IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 5 July 2024, accepted 30 August 2024, date of publication 3 September 2024, date of current version 13 September 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3453931

APPLIED RESEARCH

Steel Surface Defect Detection Method Based on Improved YOLOv9 Network

JIALIN ZOU[1] AND HONGCHENG WANG[2]
[1]School of Computer Science and Technology, Dongguan University of Technology, Dongguan 523808, China
[2]School of Electrical Engineering and Intelligentization, Dongguan University of Technology, Dongguan 523808, China

Corresponding author: Hongcheng Wang (wanghc@dgut.edu.cn)

This work was supported in part by Dongguan Science and Technology of Social Development Program under Grant 20231800940532, and in part by Songshan Lake Sci-Tech Commissioner Program under Grant 20234373-01KCJ-G.

ABSTRACT This paper introduces a new steel defect detection model CK-NET, which uses YOLOv9c as the baseline model and adopts YOLOv9's model architecture for improvement. The proposed model addresses issues of shallow information loss and insufficient feature extraction and fusion caused by network deepening. A new feature extraction module is designed to control model parameters and enhance feature extraction. Minor improvements to Convolutional Block of Attention Module (CBAM) and the introduction of deformable convolutions in the backbone network further enhance feature extraction. A new feature fusion module, combined with a self-attention mechanism (SA), elegantly fuses features from different levels to assist downstream detection tasks. The Programmable Gradient Information (PGI) auxiliary branch is also improved to better fuse features and guide model learning with gradient information. All improved modules have been integrated into CK-NET. Experiments on the NEU-DET dataset demonstrate that CK-Net achieves a 13.2% higher mAP value than YOLOv9c, reaching a 92.1% mAP value while maintaining similar model parameters, validating the model's effectiveness.

INDEX TERMS Steel surface defect detection, YOLOv9, feature extraction and fusion.

I. INTRODUCTION

Steel, an alloy primarily composed of iron and carbon, has played a crucial role in modern civilization's infrastructure. Its exceptional strength, durability, and versatility make it a vital material for applications in construction, transportation, manufacturing, equipment, and machinery. The reliability of steel directly impacts the safety and longevity of structures and products. Surface quality significantly influences steel properties like fatigue strength, corrosion resistance, and structural integrity. Defects such as cracks, scratches, pits, and inclusions can arise during various production stages, posing risks to steel performance and safety. Therefore, maintaining high surface quality is essential for economic and safety reasons in the steel industry. Traditional methods for detecting surface defects rely on subjective visual inspection, leading to inconsistent results due to human factors. The task involves classifying and locating defects, with machine learning methods increasingly being utilized for industrial defect detection. Zeng et al. [1] employed machine learning methods to categorize surface defects on wafers. The algorithms used in machine learning extract important edge features, surface textures, and other image details from the images collected for processing and generating image recognition outcomes. Although these techniques can decrease the need for manual work, they are still heavily reliant on manual feature extraction, which poses limitations. The rise of deep learning has resulted in significant progress across various industries [2], particularly in the field of industrial quality control. Deep learning, a subset of machine learning, employs artificial neural networks with multiple layers to analyze data in a hierarchical manner. In the context of identifying defects on steel surfaces, deep learning algorithms have the ability to automatically recognize complex defect patterns from images, providing a dependable and unbiased detection method.

The associate editor coordinating the review of this manuscript and approving it for publication was Wei-Yen Hsu.

In this study, a novel steel surface defect detection model named CK-Net is proposed to address existing issues in current methods. The baseline model for CK-Net is YOLOv9c [3], which has been enhanced with Programmable Gradient Information (PGI) auxiliary branches to reduce forward information loss. The YOLOv9 series algorithm has shown superior performance compared to other baseline models. Due to the complexity of steel data, extracting comprehensive features is a challenging task for models. To tackle this challenge, a new feature extraction module called CK-GELAN, based on YOLOv9's GELAN architecture, is introduced. CK-GELAN consists of multiple modules that sequentially extract features, with the features of each layer being added in parallel. Additionally, three smaller modules within the feature extraction module utilize depthwise separable convolution to manage model complexity while maintaining feature extraction capacity. Furthermore, the width of feature extraction module has been increased. CK-GELAN not only enhances the depth of the backbone network to extract more semantic information but also broadens the model's width to capture more spatial details. This study incorporates deformable convolutions [4] and Convolutional Block of Attention Module (CBAM) [5]in both the backbone network and PGI auxiliary branches. The spatial attention module of CBAM is adjusted to improve the feature extraction capability of the backbone network. Additionally, a feature fusion module, CK-FFM, is developed to address the issue of inadequate feature fusion by combining feature maps from different levels and incorporating a self-attention mechanism within the skip connections of multi-level feature connections. This module is also integrated into the PGI auxiliary branch. The PGI branch structure of YOLOv9 is modified to include a feature fusion network that performs unidirectional fusion of learned features. Unlike traditional models, this paper focuses on utilizing a bidirectional feature fusion architecture to integrate richer semantic and spatial features, aiming to enhance the accuracy of backward gradient propagation parameters and improve the learning process of the backbone network.

II. RELATED WORK

This section provides a comprehensive review of the existing literature on deep learning applications in classifying and detecting steel surface defects and industrial defects.

He et al. [6] proposed an end-to-end strategy for defect detection on steel surfaces. They introduced a multilevel feature fusion network (MFN) to combine different hierarchical features into a single feature. The authors also introduced the DF-ResNeSt50 network model [7], which incorporates the visual attention mechanism inspired by the bionic algorithm. This model merges the feature pyramid network and split-attention network, optimizing them through techniques such as data augmentation, multi-scale feature fusion, and network structure enhancements. These improvements resulted in enhanced detection performance and
efficiency.

Zou et al. [8] improved the YOLOv5 network model by integrating an attention mechanism to identify and extract more important features, reducing algorithm errors and enhancing model accuracy. Lin et al. [9] utilized convolutional neural networks for LED chip defect detection. Chen et al. [10] introduced a lightweight convolutional neural network model for wafer defect classification, incorporating multiple 1×1 convolutional kernels to increase channel numbers. To reduce parameters, they included a global average pooling layer and depthwise separable convolution. Lv and Xu [11] integrated an SE attention mechanism into the YOLOv5 network for improved target attention and replaced the original SPPF module with an SPPFCSPC module to enhance feature processing. This technical advancement significantly enhances precise real-time detection in industrial steel plate production.

Li et al. [12] proposed the MSFE feature extraction module and the EFF feature fusion module, which leverage depthwise separable convolution and residual structure to extract more comprehensive features while keeping model parameters in check. Huang et al. [13] employed the K-means++ algorithm to re-cluster the steel dataset and generate appropriate bounding boxes. They also integrated a deformable convolutional module into the backbone network to improve feature extraction capabilities. Furthermore, the CBAM module was added to dynamically extract regions of interest within the network. The use of the Focal EIOU loss function during model training was intended to tackle imbalanced samples, although its efficacy was somewhat limited.

Song et al. [14] proposed an improved Faster-RCNN method that incorporates deformation convolution and ROI alignment to detect surface defects on steel plates. This approach notably boosts detection performance for large-scale defects with complex and irregular shapes. Furthermore, a new background suppression technique was introduced to enhance discrimination between foreground and background, tackling the challenge of low detection accuracy caused by the similarity between defect and normal areas on steel plate surfaces. However, despite the improved detection accuracy, the utilization of multiple variable convolutions resulted in a decrease in forward inference speed and an expansion of network parameters. Imoto et al. [15] introduced a method based on CNN and transfer learning, utilizing the inception model for the automatic classification of defects. The method primarily concentrates on analyzing defects to ascertain the causes of yield reduction, drawing upon the outcomes of defect classification. Yeung and Lam [16] proposed a fusion attention network, in which the attention mechanism was applied to a single balanced feature map to improve the accuracy of the detection network and maintain the detection speed. Kou et al. [17] designed an anchor-free network with dense convolutional blocks for steel surface defect detection.

III. PROPOSED METHOD

A. CK-GELAN FEATURE EXTRACTION MODULE

The CK-GELAN feature extraction module is an improved design based on the architecture of the feature extraction module in the YOLOv9 network. This article has made a series of improvements to the structure of internal modules. As the depth of the feature extraction network increases, it becomes capable of extracting more semantic information. However, there is a risk of losing spatial details. To address this, widening the network alongside deepening it can aid in capturing detailed texture features and mitigating gradient vanishing issues to some extent. It is important to note that overly deep networks can result in a large number of model parameters. Hence, choosing the right feature extraction module is essential. This article introduces CK-GELAN, a novel feature extraction module inspired by the architecture of YOLOv9's GELAN module. CK-GELAN strikes a fine balance between feature extraction capacity, network depth and width, and model complexity. This enables the model to extract comprehensive features while keeping the parameter count in check. Additionally, the internal feature fusion and stitching within the module help mitigate information loss during forward propagation. Its main function is to enhance the model's ability to extract image features, appearing as a feature extraction module in every part of the network. Figure 1 illustrates the feature extraction architecture of CK-GELAN.

Figure 1 (a) depicts the architectural design of CK-GELAN, featuring two primary branches. The left branch initiates with a 3 × 3 convolution operation (Convolution, BN, and SiLU) on the input feature map for enhanced feature extraction, doubling the channel count without downsampling. Subsequently, a 1 × 1 convolution reduces the channel number. Assuming the initial input feature map F has dimensions of H×W×C, post the initial convolution, F transitions to H×W×2C. Following the 1×1 convolution, F reverts to its original dimensions of H×W×C, ensuring effective feature extraction while retaining control over feature map dimensionality for subsequent operations. In the input feature map F, a 3×3 convolution is initially applied for feature extraction, altering the dimension to H×W×2C. The feature map then progresses through a cascaded module comprising b1, b2, and b3. The outcomes of these modules are merged to generate the final feature map of the right branch. Ultimately, the feature maps from the last two branches are concatenated in the channel dimension to yield the output of a CK-GELAN feature extraction module. The cascading module encompasses three branches: b1, b2, and b3. Both b1 and b2 branches yield two outputs each – one for feature addition and the other as input for the subsequent module. The third branch, b3, generates a single output. The resultant feature map size of the cascading module is H×W×C. Upon concatenating the outputs of the two branches in the channel, the output feature map size of CK-GELAN expands to H×W×2C. The computation of CK-GELAN is as follows:

$$F' = Concat(Conv_{1\times1}\left(Conv_{3\times3}\left(x\right)\right), x_1 + x_2 + x_3), \quad (1)$$

$$x_1 = b_1(Conv_{3\times3}(x)), \quad (2)$$

$$x_2 = b_2(x_1), \quad (3)$$

$$x_3 = b_3(x_2), \quad (4)$$

where x represents the input feature map F, and F' represents the output result of CK-GELAN. $Conv_{1\times1}$ denotes a 1 × 1 convolution operation, while $Conv_{3\times3}$ denotes a 3 × 3 convolution operation. x_1, x_2, and x_3 represent the output results of cascade modules b1, b2, and b3, respectively, as shown in (2), (3), and (4). b_1, b_2, and b_3 represent the processing of modules b1, b2, and b3, respectively. *Concat* represents concatenation in the channel dimension. $Conv_{n\times n}$ contains n×n convolution, BN, and SiLU activation functions.

Figures 1 (b), (c), and (d) illustrate schematic diagrams of modules b1, b2, and b3, respectively. In Figure 1 (b), F represents the input feature map. The initial convolution module in the right branch of CK-GELAN adjusts the feature map size to H×W×2C. Following this, the 1 × 1 convolution within module b1 alters the feature map size to H×W×C. The right branch of b1 features a structure that combines two 3×3 depthwise separable convolutions [18] and one 1×1 convolution. Depthwise separable convolution aids in feature extraction, while the 1 × 1 convolution reduces the number of channels to achieve a feature map size of H×W×C, which is advantageous for subsequent addition operations. The use of 1 × 1 convolution for channel compression ensures consistency in the feature map size of internal modules, simplifying subsequent operations. The output of b1 has two branches. The first part will be temporarily stored for addition to the outputs of b2 and b3. The other part serves as input for b2. In Figure 1 (c), the input feature map F undergoes a 3 × 3 convolution to extract features and increase the number of channels, resulting in a feature map dimension of H× W×2C. The subsequent structure follows the b1 module, with the left side utilizing a 1×1 convolution to reduce the number of channels while preserving the original features. On the right side, a combination of 3×3 depth separable convolution and 1 × 1 convolution is used for feature extraction and channel control. Finally, an addition operation is performed to generate the output of b2.

The role of b1 and b2 is for conventional feature extraction. b2 has the same feature extraction structure as b1 after the convolutional compression channel, and the effect of overlaying two similar structures to extract features improves the network. Figure 1 (d) illustrates the structural diagram of b3, which sets itself apart from b1 and b2 by featuring three branches to enhance the network structure and improve the comprehensiveness of the extracted information. The input feature map F of b3 is derived from the output of b2. F undergoes feature extraction via a 3 × 3 convolution, followed by channel control using a 1 × 1 convolution. The remaining two branches utilize depthwise separable convolution and 1 × 1 convolution for feature extraction. Subsequently, the extracted features from both branches are concatenated in the channel dimension and added to the output of the left branch. The resulting output feature map has dimensions H×W×C.

Notably, the middle branch employs a 3 × 3 depth separable convolution, while the right branch employs a 5 × 5 depth separable convolution to expand the network's receptive field, integrate features from various receptive fields, and mitigate the loss of texture features for smaller targets.

The main purpose of submodule b3 is to avoid insufficient feature extraction for b1 and b2. At the same time, b3 also has a 5 × 5 depth separable convolution, which appropriately enhances the receptive field of the model and enhances its feature extraction ability. The combination of three submodules can effectively extract the features of images, allowing the network to fully utilize its ability to analyze defect features. b1, b2 and b3 are computed as:

$$x_1 = Conv_{1\times1}\left(x'\right) + Conv_{1\times1}(DConv_{3\times3}(DConv_{3\times3}(x')), \tag{5}$$

$$x' = Conv_{3\times3}(x), \tag{6}$$

$$x_2 = Conv_{1\times1}\left(Conv_{3\times3}\left(x_1\right)\right) + Conv_{1\times1}(DConv_{3\times3}(DConv_{3\times3}(Conv_{3\times3}(x_1)))), \tag{7}$$

$$x_3 = Conv_{1\times1}\left(Conv_{3\times3}\left(x_2\right)\right) + Concat(Conv_{1\times1}(DConv_{3\times3}\left(DConv_{3\times3}\left(x_2\right)\right)), Conv_{1\times1}(DConv_{5\times5}(DConv_{5\times5}(x_2)))), \tag{8}$$

where x' in (5) is fed into module b1, which is defined by (6). $DConv_{3\times3}$ refers to a depthwise separable convolution using a 3 × 3 kernel size, and a separate instance of $DConv_{5\times5}$ uses a 5 × 5 kernel size.

The feature extraction module CK-GELAN showcases robust capabilities by incorporating three cascaded modules. Its structured approach ensures seamless integration between levels, minimizing the risk of information loss and leading to a more stable feature extraction process in the short term. CK-GELAN excels in both depth and width, offering multi-scale feature extraction that efficiently merges features from different receptive fields, making it a superior module for feature extraction.

B. THE IMPROVED CBAM ATTENTION MODULE IN BACKBONE NETWORK

The accuracy of defect detection in steel data is hindered by its low resolution. To address this issue, CBAM [5] was incorporated into the backbone network for enhanced feature extraction. Making appropriate adjustments to CBAM and applying it to the backbone network of CK-NET for feature extraction will promote the effectiveness of steel defect detection. CBAM comprises two main components: the channel attention module and the spatial attention module. By combining spatial and channel attention mechanisms, CBAM dynamically extracts features, directing the model's focus to regions of interest. This improves the network's capability to concentrate on crucial information, ultimately enhancing network performance. This approach is particularly beneficial for analyzing low-resolution or poor-quality images, such as steel data, as it significantly boosts feature recognition and prevents inadequate defect feature extraction resulting

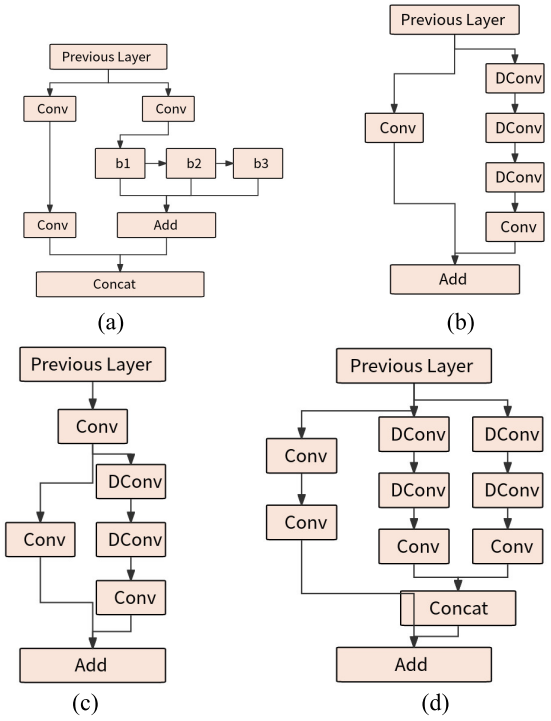

FIGURE 1. (a) The diagram of CK-GELAN. (b) b1 in CK-GELAN. (c) b2 in CK-GELAN. (d) b3 in CK-GELAN.

from focusing solely on a small subset of defects. Through the sequential integration of channel and spatial attention, CBAM helps networks effectively prioritize essential features and spatial regions. The channel attention mechanism and spatial attention mechanism of CBAM are mathematically represented in (9) and (10).

$$M_C\left(F\right) = \sigma(MLP\left(AvgPool\left(F\right)\right) + MLP(MaxPool(F))), \tag{9}$$

$$M_S\left(F'\right) = \sigma(Conv_{7\times7}(AvgPool\left(F'\right); MaxPool(F'))), \tag{10}$$

where F represents the input feature map, $AvgPool$ and $MaxPool$ refer to mean pooling and maximum pooling, respectively. MLP represents a basic neural network, σ denotes the sigmoid activation function, and M_C represents the output feature map of the channel attention module. The feature map F' is refined through the channel attention mechanism. $Conv_{7\times7}$ denotes a convolution operation using a 7 × 7 kernel, σ refers to the sigmoid activation function, and M_S represents the output of the spatial attention mechanism.

In this paper, the 7 × 7 convolution kernel is replaced with 3 × 3, 5 × 5, and 1 × 1 convolution kernels, while enhancing the spatial attention module of CBAM. This modification not only improves parameter efficiency and computational performance but also enhances the model's expressive power. The use of smaller convolutional kernels reduces the number of parameters in each layer, allowing for the creation of deeper and lighter networks while reducing the risk of

overfitting. Substituting a 7 × 7 convolution kernel with smaller ones not only reduces parameter count but also increases network depth, enhancing its learning capacity.

Using multiple smaller convolution kernels to replace a larger one, the most obvious feature is the addition of an activation layer. As the activation layer increases, the model's ability to fit nonlinear data improves, and the accuracy of data learning also improves.

Stacking these smaller convolutional layers enables the network to learn more complex mappings and improve its expressive power by applying nonlinear activation functions after each layer. This method effectively deepens network complexity without significantly increasing computational costs. By utilizing smaller convolutional kernels, the network can expand its receptive field while maintaining a low parameter count, thereby improving computational efficiency. This flexible approach to convolution operations facilitates the development of network architectures with higher computational efficiency and adaptability to various task requirements. The use of smaller convolution kernels, instead of larger ones, can enhance network performance and efficiency by increasing nonlinearity, reducing parameters, and improving the network's learning capabilities. This strategy has been widely implemented in efficient convolutional network architectures like VGG [19], GoogLeNet [20], and ResNet [21]. The improved expression for the spatial attention module is given by (11).

$$M_S\left(F'\right) = \sigma(Conv_{1\times1}(Conv_{5\times5}(Conv_{3\times3}\left(AvgPool\left(F'\right);\right. \\ \left. MaxPool\left(F'\right)\right))), \tag{11}$$

CBAM enhances the model's representation ability by improving attention towards important feature channels and spatial regions, allowing the network to extract richer and more discriminative features. As a module, CBAM can be seamlessly integrated into existing convolutional networks, boosting network performance without a significant increase in computational costs. Its adaptive adjustment of channel and spatial weights based on input features gives CBAM strong generalization ability and applicability. By introducing a focused attention mechanism in critical areas of the model, CBAM enables the network to efficiently and accurately recognize and process important information, leading to a significant improvement in model performance. In this article, CBAM has been repeatedly utilized to incorporate attention mechanisms and enhance the model's feature extraction capabilities. The backbone network of CK-Net is illustrated in Figure 2.

CK-GELAN4 represents the continuous stacking of four CK-GELAN feature extraction modules. Three feature maps, namely C1, C2, and C3, were extracted from the backbone network of CK-Net for subsequent multi-scale feature map detection. To enhance the network's attention and improve information capture on steel defects, CBAM is added before the feature extraction module each time a multi-scale feature map is extracted. Additionally, a deformable convolution

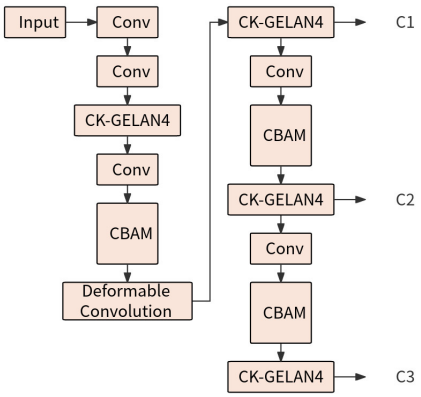

FIGURE 2. CK-Net's backbone network.

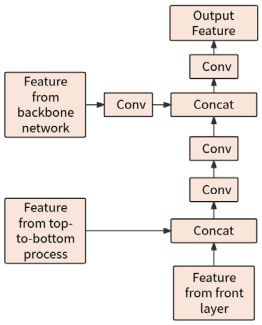

FIGURE 3. The diagram of CK-FFM.

module [4] is included after the first CBAM, enhancing the standard convolution's capabilities by incorporating additional offsets to adapt to geometric changes in the input feature map. This deformable convolution mechanism allows for more flexible capture of irregular shapes and objects in images. In this study, the use of multiple deformable convolution modules did not show a significant improvement in feature extraction but rather increased the model parameters. However, when using a single deformable convolution module, both the feature extraction effectiveness and model parameter increase were found to be significant [13].

C. THE FEATURE FUSION MODULE OF CK-Net

Adding new feature fusion modules to the network can enhance the effectiveness of feature detection. The fusion of multi-layer features can not only preserve the original features of the image, but also fuse the features extracted from deep features. A novel feature fusion module has been introduced to tackle the issue of significant information discrepancy that arises from feature information loss during extraction from the backbone network and fusion from the feature fusion network. This module effectively combines different sequences and improves feature fusion for subsequent detection tasks. Figure 3 illustrates the CK-FFM feature fusion module proposed in this study.

CK-FFM combines feature maps from three dimensions: the feature map from the previous layer of CK-FFM, the same

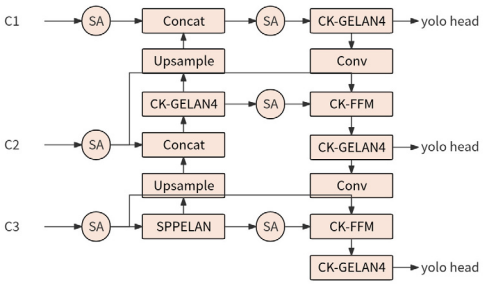

FIGURE 4. CK-Net's neck network.

level feature map sampled from top to bottom in the feature fusion network, and the same level feature map in the backbone network. The front layer feature map and the top-down sampled feature map are concatenated in the channel dimensions and then undergo a 3×3 convolution operation to extract features from the fused feature map. Next, a 1×1 convolution compresses the channel dimension and fuses it with the backbone network feature map processed by a 3×3 convolution, enriching the feature map with multi-dimensional features. Finally, the output features are obtained by reducing the number of channels through a 1×1 convolution.CK-FFM is computed as:

$$\begin{aligned} &CK-FFM\left(F_1,F_2,F_3\right)\\ &=Conv_{1\times1}(Concat\left(Conv_{3\times3}\left(F_3\right),\right.\\ &\quad\left.Conv_{1\times1}\left(Conv_{3\times3}\left(Concat\left(F_1,F_2\right)\right)\right)\right)), \end{aligned} \tag{12}$$

where F_1, F_2, and F_3 are the feature maps of the previous layer of CK-FFM, the same level feature maps sampled from top to bottom in the feature fusion network, and the same level feature maps in the backbone network, respectively.

In this article, the CK-FFM feature fusion module is utilized in the bottom-up feature fusion component of the neck network. Figure 4 depicts a schematic diagram of the CK-Net's neck network.

Following the passage through the backbone network, C1, C2, and C3 are integrated in the neck network to merge features of different scales before being processed by the conventional YOLO decoupling detection head. The network's skip connection structure includes a self-attention mechanism (SA) to blend C1, C2, and C3 within the neck network and enhance the connection between the top-down and bottom-up sections of the network. This self-attention mechanism aids in maintaining information integrity and partially addresses the issue of gradient vanishing. In the up-sampling phase of the neck network, all components, except for the feature extraction module, follow a structure similar to the YOLOv9 network. However, CK-Net introduces a novel feature extraction module CK-FFM and an attention mechanism (SA). During the downstream feature fusion process, the combined feature maps processed by CK-FFM and SA contain rich network information, effectively extracting detailed features of steel defects, which is advantageous for subsequent classification and localization tasks.

D. THE IMPROVED PGI AUXILIARY BRANCH

As the depth of the backbone network increases, there is a risk of losing original information as network features evolve. This can result in a significant deviation in loss during reverse gradient updates, impacting network learning effectiveness. To address this issue, YOLOv9 introduced Programmable Gradient Information (PGI) [3]. PGI involves extracting shallow features from feature maps to provide accurate gradient information for network learning, preventing bias and mitigating shallow feature loss. By integrating multi-scale feature maps C1, C2, C3 with shallow features from the PGI branch, a feature fusion network combines relevant features for detection tasks. Failure to fuse features from different scales may lead to misclassification of large objects as background, highlighting the importance of incorporating multi-level features. However, increasing the number of fused features may hinder the efficiency of multi-layer feature merging in the PGI network.

The PGI auxiliary branch can help the model focus on the region that is most likely to contain the target by predicting candidate regions, thereby reducing unnecessary calculations. This approach can to some extent reduce the computational complexity of the detection model and accelerate the inference process. It can locate and identify target objects more accurately through additional predictive information. In PGI, candidate regions are first generated through the backbone detection network. These candidate regions are usually areas with potential targets in the image, rather than covering the entire image.

The PGI auxiliary branch receives feature maps from the backbone detector as input. These feature maps typically contain semantic and positional information about each region in the image. The task of PGI auxiliary branches is to further evaluate and screen these candidate regions. It may apply some Region Proposal Network (RPN) or similar methods to generate confidence scores or bounding box predictions for candidate regions. The confidence score or bounding box prediction generated by PGI assisted branches can help the main detector filter out areas that are unlikely to contain real targets, thus focusing on the areas that are most likely to contain real targets.

This filtering can reduce the number of regions for subsequent processing and improve the efficiency and accuracy of the subsequent object detection module. The PGI auxiliary branch not only accelerates the inference speed of the model but also helps to improve the accuracy of detection by introducing additional candidate region generation and screening mechanisms.

This article proposes enhancements to the PGI module to improve feature fusion, as illustrated in Figure 5.

CBAM and deformable convolution are integrated into the shallow feature extraction module of PGI to improve feature extraction, similar to CK-Net's backbone network. The output from shallow feature extraction is combined with three multi-scale features from the SA backbone network, with the attention mechanism aiming to prevent gradient

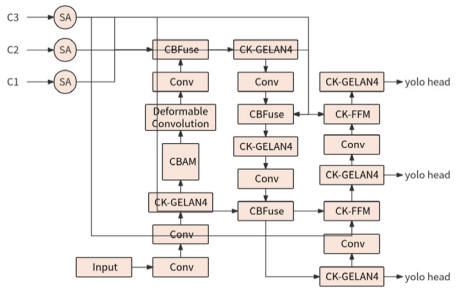

FIGURE 5. Improved PGI auxiliary branch.

disappearance. Following YOLOv9's PGI approach, the fused features undergo further extraction and fusion with semantically strong feature maps C2 and C3. A feature fusion module called CK-FFM is added after the original PGI module to enhance feature fusion within the branch, inspired by CK-Net's neck network. The CK-FFM module adjusts channel numbers on a mixed feature map using a 1×1 convolution, improving the mixed features of PGI. Finally, the YOLO decoupling detection head is utilized for detecting the mixed features, providing gradient information, and aiding in network learning. The key enhancement in the improved PGI is the expansion of the feature fusion component from one branch to two branches. It is worth noting that single branch feature fusion networks have limited fusion capabilities, while dual branch feature fusion can comprehensively merge mixed features, which is crucial for subsequent detection tasks [22].

IV. EXPERIMENTAL DATASET

The study utilizes the NEU-DET [6] dataset for detecting defects on steel surfaces, which includes six categories: crazing, inclusion, patches, pitted_surface, rolled-in_scale, and scratches. Each category contains 300 images, making a total of 1800 images. The dataset was divided into three subsets - training set, validation set, and test set - in an 8:1:1 ratio. Figure 6 displays the defect data for the six types in the NEU-DET dataset.

Data augmentation [23] is utilized to improve the model's ability to learn challenging target regions. The distribution of samples per class post data augmentation is illustrated in Figure 7. Targeted data augmentation on challenging samples can effectively mitigate training challenges arising from the varying degrees of difficulty in learning samples.

V. EXPERIMENTAL RESULTS AND ANALYSIS

A. EVALUATION CRITERIA

This study assesses the model performance using metrics such as Parameters (Params), Precision (P), Recall (R), and mAP. Parameters indicate the number of model parameters, with a higher count leading to slower inference speed, while fewer parameters result in faster inference speed, making the model more lightweight. Precision reflects the accuracy of the model's defect classification, while Recall measures the model's ability to comprehensively detect defects, reducing the chances of missed detections. The mAP value serves as a crucial indicator for evaluating object detection model performance. It represents the mean average precision, balancing Precision and Recall for each category detected by the model. A higher mAP value indicates superior model performance. The formulas for Precision, Recall, and mAP are as follows:

$$Precision = \frac{TP}{TP + FP}, \tag{13}$$

$$Recall = \frac{TP}{TP + FN}, \tag{14}$$

$$AP_i = \int_0^1 P(R)d(R), \tag{15}$$

$$mAP = \frac{1}{N}\sum_{i=1}^{N} AP_i. \tag{16}$$

In academic research, TP stands for True Positive, where the predicted value is 1 and the actual value is 1, indicating a correct prediction. FP stands for False Positive, where the predicted value is 1 but the actual value is 0, indicating an error in prediction. FN stands for False Negative, where the predicted value is 0 but the actual value is 1, also indicating an error in prediction. The paper calculates the Average Precision (AP) as a comprehensive indicator of Precision and Recall for each category, and then computes the mean Average Precision (mAP) as the arithmetic mean of AP. The article mentions that there are a total of six categories (N=6) being considered.

B. COMPARATIVE EXPERIMENT

Utilizing Faster-RCNN [24], YOLOv8s [25], YOLOv9c, and YOLOv9e [3] as comparative experimental models alongside the CK-Net model, this study conducted comparative experiments on the NEU-DET dataset for 300 epochs with an input size of 640×640 for each image. The performance comparison of the different models is presented in Table 1.

The comparison of various object detection models highlights that the Faster-RCNN with ResNet50 backbone has the highest number of parameters, while YOLOv8s has the fewest. YOLOv9c exhibits the lowest precision, whereas YOLOv9e demonstrates the lowest recall. In contrast, the CK-Net model in this study showcases superior precision and recall. YOLOv8s surpasses Faster-RCNN by 1.6 percentage points in mAP, despite having significantly fewer parameters. YOLOv9c, with more than double the parameters of YOLOv8s, achieves a 2.4 percentage points higher mAP. The model parameter count of YOLOv9e is much higher than that of YOLOv9c. Although the mAP value of the former is slightly higher than that of the latter, the magnitude of the change is not significant, and the recall rate of YOLOv9c is 6.3% higher than that of YOLOv9e.The findings suggest that as models become more complex, the improvement in mAP becomes limited, indicating a plateau in the model's learning capacity for steel defect data. The paper also delves into discussions on data augmentation and structural enhancements that have further boosted the model's learning capacity.

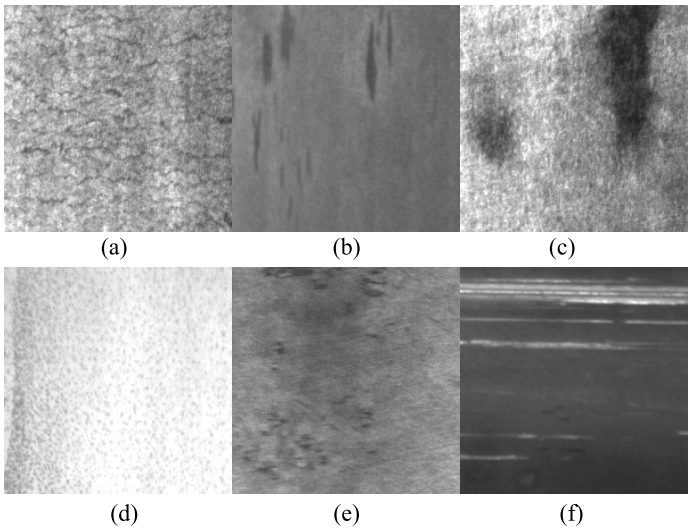

FIGURE 6. Six defects of NEU-DET dataset: (a) crazing, (b) inclusion, (c) patches, (d) pitted_surface, (e) rolled-in_scale, (f) scratches.

TABLE 1. Performance comparison of different models.

Model	Params (M)	Precision (%)	Recall (%)	mAP (%)
Faster-RCNN(ResNet50)	230	72.3	72.5	74.9
YOLOv8s	11.2	74.6	72.7	76.5
YOLOv9c	25.5	71.6	78.3	78.9
YOLOv9e	58.1	77.4	72	79
CK-Net (Based on YOLOv9c)	24.2	90.6	83.8	92.1

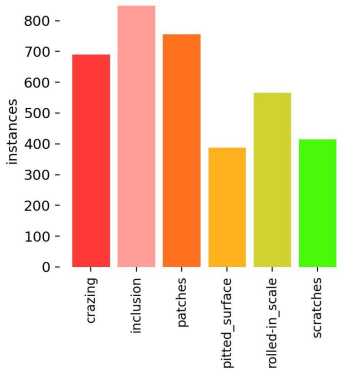

FIGURE 7. Number of samples for each category after data augmentation.

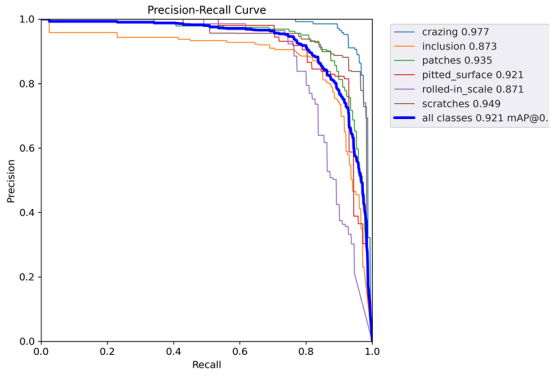

FIGURE 8. P-R curve of CK-Net.

Notably, the CK-Net model achieved an impressive mAP of 92.1% on the NEU-DET dataset in this study.

Figure 8 illustrates the precision-recall curve throughout the CK-Net training process, highlighting the variations in precision and recall for each category over time, along with a comparison of training outcomes across different categories.

The model's performance is evaluated based on a combination of precision and recall, with higher values indicating better performance. After data augmentation on challenging samples like crazing and pitted surfaces, CK-Net demonstrated strong performance in both categories, achieving AP values exceeding 90%. Among the six data categories, the model performed least effectively in the rolled-in_scale

TABLE 2. Results of ablation experiment.

Model	Params(M)	mAP(%)
a).YOLOv9c	25.5	78.9
b).a + CK-GELAN	22.3	80.1
c).b+Improved CBAM	22.4	81.2
d).c+Deformable Convolution	23	82.9
e).d+CK-FFM with SA	23.5	83.3
f).e+Improved PGI(CK-Net)	24.2	83.8
g).f+data augmentation	24.2	92.1

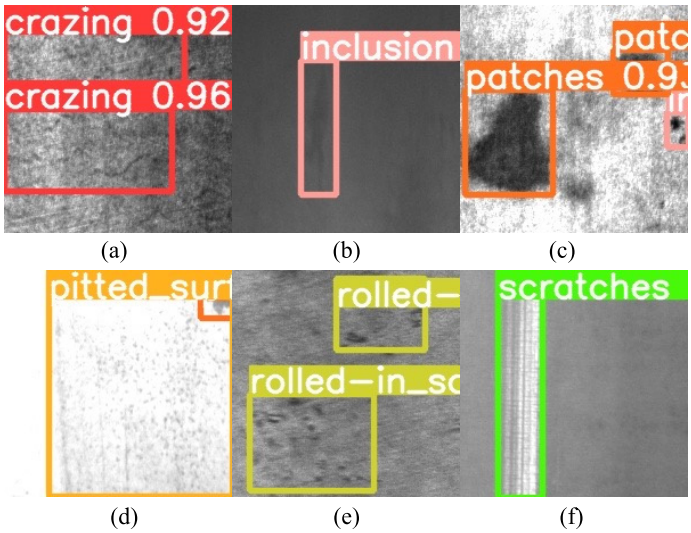

FIGURE 9. Examples of detection for six defect categories: (a) crazing, (b) inclusion, (c) patches, (d) pitted_surface, (e) rolled-in_scale, (f) scratches.

category, with an AP value of 87.1%. The model excelled in the crazing category, achieving an AP value of 97.7%. It is worth noting that only the inclusion and rolled-in scale categories had AP values slightly below 90%, while the other categories showed very high values. Overall, the model's training results were highly favorable.

C. ABLATION EXPERIMENT

A series of ablation experiments were carried out on the baseline model YOLOv9c in this study to showcase the efficacy of the model enhancements. The primary comparative metrics assessed included the model parameters count and mAP. The findings of the ablation experiment are detailed in Table 2.

The reason why YOLOv9c was chosen as the baseline model instead of YOLOv9e in this article is that although YOLOv9e has higher model precision than YOLOv9c, its mAP value is only 0.1% higher than YOLOv9c in the performance of the steel defect detection dataset in this paper. At the same time, the parameter count of YOLOv9e is much higher than that of YOLOv9c. This article adheres to the concept of appropriately improving the detection accuracy of the model while minimizing the number of control model parameters, and therefore chooses YOLOv9c as the baseline model for this article.

Utilizing CK-GELAN as a feature extraction module alongside YOLOv9c results in a reduction of 3.2M parameters. This study integrates a significant number of depthwise separable convolutions within CK-GELAN to replace traditional convolution operations, effectively reducing the model's parameter count. Moreover, there was a 1.2% increase in mAP value, demonstrating the effectiveness of the CK-GELAN module in feature extraction. The incorporation of an enhanced CBAM structure into the backbone feature extraction network led to a 1.1% mAP value increase without a substantial increase in parameters. The variable convolution module improved the mAP value by 1.7 while only adding 0.6M parameters, enhancing the model's feature extraction capabilities. The feature fusion module CK-FFM, which includes a self-attention mechanism (SA), increased the model parameters by 0.5M but also improved the mAP value by 0.4%. By enhancing the PGI auxiliary branch of

YOLOv9 and integrating superior feature extraction modules and feature fusion networks, transitioning from unidirectional to bidirectional fusion resulted in a 0.7M parameter increase and a 0.5% mAP value improvement. Targeted data enhancement to address the imbalance in learning difficulty among samples significantly improved training effectiveness, leading to an 8.3% increase in mAP value. Appropriate data augmentation [23] methods can indeed improve the training effectiveness of the model during training.

A series of ablation experiments have demonstrated the effectiveness of the enhanced YOLOv9 model, CK-Net, in detecting low-quality data such as steel. By integrating multiple modules for feature extraction and processing, the improved model shows superior training performance when compared to the baseline model YOLOv9c. This highlights the significant value of the enhanced model in the field of industrial defect detection.

D. STEEL DATASET DETECTION RESULTS

The deep learning environment used in this article is Python 3.8, Pytorch 1.8.2+cu111, NVIDIA RTX 3070 with 8G gpu, and Intel Core i9-9900K CPU. After completing the training phase, testing was conducted on the test set. The test set contains 180 images. The pre-processing of the test image takes 0.2 milliseconds, model inference takes 27.9 milliseconds, and NMS takes 0.9 milliseconds on the test image with dimension (1, 3, 640, 640).

Partial detection results are shown in Figure 9. Among them, (a) shows the detection results for the ''crazing'' category, (b) for the ''inclusion'' category, (c) for the ''patches'' category, (d) for the ''pitted_surface'' category, (e) for the ''rolled-in_scale'' category, and (f) for the ''scratches'' category. As shown in the figure, various types of steel defects have been accurately identified and classified, and the detection effect is excellent. The detection precision of the CK-NET model in this paper on the test dataset is 99%, which fully proves the effectiveness of CK-NET in the task of detecting surface defects on steel.

VI. CONCLUSION

This paper presents the design of a steel surface defect detection network, CK-Net, based on the YOLOv9 model architecture. The development includes a feature extraction module, CK-GELAN, that integrates multiple modules. Additionally, an efficient feature fusion module, CK-FFM, is introduced along with a self-attention mechanism (SA) to enhance the fusion of network features of different scales, ensuring spatial and semantic information richness. Minor enhancements were made to the CBAM, which combined with deformable convolution in the feature extraction network to improve the model's ability to learn regions of interest and enhance feature extraction capabilities. The programmable gradient information branch (PGI) of YOLOv9 was also improved by replacing the original unidirectional fusion branch with a bidirectional fusion branch. This further strengthens the multi-level feature fusion capability, allowing backpropagation gradient information to guide the backbone network in learning features effectively, thus avoiding forward information loss due to network deepening and ultimately improving network learning performance. Experimental results show that the CK-Net model achieved an mAP value of 92.1% on the steel surface defect detection dataset NEU-DET, making significant contributions to the field of industrial defect detection.

REFERENCES

[1] Z. Zhen, D. Shuguang, and M. Ping'an, ''Wafer defects detecting and classifying system based on machine vision,'' in *Proc. 8th Int. Conf. Electron. Meas. Instrum.*, Aug. 2007, doi: 10.1109/ICEMI.2007.4351196.

[2] M. Salehi and S. Mirzakuchaki, ''A novel approach to speech enhancement based on deep neural networks,'' *Adv. Electr. Comput. Eng.*, vol. 22, no. 2, pp. 71–78, 2022, doi: 10.4316/AECE.2022.02009.

[3] C.-Y. Wang, I.-H. Yeh, and H.-Y. M. Liao, ''YOLOv9: Learning what you want to learn using programmable gradient information,'' 2024, *arXiv:2402.13616*.

[4] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei, ''Deformable convolutional networks,'' in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Venice, Italy, Oct. 2017, pp. 764–773.

[5] S. Woo, J. Park, J. Y. Lee, and I. S. Kweon, ''CBAM: Convolutional block attention module,'' in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Munich, Germany, Sep. 2018, pp. 3–19.

[6] 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, doi: 10.1109/TIM.2019.2915404.

[7] Z. Hao, Z. Wang, D. Bai, B. Tao, X. Tong, and B. Chen, ''Intelligent detection of steel defects based on improved split attention networks,'' *Frontiers Bioeng. Biotechnol.*, vol. 9, Jan. 2022, doi: 10.3389/fbioe.2021.810876.

[8] Z. Zou, K. Chen, Z. Shi, Y. Guo, and J. Ye, ''Object detection in 20 years: A survey,'' *Proc. IEEE*, vol. 111, no. 3, pp. 257–276, Mar. 2023, doi: 10.1109/JPROC.2023.3238524.

[9] H. Lin, B. Li, X. Wang, Y. Shu, and S. Niu, ''Automated defect inspection of LED chip using deep convolutional neural network,'' *J. Intell. Manuf.*, vol. 30, no. 6, pp. 2525–2534, Aug. 2019, doi: 10.1007/s10845-018-1415-x.

[10] X. Chen, J. Chen, X. Han, C. Zhao, D. Zhang, K. Zhu, and Y. Su, ''A lightweighted CNN model for wafer structural defect detection,'' *IEEE Access*, vol. 8, pp. 24006–24018, 2020, doi: 10.1109/ACCESS.2020.2970461.

[11] X. Lv and J. Xu, ''Steel plate surface defect detection method based on improved YOLO algorithm,'' in *Proc. China Automat. Congr. (CAC)*, 2023, doi: 10.1109/CAC59555.2023.10450211.

[12] Z. Li, X. Wei, M. Hassaballah, Y. Li, and X. Jiang, ''A deep learning model for steel surface defect detection,'' *Complex Intell. Syst.*, vol. 10, no. 1, pp. 885–897, Feb. 2024, doi: 10.1007/s40747-023-01180-7.

[13] B. Huang, J. Liu, X. Liu, K. Liu, X. Liao, K. Li, and J. Wang, ''Improved YOLOv5 network for steel surface defect detection,'' *Metals*, vol. 13, no. 8, p. 1439, Aug. 2023, doi: 10.3390/met13081439.

[14] C. Song, J. Chen, Z. Lu, F. Li, and Y. Liu, ''Steel surface defect detection via deformable convolution and background suppression,'' *IEEE Trans. Instrum. Meas.*, vol. 72, pp. 1–9, 2023, doi: 10.1109/TIM.2023.3277989.

[15] K. Imoto, T. Nakai, T. Ike, K. Haruki, and Y. Sato, ''A CNN-based transfer learning method for defect classification in semiconductor manufacturing,'' *IEEE Trans. Semicond. Manuf.*, vol. 32, no. 4, pp. 455–459, Nov. 2019.

[16] C.-C. Yeung and K.-M. Lam, ''Efficient fused-attention model for steel surface defect detection,'' *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–11, 2022.

[17] X. Kou, S. Liu, K. Cheng, and Y. Qian, ''Development of a YOLO-V3-based model for detecting defects on steel strip surface,'' *Measurement*, vol. 182, Sep. 2021, Art. no. 109454.

[18] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, ''MobileNets: Efficient convolutional neural networks for mobile vision applications,'' 2017, *arXiv:1704.04861*.

[19] K. Simonyan and A. Zisserman, ''Very deep convolutional networks for large-scale image recognition,'' 2014, *arXiv:1409.1556*.

[20] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1–9, doi: 10.1109/CVPR.2015.7298594.
[21] M. Shafiq and Z. Gu, "Deep residual learning for image recognition: A survey," *Appl. Sci.*, vol. 12, no. 18, p. 8972, 2022, doi: 10.3390/app12188972.
[22] S. Yun, D. Han, S. Chun, S. J. Oh, Y. Yoo, and J. Choe, "CutMix: Regularization strategy to train strong classifiers with localizable features," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6022–6031, doi: 10.1109/ICCV.2019.00612.
[23] L. Liu, H. Zhang, X. Xu, Z. Zhang, and S. Yan, "Collocating clothes with generative adversarial networks cosupervised by categories and attributes: A multidiscriminator framework," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 9, pp. 3540–3554, Sep. 2020, doi: 10.1109/TNNLS.2019.2944979.
[24] 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, doi: 10.1109/TPAMI.2016.2577031.
[25] Y. Zhou, W. Zhu, Y. He, and Y. Li, "YOLOv8-based spatial target part recognition," in *Proc. IEEE 3rd Int. Conf. Inf. Technol., Big Data Artif. Intell. (ICIBA)*, vol. 3, May 2023, pp. 1684–1687, doi: 10.1109/ICIBA56860.2023.10165260.

JIALIN ZOU received the bachelor's degree from the School of Computer Science, Guangdong University of Petrochemical Technology, in 2023. He is currently pursuing the degree with the School of Computer Science and Technology, Dongguan University of Technology. His current research interests include deep learning, computer vision, industrial defect detection, and diffusion model.

HONGCHENG WANG received the Ph.D. degree in optics from Sun Yat-sen University, in 2007. He is currently working as a Professor with the School of Electrical Engineering and Intelligentization, Dongguan University of Technology. His research interests include artificial intelligence technology and micro/nano optical devices.