MAFSRM: A Multiangle Feature Separation and Reconstruction Module for Industrial Defect Detection

Zongshuai Zhang, Ying Gao, Lvwei Zhu, Eric Rigall, Xinjing Wang, and Junyu Dong, *Member, IEEE*

***Abstract*—In industrial manufacturing, the measurability of surface defects directly impacts the accuracy of quality assessment and the effectiveness of process control. While object detection-based methods can rapidly localize defects, they often fall short in supporting high-fidelity visual measurement due to several critical challenges: the strict subsecond timing required by high-speed production lines, the interference from complex backgrounds that distorts geometric and edge information, and the low salience and diversity of small-scale defects, which hampers the extraction of stable, measurement-relevant features. To address these issues, we propose a plug-and-play multiangle feature separation and reconstruction module (MAFSRM), designed to enhance the quality and robustness of defect features from a measurement perspective. MAFSRM consists of three submodules: 1) spatial pyramid pooling-fast separation reconstruction (SPPFSR) for fast, scale-adaptive feature extraction; 2) weight separation reconstruction (WSR) for isolating defect regions from background interference in the spatial domain; 3) feature separation reconstruction (FSR) for enhancing defect distinctiveness by refining channel-level feature representations. The module can be seamlessly integrated into mainstream single-stage detection frameworks, enabling easy deployment without altering the existing pipeline. Experiments on NEU-DET, GC10-DET, PCB, and a self-constructed Tire-DET dataset show that MAFSRM significantly improves both detection accuracy and feature stability, offering highly reliable inputs for downstream measurements of defect size and morphology.**

***Index Terms*—Deep learning, defect measurement, separation reconstruction, spatial feature pyramid, vision-based inspection.**

I. Introduction

IN INDUSTRIAL manufacturing, vision-based measurement systems have become a vital component of nondestructive inspection, enabling the detection of surface defects with high efficiency and automation. Compared with traditional contact-based instruments, these systems offer notable advantages such as real-time processing, noncontact operation, and scalability, making them particularly suitable for large-scale production lines. However, ensuring high-precision measurement performance in such systems remains challenging due to three key factors. First, the strict real-time constraints of industrial workflows require that measurement and analysis be completed without disrupting the production line. Second, as illustrated in Fig. 1, the complex visual background in industrial environments introduces significant noise, making it difficult to isolate the true measurement region. Third, many defects are small, diverse in appearance, and visually subtle, leading to weak signal responses that reduce the accuracy and reliability of feature extraction. These challenges significantly hinder the measurement fidelity and robustness of visual systems in industrial inspection tasks.

Most existing approaches to defect detection rely on general-purpose object detection frameworks, such as YOLO [1] and Faster R-CNN [2]. While these models demonstrate strong performance in standard visual recognition tasks, they are not explicitly designed from the perspective of industrial measurement requirements. Specifically, they lack optimization for robust feature extraction and stable measurement representation, which are critical in high-precision visual inspection systems. In addition, current research largely emphasizes improvements in network depth, parameter efficiency, or inference speed, while paying limited attention to the measurement consistency, repeatability, and signal-to-noise sensitivity that are essential in practical inspection scenarios. Furthermore, these detection models are rarely designed as modular, embeddable measurement enhancement components, limiting their direct integration into real-world industrial instrumentation systems.

Although several feature enhancement strategies have been explored in computer vision—such as attention mechanisms [3], [4], multiscale fusion [5], [6], or feature refinement blocks [7], [8]—they are predominantly optimized for recognition accuracy or computational efficiency in generic tasks. Crucially, they treat feature enhancement as a performance

Received 17 October 2025; revised 31 December 2025; accepted 2 January 2026. Date of publication 30 January 2026; date of current version 17 February 2026. This work was supported in part by the National Natural Science Foundation of China under Grant 62401310, in part by Shandong Province University Youth Innovation Technology Support Program under Grant 2024KJG053, and in part by the opening project of Shandong Province Engineering Research Centre (Qingdao University of Science and Technology) under Grant KF2024SD003. The Associate Editor coordinating the review process was Dr. Maryam Shamgholi. *(Corresponding author: Ying Gao.)*

Zongshuai Zhang, Ying Gao, Lvwei Zhu, and Xinjing Wang are with the School of Data Science, Qingdao University of Science and Technology, Qingdao 266061, China (e-mail: e402318005@mails.qust.edu.cn; gaoying@qust.edu.cn; 4023112073@mails.qust.edu.cn; 2129730221@mails.qust.edu.cn).

Eric Rigall is with the School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou 510006, China (e-mail: rigall@mail.sysu.edu.cn).

Junyu Dong is with the College of Information Science and Engineering, Ocean University of China, Qingdao 266100, China (e-mail: dongjunyu@ouc.edu.cn).

Digital Object Identifier 10.1109/TIM.2026.3659550

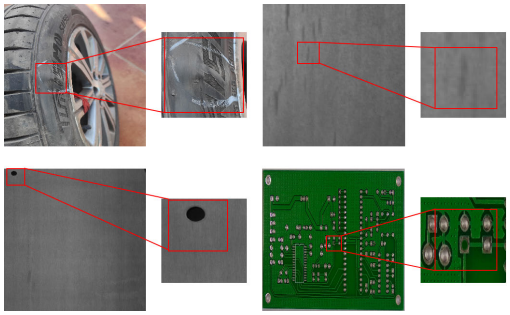

Fig. 1. Defect samples from the four datasets, illustrating the challenges of low visual salience and strong background similarity. From top to bottom and left to right, they come from the tire-defect, NEU-DET, GC10-DET, and PCB datasets, respectively.

booster for classification or localization, not as a measurement-oriented signal processing operation. In contrast, our work is grounded in the principles of industrial metrology: we view defect features as weak signals embedded in noisy backgrounds and adopt a separation–reconstruction paradigm inspired by signal denoising and measurement traceability. This distinction is fundamental rather than merely reweighting or fusing features, and we explicitly decouple and reconstruct defect-relevant representations along three orthogonal measurement dimensions: spatial scale, local geometry, and channel semantics. This ensures not only higher detection accuracy but also improved repeatability, robustness to background interference, and sensitivity to subpixel anomalies—properties rarely prioritized in mainstream detection architectures.

To address the aforementioned challenges, this article proposes a generalized, plug-and-play feature enhancement module, termed the multiangle feature separation and reconstruction module (MAFSRM). Designed for seamless integration with mainstream object detection architectures, MAFSRM enhances the accuracy, stability, and reliability of feature representations used in visual measurement tasks. By decomposing and reconstructing defect-related features from multiple perspectives—scale, spatial locality, and channelwise distribution—the module improves the measurement system's ability to distinguish subtle, low-contrast defects within complex backgrounds. MAFSRM consists of three submodules: spatial pyramid pooling-fast separation reconstruction (SPPFSR), weight separation reconstruction (WSR), and feature separation reconstruction (FSR). The entire module is designed to be pluggable, offering excellent versatility and deployability, and is easy to integrate into existing industrial measurement systems. The contributions of this work are summarized as follows.

1) We design a universal multiangle feature separation and reconstruction module that enhances the extraction of defect-relevant regions in visual measurement systems. The proposed module effectively addresses three core challenges: real-time performance, background interference, and weak defect signal characteristics.
2) The module incorporates three complementary submodules that process feature information from the perspectives of spatial location, channel structure, and scale, respectively. This multiangle strategy not only enables high adaptability and integration flexibility across various detection architectures but also allows the module to be efficiently inserted into standard pipelines without major redesign.
3) Extensive experiments conducted on multiple public industrial defect datasets and a self-constructed dataset demonstrate that MAFSRM significantly improves detection accuracy and measurement consistency, validating its practicality in real-world industrial measurement scenarios.

II. Related Work

A. Defect Detection

Defect detection in industrial production addresses inevitable flaws caused by various factors, such as technology limitations and worker errors due to subjective factors. Historically reliant on manual inspection, defect detection has shifted toward computer-based methods. Traditional automatic inspection approaches use feature-based computer vision algorithms, categorized into texture, color, and shape-based features. For example, Suresh et al. [9] pioneered a statistical classification system for steel plate defects. These traditional methods, while easily interpretable, often struggle with complex defects, due to their reliance on manually designed features. Deep learning has since become dominant in defect detection. Wang et al. [10] introduced domain generalization and noise regularization strategies to effectively detect defects with limited samples. Due to its ability to optimize the feature selection process and its lightweight design, LiFSO-Net [11] stands out for the detection of flat metal defects with complex dimensions. Liu and He [12] proposed a multilevel feature extraction and focused on context module to enhance the capture of defect information. Xiao et al. [13] have developed GRA-Net, a feature interaction attention module, to leverage the relatively few defective pixels. Zhao et al. [14] proposed the ICA-Net, an industrial defect detection network based on convolutional attention guidance and multiscale feature aggregation, to balance speed and accuracy, and EffNet-PCB, proposed by Hou and Zhang [15] for industrial PCB defect detection, also surpasses the popular object detection network. With the gradual adoption of transformers in object detection tasks, ETDNet [16]—based on lightweight transformers—has also achieved efficient defect detection. In addition, Chao et al. [17] introduced the parallel multiscale feature pyramid network (PMFPN) to achieve better feature extraction in complex scenes.

To address semantic ambiguity in industrial images, TSKD [18] distills relational, decoupled, and response knowledge into a lightweight student network. VLCIM [19] fuses visual and textual cues through a cyclic vision–language interaction module. For MFL pipeline inspection, TOPC-Net [20] jointly optimizes a physical model and a deep network for interpretable defect diagnosis.

B. Structural Reparameterization

Structural reparameterization, introduced in RepVGG by Ding et al. [7], optimizes models by transforming their

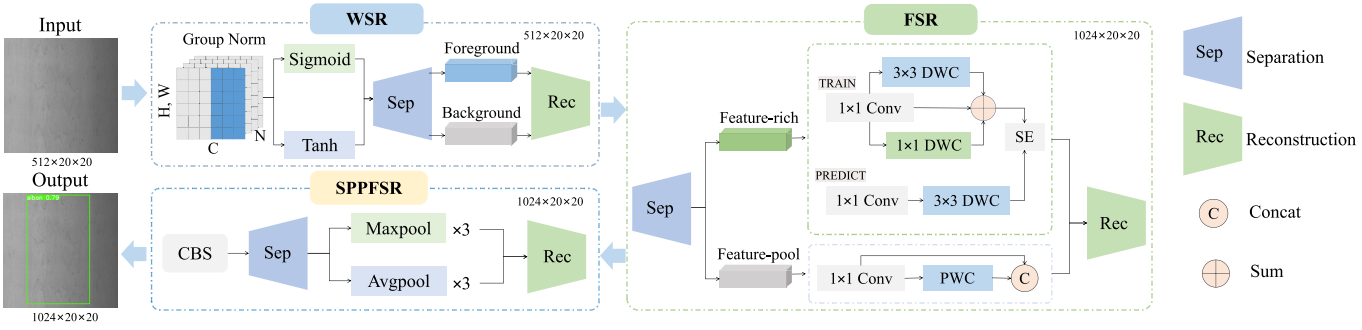

Fig. 2. Overall structure of MAFSRM. Input features are processed sequentially by three modules that operate at different levels: spatial (WSR), channel (FSR), and scale (SPPFSR). WSR (interference removal) → FSR (semantic enhancement) → SPPFSR (scale fusion).

structure to enhance flexibility and representation. This technique involves the use of a more complex network structure during training to improve feature extraction and a simpler structure during inference to speed up processing. Specifically, RepVGG trains with parallel 1×1 and 3×3 convolutional branches and then unifies them into 3×3 branches at inference. Structural reparameterization has also been applied in RepMLP by Ding et al. [21]. This method significantly enhances inference speed without compromising detection accuracy, which is crucial for real-time defect detection tasks.

C. Feature Reconstruction

Feature reconstruction involves enhancing a model's performance or interpretability by modifying how features are represented. This includes techniques such as feature selection, extraction, and transformation. Li et al. [22] introduced spatial and channel reconstruction convolution to address feature redundancy. Their model improves performance by reconstructing features from both the spatial and channel angles while reducing redundant information. Liu et al. [23] proposed an adaptive feature reconstruction algorithm to mitigate performance deterioration in small-sample object detection when the sample size is insufficient for new class detection. Wertheimer et al. [24] tackled the small-sample classification problem by reconstructing features in a latent space and introduced a mechanism for direct regression from support sample features to query sample features. Zhang et al. [25] proposed FFAGR-Net, which uses "grouped feature reconstruction (GFR)" to split aggregated features into multiple sublevel features and autonomously learn the channel-spatial layout of targets, achieving a 7.6% mAP50 improvement on UAV small-target datasets. Qiu et al. [26] proposed HFCR-Net, which performs bidirectional feature reconstruction between support and query sets through a "channel–spatial dual-reconstruction" mechanism, amplifying fine-grained differences under few-shot conditions and providing the latest 2025 paradigm for subtle-defect feature discrimination. These approaches highlight that feature reconstruction can enhance model generalization and improve detection for small objects, which is crucial for effectively extracting defect features and differentiating defects from background regions in defect detection.

III. Method

The overall architecture of MAFSRM is illustrated in Fig. 2. The WSR, FSR, and SPPFSR modules are arranged sequentially to progressively refine feature representations from distinct perspectives. Specifically, WSR first emphasizes spatially salient defect regions by suppressing background interference. Subsequently, FSR enriches channel-level representations by separating and reconstructing features from different network layers, thereby achieving more comprehensive defect feature mining. Finally, SPPFSR extracts and fuses multiscale features to enhance scale-level information.

These three submodules form a complementary and progressive refinement process, strictly following a causal chain of interference removal, semantic enhancement, and scale fusion. This cascading approach effectively prioritizes denoising before enhancing sensitivity and expanding scale, rendering WSR, FSR, and SPPFSR both functionally independent and mutually dependent. For instance, without the preprocessing of WSR, FSR might misidentify background noise as defects; similarly, without the channel recalibration provided by FSR, the multiscale fusion in SPPFSR could amplify redundant textures. Conversely, the fusion output of SPPFSR validates the suppression and enhancement effects of the preceding modules in their respective dimensions, establishing a closed-loop complementarity that significantly improves defect edge preservation and measurement consistency.

A. Weight Separation Reconstruction

In industrial production, defective regions often exhibit inconspicuous characteristics, making them difficult to be distinguished from the background. This nonsignificance contributes considerably to the poor performance of many general-purpose object detection systems. To address this issue, we propose the WSR module. This module amplifies the difference between the defective region and the background region through group normalization (GN) and processes the two separately from the spatial position angle, retaining most of the defective information as the main feature, complemented by a small portion of the background information as a supplement feature. Combining these two types of features in a certain proportion can strengthen the features of the defective region that are not significant, as illustrated in Fig. 3.

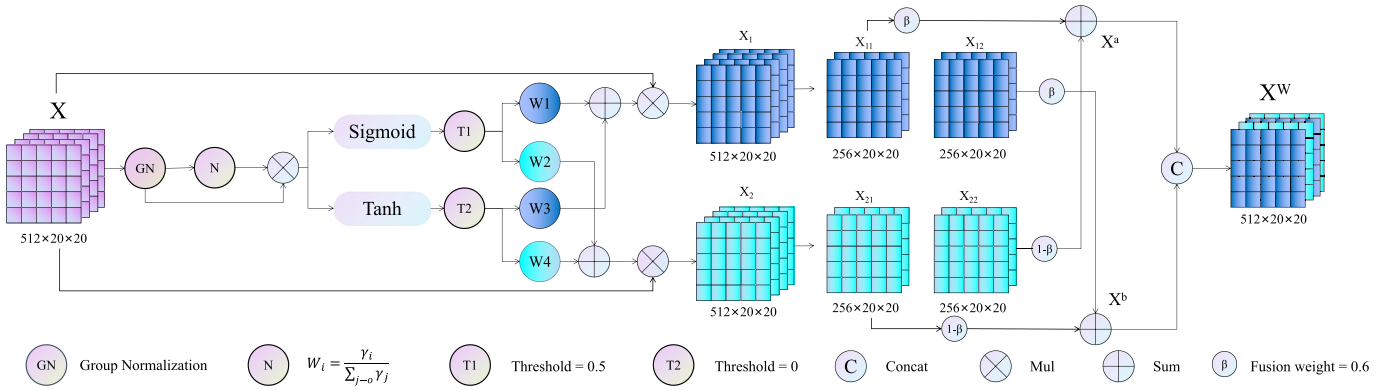

Fig. 3. Structure of the WSR. The input features are first processed through GN to obtain the variable parameter γ. This parameter is used to distinguish between defective regions and background regions based on its value. Following the concept of separation, the reweighted results are mapped to different ranges using the sigmoid and tanh activation functions, respectively, with two distinct thresholds, T_1 and T_2, set to divide the weights accordingly. After that, the results above and below the threshold are summed and multiplied with the initial feature map to obtain the defect feature map and the background feature map. Finally these two parts are reconstructed and concatenated according to a certain ratio to obtain the final feature map.

The separation operation in this study uses parameters from the GN [27] layer. For a given input feature map (X) with batch size (N), number of channels (C), height (H), and width (W), GN(X) is obtained by normalizing (X) through GN, as described by the following equation:

$$X_{\text{out}} = \text{GN}(X) = \gamma \frac{X - \mu}{\sqrt{\sigma^2 + \varepsilon}} + \beta \tag{1}$$

where μ is the mean, σ is the standard deviation, ϵ is a small positive constant, and γ and β are trainable parameters. GN divides the channels into groups and normalizes them by calculating the mean and variance within each group. The richer spatial information reflects more variation in spatial pixels contributing to a larger γ. Defective regions tend to have richer spatial information, which means that the defective region and the background region can be distinguished by γ. The weights of different feature maps can be determined by the following equation:

$$W_\gamma = \{w_i\} = \frac{y_i}{\sum_{j=1}^{C} y_j}, \quad i, j = 1, 2, \ldots, C. \tag{2}$$

The obtained weights based on γ are subjected to a reweighting operation. Based on the concept of separation, the reweighted results are mapped to the range $(0, 1)$ through the sigmoid function and to the range $(-1, 1)$ through the tanh function, respectively, and two different thresholds T_1 and T_2 dividing the weights. Due to the differences in the mapping ranges of the two functions, the threshold T_1 is set to be 0.5, the threshold T_2 is set to 0. These values correspond to the natural decision boundaries of the two activation functions. Using these inherent thresholds eliminates the need for additional parameters and avoids extra hyperparameter tuning, while still achieving stable performance across all tested datasets. W_1 and W_3 are the weights greater than the threshold, and W_2 and W_4 are the weights less than the threshold, and the specific process is shown in the following equation:

$$W_{1,2} = T_1\left(\text{Sigmoid}\left(W_\gamma\left(\text{GN}\left(X\right)\right)\right)\right) \tag{3}$$

$$W_{3,4} = T_2\left(\text{Tanh}\left(W_\gamma\left(\text{GN}\left(X\right)\right)\right)\right). \tag{4}$$

After obtaining the new weights, we add the weights W_1 and W_3 together that were above their corresponding thresholds, and the same applies to the weights W_2 and W_4. These added weights are then multiplied with the input feature X, respectively producing two feature maps: X_1, which contains a wealth of defect-related information, and X_2, which contains background information. At this stage, we have achieved an initial weight separation.

Although X_1 contains a significant amount of defect information, some features may inevitably be missing. Therefore, instead of discarding X_2 entirely, X_2 should be used to supplement X_1. The next step involves performing a reasonable and effective feature reconstruction on these separated features. For this, X_1 and X_2 are further divided into two equal parts along the channel dimension, resulting in four components: X_{11} and X_{12} for X_1, X_{21}, and X_{22} for X_2. A component from one side is combined to one of the other side, according to a specific ratio. Since X_1 is rich in defect information, higher weights are assigned to X_{11} and X_{12} during the summation process, while lower weights are assigned to X_{21} and X_{22}. The two weighted sums are then concatenated to obtain the final reconstructed feature X^w. The specific feature reconstruction operation is shown in the following equation:

$$X^a = \beta X_{11} \oplus 1 - \beta X_{22} \tag{5}$$

$$X^b = \beta X_{12} \oplus 1 - \beta X_{21} \tag{6}$$

$$X^w = \text{concat}\left(X^a, X^b\right). \tag{7}$$

Here, a fusion weight $\beta = 0.6$ (correspondingly $1 - \beta = 0.4$) is adopted as the cross-dataset universal optimal value. As shown in Fig. 4, experiments across four datasets indicate that $\beta \in [0.55, 0.65]$ yields the best performance. Therefore, we use the single value $\beta = 0.6$ as the optimal hyperparameter, which can be directly transferred to all the industrial datasets. This separated and reconstructed feature map enables a more comprehensive extraction of defect features and more clearly distinguishes the defective region from the background. This improvement facilitates the defect detection task.

Compared with mainstream spatial-attention modules such as CBAM [3] or SE [4], WSR does not learn pixelwise gating

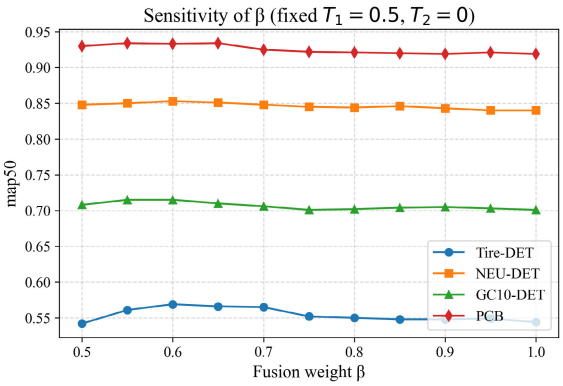

Fig. 4. Sensitivity of fusion weight β in WSR. After fixing T_0 and T_1, experiments on $\beta \in [0.5, 1]$ were conducted on the tire-DET, NEU-DET, GC10-DET, and PCB datasets, and the results showed that the best performance was achieved when $\beta \in [0.55, 0.65]$.

from scratch; instead, it reuses the γ coefficient of GN to split foreground/background, and then reconstructs the two parts with a learned ratio. This yields two advantages: 1) zero extra parameters and latency; and 2) stronger signal-to-noise ratio for low-contrast defects, because the separation is driven by physical statistics (GN) rather than black-box MLPs.

B. Feature Separation Reconstruction

Due to the diversity and complexity of defects in industrial products, conventional feature extraction algorithms often require deeper networks to effectively perform defect detection. However, these complex networks can result in a large number of parameters, leading to high latency during detection tasks. And because defects are often inconspicuous, very deep networks may struggle to accurately identify them. In contrast, shallower networks sometimes perform better on defect detection. Therefore, reducing network complexity and accelerating model inference speed while still effectively extracting defect features is a primary goal. To address this, we have designed the feature separation reconstruction (FSR) module from channelwise information angle, as shown in Fig. 5.

The advantage of this module is to strengthen the accuracy and speed in defect detection. In terms of accuracy, for the defect information with complex features, the deep convolution with multitopology is used for reinforcement training to obtain richer semantic information, which is combined with the richer position information obtained by using the shallow convolution with a single path to obtain more effective defect feature information. In terms of speed, on one hand, dividing part of the feature information and using simple network processing for it reduces the amount of computation; and on the other hand, when predicting, the 1×1 convolutional branch and the constant mapping branch are equated to 3×3 branches, which reduces the number of parameters.

Given an input feature map $Y \in R^{C\times H\times W}$, where C represents the number of channels, H is the height, and W is the width, the channels are divided into two parts: aC channels on one part and $(1-a)C$ channels on the other part, where the default value of a is set to 7/8. It is important to note that a must be greater than 0.5, as the part with aC channels receives more complex processing down the line. This separation approach enables the model to adopt different processing strategies for different types of feature regions, avoiding uniform complex processing for all the channels. As a result, it improves the efficiency of feature extraction and reduces unnecessary computational costs.

For the part with aC channels, this article adopts different network architectures for training and inference, inspired by the VGG network [28]. The rationale behind this is twofold: first, the use of multibranch topology can increase the width and depth of the model during the training process, thus improving the model's expressive ability. The multibranch structure captures more features at different levels, which helps the model learn a richer representation of features, which is especially important for complex tasks such as defect detection. Second, in the inference phase, the use of a simple network structure (single-path convolution) can significantly reduce the amount of computation and improve the running speed of the model. This is important for real-world scenarios that require fast detection, especially in resource-constrained environments. Both the phases use a maximum convolution kernel of 3×3.

A 1×1 convolution is first applied to compress the aC feature channels and improve the computational efficiency, with a default squeeze ratio of 2. Afterward, the process splits into two modes: training and inference. During training, a multibranch topology is used, where a 1×1 depthwise convolution (DWC) is processed in parallel with a 3×3 DWC. The results from these two convolutions are summed with the results of the first 1×1 convolution to obtain a new feature map. For inference, only a single 3×3 DWC, following the first 1×1 convolution, is used to generate the new feature map.

The conversion of training parameters to inference parameters in this context is inspired by the structural reparameterization method proposed by RepVGG [7]. During training, the parameters of the 3×3 DWC remain unchanged. To convert the 1×1 DWC into a 3×3 DWC for inference, a ring of zeros is added around its weights. It means that the 1×1 kernel component is kept, while the eight surrounding elements of a 3×3 spatial kernel are filled with zeroes, and then added to the parameters of the 3×3 DWC from the training phase. This process yields the parameters of the 3×3 DWC used during inference. Note that the reparameterization design of FSR has been repeatedly tested on computationally intensive hardware such as GPUs and CPUs, showing a 10%–30% reduction in latency and a 20%–40% decrease in memory usage. However, on NPUs, TPUs, or highly fragmented mobile devices, due to operator library and bandwidth limitations, the acceleration effect diminishes or even reverses. Therefore, it is not universally applicable and should be tested on the target platform and task before deciding whether to enable it.

After the end of feature extraction, the squeezing-and-excitation (SE) module [4] is applied to learn the nonlinear relationships between channels, enhancing channel dependencies, and finally producing Y_1. The SE module assigns different weights to image locations via a weighting matrix,

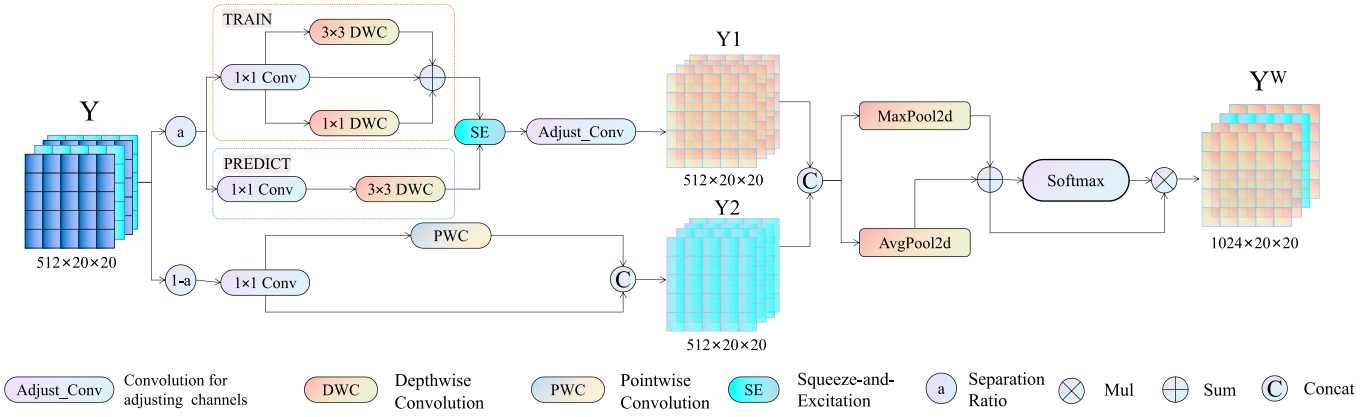

Fig. 5. Structure of feature separation reconstruction (FSR). The input feature map is divided into two parts along the channel dimension. The above part, considered feature-rich, undergoes feature extraction using a multibranch topology, while the bottom part, considered low-information, is processed using a single-path, lightweight convolution. Both of them complement each other, and after concatenating them into a complete feature map, the global maximum pooling and global average pooling operations are performed, respectively, and the feature map is obtained after Softmax.

emphasizing important features, and the specific process is shown in the following equation:

$$\text{out}_{\text{train}} = \text{SE}\left(\text{DWC}_{3\times3}\left(\text{Conv}_{1\times1}(\text{in})\right) + \text{DWC}_{1\times1}\left(\text{Conv}_{1\times1}(\text{in})\right)\right) \quad (8)$$

$$\text{out}_{\text{predict}} = \text{SE}\left(\text{DWC}_{3\times3}\left(\text{Conv}_{1\times1}(\text{in})\right)\right). \quad (9)$$

For the $(1-a)C$ shallow features extracted by single-path convolution branch, first the same as above with a 1×1 convolution to compress the feature channel, thus improving the computational efficiency, and then a pointwise convolution, PWC, is used to replace the parameter-complex convolution operation, with the input residuals connected to the channel dimensions. The concat operation is performed with the input residuals connected in the channel dimension to obtain the shallow feature map Y_2.

The resulting deep features Y_1 and shallow features Y_2 are then combined by concatenating them along the channel dimension to form Y_3. This concatenation uses the high-level semantic information from Y_1 and the low-level spatial details from Y_2, improving the comprehensiveness of complex defect features. Following it, Y_3 undergoes separately two types of global pooling operations: a global maximum pooling and a global average pooling. Their results are combined and processed through the Softmax function, and then multiplied by the Sotfmax function input to produce the final reconstructed feature map Y^w.

Unlike SE [4] or CBAM [3] that simply reweights channels, FSR first splits channels into "feature-rich" and "low-information" groups, applies different topological complexities during training, and collapses them into a single 3×3 branch at inference through structural reparameterization. Thus, FSR enjoys the expressiveness of BiFPN [29] or RepVGG [7] but keeps the parameter count identical to the vanilla block, making it more friendly to resource-limited industrial edge devices.

C. SPP-Fast Separation Reconstruction

The original spatial pyramid pooling (SPP) structure, introduced by He et al. [30] in a parallel form, uses three max-pooling layers of varying sizes to effectively mitigate image distortion caused by operations such as cropping or scaling, offering fast feature extraction and low computational costs. Building on SPP, LLC [31] proposed the faster SPPF in YOLOv5, which transitioned from a parallel to a serial structure, thus reducing computations. The SPPFSR module introduced in this article adopts a hybrid serial–parallel synchronization structure, as shown in Fig. 6.

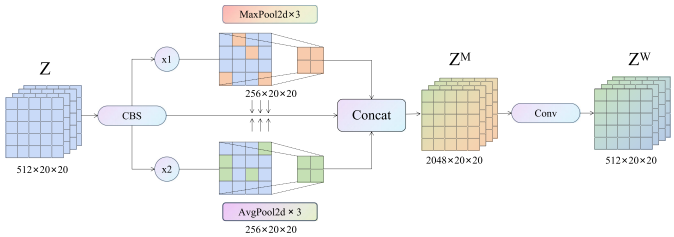

Fig. 6. Structure of SPPFSR. The input features are convolved through a CBS layer and separated to obtain two parts: $x1$ and $x2$, where three consecutive maximum pooling operations are performed on $x1$, while three consecutive average pooling operations are performed on $x2$. Finally, the eight dimensional features are concatenated, and the feature map is reduced to the input size by fusion with a Conv.

Initially, the input image is processed through a CBS convolutional layer to enhance the model's receptive field and feature representation, and CBS means Conv + BN + SiLU. The output is then divided into two parts: x_1 and x_2. x_1 undergoes serial max-pooling, with intermediate feature maps retained for subsequent concatenation. Concurrently, x_2 undergoes serial average pooling, with its intermediate feature maps similarly retained. In the final step, the retained feature maps from x_1 and x_2 are all concatenated together along the channel dimension, resulting in 3×8 dimensional feature splice.

After feature splicing, another conv module upscaling is performed to maintain the scale consistency of the input and output feature maps, thereby obtaining rich semantic information, and the specific process is shown in the following equation:

$$k = 1, 2, 3 \quad (10)$$

$$M^{(k)} = \text{MaxPooling}^{k}(x_1) \tag{11}$$

$$A^{(k)} = \text{AvgPooling}^{k}(x_2) \tag{12}$$

$$O_{\text{final}} = \text{Conv}_{\text{up}}\left(\text{Concat}\left(x_1, x_2, M^{(k)}, A^{(k)}\right)\right). \tag{13}$$

The advantage of SPPFSR module lies in its powerful multiscale feature extraction capability, which is able to extract features from different scales through a combination of serial and parallel pooling operations, capturing rich detail information and overall structure information from scale angle. Meanwhile, the multidimensional feature fusion approach enables the integration of feature information from different scales, which further enhances the feature expression capability and provides a more powerful feature base for subsequent defect detection. In addition, the module retains the fastness of the SPP structure for processing inputs of different scales, has good scale adaptability and robustness, can better handle defect features of different sizes and shapes, and exhibits higher stability and reliability in the face of complex and changing industrial scenes.

Compared with existing pooling modules such as PMFPN [17] and ASPP [32], our proposed SPPFSR focuses more on lightweight enhancement of global context and edge saliency. Unlike cross-layer, heavy-fusion Neck structures such as PMFPN, Our SPPFSR is positioned as a single-stage, lightweight, hardware-friendly receptive field enhancer. It aims to achieve a triple tradeoff of faster processing, lighter weight, and higher accuracy without increasing latency or expanding the number of parameters.

IV. Experimental Results and Analyses

A. Dataset

In this article, we conduct ablation experiments mainly on our Tire-DET dataset, and also select three publicly available defect datasets: NEU-DET, GC10-DET, and PCB to test the comprehensive performance of our model and to validate its generalization performance on defect data.

1) Tire-DET Dataset: The Tire-DET dataset comes from the high-quality tire image dataset provided by Sailun Group, and the dataset is divided into three categories: burst tire, bulge, and normal. There are a total of 3814 RGB images in the dataset, and 3051 images are randomly selected as the training set, 381 images as the validation set, and 382 images as the test set.

2) NEU-DET Dataset: The NEU-DET dataset [33] is a surface defect database published by Northeastern University (NEU) that collects six typical surface defects of hot rolled steel strip, namely, rolled-in scale (RS), patches (Pa), cracks (Cr), pitting surfaces (PS), inclusions (In), and scratches (Sc). The database includes 1800 grayscale images: six different types of typical surface defects, each containing 300 samples.

3) PCB Dataset: The PCB dataset [34] is a publicly available synthetic PCB dataset released by Peking University, containing 1386 RGB images and six types of defects (missing holes, rat bites, open circuits, short circuits, spurs, and stray copper) for inspection, classification, and alignment tasks.

4) GC10-DET Dataset: GC10-DET [35] is a dataset of surface defects collected in a real industry. It contains ten types of surface defects, i.e., Punch (Pu), Weld (Wl), Crescent Gap (Cg), Water Spot. Oil Spot (Os), Silk Spot (Ss), Inclusions (In), Rolled Pits (Rp), Creases (Cr), and Waist Folds (Wf). The collected defects are on the surface of the steel plate. The dataset consists of 3570 grayscale images.

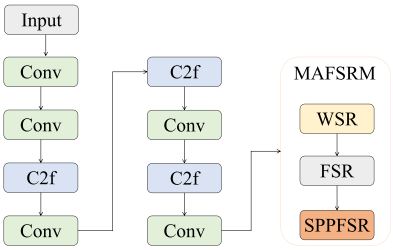

Fig. 7. YOLOv8 backbone model enhanced with MAFSRM. The improvement over YOLOv8 lies in replacing the final C2f and SPPF layers in the original backbone with MAFSRM.

In this article, we use PyTorch to implement our entire model, and train and test it on an NVIDIA RTX 3080 GPU (with 10 GB of memory). In this article, the model takes YOLOv8 improved by MAFSRM as an example (as shown in Fig. 7), and the pretraining weights of YOLOv8n were directly introduced into the training model during the training process, which can shorten the training time of the model and accelerate the speed of convergence. The initial learning rate is set to 0.01, the SGD optimizer's weights decay is set to 0.0005 with a learning rate momentum of 0.937, a batch size of a uniform 8, and all the subsequent experimental rounds are no more than 300 epochs. In terms of data preprocessing, Mosaic data augmentation, adaptive anchor computation, and adaptive image scaling are uniformly applied to all the datasets, and Mosaic [36] data augmentation is turned off in the last ten rounds of training to improve the model's generalization ability.

B. Implementation Details

Evaluation metrics primarily include mean average precision (mAP), while also covering the number of parameters (Param), floating-point performance (GFLOPs), and real-time inference speed (FPS). mAP serves as the core metric for assessing the accuracy of object detection models; Param measures the quantity of model parameters, typically exhibiting a positive correlation with GFLOPs; frames per second (FPS) indicates the number of images a model can process per second, directly reflecting the real-time performance required for industrial production lines.

To validate deployability in real-world factory environments, the MAFSRM-enhanced YOLOv8n model was first exported to the ONNX format and then compiled into a TensorRT 8.6 engine. This optimized engine was subsequently packaged and deployed on an edge PC powered by an Intel i5-1235U CPU and an RTX 3080 10-GB GPU, positioned alongside the conveyor belt in a tire manufacturing plant. In addition, a custom trigger program was developed to interface with an optical sensor, automatically activating the industrial camera when a tire enters the field of view. Once triggered, the

TABLE I
INFLUENCE OF OUR THREE PROPOSED MODULES, ON THE FOUR DATASETS

YOLOv8	SPPFSR	WSR	FSR	Param	GFLOPs	PCB	GC10	NEU	Tire
✓	-	-	-	3.0M	8.204	0.922	0.654	0.800	0.530
✓	✓	-	-	3.0M	8.203	0.935	0.676	0.839	0.543
✓	-	✓	-	3.0M	8.077	0.928	0.656	0.830	0.540
✓	-	-	✓	3.0M	8.082	0.933	0.685	0.844	0.562
✓	✓	✓	✓	3.0M	**8.081**	**0.944**	**0.691**	**0.855**	**0.582**

Fig. 8. MAFSRM detection results under different light conditions. From left to right: original lighting, bright light, low light, and side light. The detection performance remains unchanged under different lighting conditions.

captured 1280 × 720 color image is immediately transmitted to the model for real-time defect detection.

Operating with a batch size of 1, the system achieved an average end-to-end latency of 8.1 ms, corresponding to approximately 123 FPS, during 30 consecutive working days of stable operation. Peak resource consumption was recorded at 2.3 GB of GPU memory and 1.1 GB of CPU RAM. Throughout the 30-day period, no manual restarts or memory leaks were observed. The system inspected approximately 126 000 tires on a two-shift production line operating 16 h per day. In total, the model reported 443 defects. Of these, 392 overlapped with the 397 manually audited ground-truth defects, resulting in a false-negative rate of 1.26% (corresponding to five missed defects). The remaining 51 reports consisted of 38 false alarms and 13 valid defects that had been overlooked by human inspectors. Consequently, the false-positive rate was calculated at 0.03% (38 of 126 000). Note that the 13 additional true positives were treated as supplementary findings and excluded from the denominator when computing the false-positive rate. Both the error rates satisfy the factory's inspection specifications, and the average inspection time of 8.1 ms is well within the strict requirement of less than 10 ms. This large-scale deployment confirms the reliability and practical usability of MAFSRM in high-speed, long-duration industrial environments.

In parallel, we conducted experimental verification of MAFSRM under varying lighting conditions, as illustrated in Fig. 8. Our method consistently outperformed the baseline despite these interferences, demonstrating the robustness of MAFSRM in typical industrial scenarios. Furthermore, to assess measurement precision, we randomly selected 100 defective samples and used the dimensions (length and width) of the annotated bounding boxes as ground truth to calculate the relative error of the predicted boxes. The mean absolute percentage error (MAPE) for the YOLOv8 baseline was 7.8%; this metric decreased to 4.2% with the integration of MAFSRM, successfully meeting the rigorous industrial tolerance requirement of ±5%.

C. Ablation Study

We then conducted ablation experiments to assess the effectiveness of each proposed module. We experimented by adding SPPFSR, WSR, and FSR sequentially to the backbone network of yolov8, and the experimental results on different datasets are shown in Table I.

As can be seen in Table I, the MAFSRM proposed in this article has the highest mAP50 on Tire-DET dataset, as well as on the two publicly available datasets, GC10-DET and NEU-DET, while its number of parameters remains largely unchanged. The addition of each module improves the model's effectiveness to a certain degree, validating their individual contributions to the measurement task. Specifically, the WSR module effectively isolates defect regions from complex backgrounds by leveraging spatial weight separation, which enhances the signal-to-noise ratio. The FSR module refines channel-level representations through multibranch topology, enabling the model to extract richer semantic features from low-contrast defects. Furthermore, the SPPFSR module outperforms the standard SPPF baseline by introducing a dual-pooling strategy, and it captures both the salient defect details through max pooling and global contextual information via average pooling, thereby mitigating background interference. Consequently, the mAP50 reaches its highest value in all the three datasets when SPPFSR, WSR, and FSR are added simultaneously to the model. For the public dataset PCB, although the improvement is not significant, our model also exhibits good performance improvement compared with the original network structure. Regarding the limited improvement of MAFSRM on the PCB dataset, we plotted the corresponding *P–R* curve, as shown in Fig. 9. We carried out two post hoc analyses on the existing statistical results and the characteristics of the PCB dataset itself: first, the PCB dataset consists of synthetic board diagrams with a regular grid background, uniform colors, and high signal-to-noise ratio. The baseline model can already accurately locate using high-contrast edges, so the additional benefits brought by channel recalibration in FSR are compressed by the "ceiling effect", and second, in the dataset, short circuits and spurious copper account for approximately 55% of the samples, while missing holes and mouse bites make up less than 8%. The *P–R* curve is dominated by the high-frequency categories, which "flattens"

TABLE II
INFLUENCE OF DIFFERENT VALUES FOR a, ON THE FOUR DATASETS

a	Params	GFLOPs	mAP50				mAP50-95			
(Ratio)	(M)	(G)	Tire	NEU	GC10	PCB	Tire	NEU	GC10	PCB
1/2	3.0	8.1	0.502	0.790	0.682	0.921	0.190	0.465	0.335	0.493
5/8	3.0	8.1	0.530	0.797	0.684	0.924	0.219	0.490	0.343	**0.499**
3/4	3.0	8.1	0.542	0.834	0.705	0.930	0.200	0.494	0.348	0.491
7/8	3.0	8.1	**0.569**	**0.853**	**0.715**	**0.934**	**0.241**	**0.497**	**0.359**	**0.499**
1	3.0	8.1	0.540	0.839	0.674	0.926	0.191	0.487	0.338	0.492

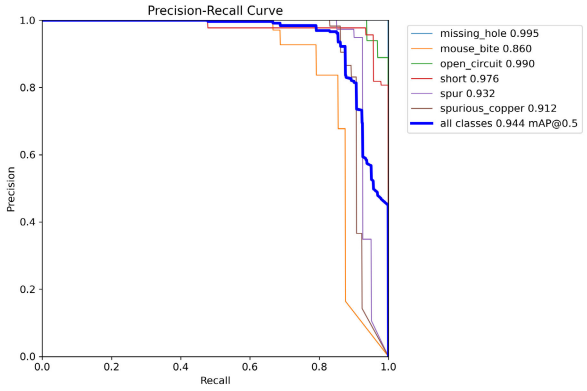

Fig. 9. *P–R* curve of the improved YOLOv8 on the PCB dataset. The model performs well in the missing_hole and open_circuit categories, but its performance is average in the mouse_bite category.

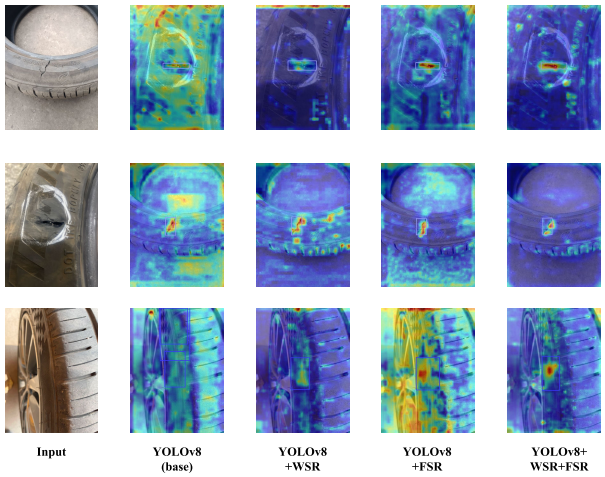

Fig. 10. Heatmap visualization on the tire-DET dataset. From left to right: input image, YOLOv8(base), YOLOv8 + WSR, YOLOv8 + FSR, and YOLOv8 + WSR + FSR visualization results.

the macro mAP50 improvement, and the rare types result in the overall accuracy improvement being not significant. In summary, the limited improvement of PCB is not due to module failure, but rather because the high contrast and low diversity nature of the dataset itself offers little room for optimization by the detector.

It is worth noting that although the absolute improvement on the PCB dataset (2.2%) appears smaller than that on the other datasets, this result should be interpreted in the context of performance saturation and model efficiency. First, the baseline YOLOv8 already achieves a high mAP50 of 92.2% on this synthetic dataset, leaving less than 8% absolute headroom for further optimization. Under such saturation, the 2.2% gain brought by MAFSRM corresponds to a meaningful reduction of the remaining hard cases, for example, subtle defects such as mouse bites, rather than a trivial fluctuation. Second, with respect to model complexity, Table I shows that MAFSRM attains this improvement with negligible additional computational cost, as both the parameter count and GFLOPs remain essentially unchanged at 3.0 M. This observation indicates that the proposed feature separation and reconstruction mechanism refines feature representations in a highly efficient manner, instead of relying on increasing model size to trade for accuracy. Consequently, the PCB results provide complementary evidence: they confirm that MAFSRM preserves robustness and avoids performance degradation on high-SNR, lower complexity tasks, while at the same time delivering substantial gains on more challenging, real-world industrial scenarios such as Tire-DET.

During the experiments, it is found that the results of using the WSR module alone are not very satisfactory, and we believe this is because the WSR module only considers using the spatial aspects of the features for the separation and reconstruction and ignores the factors of the features on the channel side. Since the FSR module is separated and reconstructed at the channel level, it can effectively make up for the shortcomings of WSR in channel feature extraction, while the shortcomings of FSR in the spatial dimensions are compensated by WSR. To further demonstrate the effectiveness of WSR and its synergistic interaction with FSR, we used Grad-CAM for feature visualization, as shown in Fig. 10, to compare the feature activation maps of WSR, FSR, and their cascaded model. The results show that WSR more accurately suppresses background interference and highlights defect edges in the spatial dimension, but exhibits scattered responses in the channel dimension. Conversely, FSR significantly enhances semantic consistency in defect regions through channel-level reconstruction. When cascaded, the activation maps demonstrate superior spatial focus and semantic consistency compared with individual modules, intuitively validating their complementary nature. This visual comparison corroborates that the WSR → FS R sequence alone produces a complementary effect—WSR cleans the spatial input for FSR, and FSR returns channel-focused activations that reinforce WSR's edge suppression—thereby validating the necessity of their cascading order.

TABLE III
FIVEFOLD CROSS-VALIDATION RESULTS (MAP50) AND PAIRED T-TEST SIGNIFICANCE ACROSS FOUR DATASETS

Dataset	YOLOv8 mAP50 (mean ± std)	MAFSRM mAP50 (mean ± std)	Δ	t-test p	Significant ($p < 0.05$)
PCB	0.922 ± 0.002	0.944 ± 0.002	+0.022	0.00039	✓
GC10	0.654 ± 0.003	0.691 ± 0.002	+0.037	0.00022	✓
NEU	0.800 ± 0.002	0.855 ± 0.002	+0.055	<0.0001	✓
Tire	0.530 ± 0.002	0.582 ± 0.002	+0.052	0.00031	✓

In addition, the channel segmentation parameter a in FSR can be adjusted, where the larger the value of a, the larger the defective feature region is sent into the multibranch complex convolutional feature extraction process. Due to the higher complexity of the convolutional layers in this multibranch process, the number of parameters will rise as the value of a increases, and at the same time the performance of the feature extraction will be improved. It is worth noting that the value of a ranges from 0.5 to 1. Therefore, as can be seen in Table II, the default value of a is set to 7/8 to achieve a satisfactory result while trying not to increase the number of parameters or the GFLOPs.

To further verify the effectiveness and stability of the proposed MAFSRM, we conducted a fivefold cross-validation across the four datasets and performed a statistical significance analysis on the resulting mAP50 scores. Specifically, for each fold, we computed the mAP50 for both the baseline YOLOv8 and MAFSRM, applying a paired t-test to examine whether the observed improvements were statistically significant. As summarized in Table III, MAFSRM consistently outperforms the baseline on all the datasets, with average improvements ranging from 0.022 to 0.055. Crucially, all the t-test p-values are far below 0.05, with the majority being less than 0.001, confirming that the performance gains are statistically significant rather than artifacts of random variations or data partitioning.

These results demonstrate that the enhanced performance brought by MAFSRM is stable across different data splits, thereby proving the reliability of the proposed modules. It should also be emphasized that the mAP50 results reported in the subsequent experiments follow the same evaluation protocol and statistical validation procedure, ensuring the consistency and credibility of all the experimental conclusions in this work.

D. Comparison Experiments

To validate the generalization ability and effectiveness of the model proposed in this article for defect data, tests were carried out on the Tire-DET, NEU-DET, GC10-DET, and PCB datasets, and the experimental results were compared with the benchmarking results on the corresponding datasets.

First, several feature extraction modules, attention mechanisms, and novel convolutional modules proposed in recent years have been selected and effectively added to the network of YOLOv8. For example, AKConv is a changeable kernel convolution, EMA module is an efficient multiscale attention mechanism, and BiFPN is a feature pyramid network.

TABLE IV
RESULTS OBTAINED AFTER TRAINING YOLOV8 ON THE TIRE-DET DATASET, THEN MODIFIED BY ADDING DIFFERENT MODULES

	Precision	Recall	mAP50	mAP50-95
YOLOv8	0.679	0.391	0.530	0.236
AKConv [37]	0.612	**0.517**	0.565	0.207
Bifpn [29]	0.573	0.457	0.520	0.216
Repghost [38]	0.698	0.500	0.568	0.227
EMA [39]	0.691	0.443	0.549	0.212
Repvitblock+EMA [39]	0.817	0.448	0.563	0.202
Glod [40]	0.594	0.414	0.463	0.187
Swintransformer [41]	0.793	0.331	0.439	0.185
ODConv [42]	**0.827**	0.425	0.515	0.203
MAFSRM(Ours)	0.811	0.501	**0.582**	**0.246**

Under the premise that the training environment and the hyperparameters remain the same, these modules are trained on the same Tire-DET data separately and then compared with the performance of our MAFSRM proposed in this article. The evaluation metrics selected for comparison are P (precision), R (recall), mAP50, and mAP50-95, whose results are shown in Table IV.

As presented in Table IV, compared with other competing modules, our MAFSRM achieves the highest mAP50 and mAP50-95, alongside superior precision and recall. These results demonstrate that MAFSRM outperforms existing models across all the key metrics, exhibiting exceptional accuracy, generalization capability, and robustness. Specifically, when compared with the original YOLOv8 baseline, the integration of MAFSRM yields substantial improvements: precision increases by approximately 19.4%, recall by 28.1%, mAP50 by 9.8%, and mAP50-95 by 4.2%.

To further illustrate these practical advantages, we provide a visual comparison in Fig. 11. As shown, the baseline model suffers from frequent false positives caused by complex background interference, particularly in the PCB and GC10-DET datasets, as well as missed detections of subtle defects such as microcracks in Tire-DET and faint scratches in NEU-DET. In contrast, when MAFSRM is integrated, these issues are effectively mitigated. The WSR module suppresses background clutter to reduce false alarms, while the FSR and SPPFSR modules enhance feature responses for low-salience defects, leading to significantly improved detection accuracy and localization fidelity.

Moreover, MAFSRM is trained and tested on the NEU-DET and PCB datasets to compare it with various object detec-

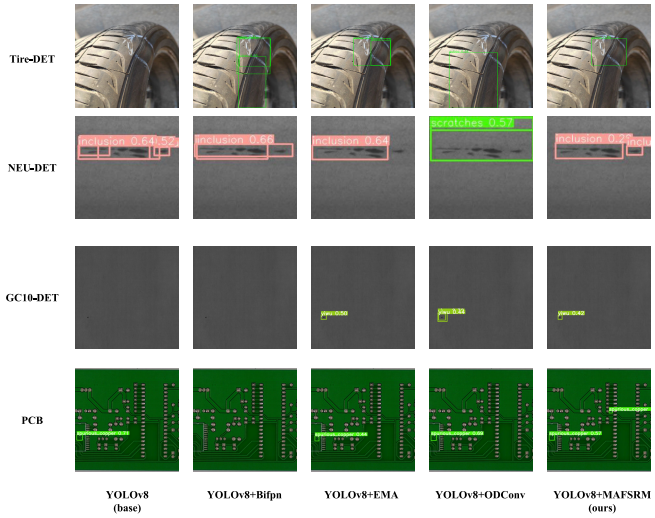

Fig. 11. Performance of different modules on four datasets. The horizontal axis represents module names: YOLOv8 (base), YOLOv8 Bifpn, YOLOv8 EMA, YOLOv8 ODConv, and YOLOv8 MAFSRM (ours); the vertical axis represents dataset names: Tire-DET, NEU-DET, GC10-DET, and PCB.

TABLE V
PERFORMANCE COMPARISON OF DIFFERENT OBJECT DETECTION ARCHITECTURES ON THE NEU-DET DATASET

Methods	Backbone	mAP50	FPS
SSD [43]	VGG16	71.6%	73
SSD-MAFSRM(Ours)	VGG16	75.4%	71
Faster R-CNN [2]	ResNet-34	70.0%	20
Faster R-CNN	ResNet-50	76.6%	18
Faster R-CNN-MAFSRM(Ours) [2]	ResNet-50	78.2%	17
DEA-RetinaNet [44]	ResNet-50	79.1%	13
DEA-RetinaNet-MAFSRM(Ours)	ResNet-50	82.2%	13
CABF-FCOS [45]	ResNet-50	76.7%	18
CABF-FCOS-MAFSRM(Ours)	ResNet-50	81.7%	18
SwinTransformer	Swin-T	78.3%	19
DDN [46]	VGG16	76.3%	15
DDN	ResNet-50	82.1%	11
ES-Net [47]	CSPDarknet53	79.1%	56
ES-Net-MAFSRM(Ours) [47]	CSPDarknet53	81.8%	54
YOLOv5n [31]	CSPDarknet53	74.9%	73
YOLOv5n-MAFSRM(Ours)	CSPDarknet53	82.8%	72
YOLOv8n [1]	CSPDarknet53	80.0%	119
YOLOv8n-MAFSRM(Ours)	CSPDarknet53	**85.5%**	111
YOLO11n [48]	CSPDarknet53	80.5%	**129**
YOLO11n-MAFSRM(Ours)	CSPDarknet53	81.5%	120
RT-DETRv3 [49]	ResNet-34	81.1%	118
YOLOv13n [50]	DS-C3k2	81.8%	116
YOLOv13n-MAFSRM(Ours)	DS-C3k2	85.0%	110

tion methods, where the corresponding results are shown in Tables V and VI.

As can be seen from these two tables, our model also achieves high performance on the public datasets NEU-DET and PCB. Since the purpose of this article is to construct a common defect detection model, the network model is not adapted to different datasets, but as far as the results are concerned, our MAFSRM still has a good performance among the more advanced detection methods at present.

TABLE VI
PERFORMANCE COMPARISON OF DIFFERENT OBJECT DETECTION ARCHITECTURES ON THE PCB DATASET

Methods	Backbone	Param	mAP50	FPS
Faster R-CNN [2]	VGG16	315.4M	56.8%	16
Faster R-CNN	ResNet-101	59.5M	93.1%	6
FPN [5]	ResNet-101	47.5M	92.7%	13
YOLOv4 [36]	CSPDarknet-53	63.9	79.1%	49
Improved YOLOv4 [51]	CSPDarknet-53	244.2M	96.3%	24
ES-Net [47]	CSPDarknet-53	147.9M	**97.5%**	56
YOLOv5m [31]	CSPDarknet-53	42.3M	88.1%	73
YOLOv5m-MAFSRM(Ours)	CSPDarknet-53	42.4M	92.3%	72
YOLOv8n [1]	CSPDarknet-53	3.0M	92.2%	119
YOLOv8n-MAFSRM(Ours)	CSPDarknet-53	3.0M	95.5%	111
YOLO11n [48]	CSPDarknet-53	2.6M	87.4%	**129**
YOLO11n-MAFSRM(Ours)	CSPDarknet-53	2.6M	92.7%	120
RT-DETRv3 [49]	ResNet-34	31.1M	88.3%	118
YOLOv13n [50]	DS-C3k2	**2.5M**	88.7%	116
YOLOv13n-MAFSRM(Ours)	DS-C3k2	**2.5M**	93.5%	110

TABLE VII
INSERTION POSITIONS OF MAFSRM IN DIFFERENT NETWORKS

Original Network	Insertion Position
SSD (VGG16)	After conv4_3 and before pool4
Faster R-CNN (ResNet-50)	After C4 and before RPN
DEA-RetinaNet (ResNet-50)	After C5 and before FPN construction
CABF-FCOS (ResNet-50)	After C4 and before CABF cross-attention
ES-Net (CSPDarknet53)	After backbone and before light neck

Especially on the NEU-DET dataset, our MAFSRM achieves the highest mAP50 of 85.5%, which is much higher than the previous detection algorithms by a significant margin. Moreover, among the compared methods, DDN, DEA-RetinaNet, and CABF-FCOS were specially designed for the NEU-DET dataset, yet our model surpasses their performance results. On the PCB dataset, as shown in Table VI, the mAP50 of the MAFSRM is also in the top rank. To demonstrate MAFSRM's plug-and-play capability, we integrated it into mainstream network architectures such as SSD, Faster R-CNN, DEA-RetinaNet, and CABF-FCOS. The specific integration locations are shown in Table VII. These networks cover one-stage, two-stage, anchor-based, anchor-free, lightweight, and real-time detector families, providing a comprehensive validation of the portability of MAFSRM. As shown in Table V, inserting MAFSRM at the backbone mid-level feature outputs consistently improves the mAP50 of all the architectures. This is because these intermediate layers preserve an effective balance between spatial resolution and semantic abstraction, allowing the WSR, FSR, and SPPFSR modules to strengthen defect-relevant representations while maintaining computational efficiency.

From Table VI, it can be seen that MAFSRM's improved YOLOv8 is much smaller than the other top four object detection networks in terms of number of parameters, and for the best performing ES-Net, its mAP50 is slightly higher

TABLE VIII
NUMBER OF PARAMETERS AND MAP50 OF LIGHTWEIGHT VERSIONS OF OBJECT DETECTION NETWORKS ON THE GC10-DET DATASET

Methods	Param	mAP50	FPS
YOLOv3-tiny [52]	8.67M	61.5%	**188**
YOLOv4-tiny [36]	6.27M	58.1%	176
YOLOv5s [31]	7.03M	68.6%	95
YOLOv7-tiny [53]	6.02M	67.6%	99
SE-YOLOv5 [54]	7.05M	69.1%	85
YOLOv8n [1]	3.01M	65.4%	119
YOLOV8n-MAFSRM(Ours)	3.01M	**69.1%**	111
YOLO11n [48]	2.60M	66.4%	129
YOLO11n-MAFSRM(Ours)	2.60M	67.8%	120
YOLOv13n [50]	**2.50M**	67.5%	116
YOLOv13n-MAFSRM(Ours)	**2.50M**	68.3%	110

than that of MAFSRM's improved YOLOv8, yet its number of parameters is nearly 60 times that of MAFSRM's improved YOLOv8. For defect detection in industrial production, the importance of real-time and accuracy is similarly high. From the FPS, it is evident that under the premise of similar accuracy, the MAFSRM-enhanced YOLOv8 has a higher FPS, which better meets the requirements of real-time performance and accuracy in industrial production.

To further demonstrate the advantage of small network complexity of MAFSRM's improved YOLOv8, on the GC10-DET dataset, we carried out on comparative experiments on the number of parameters and mAP50 of various types of lightweight object detection networks, as shown in Table VIII.

It can be seen that in terms of accuracy, our model has the same highest mAP50 with the SE-YOLOv5 model, while in terms of computational complexity, it has the same lowest number of parameters with YOLOv8n. It can be clearly seen that the MAFSRM's improved YOLOv8 surpasses the compared lightweight networks on both the number of parameters and the accuracy.

In summary, the MAFSRM proposed in this article achieves remarkably high performance on our Tire-DET dataset and the three public defect datasets. By inserting the MAFSRM module into different object detection networks, we have demonstrated the plug-and-play capability of our model. Besides, the qualitative detection results are shown in Fig. 9. It can be seen that the model has good detection effect on defects in various different scenarios.

V. CONCLUSION

In this work, we present MAFSRM, a modular enhancement unit that shifts the focus of industrial defect detection from mere localization to measurement-oriented feature extraction. By performing multiangle feature separation and reconstruction across scale, spatial, and channel dimensions, MAFSRM significantly improves the clarity, stability, and relevance of defect features for downstream measurement tasks. Specifically, the SPPFSR module enhances multiscale feature acquisition and detection speed, the WSR module improves spatial focus by reducing background interference, and the FSR module enriches channelwise representations to better capture subtle or irregular defects. Owing to its modular and plug-and-play architecture, MAFSRM can be easily embedded into existing single-stage detectors, enabling their extension into fully integrated detection-to-measurement systems without altering the production workflow. The experimental results on the NEU-DET, GC10-DET, PCB, and Tire-DET datasets confirm that MAFSRM provides more measurement-reliable feature outputs while maintaining real-time performance, demonstrating its practicality and scalability for industrial visual measurement applications.

Currently, the primary bottlenecks in defect detection stem from two key challenges: extreme small-scale targets and sample scarcity. In industrial scenarios, defects typically exhibit minute dimensions, while the volume of defect data collected from production lines is inherently limited. These factors severely constrain model accuracy and transferability. Consequently, our future work will focus on small-target defect detection and few-shot learning, aiming to develop novel frameworks that balance sensitivity with generalization to provide more robust solutions for industrial visual inspection.

REFERENCES

[1] R. Varghese and M. Sambath, "YOLOv8: A novel object detection algorithm with enhanced performance and robustness," in *Proc. Int. Conf. Adv. Data Eng. Intell. Comput. Syst. (ADICS)*, Apr. 2024, pp. 1–6.

[2] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.

[3] S. Woo, J. Park, J. Lee, and I. S. Kweon, "CBAM: Convolutional block attention module," in *Proc. ECCV*. Cham, Switzerland: Springer, 2018, pp. 3–19.

[4] J. Hu, L. Shen, S. Albanie, G. Sun, and E. Wu, "Squeeze-and-excitation networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 8, pp. 2011–2023, Aug. 2020.

[5] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 936–944.

[6] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8759–8768.

[7] X. Ding, X. Zhang, N. Ma, J. Han, G. Ding, and J. Sun, "RepVGG: Making VGG-style ConvNets great again," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 13728–13737.

[8] K. Han, Y. Wang, Q. Tian, J. Guo, C. Xu, and C. Xu, "GhostNet: More features from cheap operations," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 1577–1586.

[9] B. R. Suresh, R. A. Fundakowski, T. S. Levitt, and J. E. Overland, "A real-time automated visual inspection system for hot steel slabs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vols. PAMI–5, no. 6, pp. 563–572, Nov. 1983.

[10] H. Wang, Z. Li, and H. Wang, "Few-shot steel surface defect detection," *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–12, 2022.

[11] H. Zhong et al., "LiFSO-Net: A lightweight feature screening optimization network for complex-scale flat metal defect detection," *Knowledge-Based Syst.*, vol. 304, Nov. 2024, Art. no. 112520.

[12] T. Liu and Z. He, "TAS2-Net: Triple-attention semantic segmentation network for small surface defect detection," *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–12, 2022.

[13] M. Xiao, B. Yang, S. Wang, F. Mo, Y. He, and Y. Gao, "GRA-Net: Global receptive attention network for surface defect detection," *Knowl.-Based Syst.*, vol. 280, Nov. 2023, Art. no. 111066.

[14] S. Zhao, G. Li, M. Zhou, and M. Li, "ICA-Net: Industrial defect detection network based on convolutional attention guidance and aggregation of multiscale features," *Eng. Appl. Artif. Intell.*, vol. 126, Nov. 2023, Art. no. 107134.

[15] Y. Hou and X. Zhang, "A lightweight and high-accuracy framework for printed circuit board defect detection," *Eng. Appl. Artif. Intell.*, vol. 148, May 2025, Art. no. 110375.

[16] H. Zhou, R. Yang, R. Hu, C. Shu, X. Tang, and X. Li, "ETDNet: Efficient transformer-based detection network for surface defect detection," *IEEE Trans. Instrum. Meas.*, vol. 72, pp. 1–14, 2023.
[17] C. Chao, X. Mu, Z. Guo, Y. Sun, X. Tian, and F. Yong, "IAMF-YOLO: Metal surface defect detection based on improved YOLOv8," *IEEE Trans. Instrum. Meas.*, vol. 74, pp. 1–17, 2025.
[18] Z. Wen, J. Liu, H. Zhao, and Q. Wang, "A triple semantic-aware knowledge distillation network for industrial defect detection," *Comput. Ind.*, vol. 166, Apr. 2025, Art. no. 104252.
[19] X. Shen et al., "VLCIM: A vision-language cyclic interaction model for industrial defect detection," *IEEE Trans. Instrum. Meas.*, vol. 74, pp. 1–13, 2025.
[20] X. Shen et al., "A task-oriented physical collaborative network for pipeline defect diagnosis in a magnetic flux leakage detection system," *Comput. Ind.*, vol. 169, Aug. 2025, Art. no. 104290.
[21] X. Ding, C. Xia, X. Zhang, X. Chu, J. Han, and G. Ding, "RepMLP: Re-parameterizing convolutions into fully-connected layers for image recognition," 2021, *arXiv:2105.01883*.
[22] J. Li, Y. Wen, and L. He, "SCConv: Spatial and channel reconstruction convolution for feature redundancy," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2023, pp. 6153–6162.
[23] D. Liu, X. Liu, D. Yu, B. Yang, and W. Li, "Adaptive feature reconstruction algorithm for few-shot object detection," *J. Shandong Univ., Eng. Sci.*, vol. 52, no. 3, pp. 115–122, Jun. 2022.
[24] D. Wertheimer, L. Tang, and B. Hariharan, "Few-shot classification with feature map reconstruction networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 8008–8017.
[25] Y. Zhang, C. Wu, T. Zhang, and Y. Zheng, "Full-scale feature aggregation and grouping feature reconstruction-based UAV image target detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 62, 2024, Art. no. 5617014.
[26] S. Qiu, W. Yang, and M. Yang, "Hybrid feature collaborative reconstruction network for few-shot fine-grained image classification," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2025, pp. 1–5.
[27] Y. Wu and K. He, "Group normalization," *Int. J. Comput. Vis.*, vol. 128, no. 3, pp. 742–755, Mar. 2020.
[28] K. Simonyan, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Represent.*, Jan. 2015, pp. 1–14.
[29] M. Tan, R. Pang, and Q. V. Le, "EfficientDet: Scalable and efficient object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 10778–10787.
[30] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1904–1916, Sep. 2015.
[31] U. LLC. (2020). *YOLOv5*. [Online]. Available: https://github.com/ultralytics/yolov5
[32] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, Apr. 2018.
[33] K. Song. (2021). *Neu Surface Defect Database*. [Online]. Available: http://faculty.neu.edu.cn/songkechen/zh_CN/zdylm/263270/list/
[34] L. Dai. (2019). *PKU-Market-PCB*. [Online]. Available: https://robotics.pkusz.edu.cn/resources/dataset/
[35] X. Lv, F. Duan, J.-J. Jiang, X. Fu, and L. Gan, "Deep metallic surface defect detection: The new benchmark and detection network," *Sensors*, vol. 20, no. 6, p. 1562, Mar. 2020.
[36] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal speed and accuracy of object detection," 2020, *arXiv:2004.10934*.
[37] X. Zhang et al., "AKConv: Convolutional kernel with arbitrary sampled shapes and arbitrary number of parameters," 2023, *arXiv:2311.11587*.
[38] C. Chen, Z. Guo, H. Zeng, P. Xiong, and J. Dong, "RepGhost: A hardware-efficient ghost module via re-parameterization," 2022, *arXiv:2211.06088*.
[39] D. Ouyang et al., "Efficient multi-scale attention module with cross-spatial learning," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Jun. 2023, pp. 1–5.
[40] C. Wang et al., "Gold-YOLO: Efficient object detector via gather-and-distribute mechanism," 2023, *arXiv:2309.11331*.
[41] J. Huang et al., "Swin transformer for fast MRI," *Neurocomputing*, vol. 493, pp. 281–304, Jul. 2022.
[42] C. Li, A. Zhou, and A. Yao, "Omni-dimensional dynamic convolution," 2022, *arXiv:2209.07947*.
[43] W. Liu et al., "SSD: Single shot multibox detector," in *Computer Vision—ECCV 2016*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds., Cham, Switzerland: Springer, Oct. 2016, pp. 21–37.
[44] X. Cheng and J. Yu, "RetinaNet with difference channel attention and adaptively spatial feature fusion for steel surface defect detection," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–11, 2021.
[45] J. Yu, X. Cheng, and Q. Li, "Surface defect detection of steel strips based on anchor-free network with channel attention and bidirectional feature fusion," *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–10, 2022.
[46] Y. He, K. Song, Q. Meng, and Y. Yan, "An end-to-end steel surface defect detection approach via fusing multiple hierarchical features," *IEEE Trans. Instrum. Meas.*, vol. 69, no. 4, pp. 1493–1504, Apr. 2020.
[47] X. Yu, W. Lyu, D. Zhou, C. Wang, and W. Xu, "ES-Net: Efficient scale-aware network for tiny defect detection," *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–14, 2022.
[48] R. Khanam and M. Hussain, "YOLOv11: An overview of the key architectural enhancements," 2024, *arXiv:2410.17725*.
[49] S. Wang, C. Xia, F. Lv, and Y. Shi, "RT-DETRv3: Real-time end-to-end object detection with hierarchical dense positive supervision," 2024, *arXiv:2409.08475*.
[50] M. Lei et al., "YOLOv13: Real-time object detection with hypergraph-enhanced adaptive visual perception," 2025, *arXiv:2506.17733*.
[51] H. Xin, Z. Chen, and B. Wang, "PCB electronic component defect detection method based on improved YOLOv4 algorithm," *J. Phys., Conf. Ser.*, vol. 1827, no. 1, Mar. 2021, Art. no. 012167.
[52] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*.
[53] Y. Hui, J. Wang, and B. Li, "WSA-YOLO: Weak-supervised and adaptive object detection in the low-light environment for YOLOV7," *IEEE Trans. Instrum. Meas.*, vol. 73, pp. 1–12, 2024.
[54] L. Zheng, X. Wang, Q. Wang, S. Wang, and X. Liu, "A fabric defect detection method based on improved YOLOv5," in *Proc. 7th Int. Conf. Comput. Commun. (ICCC)*, Dec. 2021, pp. 620–624.