{i,j}\right)+\epsilon}{\sum_{i,j}\left(P_{edge}^{i,j}\right)^2+\sum_{i,j}\left(G_{edge}^{i,j}\right)^2+\epsilon} \tag{6}$$

where ϵ is a smoothing constant introduced to prevent division by zero. This loss continuously drives steep gradient propagation toward low-contrast contour regions.

Simultaneously, the Imbalance Calibration (IC) strategy mitigates the long-tail bias based on the effective sample volume theory. Let N_k denote the total number of samples in the category k. The effective sample weighting factor ù $_k$ is computed to suppress the dominant gradient influence of the majority classes:

$$\omega_k=\frac{1\text{-}\gamma}{1\text{-}\gamma^{N_k}} \tag{7}$$

where $\gamma\in[0,1)$ controls the attenuation rate. This dynamic weighting factor ω_k is incorporated into a modified Focal Loss framework to construct the calibrated classification objective L_{IC}:

$$L_{IC}=-\sum_{k=1}^{K}\omega_k\cdot y_k\cdot(1\text{-}p_k)^{\hat{o}}\log fo(p_k) \tag{8}$$

where y_k and p_k denote the ground-truth label and the predicted probability, respectively, and ô is the focusing parameter.

2.4. Unified joint loss function

To establish an end-to-end multi-objective optimization framework, the standard YOLO loss is comprehensively reconstructed. The total joint loss function L_{Total} The proposed YOLO11-SPMB is defined as a weighted aggregation of bounding box regression, mask segmentation, boundary enhancement, and calibrated classification:

$$L_{Total}=\lambda_1 L_{CIoU}+\lambda_2 L_{mask\_bce}+\lambda_3 L_{BE}+\lambda_4 L_{IC} \tag{9}$$

where L_{CIoU} denotes the Complete Intersection over Union loss for robust spatial bounding box localization, and L_{mask_bce} represents the standard binary cross-entropy loss for interior mask regions. The hyperparameter weights $(\lambda_1,\lambda_2,\lambda_3,\lambda_4)$ are empirically optimized via grid search to ensure stable gradient convergence across competing spatial and semantic objectives.

3. Experiments

3.1. Dataset and implementation details

To evaluate the YOLO11-SPMB network, experiments were performed on a custom tire defect dataset named TireDefect-Self. The original images were collected using a 3D laser scanner to avoid ambient light interference, and the 3D point cloud data was converted into 2D height grayscale images. The dataset includes three common tire defect types: cracks, scratches, and impurities or bulges. Industrial datasets usually present data imbalance where certain defects are rare. To prevent model overfitting caused by this imbalance, data augmentation methods such as geometric flipping, random noise injection, Mosaic, and MixUp were applied. The final augmented dataset contains 4,000 images. We sorted the dataset by tire IDs to avoid data leaks. This gave us 3,200 images for training, 400 for validation, and 400 for testing.

We ran all tests on an Ubuntu 20.04 machine. It had an Intel Xeon CPU and one RTX 4090 GPU (24GB). We wrote the code in PyTorch 2.0 and used CUDA 11.8. We resized all input images to 640 × 640 pixels. We trained the network for 300 epochs with a batch size of 32. We used the AdamW optimizer. The starting learning rate was 0.001, and we lowered it step by step using cosine annealing. For testing, we set the NMS IoU limit to 0.45 and the confidence limit to 0.25.

3.2. Evaluation metrics

We measure both bounding box detection and mask quality using five metrics: *mAP@0.5, mAP@0.5:0.95, Recall, mIoU*, and *FPS*. Each metric handles a specific task. The *mAP@0.5* score checks if the model finds the general area of a defect using a loose 50% overlap rule. The *mAP@0.5:0.95* score measures box accuracy strictly across multiple overlap levels from 50% to 95%. Recall counts the percentage of real defects the model successfully catches to monitor missed flaws. The *mIoU* metric calculates pixel-level overlap to judge how well the predicted mask shape fits the true defect boundary. The *FPS* score counts the frames processed per second to show the running speed on the factory line.

3.3. Comparison with other methods

We compared YOLO11-SPMB with several top networks. Table 1 lists the baseline models. We chose two-stage models (Mask R-CNN, Cascade Mask R-CNN, and PointRend). We chose standard segmentation models (U-Net and DeepLabV3+). We also chose single-stage models (YOLOv9-CBAM, YOLOv10-DCNv3, and YOLOv11-seg).

Table 1 shows the test results. Two-stage models like Mask R-CNN are too slow. They only run at 9 to 15 FPS because their structure is too large. Factories cannot use them on fast lines. Models like YOLOv10-DCNv3 add attention or deformable convolutions. This improves accuracy but drops the speed to 42 FPS due to the extra math. See **Table 1** and **Figure 4**.

Table 1. Performance comparison table of mainstream instance segmentation networks

Model	*mAP@0.5*	*mAP@0.5:0.95*	*Recall*	*mIoU*	*FPS*
Mask R-CNN	0.902	0.612	0.861	0.728	12
Cascade Mask R-CNN	0.915	0.635	0.874	0.735	9
PointRend	0.912	0.648	0.870	0.758	15
U-Net	0.901	0.648	0.861	0.747	58
DeepLabV3+	0.922	0.632	0.884	0.715	35
YOLOv9-CBAM	0.928	0.655	0.894	0.753	47
YOLOv10-DCNv3	0.933	0.665	0.902	0.760	42
YOLOv11-seg	0.926	0.658	0.895	0.751	60
YOLO11-SPMB	0.954	0.702	0.928	0.782	60

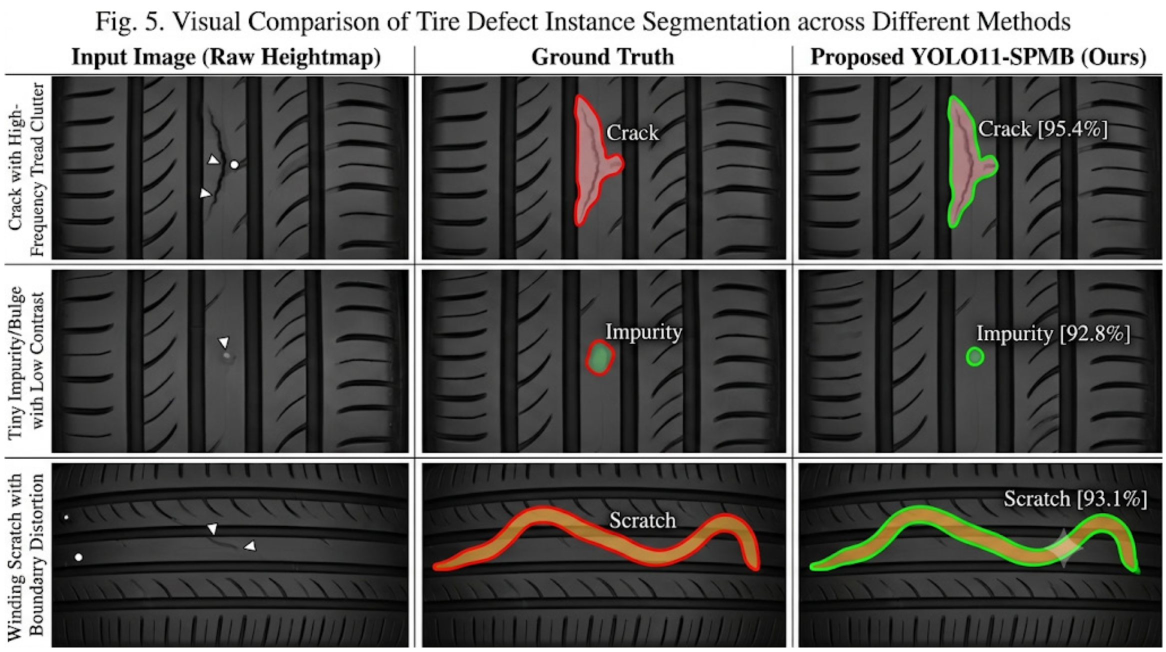

Figure 4. Tire defect images.

YOLO11-SPMB retains both speed and precision. The model reaches an mAP@0.5:0.95 of 0.702. The Recall score increases to 0.928, which means fewer small cracks are missed. The mIoU is 0.782. This score shows that the predicted masks fit the real defect shapes well. The MS-AFF and BE-IC modules add very little computing time. YOLO11-SPMB runs at 60 FPS, matching the baseline YOLOv11-seg exactly. The model balances mask quality and speed. It works well for automated tire checks in factories.

3.4. Ablation study

We ran an ablation study on the MS-AFF, BE, and IC modules. The baseline is the standard YOLOv11-seg network. Table 2 shows the step-by-step results. Adding the MS-AFF module increases the Recall score from 0.895 to 0.912. The module filters tread noise and finds small defects. Adding the BE mechanism uses Dice loss to improve boundary shapes. This increases the mIoU score to 0.770. The IC strategy handles the data imbalance from rare defects. The final mAP@0.5:0.95 score reaches 0.702. See **Table 2**.

Table 2. Ablation study results of the proposed modules

MS-AFF	*BE*	*IC*	*mAP@0.5:0.95*	*Recall*	*mIoU*
			0.658	0.895	0.751
✓			0.681	0.912	0.762
✓	✓		0.685	0.915	0.770
✓	✓	✓	0.702	0.928	0.782

4. Conclusion

This paper presents YOLO11-SPMB, a single-stage instance segmentation network designed to inspect tire surface defects. It handles common factory issues like busy tread patterns, very small flaws, and faint defect edges. We updated the feature fusion neck and the prediction head of the standard YOLOv11 model to get better pixel-level results. Inside the network, the Multi-Scale Adaptive Feature Fusion (MS-AFF) module blocks out the repeating noise from normal tire treads and makes tiny defects easier to see. At the same time, the Boundary Enhancement and Imbalance Calibration (BE-IC) strategy uses a boundary Dice loss to fix messy mask edges and changes class weights to handle uneven data. Tests on our tire dataset show that YOLO11-SPMB balances speed and accuracy well. It gets a mAP@0.5:0.95 of 0.702, a recall of 0.928, and a mIoU of 0.782, while running at 60 FPS. We also tested it on the NEU-Seg dataset, and the results show the model works well on other industrial surfaces too.

Disclosure statement

The authors declare no conflict of interest.

References

[1] Zuo Y, Wang J, Song J, 2021, Application of YOLO Object Detection Network in Weld Surface Defect Detection. 2021 IEEE 11th Annual International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER), 704–710.

[2] Xikun X, Changjiang L, Meng X, 2021, Application of Attention YOLOV4 Algorithm in Metal Defect Detection. 2021 IEEE International Conference on Emergency Science and Information Technology (ICESIT), 465–468.

[3] Mišković M, et al., 2021, Automated Defect Detection From Ultrasonic Images Using Deep Learning. IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control, 68(9): 3126–3134.

[4] Huang W, Wei P, Zhang M, 2021, PCB Defect Detection Using Deep Learning Methods. 2021 IEEE 24th International Conference on Computer Supported Cooperative Work in Design (CSCWD), 1–6.

[5] Sun S, Liu W, Cui R, 2022, YOLO Based Bridge Surface Defect Detection Using Decoupled Prediction. 2022 7th Asia-Pacific Conference on Intelligent Robot Systems (ACIRS), 117–122.

[6] Zhang Y, Zhang Z, Fu K, et al., 2022, Adaptive Defect Detection for 3-D Printed Lattice Structures Based on Improved Faster R-CNN. IEEE Transactions on Instrumentation and Measurement, 71: 1–9.

[7] Li J, et al., 2022, Surface Defect Detection Based on Improved YOLOv3-Tiny Algorithm. 2022 7th International Conference on Image, Vision and Computing (ICIVC), 1–6.

[8] Ren X, He X, 2023, An Enhanced Instance Segmentation Network for Surface Defect Detection. 2023 9th International Conference on Computer and Communications (ICCC), 2098–2103.

[9] Chen D, Xie S, Xie Y, 2023, A Contour-Based Instance Segmentation Method for Copper Tube. 2023 China Automation Congress (CAC), 7864–7869.

[10] Shen M, Cai F, Xie J, 2023, A Multi-Scale Defect Detection for Steel Surface Based on Improved Faster R-CNN. 2023 China Automation Congress (CAC), 7448–7453.

[11] Malik P, Lakshakar H, Hazra J, 2023, Aircraft Defect Localization and Detection Using Object Detection Techniques. 2023 3rd International Conference on Applied Electromagnetics, Signal Processing, & Communication (AESPC), 1–6.

[12] Haobo Y, 2024, A Survey of Industrial Surface Defect Detection Based on Deep Learning. 2024 International Conference on Cyber-Physical Social Intelligence (ICCSI), 1–6.

[13] Raj A, Ahlawat P, 2024, Refined YOLOv8 for Enhanced Small Object Detection in PCB Defect. 2024 IEEE International Conference on Intelligent Signal Processing and Effective Communication Technologies (INSPECT),

1–6.
[14] Yu L, Yan P, 2024, YOLOv8-LSD: A Steel Surface Defect Detection Algorithm Based on YOLOv8. 2024 International Conference on Image Processing, Computer Vision and Machine Learning (ICICML), 1394–1399.
[15] Wang C, Yeh I, Liao H, 2024, YOLOv9: Learning What You Want to Learn Using Programmable Gradient Information. 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 6222–6232.
[16] Sun H, Yao G, Zhu S, 2025, SOD-YOLOv10: Small Object Detection in Remote Sensing Images Based on YOLOv10. IEEE Geoscience and Remote Sensing Letters, 22: 1–5.
[17] Mishra A, Narendra M, Sinha A, et al., 2025, Dynamic Backbone Optimization of YOLOv10 for Real-Time Object Detection in UAV-Based Search and Rescue Missions. IEEE Access, 13: 195975–195986.
[18] Dang Z, Wang X, 2025, FD-YOLO11: A Feature-Enhanced Deep Learning Model for Steel Surface Defect Detection. IEEE Access, 13: 63981–63993.
[19] Shen P, Mei K, Cao H, 2025, LDDFSF-YOLO11: A Lightweight Insulator Defect Detection Method Focusing on Small-Sized Features. IEEE Access, 13: 90273–90292.
[20] Selvam P, Rajasekar R, Gunasundari C, 2025, YOLO-DefXpert: An Advanced Defect Detection on PCB Surfaces Using Improved YOLOv11 Algorithm. IEEE Access, 13: 143085–143101.

Publisher's note

Bio-Byword Scientific Publishing remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.