PCP-YOLO:An approach integrating non-deep feature enhancement module and polarized self-attention for small object detection of multiscale defects

Penglin Wang[1], Donghui Shi[1,2*], Jose Aguilar[3,4,5]

[1*]School of Electronic and Information, Anhui Jianzhu University, Hefei, 230601, Anhui, China.
[2]Anhui Province Key Laboratory of Intelligent Building and Building Energy Saving, Anhui Jianzhu University, Hefei, 230022, Anhui, China.
[3]Grupo de Investigación en I+D+i en TIC, Universidad EAFIT, Medellín, Colombia.
[4]Centro de Estudios en Microelectrónica y Sistemas Distribuidos, Universidad de Los Andes, Merida, Venezuela.
[5]IMDEA Networks Institute, Anhui Jianzhu University, Madrid, Spain.

*Corresponding author(s). E-mail(s): sdonghui@gmail.com;
Contributing authors: wpl_aqnu@163.com; aguilar@ula.ve;

Abstract

Industrial defect detection is crucial for ensuring the quality of industrial products. Defects on the surface of industrial products are characterized by multi-scale, multi-type, rich small objects, and complex background interference. Particularly, detecting small objects in multi-scale defects under complex background interference poses significant challenges for defect detection tasks. Improving the algorithm's ability to detect industrial defects, especially enhancing the detection capability for small-size defects while ensuring that the inference speed is not excessively impacted, is a longstanding challenge. To achieve accurate and rapid detection in the field of industrial defect detection, this paper proposes a PCP-YOLO anchor-free network method for small object detection. Initially, the anchor-free YOLOv8 is used as the detection framework, eliminating the influence of anchor-related hyperparameters and improving the detection capability for multi-scale and small-size defects. Subsequently, a lightweight and non-deep feature extraction module, PotentNet, is designed and introduced in the backbone network to enhance the extraction of fine-grained defect features in images. Then, in the neck network, a feature fusion module with polarized self-attention, C2f_ParallelPolarized, is designed to enhance the model's ability to fuse features of small-size defects in images from the perspectives of polarized filtering and increasing the dynamic range of attention. Finally, CARAFE is used to replace the original upsampling module in the neck network to enhance the model's ability to utilize semantic information around points near features in images. This method has been evaluated on public datasets NEU-DET, PCB-DET, and the real industrial scene dataset GC10-DET. The mAP@0.5 values are 79.4%, 96.1%, and 77.6% respectively, which are 2.7%, 2.4%, and 2.7% higher than those of the YOLOv8 detection method, significantly outperforming the SOTA detection methods. The inference speed of this method ranks second among 13 models. The results show that PCP-YOLO is promising for real-time defect detection in industry.

Keywords: Industrial defect, YOLOv8, Small object detection, Feature enhancement, Feature fusion Anchor-free network

1 Introduction

Defect detection is an indispensable part of industrial production and plays a crucial role in ensuring product quality. During the production and usage of industrial products, various scales of surface defects may occur due to factors like material quality, production equipment, and manufacturing processes[1]. These defects can affect the appearance and performance of industrial products, thereby reducing production efficiency and even causing engineering safety incidents[2]. However, if these defects can be accurately and rapidly identified during the industrial production process, such potential issues can be somewhat mitigated[3].

Nevertheless, defect detection is not an easy task. Industrial defects often exhibit more complex characteristics, generally characterized by varying defect scales, a prevalence of small-sized defects, and complex background interference[4]. These characteristics pose significant challenges for industrial defect detection. Moreover, in practical industrial applications, the speed of defect detection models is also critical, necessitating tools capable of real-time industrial defect detection tasks[5].

Evolved from convolutional neural networks, the SSD[6], YOLO series[7], and RCNN series[8] are representative baseline models for object detection, widely applied in various scenarios. These detection models include RCNN, Fast-RCNN[9], Faster-RCNN[10], and others. Although these models are accurate and precise, their slow speed does not meet the real-time requirements of industrial scenes. Single-stage detection models, which balance detection precision and speed, are capable of real-time defect detection[11]. Therefore, compared to two-stage detection models, single-stage models better meet the growing detection needs of modern industry and are easier to apply in practical industrial scenarios.

To enhance the feature extraction capability of the model, related teams have added large parameter modules such as attention mechanisms, transformers, and residual connections to the backbone network of single-stage detection models. While these modules do enhance feature extraction to some extent, they usually involve many parameters and multiple modules, inevitably increasing the network's depth and impacting detection speed. However, the feature maps in the backbone network do not contain rich position and semantic information. Merely enhancing the feature extraction capability of the backbone network is insufficient to meet the needs of defect detection applications for position and semantic information[12]. Even if some feature information is enhanced, as it is transmitted to deeper feature extraction layers and the feature fusion network in the neck, the enhanced feature information often gets lost again, especially details and positional features related to small objects. Therefore, enhancing only the backbone network's feature extraction capability is inadequate, and emphasis should also be placed on lightweight enhancements.

For feature fusion optimization, typical methods design new feature propagation and interaction paths to enhance the network's capability for feature fusion from the perspective of enhanced feature extraction[13]. Although this method can effectively improve the detection of multi-scale defects, it may not substantially enhance the detection capability for small-size defects under complex background interference due to limited small-size defect feature information and excessive interference features[14].

Considering the applicability of these two mainstream improvement methods, this study starts with how to rationally improve the backbone network of the base model without introducing excessive parameters, enhance the feature extraction capability of the backbone network, and improve the components used for feature fusion in the neck network of the base model, filtering out interference features to enhance the detection capability of small targets in multi-scale defects. Consequently, this paper develops a new deep learning method, the anchor-free PCP-YOLO, aimed at achieving precise and rapid detection of industrial multi-scale defects.

In summary, the main contributions of this paper are as follows:

Employing the PCP-YOLO method for detecting small target defects in multi-scale defect backgrounds, which is more precise and lightweight compared to traditional models. Moreover, the use of an anchor-free structure eliminates the influence of artificially designed prior anchors, allowing more flexible detection of multi-scale and small-scale defects.

Designing and introducing lightweight and shallow feature extraction modules, PotentNet, within the backbone network to enhance the model's capability to extract fine-grained defect features from images.

Enhancing the model's feature fusion capability for small-size defects in images through the design of the Polarized Self-Attention module, C2f_ParallelPolarized, which strengthens feature fusion by polarized filtering and expanding the dynamic range of attention, while employing CARAFE to replace the original upsampling module in the neck network, enhancing the model's ability to utilize semantic information around points near image features.

2 Related work

Compared to two-stage object detection algorithms, single-stage object detection algorithms strike a better balance between precision and speed and are widely applied in the industrial detection field. Represented by SSD and the YOLO series, these single-stage algorithms use a standalone network to directly classify and adjust the predicted bounding boxes through anchors[15]. The latest YOLOv8 algorithm achieves state-of-the-art (SOTA) performance in the object detection field, offering faster inference speed, higher detection precision, and smaller model size. Unlike previous YOLO series algorithms, YOLOv8 employs an anchor-free mechanism, which helps enhance its performance in defect detection and reduces reliance on manually designed components. However, YOLOv8 still faces limitations inherent to single-stage object detection algorithms, such as false positives, missed detections, and limited accuracy in tasks targeting defects with complex backgrounds and multiple scales[16]. These issues stem from the YOLO series' backbone network's limited ability to extract features of multi-scale defects in complex backgrounds and the neck network's insufficient integration of fine-grained, small-size features.

Various teams have made series of improvements based on the YOLO model series. These improvements focus on enhancing the original algorithm's backbone network, neck network, and anchor design. Kou et al.[17] proposed a defect detection algorithm based on YOLOv3, incorporating dense convolutional blocks and an anchor-free feature selection mechanism into YOLOv3's backbone network, capable of performing simple defect detection tasks but with low precision, making it unsuitable for detecting multi-scale defects in actual industrial scenes; Dong et al.[18] introduced a parallel hybrid attention mechanism into the backbone network of an improved YOLOv5 algorithm to enhance feature extraction, but this method's attention mechanism, involving deeper network layers, affected the model's computational speed and still fell short in integrating features of various defect scales; Xu et al.[19] enhanced the feature extraction ability of YOLOv5's backbone network by integrating a CA attention mechanism, but the method still showed limitations in detecting multi-scale defects; Lu et al.[20] replaced the original algorithm's backbone network with a ShuffleNet network and added an SE attention module, effectively enhancing the backbone network's feature extraction capability, but without rich feature mapping of position and semantic information in the backbone network, solely strengthening the feature extraction capability is insufficient for the demands of defect detection tasks; Zhao et al.[21] introduced a ResNet module into the backbone network of a steel surface defect detection algorithm based on YOLOv5 and modified the FPN structure to DFPN in the neck network, which improved the overall detection precision and feature fusion capability of the model, but it still failed to effectively enhance the detection capability for multi-scale defect features; Wang et al.[22] proposed a method based on an improved YOLOv7 for detecting defects on steel surfaces, utilizing a weighted bidirectional feature fusion network to further integrate multi-scale features, but this method only considered improving feature propagation paths and failed to effectively filter invalid information interference from complex backgrounds, limiting the model's ability to recognize small-scale defect features; Yang et al.[23] developed a defect feature detection method based on an improved YOLOv8, which involved replacing the backbone network with a Swin-Transformer module and introducing a new bounding box loss calculation method, effectively improving the overall detection accuracy of the original algorithm, but this method, using deeper network layers and introducing a large number of parameters, overlooked the need for model lightness; Weining Xie et al.[24] proposed a steel surface defect detection algorithm based on an improved YOLOv8, enhancing the model's ability to extract features of multi-scale defects, but the algorithm model introduced a large number of parameters, resulting in slower detection speed; Qian et al.[25] proposed an improved YOLOX algorithm based on a lightweight feature fusion network, significantly enhancing the model's detection speed, but still showing insufficient ability to detect multi-scale defect features; Ling et al.[26] proposed a PCB board defect detection method based on an improved YOLOv8, enhancing the backbone network's feature extraction capability through an improved C2f and introducing lightweight Ghost convolutional modules to enhance the neck network's feature fusion capability, although this method improved the overall detection capability compared to the original algorithm, it still failed to effectively filter interference from complex background information and showed insufficient performance in detecting small-scale defect features in complex backgrounds.

Despite the improvements proposed by related teams in detection performance, their ability to detect defects in complex backgrounds with multi-scale defects remains insufficient, especially in recognizing and locating small object defects. Therefore, this paper proposes the PCPC-YOLO model, primarily aimed at detecting multi-scale defects in complex background interference and effectively identifying small object defects within multi-scale defects. Unlike the aforementioned methods, we thoroughly analyze the correlation between small-size defects in multi-scale defect backgrounds and the feature fusion network, focusing on the challenges of complex background information interference. Without introducing excessive additional parameters and structures, we enhance the model's capability to extract features of multi-scale defects and integrate small-size defect features, with virtually no loss in detection speed. Additionally, we utilize the flexibility of the anchor-free mechanism to detect small-size and multi-scale defects, eliminating the model's dependence on human experience.

3 Developed methods for small object detection of multiscale defects

PCP-YOLO is an algorithm model developed based on the YOLOv8n architecture, specifically designed to perform multi-scale defect detection tasks. Therefore, in this section, the developed PCP-YOLO architecture is introduced, detailing the PotentNet module used to enhance the multi-scale feature extraction capability, the C2f_ParallelPolarized module designed to augment the fusion of small target defect features, and the Carafe module which increases the model's understanding of the semantic information around defect features.

3.1 Lightweight and non-deep feature extraction module

While the backbone network of the base model YOLOv8 already exhibits good feature extraction capabilities, it still faces challenges in adequately extracting features from multi-scale defects[27, 28]. A key consideration is how to enhance the model's capability to extract features of multi-scale defects without introducing too many parameters and while maintaining the model's lightweight nature.

We have drawn inspiration from the initial block design of the high-performance non-deep network, Rep-VGG[29], and modified it to better suit the task of extracting multi-scale features in real-time defect detection. This modified module, designed to enhance feature extraction capabilities, is called PotentNet. For a non-deep network with only 3*3 convolutions, one challenge is its relatively limited receptive field. To address this issue, we constructed a layer based on the Squeeze-and-Excitation (SE) design, termed the Skip-Pooling-Squeeze-Excitation (SPSE) layer. Traditional mechanisms for enhancing feature extraction often involve deeper networks with more parameters, which can impact detection speed. Therefore, we use the SPSE design, which is applied together with skip connections and utilizes a single fully connected layer. In the experimental section of this paper, we found that this design helps to improve performance. Figure 1 provides a schematic of the PotentNet module structure with the SPSE mechanism.

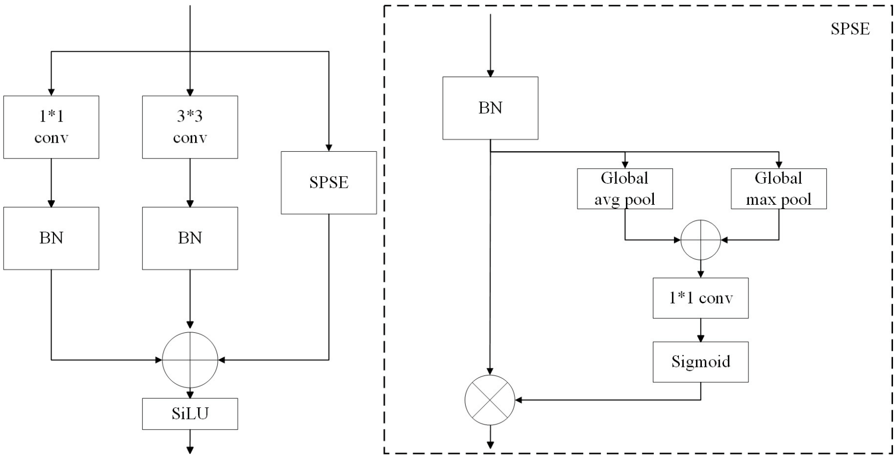


Fig. 1 Schematic diagram of the lightweight and non-deep PotentNet module structure

As shown in Figure 1, the PotentNet module consists of a 1*1 convolutional layer , a 3*3 convolutional layer , an SPSE module, and a linear activation layer. The SPSE module includes a batch normalization layer, a global average pooling layer, a global max pooling layer, a 1*1 convolutional layer, and a linear activation layer. When the PotentNet module receives image feature information processed by the upper layers of the backbone network, it first sends the information to both the 1*1 and 3*3 convolutional modules and the SPSE module for processing. The information processed by the two convolutional modules and the SPSE module is then fused, and finally processed by the linear activation layer SiLU before being sent into the neck network. The SPSE module first performs batch normalization, feature fusion of global average pooling and max pooling, convolutional feature extraction, and linear activation on the incoming information, then performs a skip connection with the batch-normalized information, and finally outputs the information processed by the SPSE module. The PotentNet module does not introduce excessive network layers, ensuring the model's lightweight nature to a certain extent, while effectively extracting feature information through the rational use of convolutional modules, global pooling, and skip connection mechanisms.

3.2 Polarized Feature Fusion Module

Although the neck network of the base model YOLOv8 already performs well in terms of feature fusion capability, it still faces challenges in sufficiently capturing small-size feature information streams for multi-scale defect features, especially for small target defects, which subsequently impacts the performance of feature fusion.

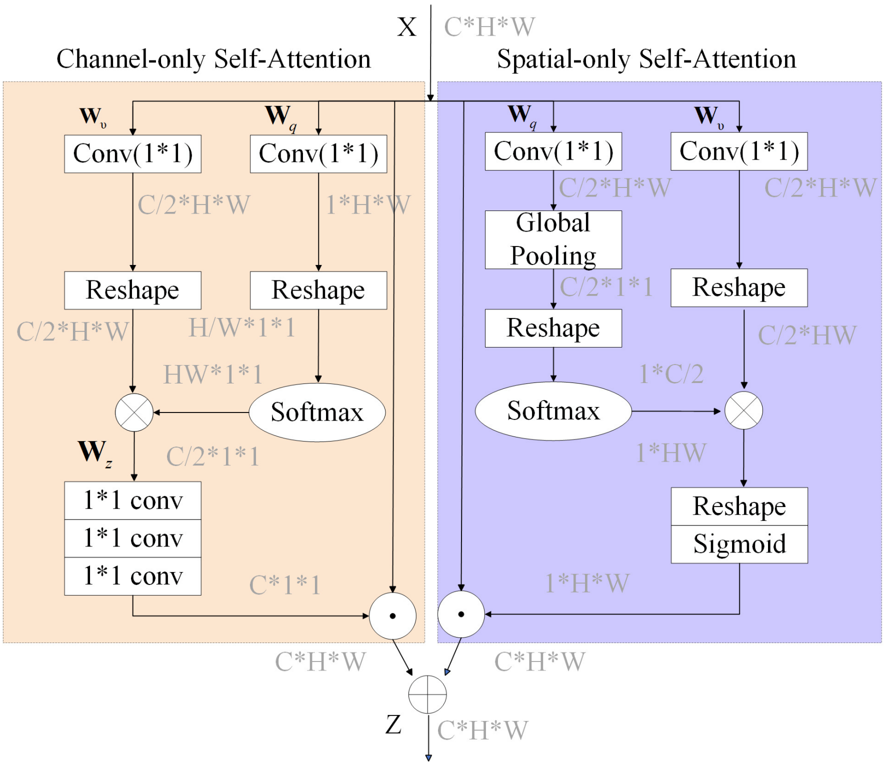


Fig. 2 Schematic Diagram of the Polarized-Self-Attention Network Structure

To enhance the neck network's ability to fuse feature information, this study introduces a Polarized-Self-Attention mechanism (PSA)[30], suitable for integrating small target defect features, and designs the C2f_ParallelPolarized module equipped with the PSA mechanism to replace the standard C2f. In the field of computer vision, research on self-attention mechanisms primarily focuses on operating on the input tensor X with self-attention blocks to highlight or suppress features. Traditional self-attention mechanisms struggle to capture rich gradients of small-scale defect features. The inspiration for the polarized self-attention mechanism presented in this paper comes from scenarios encountered in camera photography. For instance, when improving the quality of photos taken by cameras, the camera always encounters random light causing glare and reflections across the horizontal direction. Using a filtering mechanism that allows light to pass only perpendicular to the horizontal direction can effectively enhance the photo's contrast. However, such filtered light usually has a smaller dynamic range, thus requiring additional enhancement, such as through High Dynamic Range (HDR) methods, which can restore details of the original scene.

Based on this concept of filtering light and enhancing the dynamic range of light, the C2f_ParallelPolarized module with the PSA mechanism was designed, incorporating two key steps: polarized filtering and expanding the attention dynamic range. Polarized filtering involves processing feature information from both the channel and spatial dimensions while maintaining the transmission of the original feature information. The method to expand the attention dynamic range involves processing the feature information from both spatial and channel dimensions, increasing the attention dynamic range of the bottleneck tensor (the two smallest feature tensors from the polarized filtering process) through a Softmax layer, and implementing feature mapping through a Sigmoid layer. This method draws inspiration from HDR photography techniques.

The structure of the PSA is shown in Figure 2, consisting of a channel self-attention processing group and a spatial self-attention processing group. The channel self-attention processing group includes five 1*1 Conv convolutional layers, two Reshape layers for processing tensor dimensions, and one Softmax layer. The spatial self-attention processing group comprises two 1*1 Conv convolutional layers, one GlobalPooling layer, three Reshape layers for processing tensor dimensions, one Sigmoid activation function layer, and one Softmax activation function layer.

As shown in Figure 2, the principle of feature information processing by the PSA module is as follows: When the PSA module receives the tensor X information from the input, the tensor information is transmitted to both the channel self-attention processing group and the spatial self-attention processing group for parallel processing. The specific process is detailed in the following steps:In the channel self-attention processing group: First, tensor X simultaneously undergoes operations w_v and w_q, which involve simultaneous 1*1 convolution (Conv) operations. Then, the tensors obtained from wv and wq operations are modified in different dimensions, followed by matrix multiplication of the dimensionally altered information. Finally, the information goes through three sequential 1*1 Conv operations before outputting the feature information returned by the channel self-attention processing group.In the spatial self-attention processing group: Initially, tensor X simultaneously undergoes operations w_v and w_q, also involving simultaneous 1*1 Conv operations. Subsequently, the tensor from the w_v operation undergoes global pooling (GlobalPooling). Then, the globally pooled tensor and the tensor from the w_q operation are altered in different dimensions, followed by matrix multiplication of this dimensionally altered information. Lastly, the process concludes with a tensor dimension change (Reshape) and activation via the Sigmoid function, before outputting the feature information returned by the spatial self-attention processing group. The feature information from both the channel self-attention and spatial self-attention processing groups are added and merged, resulting in enriched feature gradient flow information for small targets and enhanced feature fusion capability of the neck network.The feature information from both the channel self-attention and spatial self-attention processing groups are added and merged, resulting in enriched feature gradient flow information for small targets and enhanced feature fusion capability of the neck network.

The PSA module's respective calculation methods in the channel self-attention processing group and the spatial self-attention processing group are shown in formulas (1) and (2). In formula (1), *ch* represents processing related to the channel, while in formula (2), *sp* denotes spatial modulation processing. In formulas (1) and (2),w_v, w_q, and w_z represent convolution operations with different parameters and depths. The symbols σ_1, σ_2, and σ_3 in formulas (1) and (2) represent different tensor reshaping operators in the channel and spatial attention processing groups, used for Reshape operations. The output tensor dimensions from the channel self-attention processing group and the spatial self-attention processing group are both $C * H * W$. The feature vectors output from both groups are added together, resulting in a fused feature vector after the application of the polarized self-attention mechanism. The calculation methods for F_{SG} and F_{SM} are shown in formulas (3) and (4), representing the processes of the Sigmoid activation function and the Softmax normalization layer, respectively. In formula (4), x_j refers to the vector being processed, and N_p indicates the total number of categories in the normalization process.

$$A^{ch}(X)=F_{SG}[\mathbf{W}_{z|\Theta_1}((\sigma_1(\mathbf{W}_v(X))^* F_{SM}(\sigma_2(\mathbf{W}_q(X)))))] \tag{1}$$

$$A^{sp}(X)=F_{SG}[\sigma_3(F_{sM}(\sigma_1(F_{GP}(\mathbf{W}_q(X))))^*\sigma_2(\mathbf{W}_v(X)))] \tag{2}$$

$$F_{SG}(X) = \frac{1}{1+e^{-x}} \tag{3}$$

$$F_{SM}(X) = \sum_{j=1}^{N_p} \frac{e^{x_j}}{\sum_{m=1}^{N_p} e^{x_m}} x_j \tag{4}$$

As shown in Figure 3, the C2f_ParallelPolarized module primarily consists of CBS units and PolarizedBottleneck units. The PolarizedBottleneck comprises two CBS units and one PSA unit. This process extracts features of different levels and abstraction from the input data through CBS units and multiple PolarizedBottleneck units, and fuses the resampled features using element-wise addition. The C2f_ParallelPolarized module effectively utilizes the bottleneck module to expand gradient branches, ensuring a lightweight structure while obtaining richer gradient flow information. Such a branch design helps increase the network's non-linear capacity and representational ability, thereby enhancing the network's capability to model complex data and achieve better feature fusion results.

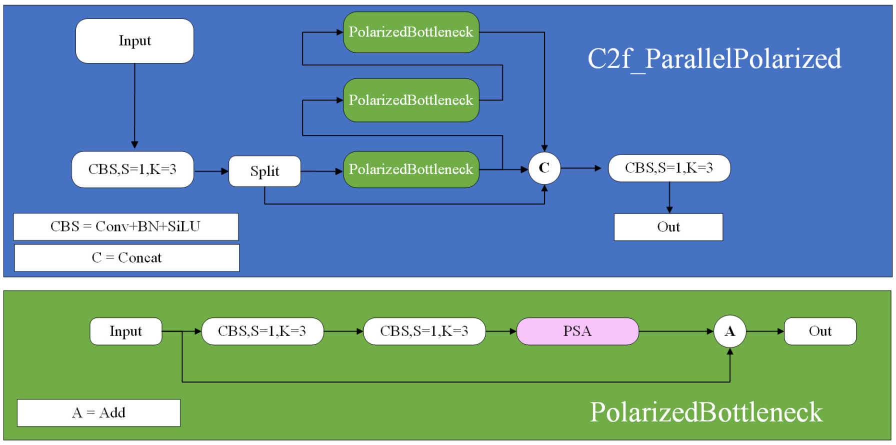


Fig. 3 Schematic Diagram of the C2f_ParallelPolarized Module

To further highlight the feature fusion capabilities of the C2f_ParallelPolarized module, research was conducted using both the C2f module and the C2f_ParallelPolarized module for feature map visualization on the NEU-DET dataset. As shown in Figure 4, column (a) represents the defect detection results of the corresponding detection models, while columns (b) and (c) show the visualization effects of feature maps at specific network layers for the corresponding models. From Figure 4, it is evident that the network with the C2f_ParallelPolarized module performs better, accurately locating defect areas and suppressing irrelevant regions. In contrast, the C2f module fails to precisely locate defect areas and erroneously focuses on some background regions.

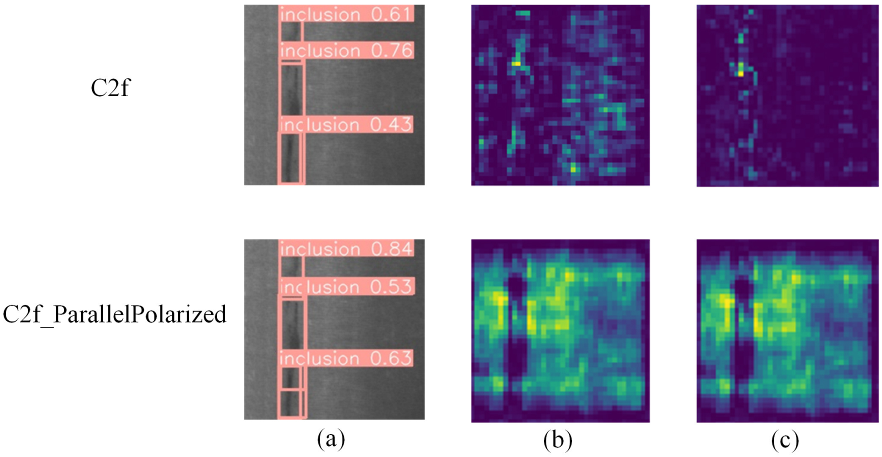


Fig. 4 Feature map visualization comparison of C2f and C2f_ParallelPolarized modules on the NEUDET dataset.Column (a) represents the defect detection results of the corresponding detection models, while columns (b) and (c) show the visualization of feature maps at specific network layers for the corresponding models.

3.3 Upsampling Module with Enhanced Understanding of Semantic Features

In the feature fusion process of the base model's neck network, upsampling is a crucial step. The purpose of upsampling is to scale low-resolution feature maps to the same size as high-resolution feature maps to facilitate feature fusion or multi-scale object detection. However, the base model still uses nearest neighbor interpolation for upsampling, which determines the upsampling kernel solely based on the spatial position of pixels and does not utilize the semantic information of the feature maps. This method ignores the potential influence of surrounding feature points and has a small receptive field, resulting in poor quality of the upsampled images. This paper adopts the CARAFE, a lightweight upsampling operator with a larger receptive field[31], replacing the original upsampling operator in the base model. This change maintains the lightweight nature of the neck network while better utilizing the semantic information of the feature maps.

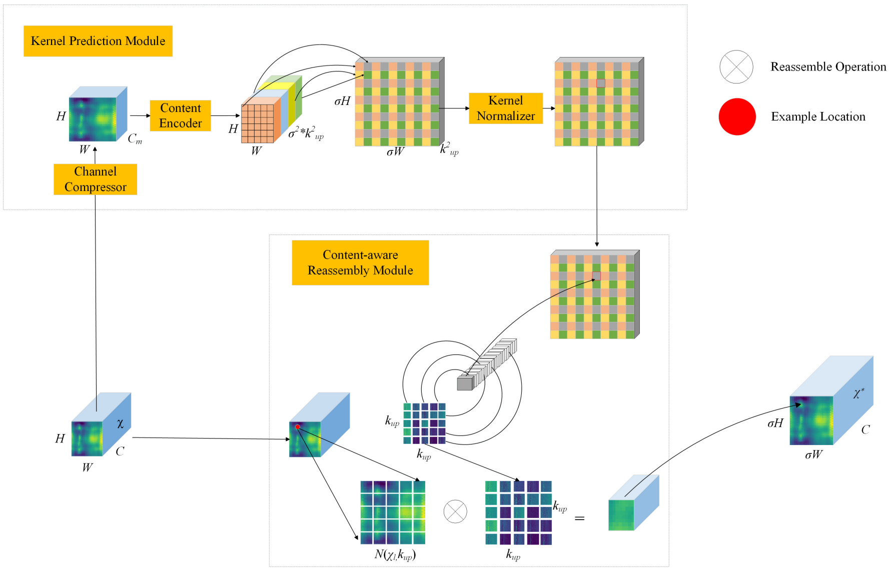


Fig. 5 Structural diagram of the upsampling operator CARAFE

The overall sampling process of CARAFE is shown in Figure 5. First, for the input feature map χ with shape $H * W * C$, a 1*1 convolutional layer is used to compress the input channels from C to C_m to reduce the model's parameters and computational costs. Next, based on the feature map with shape $H * W * C_m$, convolutional layers are used as content encoders to predict the upsampled kernel, resulting in a reshaped upsampled kernel of shape $H * W * \sigma^2 k_{up}^2$, where σ is the upsampling factor and k_{up} is the receptive field size of the feature recombination process. Then, the channels are expanded in the spatial dimension, resulting in an upsampled kernel of shape $\sigma H * \sigma W * k_{up}^2$. Finally, the upsampled kernel is normalized so that the sum of convolutional kernel weights equals one. In the feature recombination module, for each position in the output feature map, it is mapped back to the input feature map, taking a $k_{up} * k_{up}$ region centered on it, and the predicted upsampled kernel is dot-multiplied to obtain the output value. The same position across different channels shares the same upsampled kernel, resulting in an upsampled feature map χ^* of shape $\sigma H * \sigma W * C$.

3.4 Architecture of the developed PCP-YOLO network for defect detection

To improve the accuracy and detection speed of small target defect detection tasks within multi-scale defects in industrial scenarios, and to facilitate the deployment of the model on edge computing devices, we propose the PCP-YOLO network structure. The network architecture of PCP-YOLO for defect detection is illustrated in Figure 6, comprising three parts: the backbone, neck, and head networks. Initially, the input part of PCP-YOLO uniformly adjusts the resolution of the input image to 640*640; next, the backbone network extracts feature information from the input image through a series of convolution-based modules; subsequently, the backbone network outputs low, medium, and high-level feature maps to the neck network for feature fusion, which then transmits three scales of fused feature maps to the head network; finally, the head network predicts the position and size of target objects based on the three feature maps outputted by the neck network.

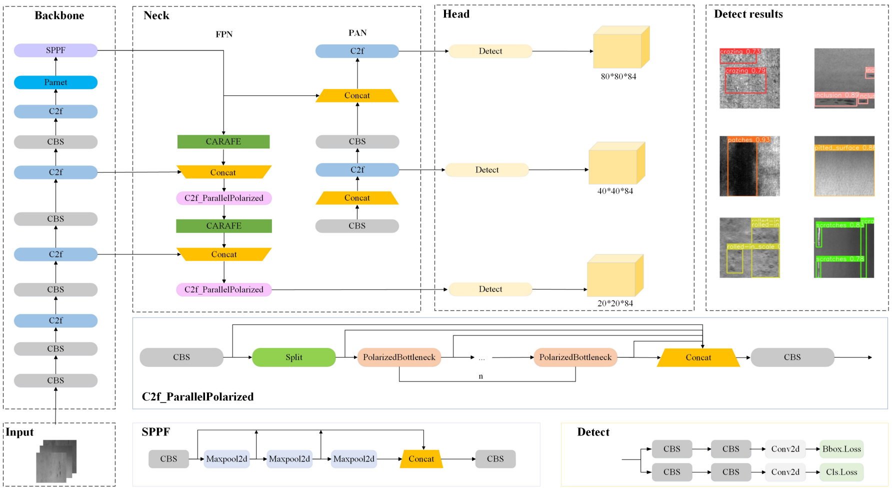


Fig. 6 Structure of DsP-YOLO network

The architecture proposed in this paper is built on the basic paradigm of YOLOv8n. Lower layers contain rich positional information, while higher layers have rich semantic information. For multi-scale defect features, improvements to the backbone network are made from the perspective of mining semantic information within multi-scale defects. Unlike the base model, the study introduces the lightweight and shallow PotentNet network module at higher levels of the backbone network to enhance its feature extraction capability. The neck network uses the classic FPN and PAN structures, with the FPN structure constructing a feature upsampling path through a horizontal connection at the top of the backbone, involving the fusion of lower and higher-level features. The PAN structure enhances the transfer of lower-level features to higher-level ones by constructing horizontal and vertical paths to fuse features of different resolutions. Differing from the base model's FPN structure using the C2f module, the study, inspired by polarized filtering and High Dynamic Range (HDR) enhancement, designs the C2f_ParallelPolarized module equipped with the PSA mechanism. This module captures the flow of small target defect features within multi-scale defects more effectively than C2f and is more lightweight, resulting in better fusion of small target defect features within multi-scale defects. Additionally, the original nearest neighbor interpolation upsampling method of the base model is replaced with the Carafe upsampling operator. This operator is more lightweight compared to the original network's upsampling operator and better aligns low-resolution feature maps with high-resolution feature maps of the same size, further enhancing the neck network's feature fusion capability.

4 Experiment

The research uses PCP-YOLO for training, validation, and testing on the NEU-DET, PCB-DET, and GC10-DET datasets. Additionally, PCP-YOLO's detection performance is compared with models from related research, and an analysis of the experimental data is conducted. Descriptions of the datasets are provided in Section 4.1 of this chapter, while the experimental environment, evaluation metrics, and detailed experimental results are presented in Sections 4.2, 4.3, and 4.4, respectively.

4.1 Dataset description

To fully verify the effectiveness and generalization performance of this method, this paper utilizes three popular public datasets: the NEU-DET and GC10-DET datasets for steel surface defect detection, and the PCB-DET dataset for PCB defect detection. Detailed information about these datasets is shown in Table 1. The table includes columns for Dataset, representing the name of the dataset; Defect type, indicating the types of defects in each dataset; Total images, showing the number of images in the dataset; Image size, displaying the dimensions of the images; and separate columns for Train, Validation, and Test, which indicate the number of images in each dataset used for training, validation, and testing, respectively.

Table 1 Detailed information on the NEU-DET, PCB-DET, and GC10-DET datasets

Dataset	Defect type	Total images 3	Images size	Train	Validation	Test
NEU-DET	6	1800	200*200	1260	270	270
PCB-DET	6	693	3034*1586	555	555	69
GC10-DET	10	2280	2048*1000	1824	228	228

(1) NEU-DET: This is an open dataset for steel surface defect detection created by Northeastern University. It includes six typical types of surface defects on hot-rolled strip steel: crazing (Cr), inclusion (In), patches (Pa), pitted surface (Ps), rolled-in scale (Rs), and scratches (Sc). Examples of these six defects are shown in Figure 7. The dataset contains a total of 1800 images, with 300 images for each defect type. In the experiments of this paper, we divided the dataset in an 8:1:1 ratio, consisting of 1260 training images, 270 validation images, and 270 test images.

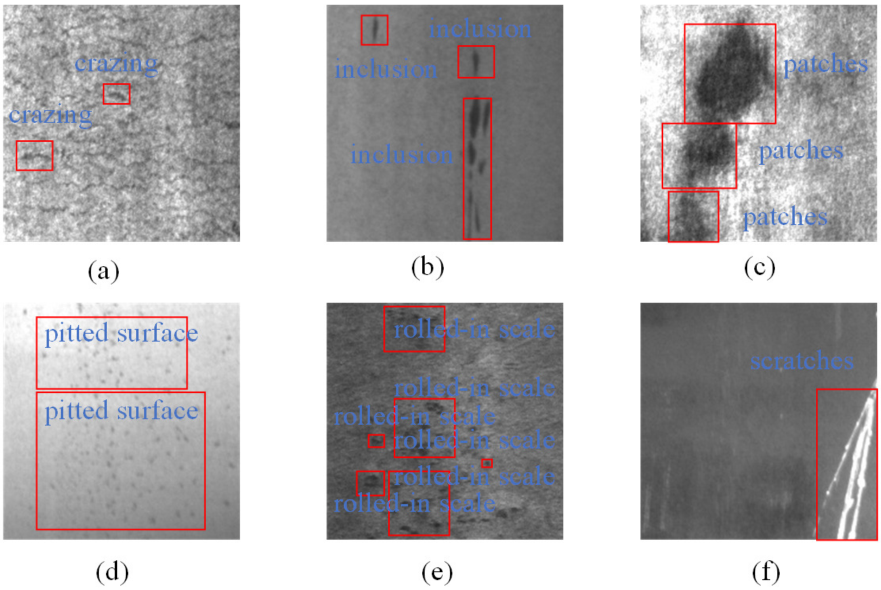
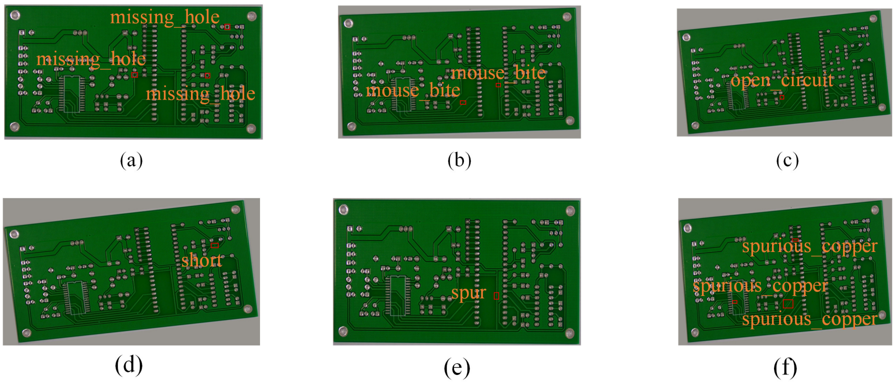


Fig. 7 Annotated representative defects in NEU-DET: (a) crazing(Cr), (b) inclusion(In), (c) patches(Pa), (d) pitted surface(Ps), (e) rolled-in scale(Rs), and (f) scratches(Sc)

(2) PCB-DET: As shown in Figure 8, this is the PCB defect dataset released by Peking University. It includes six types of defects: mouse bite (Mb), short (Sh), spur (Sp), spurious copper (Spc), missing hole (Mh), and open circuit (Oc). The dataset contains a total of 693 images. In the experiments of this paper, we divided the dataset into 555 training images, 69 validation images, and 69 test images.

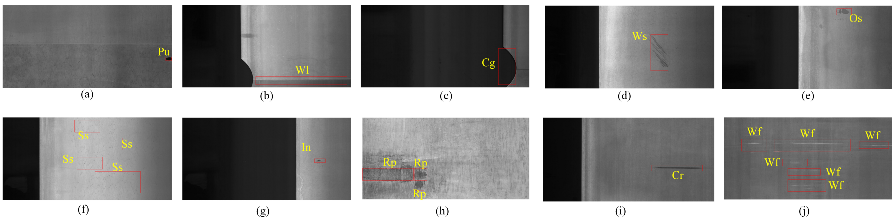
(3) GC10-DET: As shown in Figure 9, this is a public dataset of steel plate surface defects collected in a real industrial setting. It includes ten types of defects: punch hole (Ph), weld line (Wl), crescent-shaped notch (Cg), water stain (Ws), oil stain (Op), silk mark (Ss), inclusion (In), rolling pit (Rp), crease (Cr), and waist fold (Wf). The dataset contains a total of 2280 images. In the experiments of this paper, we divided the dataset into 1824 training images, 228 validation images, and 228 test below images.

These three types of datasets contain a variety of defect features across different scales, making them suitable for training and validating the capability of models to detect small and medium-sized targets in multi-scale defect detection. Additionally,



(a) (b) (c) (d) (e) (f)

Fig. 8 Annotated representative defects in PCB-DET: (a) missing_hole(Mh), (b) mouse_bite(Mb), (c) open_circuit(Oc), (d)short(Sh), (e) spur(Sp), and (f) spurious_copper(Spc)



(a) (b) (c) (d) (e) (f) (g) (h) (i) (j)

Fig. 9 Annotated representative defects in GC10-DET: (a)Pu, (b) Wl, (c) Cg, (d)Ws, (e) Os, (f) Ss, (g) In, (h)Rp, (i) Cr, and (j) Wf

as shown in Figures 7-9, the defects in these three datasets also exhibit characteristics such as background interference, lighting effects, small sample sizes, and extreme aspect ratios, which pose challenges to real-time and accurate defect detection.

4.2 Experimental environment

The experimental environment in this paper operates on a Linux operating system. The hardware specifications include an i9-13900HX CPU and an NVIDIA GeForce RTX 4060 GPU. We use the PyTorch deep learning framework to conduct our experiments. The version of the PyTorch framework is 2.1.0, and the version of CUDA is 12.1.

In this study, the deep learning methods used maintain the same hyperparameters. Likewise, although fine-tuning hyperparameters remains an unresolved challenge that requires extensive research, the focus of this paper is on developing a new industrial defect detection model with a specially designed architecture, rather than optimizing hyperparameters. The settings of the model hyperparameters are shown in Table 2.

4.3 Experimental metrics

To comprehensively evaluate the accuracy of the model, this paper employs the most classic verification metrics such as precision, recall, Average Precision (AP), and Mean

Table 2 Initialization parameters of our method

Parameters	Value	Description
Learning rate	1e-5	Initial learning rate
Decay strategy	Cosine	Description of the learning rate decline strategy
Optimizer	SGD	The type of the optimizer
Momentum	0.937	Impulse value setting during training
Weight decay	5e-4	The parameter settings for overfitting
Total epochs	200	The number of training rounds
Btach size	16	The capacity of every batch

Average Precision (mAP), with mAP as the main evaluation criterion. As shown in equations (5) to (8), the definitions of these metrics are as follows:

$$Precision = \frac{TP}{(TP+FP)} \tag{5}$$

$$Recall = \frac{TP}{(TP+FN)} \tag{6}$$

$$AP = \int_0^1 p(r)dr \tag{7}$$

$$mAP = \frac{1}{n}\sum_{i=1}^{n} AP_i \tag{8}$$

Here, TP represents the number of samples that the model correctly identifies as positive and are indeed positive, FP represents the number of samples that the model incorrectly identifies as positive but are actually negative, and FN represents the number of samples that the model incorrectly identifies as negative but are actually positive. $p(r)$ denotes the function of recall, and n is the number of samples in a certain category. AP_i represents the classification detection accuracy for the first category.

Furthermore, to assess the detection speed of the model and evaluate its real-time detection capability, the study employs Frames Per Second (FPS) to reflect the inference speed of the model. FPS indicates the number of image frames processed by the model per second. The higher the FPS value, the faster the model's defect detection speed. It is worth noting that when calculating the FPS for all models, we standardized the batch size of the model to 1.

4.4 Experiments results

4.4.1 Performance comparisons

To verify the effectiveness of the proposed method, the baseline model YOLOv8 and the model proposed in this paper, PCP-YOLO, were tested on the NEU-DET, GC10-DET, and PCB-DET datasets. The results of the models on the metrics of precision (P), recall (R), and mean Average Precision (mAP) are shown in Tables 3 and 4.

Among them, the indicator "↑5.5" indicates that the mAP of the developed PCP-YOLO is 5.5% higher on the corresponding defect type than the baseline model

Table 3 Detection performance of YOLOv8 and PCP-YOLO on NEU-DET and GC10-DET datasets.

Dataset	Methods	Defect Type	P%	R%	mAP%
NEU-DET	YOLOv8	Cr	44.7	40.7	40.3
		In	78.1	74.2	77.6
		Pa	81.3	98.6	97.1
		Ps	81.1	85.7	88.7
		Rs	66.5	61.7	64.1
		Sc	80.2	91.7	92.3
	PCP-YOLO	Cr	60.2	42.4	44.5(↑4.2)
		In	85.0	72.7	81.6(↑4.0)
		Pa	90.4	92.7	96.5(↓0.6)
		Ps	87.7	81.0	90.2(↑1.5)
		Rs	85.2	48.0	69.6(↑5.5)
		Sc	85.3	87.6	94.1(↑1.8)
GC10-DET	YOLOv8	Pu	97.8	93.9	98.8
		Wl	78.7	86.3	82.5
		Cg	85.3	99.1	96.9
		Ws	78.5	80.6	82.5
		Os	86.3	52.7	71.9
		Ss	74.5	57.1	63.4
		In	80.1	22.3	43.3
		Rp	24.3	50.1	57.3
		Cr	53.0	66.7	72.6
		Wf	68.8	55.3	80.0
	PCP-YOLO	Pu	96.1	97.0	98.8(equals)
		Wl	78.7	86.8	83.3(↑0.8)
		Cg	83.7	98.7	96.4(↓0.2)
		Ws	70.6	83.9	86.8(↑4.3)
		Os	78.9	55.5	70.9(↓1.0)
		Ss	73.2	67.9	71.9(equals)
		In	78.4	33.3	42.5(↓0.8)
		Rp	27.9	50.1	57.3(equals)
		Cr	89.9	77.8	78.1(↑5.5)
		Wf	70.4	83.3	90.0(↑1.0)

YOLOv8, ”↓0.2” represents that the mAP is 0.2% lower on the corresponding defect type than YOLOv8, and ”equals” means that the mAP is equal to that of the baseline model on the corresponding defect type. Other cells in the table follow a similar notation. A detailed discussion follows.

The PCP-YOLO model's performance on the NEU-DET dataset is shown in Table 3. Among these, except for a 0.6% decrease in the average precision (mAP) for the Pa defect, the average precision for other defects has improved. Additionally, the prediction precision (P) for all defects has improved, indicating that our approach effectively compensates for the deficiencies of the anchor-free mechanism. In these six types of defects, the Cr defect and Rs defect are typical small and medium-sized objects. The original YOLOv8's mAP values for Cr defect and Rs defect were only 40.3% and 64.1%, respectively. Due to the embedding of our proposed PotentNet structure in the backbone network, it effectively enhances the feature extraction capability of the backbone network, performing better extraction of multi-scale defect feature information.

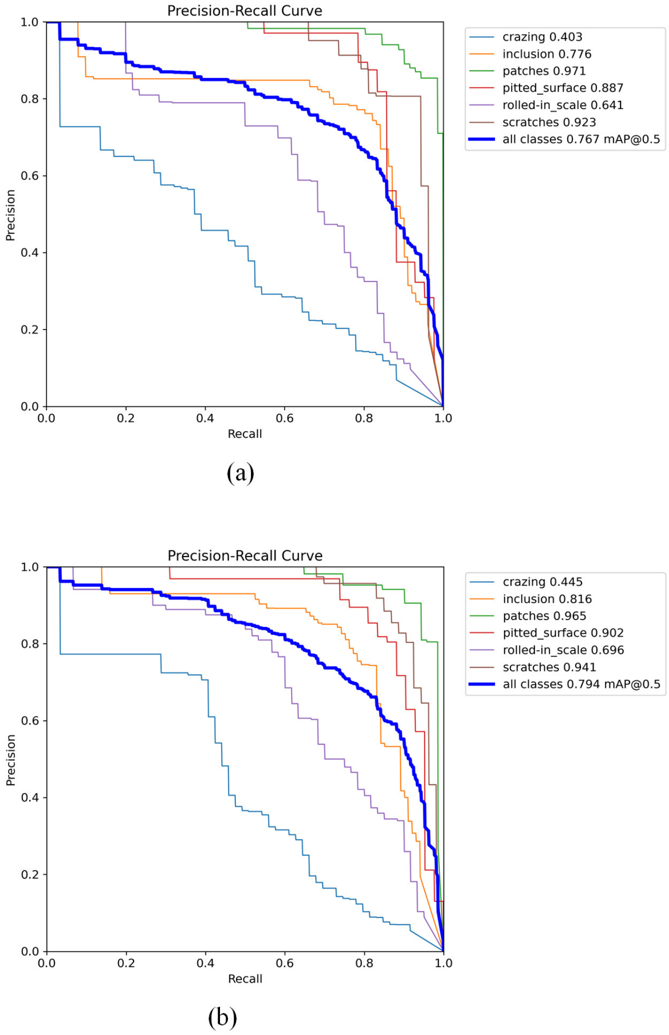
At the same time, in the neck network, we transformed C2f into C2f_ParallelPolarized, which has polarized filtering and high dynamic range enhancement capabilities, effectively capturing the feature flow information of small target defects and enhancing the neck network's ability to integrate features of small target defects. Our method increased the mAP values for Cr defect and Rs defect by 4.2% and 5.5%, respectively. Besides these two defect types, other defect types in industrial production exist in various scales, often appearing in the form of small and medium-sized objects, thus our method enhances the detection of these defects.

Table 4 Detection performance of YOLOv8 and PCP-YOLO on PCB-DET.

Dataset	Methods	Defect Type	P%	R%	mAP%
PCB-DET	YOLOv8	Mh	99.0	98.6	99.5
		Mb	96.2	78.5	84.7
		Oc	99.9	86.7	95.4
		Sh	96.7	94.9	95.7
		Sp	99.9	86.5	91.8
		Spc	95.8	95.3	96.9
	PCP-YOLO	Mh	99.0	98.3	99.5(equals)
		Mb	93.4	87.4	94.3(↑9.6)
		Oc	99.5	93.3	95.9(↑0.5)
		Sh	97.3	91.5	95.4(↓0.3)
		Sp	97.3	89.2	94.0(↑2.2)
		Spc	94.1	95.3	97.6(↑0.7)

To further understand the small object detection capabilities of the developed PCP-YOLO and the baseline model YOLOv8, the study plotted the PR curve for both the PCP-YOLO model and the baseline model based on experimental results. The PR curve, constituted by the coordinates of testing precision and recall, has an area around it known as mAP. The PR curve for both methods are shown in Figure 10, where Figure 10(a) represents the PR curve of the baseline model, and Figure 10(b) represents the PR curve of the PCP-YOLO model. According to Figure 10, the area enclosed by the PCP-YOLO is greater than that enclosed by YOLOv8. The overall mAP@0.5 of the PCP-YOLO model on the NEU-DET dataset is 79.4%, which is 2.7 percentage points higher than the baseline model. It is important to emphasize that among these six steel defects, the defects characterized by small objects are inclusion defects, patches, and pitted surface defects. From the PR curve diagram, it is evident that the PR curve area enclosed by the PCP-YOLO model for these three small target defects is significantly larger compared to YOLOv8.

To further validate the superiority of the proposed PCP-YOLO model over the baseline model, the study conducted a validation using the GC10-DET dataset from a real industrial scenario. As shown in Table 3, compared to the baseline model YOLOv8, PCP-YOLO has significantly improved detection precision for seven defects in the GC10-DET dataset. Specifically, the detection precision for Cr and Ws defects increased by 5.5% and 4.3%, respectively. This indicates that PCP-YOLO effectively enhances the detection precision for small-sized defects. Additionally, PCP-YOLO significantly improved the recall rate for all defects, a crucial indicator for assessing whether detection targets are missed, demonstrating PCP-YOLO's superior performance in detecting small objects among multi-scale defects.



(a)

(b)

Fig. 10 The PR curves for YOLOv8 and PCP-YOLO on the NEU-DET dataset. (a) The PR curves of YOLOv8 on the NEU-DET dataset. (b) The PR curves of PCP-YOLO on the NEU-DET dataset.

Given that the images in the NEU-DET and GC10-DET datasets are of steel surface defects, and the images' background information tends to be dim, further validation was conducted to verify whether the model can still outperform the baseline model under backgrounds of varying richness. The study used the PCB board defect dataset PCB-DET released by Peking University for this purpose. The PCB-DET dataset features many small-sized defects in complex backgrounds, and the dataset itself has richer background information. As shown in Table 4, for the Mb defect, our method improved the recall rate by 8.9% compared to the baseline model. This indicates that PCP-YOLO enables the model to capture more contextual information, thereby enhancing the model's recall rate. PCP-YOLO showed a slight decrease in mAP for the Sh defect but was generally consistent with the baseline model. Besides the Mb defect, the PCP-YOLO model also demonstrated excellent detection performance for the Sp defect, with a mAP improvement of 2.2% compared to the baseline model, indicating our method's good understanding of small object defect features. Moreover, the mAP for other defects also saw improvements compared to the baseline model. This proves the exceptional performance of the PCP-YOLO model on the PCB-DET dataset.

Table 5 Defect detection results on NEU-DET dataset.

Methods	mAP@0.5 of Top5/%	mAP@0.5/%	GFLOPS	Parameters	FPS
Faster R-CNN [10]	76.8	70.8	83.4	41.3 M	24.0
SSD [6]	78.1	71.0	30.6	24.5 M	43.3
RetinaNet [32]	68.2	63.2	74.5	36.4 M	28.7
YOLOX [33]	76.6	70.3	26.8	8.9 M	69.2
YOLOv7 [34]	79.7	73.2	103.2	36.5 M	40.7
YOLOv7-tiny [34]	78.9	68.6	13.1	6.0 M	86.2
YOLOv8	83.9	76.7	8.1	3.6 M	169.2
MSC-DNet [35]	85.2	79.4	78.0	34.1 M	14.1
RDD-YOLO [21]	83.1	77.6	145.6	57.0 M	48.3
DCNN [36]	83.8	76.3	89.8	40.9 M	52.0
DsP-YOLO [2]	85.8	80.4	28.5	11.1 M	86.9
PCP-YOLO	86.6	79.4	9.2	3.8 M	151.4

From Table 5, it is evident that our model significantly leads other SOTA models in mAP. In terms of model inference speed, our model ranks second among the 13 compared models. Compared to the baseline model YOLOv8, our model reduced FPS by 16.8. This is because our improvements add a minimal amount of computation to the model, which is negligible in real-world application scenarios. From lines 8, 9, and 10, it can be seen that compared to current research in defect detection models, our model achieves higher mAP@0.5 and faster inference speed in the same datasets. Additionally, our model is also optimal in terms of the number of parameters and computational complexity. From Tables 5 and 6, although PCP-YOLO's mAP@0.5 is 10.0% lower than DsP-YOLO's mAP@0.5, PCP-YOLO only performs lower than the DsP-YOLO model in the Cr defect category, but outperforms DsP-YOLO in small-scale defects Pa and Ps, with mAP@0.5 surpassing by 1.5% and 8.1%, respectively. At the same time, the PCP-YOLO model has faster inference speed and a smaller model size, making it more suitable for use in real-time industrial defect detection tasks that require speed and model size. Regarding the poor performance in the Cr category of defects, mainly due to the NEU-DET dataset's crack (Cr) images being less distinct and of lower data quality, including DsP, methods in relevant studies have not increased the mAP@0.5 for this type of feature above 55%, indicating the low image quality of Cr defects in the dataset. Therefore, the study added the calculation of the mAP@0.5 for the top five ranked features, essentially calculating the mAP@0.5 excluding Cr features, namely mAP@0.5 of Top5. From the column of mAP@0.5 of Top5, the PCP-YOLO model reached 86.6, which is the best among other SOTA models. Overall, our model is clearly superior to all other SOTA models and can satisfactorily meet the requirements of industrial defect detection tasks.

To verify our model's detection capabilities for each defect on the NEU-DET dataset, we compared the mAP@0.5 values of several mainstream models across various defects, as shown in Table 6. It can be seen that our model outperforms all SOTA models in detecting most defects. This model has improved detection outcomes, particularly for defects that are a mix of small to medium scales under complex background interference. This is because the PCP-YOLO model incorporates the PotentNet module, which further extracts higher-level features of multi-scale defects. The CARAFE module captures the semantic information around feature points, enhancing the understanding of multi-scale feature information. Moreover, C2f_ParallelPolarized in the neck network captures stronger small-scale feature flow information, achieving better feature fusion for more effective small-scale feature mapping.

Table 6 Detection results for each defect on the NEU-DET dataset.

Types	YOLOv7	YOLOv8	Faster R-CNN	SSD	RDD-YOLO	DsP-YOLO	PCP-YOLO
Crazing	40.4	40.3	41.2	35.7	50.1	54.5	44.5
Inclusion	79.6	77.6	73.9	79.3	81.7	84.0	81.6
Patches	90.2	97.1	91.7	85.2	92.6	95.0	96.5
Pitted_surface	77.7	88.7	72.0	80.9	83.7	82.1	90.2
Rolled-in scale	60.1	64.1	54.8	63.8	69.2	72.7	69.6
Scratches	90.9	92.3	92.0	81.3	88.3	94.1	94.1
Overall mAP	73.2	76.7	70.8	71.0	77.6	80.4	79.4
mAP of Top5	79.7	83.9	76.8	78.1	83.1	85.8	86.6

To further validate the generalization ability of the PCP-YOLO model, we conducted comparative experiments on each model using the real industrial scenario dataset, GC10-DET. As shown in Tables 7 and 8, the PCP-YOLO model achieved an mAP@0.5 of 77.6% on this dataset, which is a 2.7% improvement over the baseline model YOLOv8, and higher than other SOTA models. In terms of inference speed, the FPS of the PCP-YOLO model reached 134, slightly lower than the baseline model, but it already meets the requirements for industrial real-time detection. Figure 11 shows the detection effects of the PCP-YOLO model on the GC10-DET dataset. The first element of the predicted labels in the figure is numerical, sequentially corresponding to the 10 respective defect labels. Since the public dataset uses annotation names different from the English abbreviations used in this paper, the first digit in the label shown in the detection images corresponds to the defect label. For example, the label "6_siban" in the figure corresponds to the sixth defect label in this paper, the Ss label. Other labels in the illustration follow this rule. From Figure 11(a), it is known that the baseline model had issues with missed detections, however, the PCP-YOLO model avoided these issues. Figures 11(b) and 11(c) show that both PCP-YOLO and the baseline model detected the correct defect features, yet PCP-YOLO, compared to the baseline model, has higher detection accuracy for small-scale defects in complex backgrounds. From Figure 11(d), it is evident that the baseline model had false detection issues, however, PCP-YOLO avoided this problem and accurately predicted the defect labels and identified the defect locations with high precision. Therefore, the PCP-YOLO model exhibits superior performance, avoiding false and missed detections compared to the baseline model, and demonstrates higher detection accuracy for small objects within multi-scale defects.

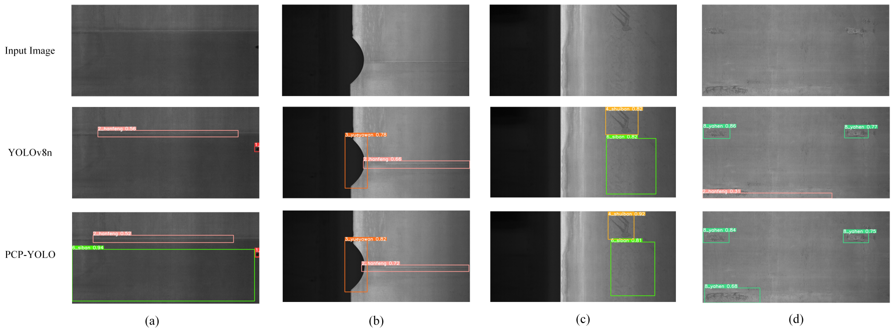


Fig. 11 The detection performance of the PCP-YOLO model on the GC10-DET dataset.

Table 7 Defect detection results on GC10-DET dataset.

Methods	mAP@0.5/%	GFLOPS	Parameters	FPS
Faster R-CNN [10]	67.3	83.4	41.3 M	23.8
YOLOv7 [34]	69.9	103.2	36.5 M	40.1
YOLOv7-tiny [34]	64.5	13.1	6.0 M	84.0
YOLOv8	74.9	8.1	3.11 M	161.0
RDD-YOLO [21]	74.9	145.6	57.0 M	47.8
DCNN [36]	74.8	89.8	40.9 M	51.2
DsP-YOLO [2]	76.3	28.5	11.1 M	85.4
PCP-YOLO	77.6	9.2	3.8 M	134

As shown in Table 8, the detection performance of different models on various defects in the GC10-DET dataset is presented. PCP-YOLO achieved the highest mAP@0.5 for the Ws, Ss, In, and Wf defects, especially for the Ws defect, where its mAP@0.5 value is significantly higher than that of the other models. This indicates that PCP-YOLO exhibits better performance on this dataset compared to the other tested models.

To further validate the robustness of the PCP-YOLO model, we conducted comparative experiments using SOTA models and our model on the PCB-DET dataset. As shown in Tables 9 and 10, our method achieved an mAP@0.5 of 96.1, which is a 2.4%

Table 8 Detection results for each defect on the GC10-DET dataset.

Types	YOLOv7	YOLOv8	Faster R-CNN	RDD-YOLO	DsP-YOLO	PCP-YOLO
Punching hole	96.4	98.8	89.9	99.0	96.7	98.5
Welding line	71.0	82.5	58.5	89.5	92.5	83.3
Crescent gap	90.4	96.9	97.9	94.0	98.7	96.4
Water spot	64.4	82.5	56.3	68.7	70.8	86.8
Oil spot	72.4	71.9	72.0	70.0	66.5	70.9
Silk spot	64.0	63.4	47.5	63.1	58.5	71.9
Inclusion	30.1	43.3	21.7	31.7	23.0	42.5
Rolled pit	35.0	57.3	54.5	63.8	79.9	57.3
Crease	99.5	72.6	94.5	99.5	94.5	78.1
Waist folding	75.4	80.0	80.0	70.2	81.6	90.0
Overall mAP	69.9	74.9	67.3	74.9	76.3	77.6

improvement over YOLOv8 and higher than other SOTA models. In terms of inference speed, YOLOv8 had the highest FPS. Compared to the baseline model YOLOv8, our model's FPS decreased by 15.7, but it still meets the performance requirements for real-time detection. Figure 12 shows the detection results of the PCP-YOLO model on the PCB-DET dataset, and it can be seen from Figure 12 that the PCP-YOLO model exhibited good performance across all defect features.

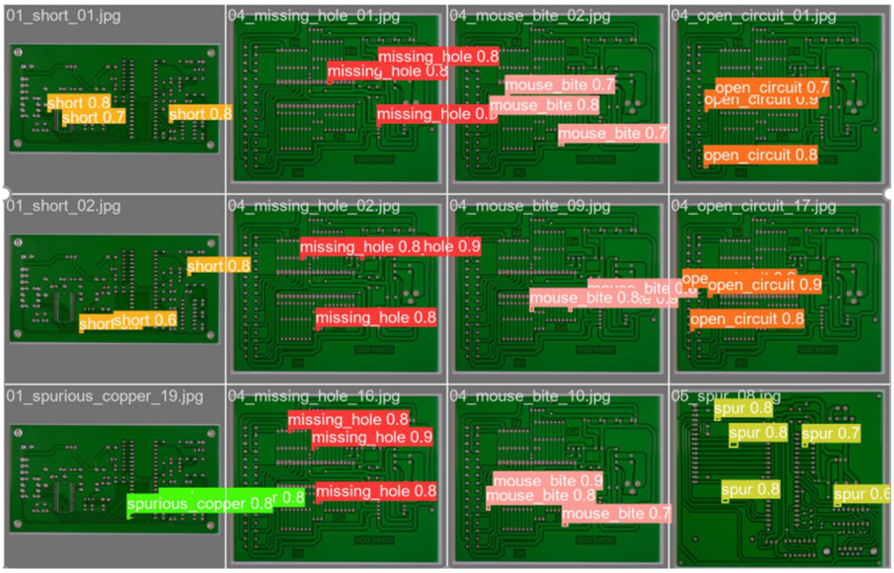


Fig. 12 The detection performance of the PCP-YOLO model on the PCB-DET dataset.

As shown in Table 10, the detection performance of different models on various defects in the PCB-DET dataset is presented. Our model achieved the highest mAP@0.5 for the Mb, Oc, and Sp defect features, and the average detection precision

Table 9 Defect detection results on PCB-DET dataset.

Methods	mAP@0.5/%	GFLOPS	Parameters	FPS
Faster R-CNN [10]	67.3	83.4	41.3 M	23.8
YOLOv7 [34]	69.9	103.2	36.5 M	40.1
YOLOv7-tiny [34]	64.5	13.1	6.0 M	84.0
YOLOv8	74.9	8.1	3.11 M	161.0
RDD-YOLO [21]	74.9	145.6	57.0 M	47.8
DCNN [36]	74.8	89.8	40.9 M	51.2
DsP-YOLO [2]	76.3	28.5	11.1 M	85.4
PCP-YOLO	77.6	9.2	3.8 M	134.0

(mAP@0.5) for the remaining features reached above 94%, slightly higher than the baseline model and comparable to the other models. This indicates that the improvements we made to the baseline model are effective, and our model has better detection performance compared to the other models.

Table 10 Detection results for each defect on the PCB-DET dataset.

Types	YOLOv7	YOLOv8	Faster R-CNN	RDD-YOLO	DsP-YOLO	PCP-YOLO
Mouse bite	83.4	99.5	80.1	93.1	92.6	99.5
Open circuit	76.9	84.7	65.5	91.8	94.3	94.3
Spur	73.3	95.4	79.3	89.4	93.4	95.9
Spurious copper	71.1	95.7	79.6	85.3	95.2	95.4
Missing hole	97.5	91.8	89.7	99.5	99.5	94.0
Short	99.4	96.9	99.8	99.0	99.5	97.6
Overall mAP	83.6	94.0	82.3	93.0	95.8	96.1

4.4.2 Ablation study

To further investigate the contributions of the PotentNet, C2f_ParallelPolarized, and CARAFE modules to the model's performance, we conducted ablation experiments on the NEU-DET dataset. The experimental results are shown in Table 11. As indicated in Table 11, the baseline model YOLOv8n achieved an mAP@0.5 of 76.7%, with 3.60 M parameters and an FPS of 169.2. When the PotentNet module was embedded in the backbone network based on the baseline model, the mAP@0.5 increased by 0.6%, but the number of parameters increased by 0.12M, and the FPS decreased by 27.0. Building on this, the neck network was modified by replacing the original nearest neighbor interpolation upsampling method with CARAFE upsampling, resulting in an mAP@0.5 increase to 78.2%, a parameter increase to 3.86M, and an FPS increase of 26.6. Further, by modifying the original C2f module in the neck network to the C2f_ParallelPolarized module with a polarization self-attention mechanism, a 1.2% improvement in mAP@0.5 was achieved, reaching 79.4%, with 3.87 M parameters and an FPS of 151.4. Compared to the other control groups in the ablation study, the PCP-YOLO model, although introducing more parameters, achieved the highest mAP@0.5, with an FPS comparable to the baseline model, meeting the dual requirements of high precision and real-time performance in industrial defect detection.

Table 11 The ablation experiments on NEU-DET.

Methods	mAP@0.5/%	Parameters	FPS
Baseline	76.7	3.60 M	169.2
+PotentNet	77.3	3.72 M	142.2
+CARAFE	77.8	3.14 M	143.5
+Polarized	77.3	3.01 M	170.0
+PotentNet+CARAFE	78.2	3.86 M	168.8
+PotentNet+C2f_ParallelPolarized	78.1	3.87 M	145.1
+CARAFE+C2f_ParallelPolarized	78.1	3.15 M	167.6
PCP-YOLO	79.4	3.87 M	151.4

4.4.3 Superiority verification of the developed PCP structure compared to other structures

To demonstrate the rationale of our proposed method, the study explored the embedding positions of the PotentNet module in the backbone network and the positions and quantities of the C2f_ParallelPolarized modules in the neck network. The experimental results are shown in Table 12. Additionally, to prove the superiority of our proposed method, we investigated the rationality of the C2f modification by embedding different attention modules in the C2f. The experimental results are presented in Table 13.

Table 12 Impact of adding same modules at different locations on model performance for the NEU-DET dataset.

Baseline	Add Module	Position	mAP@0.5%	Parameters	FPS
YOLOv8	PotentNet	P6	76.7	**3.18 M**	172.4
		P7	76.2	3.72 M	**179.2**
		P8	76.4	3.72 M	174.6
		P9	**77.3**	3.72 M	142.2
	C2f_ParallelPolarized	P13	77.0	3.02 M	165.8
		P16	77.2	3.01 M	185.6
		(P13,P16)	**77.3**	**3.01 M**	**170.0**

The research utilized the NEU-DET dataset to explore the impact of embedding the PotentNet module in the backbone network and the C2f_ParallelPolarized module in the neck network, both in terms of position and quantity. This was done by verifying the relevant metrics of the model with the same module added at different positions. The experimental results are shown in Table 12. In Table 12, the "Baseline" column represents the baseline model used, the "Add Module" column represents the added module, and the "Position" column represents the module's embedding position. In the "Position" column, "P6" indicates that the module was added to the 6th layer of the baseline model, and other values follow this rule. The initial purpose of adding the PotentNet module was to capture deep feature information in multi-scale defect features, so the PotentNet module was added to the higher layers of the backbone network. To avoid introducing a deeper network module, only one PotentNet module layer was introduced into the backbone network. According to the information in Table 12, when the PotentNet module is added to other positions in the backbone network, performance did not change significantly. However, embedding it in the 9th layer network resulted in the best mAP@0.5 performance, reaching 77.3%. The motivation for modifying the C2f was to enhance the model's ability to fuse features of small object defects in multi-scale defects under complex backgrounds. Therefore, when setting the position of the C2f_ParallelPolarized module, the focus was on the position used for detecting small objects in the detection head, resulting in two possible C2f modification positions: the 13th and 16th layers. According to Table 12, the experimental group with C2f_ParallelPolarized in both the 13th and 16th layers performed better, with the highest mAP@0.5, the lowest parameters, and the highest FPS, demonstrating good detection performance.

The research also compared adding SE, CBAM, CA, and ECA attention mechanisms to the C2f module in the neck network against the C2f_ParallelPolarized used in this study. As shown in Table 13, adding other self-attention mechanisms to the C2f did not improve network performance and instead resulted in varying degrees of decline. The experimental group with the C2f_ParallelPolarized module showed the best performance compared to the other control groups, achieving the highest mAP@0.5 of 77.3%, the fewest parameters, and an FPS comparable to the other models, thus demonstrating both lightweight and high-precision advantages.

Table 13 Detection performance of C2f embedded with various attention modules on NEU-DET dataset.

Baseline	Structure	mAP@0.5%	GFLOPs	Parameters	FPS
YOLOv8	C2f	76.7	8.1	3.60 M	169.2
	C2f+SE	75.2	8.1	3.62 M	184.9
	C2f+CBAM	73.3	8.1	3.60 M	171.6
	C2f+CA	73.8	8.1	3.71 M	173.3
	C2f+ECA	75.2	8.1	3.61 M	197.3
	C2f_ParallelPolarized	**77.3**	8.1	**3.01 M**	170.0

5 Conclusion

This article aims to address prominent issues in industrial defect detection tasks, exploring small object defect detection algorithms for multi-scale defects to ensure high accuracy and reasonable inference speeds. For this purpose, we propose the PCP-YOLO network based on YOLOv8. Drawing inspiration from the anchor-free framework of YOLOv8, we eliminate the influence of related hyperparameters such as anchor boxes, which have been a concern in previous studies. Firstly, we design and introduce a lightweight, non-deep feature extraction module, PotentNet, in the backbone network to enhance the capability of extracting fine-grained information about defects in images. Secondly, in the neck network, we design a feature fusion module with polarized self-attention, C2f_ParallelPolarized, which strengthens the model's ability to fuse features of small-sized defects in images from the perspective of polarized filtering and enhancing the dynamic range of attention. Then, we replace the original up-sampling module of the neck network with CARAFE to enhance the model's capability to utilize semantic information around points near features in images. Lastly, we apply these improvements to YOLOv8, enhancing the detection ability of small objects within multi-scale defects under complex backgrounds. Experimental results show that PCP-YOLO achieves mAP values of 79.4%, 77.6%, and 96.1% on the NEU-DET, GC10-DET, and PCB-DET datasets, respectively, which are 2.7%, 2.7%, and 2.4% higher than YOLOv8 and significantly surpass other tested models, especially in detecting small-sized defects under complex backgrounds. In terms of inference speed, our model ranks second among all tested models. The results indicate that this model meets the requirements of real-time industrial defect detection tasks and is beneficial for practical industrial applications.

In the future, we will delve deeper into issues encountered by the model in practical applications, such as irregular defects and training with small sample data. Additionally, effectively utilizing the prior knowledge associated with text and images to improve model performance is an important aspect to consider. We will also explore the use of technologies from other fields to enhance model performance, such as large language models and diffusion models.

References

[1] Luo, Q., Fang, X., Liu, L., Yang, C., Sun, Y.: Automated visual defect detection for flat steel surface: A survey. IEEE Transactions on Instrumentation and Measurement **69**(3), 626–644 (2020)

[2] Zhang, Y., Zhang, H., Huang, Q., Han, Y., Zhao, M.: Dsp-yolo: An anchor-free network with dspan for small object detection of multiscale defects. Expert Systems with Applications **241**, 122669 (2024)

[3] Dong, X., Zhang, C., Wang, J., Chen, Y., Wang, D.: Real-time detection of surface cracking defects for large-sized stamped parts. Computers in Industry **159**, 104105 (2024)

[4] Gao, Y., Gao, L., Li, X., Yan, X.: A semi-supervised convolutional neural network-based method for steel surface defect recognition. Robotics and Computer-Integrated Manufacturing **61**, 101825 (2020)

[5] Wang, R., Yu, H., Tang, J., Feng, B., Kang, Y., Song, K.: Optimal design of iron-cored coil sensor in magnetic flux leakage detection of thick-walled steel pipe. Measurement Science and Technology **34**(8), 085123 (2023)

[6] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., Berg, A.C.: Ssd: Single shot multibox detector. In: Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14, pp. 21–37 (2016). Springer

[7] Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 779–788 (2016)

[8] Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 580–587 (2014)

[9] Girshick, R.: Fast r-cnn. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1440–1448 (2015)

[10] Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. IEEE transactions on pattern analysis and machine intelligence **39**(6), 1137–1149 (2016)

[11] Tian, R., Jia, M.: Dcc-centernet: A rapid detection method for steel surface defects. Measurement **187**, 110211 (2022)

[12] Cao, Y., Pang, D., Zhao, Q., Yan, Y., Jiang, Y., Tian, C., Wang, F., Li, J.: Improved yolov8-gd deep learning model for defect detection in electroluminescence images of solar photovoltaic modules. Engineering Applications of Artificial Intelligence **131**, 107866 (2024)

[13] Su, B., Chen, H., Zhou, Z.: Baf-detector: An efficient cnn-based detector for photovoltaic cell defect detection. IEEE Transactions on Industrial Electronics **69**(3), 3161–3171 (2021)

[14] Li, L., Wang, Z., Zhang, T.: Gbh-yolov5: Ghost convolution with bottleneckcsp and tiny target prediction head incorporating yolov5 for pv panel defect detection. Electronics **12**(3), 561 (2023)

[15] Aboah, A., Wang, B., Bagci, U., Adu-Gyamfi, Y.: Real-time multi-class helmet violation detection using few-shot data sampling technique and yolov8. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5349–5357 (2023)

[16] Safaldin, M., Zaghden, N., Mejdoub, M.: An improved yolov8 to detect moving objects. IEEE Access (2024)

[17] Kou, X., Liu, S., Cheng, K., Qian, Y.: Development of a yolo-v3-based model for detecting defects on steel strip surface. Measurement **182**, 109454 (2021)

[18] Dong, H., Yuan, M., Wang, S., Zhang, L., Bao, W., Liu, Y., Hu, Q.: Pham-yolo: A parallel hybrid attention mechanism network for defect detection of meter in substation. Sensors **23**(13), 6052 (2023)

[19] Xu, L., Dong, S., Wei, H., Ren, Q., Huang, J., Liu, J.: Defect signal intelligent recognition of weld radiographs based on yolo v5-improvement. Journal of Manufacturing Processes **99**, 373–381 (2023)

[20] Lu, Q., Lin, J., Luo, L., Zhang, Y., Zhu, W.: A supervised approach for automated surface defect detection in ceramic tile quality control. Advanced Engineering Informatics **53**, 101692 (2022)

[21] Zhao, C., Shu, X., Yan, X., Zuo, X., Zhu, F.: Rdd-yolo: A modified yolo for detection of steel surface defects. Measurement **214**, 112776 (2023)

[22] Wang, Y., Wang, H., Xin, Z.: Efficient detection model of steel strip surface defects based on yolo-v7. Ieee Access **10**, 133936–133944 (2022)

[23] Yang, S., Wang, W., Gao, S., Deng, Z.: Strawberry ripeness detection based on yolov8 algorithm fused with lw-swin transformer. Computers and Electronics in Agriculture **215**, 108360 (2023)

[24] Xie, W., Sun, X., Ma, W.: A light weight multi-scale feature fusion steel surface defect detection model based on yolov8. Measurement Science and Technology (2024)

[25] Qian, X., Wang, X., Yang, S., Lei, J.: Lff-yolo: A yolo algorithm with lightweight feature fusion network for multi-scale defect detection. IEEE Access **10**, 130339–130349 (2022)

[26] Ling, Q., Isa, N.A.M., Asaari, M.S.M.: Precise detection for dense pcb components based on modified yolov8. IEEE Access (2023)

[27] Liu, Z., Abeyrathna, R.R.D., Sampurno, R.M., Nakaguchi, V.M., Ahamed, T.: Faster-yolo-ap: A lightweight apple detection algorithm based on improved yolov8 with a new efficient pdwconv in orchard. Computers and Electronics in Agriculture **223**, 109118 (2024)

[28] Li, Y., Fan, Q., Huang, H., Han, Z., Gu, Q.: A modified yolov8 detection network for uav aerial image recognition. Drones **7**(5), 304 (2023)

[29] Ding, X., Zhang, X., Ma, N., Han, J., Ding, G., Sun, J.: Repvgg: Making vgg-style convnets great again. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 13733–13742 (2021)

[30] Liu, H., Liu, F., Fan, X., Huang, D.: Polarized self-attention: Towards high-quality pixel-wise regression. arXiv preprint arXiv:2107.00782 (2021)

[31] Wang, J., Chen, K., Xu, R., Liu, Z., Loy, C.C., Lin, D.: Carafe: Content-aware reassembly of features. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 3007–3016 (2019)

[32] Lin, T.-Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2980–2988 (2017)

[33] Ge, Z., Liu, S., Wang, F., Li, Z., Sun, J.: Yolox: Exceeding yolo series in 2021. arXiv preprint arXiv:2107.08430 (2021)

[34] Wang, C.-Y., Bochkovskiy, A., Liao, H.-Y.M.: Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7464–7475 (2023)

[35] Liu, R., Huang, M., Gao, Z., Cao, Z., Cao, P.: Msc-dnet: An efficient detector with multi-scale context for defect detection on strip steel surface. Measurement **209**, 112467 (2023)

[36] Zhang, D., Hao, X., Liang, L., Liu, W., Qin, C.: A novel deep convolutional neural network algorithm for surface defect detection. Journal of Computational Design and Engineering **9**(5), 1616–1632 (2022)