Journal of Network Intelligence ©2025 ISSN 2414-8105
Taiwan Ubiquitous Information Volume **10**, Number **3**, August **2025**

Improved Yolov8 Algorithm for Steel Surface Defect Detection

Chong Guo[1], Ye-Cheng He[1,*], Yun-Fei Zhu[2]

[1]School of Information Science and Engineering, Shenyang Ligong University, Shenyang 110159, P. R. China
12652029@qq.com hyc200033@163.com

[2]Payap University, Chiang Mai 50000, Thailand
1468059575@qq.com

*Corresponding author: Ye-Cheng He

Received June 29, 2024, revised November 13, 2024, accepted February 11, 2025.

Abstract. *MRS-YOLO is an improved YOLOv8 model for steel surface defect detection, aiming to solve the problems of insufficient defect detection and low accuracy of traditional methods at different scales. The model enhances the target attention capability by introducing the MSCAAttention attention mechanism, replaces the traditional convolution with RFCAConv to improve the feature extraction accuracy, and incorporates the SPD module to enhance the processing capability for small targets and low-resolution images. In addition, a new C2f_GhostBottleneckV2 module is designed to reduce the number of model parameters. The experimental results show that MRS-YOLO achieves 92.1% and 77.4% mean accuracies (mAP) on the NEU-DET and GC10-DET datasets, respectively, which are improved by 12.5% and 9.5% compared to the original model, while the model parameters are reduced by 9.6%. These improvements make MRS-YOLO more accurate and efficient in detecting steel surface defects, providing an effective solution for industrial applications.*

Keywords: steel defect detection; object detection; attention mechanism; receptive field attention convolution; lightweighting

1. **Introduction.** Steel is the foundation and raw material for many industries and products, so the quality of the steel produced is crucial. However, during the production of steel, various types of defects such as cracks, holes, scratches, and other defects may form on the surface of the product due to the quality of raw materials, manufacturing equipment, and the production environment [1]. These defects can lead to serious accidents such as car accidents, bridge collapses and other manufacturing accidents. From the point of view of ensuring product quality control, ensuring safety in the use of steel, and controlling production costs, the development of defect detection technology for steel surfaces is crucial and in high demand. Initially, defect detection relied more on manual methods, which were not only time-consuming and laborious, but also very inefficient. However, with the development of artificial intelligence, many tasks previously performed by manual labor are gradually replaced by automated equipment [2].

The emergence and development of deep learning has led to the emergence of many target detection algorithms that can be applied in the field of defect detection. Currently, deep learning methods in the field of target detection algorithms are mainly categorized into two types according to the complexity of their detection process: two-stage

(Two Stages) target detection algorithms and one-stage (One Stage) target detection algorithms [3].

Two Stages target detection algorithms are target detection algorithms based on region suggestions, such as R-CNN [4], Fast R-CNN [5] and Faster R-CNN [6], which have a slower detection speed. Single-stage target detection algorithms, also known as regression-based target detection algorithms, do not directly generate the region of interest but directly generate the class probability and location coordinate values of the object, after a single detection can be directly obtained the final detection results, and therefore have a faster detection speed. Such as YOLO [7], YOLOv3 [8], YOLOv4 [9], YOLOv5 [10], SSD [11], RetinaNet [12] and so on.

Iron and steel surface defects inspection technology has experienced the development of manual inspection to automated inspection: manual inspection: initially, the steel surface defects detection is mainly carried out manually, through visual observation or the use of simple tools for inspection. This method has the disadvantages of low efficiency, susceptible to subjective factors, difficult to find small defects. Machine vision inspection: With the continuous development of computer vision technology, machine vision inspection technology is gradually applied to steel surface defect detection. This method automatically recognizes and classifies the defects on the steel surface through the steps of image acquisition, pre-processing, feature extraction and defect classification.

Compared with manual inspection, machine vision inspection has the advantages of high efficiency, high precision and good stability, but there are still some limitations, such as high sensitivity to environmental factors such as light and surface reflection, and difficulty in dealing with complex shapes and large-area defects.

In recent years, deep learning technology has made breakthroughs in the field of target detection and has been gradually applied to steel surface defect detection. Deep learning detection methods do not need to manually design features, can automatically learn the features in the image and defect recognition, with higher accuracy and robustness. Although deep learning detection methods have achieved significant results in the detection of steel surface defects, there are still some shortcomings: Inadequate detection of defects at different scales: the existing deep learning detection models tend to be more effective in detecting medium-sized defects, but less effective in detecting tiny defects or large-area defects. Detection accuracy needs to be improved: the steel surface defects of a wide variety of complex morphology, and there is a certain degree of similarity, which makes it difficult to identify defects, the existing detection model still has a certain false detection rate and leakage rate. The number of model parameters is large: the training of deep learning models requires a large amount of data and computational resources, and the number of model parameters is large, resulting in a high cost of model deployment and application.

Aiming at the shortcomings of existing methods, this paper proposes an improved YOLOv8 algorithm, called MRS-YOLO, for steel surface defect detection. The algorithm effectively improves the model's ability to detect defects at different scales, detection accuracy and robustness, and reduces the number of model parameters by introducing improvements such as the attention mechanism, sensory field attention convolution, SPD module, and lightweight module, which provides new ideas and methods for the development of the steel surface defect detection technology. The performance of the algorithm is validated on the public datasets NEU-DET and GC10-DET. The main contributions of this paper are:

(1) Embedding the attention module MSCAAttention into the neck of YOLOv8 to improve the feature extraction capability of the model, so that the model pays more attention to and learns useful information.

(2) Replacing Conv in the original model with the sensory field attention convolution RFCAConv to emphasize the importance of different features within the sensory field slider and to solve the problem of convolution kernel parameter sharing.

(3) Add a layer of SPD module behind each sense field attention convolution to effectively avoid the loss of fine-grained information less efficient learning of feature representation and improve the model's ability to handle low-resolution images and small objects.

(4) The C2f_GhostBottleneckV2 module is designed to replace the traditional C2f module to reduce the number of model parameters.

2. **Related work.**

2.1. **Traditional detection methods.**

The main processes of traditional machine vision-based steel surface defect detection methods include: image preprocessing, ROI (Region of Interest) detection, image segmentation, feature extraction, and defect classification [13]. Ohkubo et al. proposed an inspection system with reflective prisms and columnar mirrors as the optics, laser as the scanning light source, and photomultiplier tubes to receive the detection system [14, 15]; Liu et al. proposed the eddy current detection theory and successfully used eddy current detection technology to detect barium metal [16]; Choudhary et al. developed an online automatic inspection system for surface defects on continuous casting billets [17]; Dombrowski et al. chose a fully programmable image digitizer and a fully programmable high-speed digital signal processor to process the digital image signals and used the system for cold rolled strip steel surface, edge cracks and other edge detection [18]; Shu et al. studied the strip steel surface inspection system, analyzed the CCD(Charge-Coupled Device) detection methods and proposed some effective algorithms [19]; Hossam et al. used computer image processing technology and pattern recognition technology for image processing and defect classification algorithms for effective surface defect detection [20]. These traditional detection methods have problems such as the need for manual feature extraction, high time cost, slow detection speed and poor robustness.

2.2. **Deep Learning Methods.**

Deep learning based steel surface defect detection method will be divided into three aspects to be introduced: attention mechanism, convolution, and lightweighting. Attention mechanism can make the model more useful information, commonly used attention mechanisms include CBAM (Convolutional Block Attention Module), CA (Coordinate Attention), SE (Squeeze-and-Excitation), ECA (Efficient Channel Attention), etc., Lv et al. added SE attention network in YOLOv5 backbone network to improve the model's attention to defect targets [21]; Zeng et al. added CA attention mechanism to YOLOv5 to improve the network's performance in detecting faults of various sizes [22]; Luo et al. integrated the CBAM attention mechanism into the backbone network of YOLOv7 to enhance the model's ability to extract and process image features [23].

Convolution is an important component of neural networks to reduce redundant information and extract useful features. Song proposed an improved convolution with deformable convolution to enhance the detection performance of large-scale defects with complex and irregular shapes [24]; Li utilized zero convolution with different expansion coefficients in YOLOv4-tiny for parallel sampling to capture the multi-scale contextual information [25]; Yu can increase the sensory field of the model without increasing the computation of the model by expanding the convolution [26].

Lightweight networks imply that the number of parameters and computation is small, and the model generated by training occupies less physical storage space. Commonly used lightweight networks include MobileNetV3, GhostNet, ShuffleNetv2, etc. Hao proposed a lightweight target detection framework based on the MobileNetv3 backbone model [27]; Sun replaced the CSPDarknet of YOLOv4 with Ghostnet to enhance the ability of the backbone network to extract defective features [28]; Yan used the ShuffleNetv2 module as the backbone of YOLOv5 to reduce the number of gigaflops per second floating point operations and parameters [29]. Deep learning based methods no longer need to extract features manually compared to traditional methods, and they also have high accuracy and robustness. In this paper, we choose YOLOv8 as a benchmark model and improve it to make it more suitable for industrial defect detection scenarios.

3. Methods.

3.1. Improved YOLOv8 model.

YOLOv8 is a newly proposed model in YOLO series, which consists of three parts: feature extraction network (Backbone), feature fusion network (Neck), and detection head (Head). MRS-YOLO is a new steel surface defect detection model proposed with YOLOv8 as the baseline, and the overall framework of the network is shown in Figure 1.

3.2. Multiscale Convolutional Attention MSCAAttention.

Multi-scale Convolutional Attention (MSCAAttention) is an innovative architecture used for feature extraction in encoders, the structure of which is shown in Figure 2. The core of the MSCA module [30] consists of three parts: a depth-splittable convolution, a multi-branch depth-splittable banded convolution (used to capture multi-scale contexts), and a 1×1 convolution (used to model the relationship between different channels). These three components work in tandem to enable the module to efficiently attend to information at different scales, thus better capturing local and global features in the image. Currently, traditional spatial attention mechanisms utilize an attention map obtained by learning to highlight the importance of each feature. The spatial attention mechanism for highlighting key features can be simply expressed as follows:

$$\begin{aligned} &F_1 = X_1 \times A_1 \\ &F_2 = X_2 \times A_2 \\ &\dots \\ &F_N = X_N \times A_N \end{aligned} \tag{1}$$

Here, F_i represents the value obtained after the weighting operation. X_i and A_i represent the values of the input feature map and the learned attention map at different positions, respectively, N is the product of the height and width of the input feature map, which represents the total number of pixel values.

In contrast to the traditional spatial attention mechanism, in the MSCA module, depth-separable convolution is used to generalize local information, while multi-branch depth-separable strip convolution is used to capture contextual information at different scales. 1×1 convolution is used to model the relationship between different channels to enhance the feature representation by introducing channel correlation:

$$Att = \mathrm{Conv}_{1\times 1}\left(\sum_{i=0}^{3} \mathrm{Scale}_i(\mathrm{DW} - \mathrm{Conv}(F))\right) \tag{2}$$

$$Out = Att \otimes F \tag{3}$$

Where Att denotes the attention map, Out denotes the output, $\mathrm{Conv}_{1\times 1}$ is a 1×1 convolution operation, DW-Conv is a depth divisible convolution, and Scale_i is a multiscale

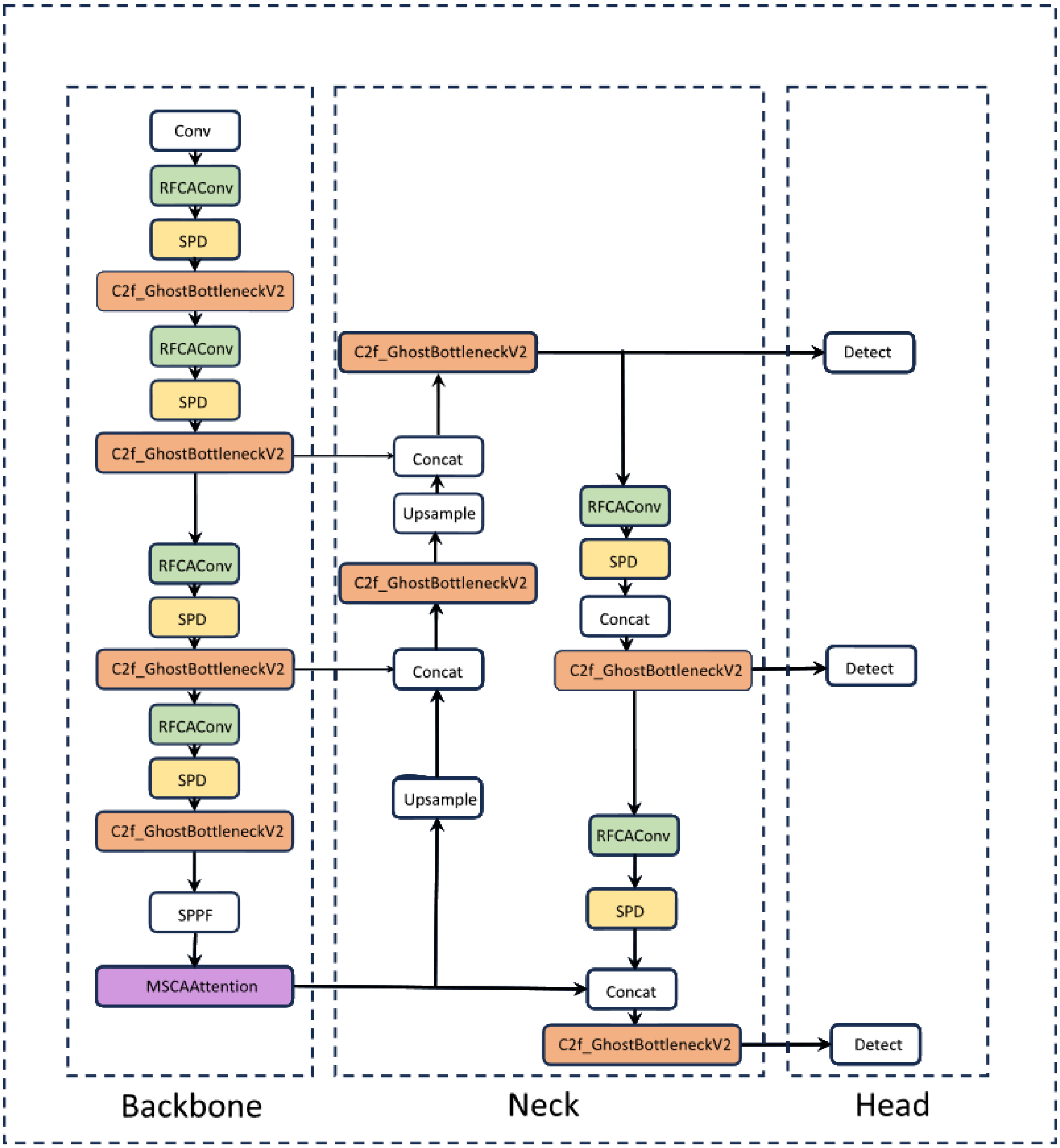


Figure 1. MRS-YOLO Network Structure Diagram

bar convolution operation for the ith branch. This design fully utilizes the lightweight nature of depth-divisible convolution and the adaptability of multi-scale bar convolution to improve the efficiency and flexibility of feature extraction.

The comparison results with the original YOLOv8 using precision (P), recall (R), mean accuracy (mAP), and parameter count (Param(M)) as evaluation metrics are shown in Table 1.

Table 1. MSCAAttention Attention Module

Scheme	P	R	mAP	Param(M)
Yolov8	0.789	0.73	0.816	3.01
Yolov8+ MSCAAttention	0.818	0.74	0.84	3.1

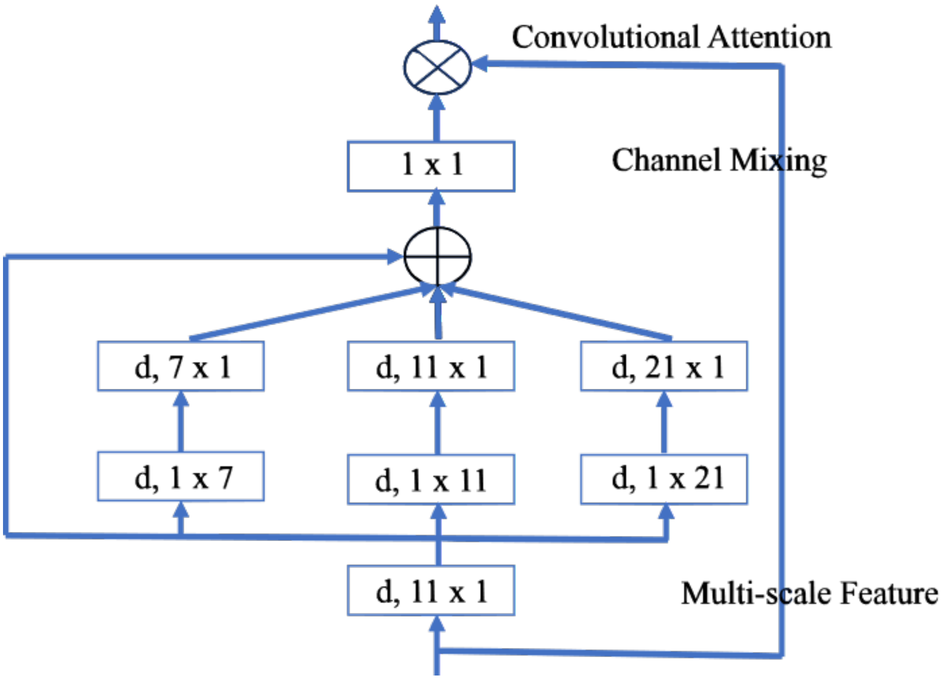


Figure 2. MSCAAttention structure diagram

3.3. **Feeling Wild Attention Convolution RFCAConv with SPD Module.** Receptive-Field Attention Convolution RFCAConv (Receptive-Field Channel Attention Convolution) is a convolutional operation in Convolutional Neural Networks (CNNs), which is designed to optimize by introducing the innovative Receptive-Field Attention (RFA) [?] concept for feature extraction, the structure of which is shown in Figure 3.

RFA is a novel receptive-field attention mechanism that emphasizes the importance of different features within the sliding window of the convolutional kernel while prioritizing the spatial features of the receptive field. By solving the convolutional kernel parameter sharing problem, the computation of RFA can be expressed as:

$$F = \text{Softmax}(g_{1\times1}(\text{Avgpool}(X))) \times \text{ReLU}(\text{Norm}(g_{k\times k}(X))) = A_{rf} \times F_{rf} \tag{4}$$

Denotes the grouped convolution of size $k \times k$, k denotes the size of the convolution kernel, Norm denotes normalization, X denotes the input feature mapping, and F is obtained by multiplying the attention map with the transformed receptive field space features.

Currently, the most commonly used kernel sizes in convolutional neural networks are 1×1 and 3×3. The convolution operation for extracting features after the introduction of the spatial attention mechanism is a 1×1 or 3×3 convolution operation. To visualize this process, the spatial attention mechanism is inserted in front of the 1×1 convolution operation. The whole process of weighting the input feature map by the attention map (reweighting ”x”) and finally extracting the slider feature information of the receiver field by the 1×1 convolution operation can be represented as:

$$\begin{gathered} F_1 = X_1 \times A_1 \times K \\ F_2 = X_2 \times A_2 \times K \\ \vdots \\ F_N = X_N \times A_N \times K \end{gathered} \tag{5}$$

Here, the convolution kernel K represents only one parameter value. The value of $A_i \times K$ is used as a new convolution kernel parameter.

However, when the spatial attention mechanism is inserted in front of the 3 × 3 convolutional operation, it will be limited. As mentioned above, if we take the value of $A_i \times K$ as a new convolution kernel parameter, the Equation below completely solves the problem of parameter sharing for large-scale convolutional kernels.

$$\begin{aligned} &F_1 = X_{11} \times A_{11} \times K_1 + X_{12} \times A_{12} \times K_2 + \cdots + X_{19} \times A_{19} \times K_9 \\ &F_2 = X_{21} \times A_{21} \times K_1 + X_{22} \times A_{22} \times K_2 + \cdots + X_{29} \times A_{29} \times K_9 \\ &... \\ &F_N = X_{N1} \times A_{N1} \times K_1 + X_{N2} \times A_{N2} \times K_2 + \cdots + X_{N9} \times A_{N9} \times K_9 \end{aligned} \tag{6}$$

Traditional convolution has two limitations: on the one hand, the parameters used to extract features in each receptive field in the convolution kernel of traditional convolution are the same in the operation process, without taking into account the difference information in different locations; on the other hand, the extraction of features in the convolution process is not sufficient and comprehensive, which greatly affects the effect of feature extraction. The traditional convolution operation utilizes a sliding window with shared parameters to extract feature information, which overcomes the problem of multiple parameters and large computation inherent in neural networks constructed with fully connected layers. Let $X \in \mathbb{R}^{H \times W \times C}$ be the input feature map, where C, H, and W are the number of channels, height, and width of the feature map, respectively. In order to clearly demonstrate the feature extraction process through the convolution kernel, using the example of $C = 1$, the convolution operation for extracting feature information from each receptive field slider can be expressed as:

$$\begin{aligned} F_1 &= X_{11} \times K_1 + X_{12} \times K_2 + \cdots + X_{1S} \times K_S \\ F_2 &= X_{21} \times K_1 + X_{22} \times K_2 + \cdots + X_{2S} \times K_S \\ &\vdots \\ F_N &= X_{N1} \times K_1 + X_{N2} \times K_2 + \cdots + X_{NS} \times K_S \end{aligned} \tag{7}$$

Here, F_i represents the value obtained by each convolutional slider after computation, X_i represents the pixel value at the corresponding position within each slider, K represents the convolutional kernel, S denotes the number of parameters in the convolutional kernel, and N represents the total number of the receptive-field slider.

RFCAConv introduces receptive field attention into the convolution operation, aiming to address the above two limitations of conventional convolution.

Space-to-depth (SPD) module is a space-to-depth layer, a feature map downsampling technique in convolutional neural networks (CNNs), which is used to downsample the feature maps while retaining all the information in the channel dimensions, thus causing no information loss, and its structure is shown in Figure 4. The method cuts the original feature maps into sub-feature maps and realizes effective downsampling of feature maps by cutting and recombining them in the spatial dimension. The basic principle of SPD [?] is to form a series of sub-feature maps by cutting on the original feature maps and connecting these sub-feature maps in the channel dimension. The calculation formula is shown below:

$$\begin{aligned}
&f_{0,0} = X[0 : S : scale, 0 : S : scale],\\
&f_{1,0} = X[1 : S : scale, 0 : S : scale], ...,\\
&f_{scale-1,0} = X[scale - 1 : S : scale, 0 : S : scale];\\
&f_{0,1} = X[0 : S : scale, 1 : S : scale],\\
&f_{1,1} = X[1 : S : scale, 1 : S : scale], ...,\\
&f_{scale-1,1} = X[scale - 1 : S : scale, 1 : S : scale];\\
&f_{0,scale-1} = X[0 : S : scale, scale - 1 : S : scale],\\
&f_{1,scale-1} = X[1 : S : scale, scale - 1 : S : scale], ...,\\
&f_{scale-1,scale-1} = X[scale - 1 : S : scale, scale - 1 : S : scale];
\end{aligned} \tag{8}$$

For an intermediate feature map X of size $S \times S \times C_1$, the sub-feature map $f_{x,y}$ consists of all those entries in the original feature map X that satisfy that $i + x$ and $j + y$ are integrable by scale. This step results in each sub-feature map downsampling the original feature map by a factor of scale. These sub-feature maps are connected along the channel dimensions to form a new feature map X_0. This shape adjustment implements a dimensional transformation of the original feature map by means of space-to-depth. With SPD, the dimensionality of the feature map can be flexibly adjusted, providing an effective method of dimensionality reduction for the model, while improving the detection accuracy of the model while maintaining certain information. The comparison results with the original YOLOv8 are shown in Table 2.

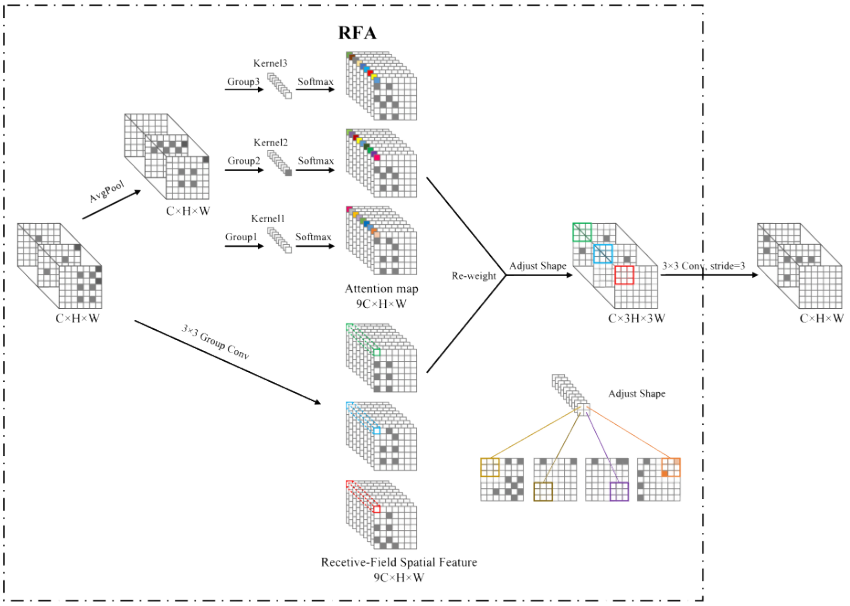


Figure 3. RFCAConv structure diagram

Table 2. Comparison of RFCAConv+SPD and Traditional Conv

Scheme	P	R	mAP	Param(M)
YOLOv8	0.789	0.73	0.816	3.01
YOLOv8+RFCAConv+SPD	0.867	0.804	0.889	3.51

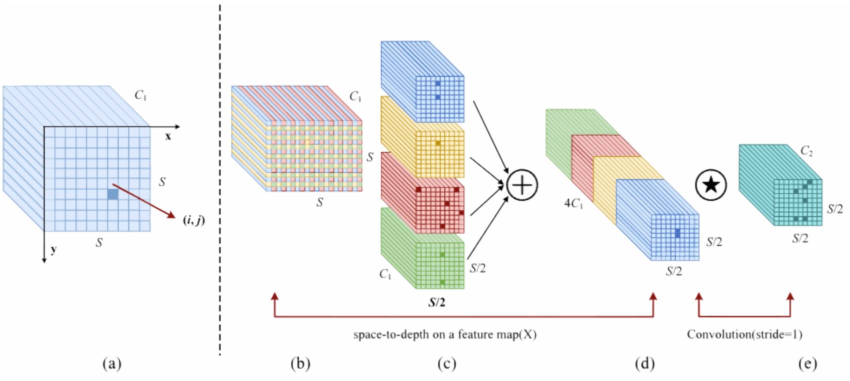


Figure 4. SPD structure diagram

3.4. **C2f_GhostV2Bottleneck Module.** The structure of C2f_GhostV2Bottleneck is shown in Figure 5. GhostNet employs a lightweight design in order to operate efficiently even when computational resources are constrained, and it mainly consists of a series of lightweight convolutional layers and feature-processing modules. The bottleneck structure of GhostV2Bottleneck is a GhostNet [33], it can generate more feature mappings to replace the original convolution. Given an input feature $X \in \mathbb{R}^{H\times W\times C}$ with height H, width W and channel C, a Ghost module can replace a standard convolution in two steps. First, the endowment features are generated using a 1×1 convolution, i.e.,

$$Y' = X * F_{1\times 1}, \tag{9}$$

where $*$ denotes the convolution operation. $F_{1\times 1}$ is the pointwise convolution and $Y' \in \mathbb{R}^{H\times W\times C'}$ is the intrinsic feature, whose size is usually smaller than the original output feature, i.e., $C' < C_{out}$. Deep convolution, etc., is then used to generate more features based on the intrinsic feature. The two parts of the feature are concatenated along the channel dimension, i.e.,

$$Y = \mathrm{Concat}\left([Y', Y' * F_{dp}]\right), \tag{10}$$

where F_{dp} is the depth-wise convolutional filter, and $Y \in \mathbb{R}^{H\times W\times C_{out}}$ is the output feature.

One of the key components utilizes a reverse residual bottleneck design. This design subtly separates the "expressiveness" and "capacity" of the model through two Ghost modules. The GhostV2 bottleneck starts with the first Ghost module generating extended features, increasing the number of channels. The Ghost module is an efficient operation that generates some of the features through cheap arithmetic, but may affect the expressiveness and capacity of the model. This is because only half of the features in the Ghost module interact with other pixels, limiting the capture of spatial information. To ameliorate this problem, DFC (Depthwise Feature Correlation) attention is introduced. DFC attention employs the operation of depthwise separable convolution, which is effective in reducing the number of parameters and computational complexity, and thus suitable for lightweight network design, by calculating the correlation matrix of the feature map in the deep convolutional network, so as to obtain the correlation information between the features. Given a feature $Z \in \mathbb{R}^{H\times W\times C}$, it can be seen as HW tokens $Z_i \in \mathbb{R}^C$, i.e., $Z = \{z_{11}, z_{12}, \ldots, z_{HW}\}$. A direct implementation of FC layer to generate the attention map is formulated as:

$$a_{hw} = \sum_{h',w'} F_{hw,h',w'} \odot z_{h'w'}, \tag{11}$$

where $\odot$ is element-wise multiplication, F is the learnable weights in the FC layer, and $A = \{a_{11}, a_{12}, \ldots, a_{HW}\}$ is the generated attention map. Equation (11) can capture the global information by aggregating all the tokens together with learnable weights, which is much simpler than the typical self-attention as well. However, its computational process still requires quadratic complexity w.r.t. feature's size, which is unacceptable in practical scenarios especially when the input images are of high resolutions. For example, the 4-th layer of GhostNet has a feature map with 3136 (56 × 56) tokens, which incurs prohibitively high complexity to calculate the attention map. Actually, feature maps in a CNN are usually of low-rank, it is unnecessary to connect all the input and output tokens in different spatial locations densely. The feature's 2D shape naturally provides a perspective to reduce the computation of FC layers, i.e., decomposing Equation (11) into two FC layers and aggregating features along the horizontal and vertical directions, respectively. It can be formulated as:

$$a'_{hw} = \sum_{h'=1}^{H} F_H(h,h') z_{h'w}, \quad h = 1,2,\ldots,H;\ w = 1,2,\ldots,W \tag{12}$$

$$a_{hw} = \sum_{w'=1}^{W} F_W(w,w') z'_{hw'}, \quad h = 1,2,\ldots,H;\ w = 1,2,\ldots,W \tag{13}$$

where F_H and F_W are transformation weights. Taking the original feature Z as input, Equation (12) and Equation (13) are applied to the features sequentially, capturing the long-range dependence along the two directions, respectively. This operation is Decoupled Fully Connected (DFC) Note.

The input features $X \in \mathbb{R}^{H\times W\times C}$ are sent to two branches, one where the Ghost module produces the output features Y (Equation (9) and Equation (10)), and the other is the DFC module that produces the attention map A (Equation (12) and Equation (13)), using a 1 × 1 convolution that converts the module's input X to the DFC's input z. The module's final output is the product $O \in \mathbb{R}^{H\times W\times C}$ of the outputs of the two branches, i.e.,

$$O = \mathrm{Sigmoid}(A) \circ V(X) \tag{14}$$

where $\circ$ is the element-wise multiplication and Sigmoid is the scaling function to normalize the attention map A into range (0, 1). $V(X)$ is the feature map obtained by linear transformation of the input feature X.

GhostV2 bottleneck better balances the expressiveness and capacity of the model while maintaining lightweight and efficiency. The results of the comparison experiments in which the C2f module is replaced with the C2f_GhostV2Bottleneck module are are shown in Table 3.

Table 3. Comparison of C2f_GhostV2Bottleneck and C2f

Scheme	**P**	**R**	**mAP**	**Param(M)**
YOLOv8	0.789	0.73	0.816	3.01
YOLOv8+C2f_GhostV2Bottleneck	0.724	0.666	0.741	2.12

4. **Experiment.**

4.1. **Experimental environment and dataset.**

This experiment is done in PyTorch framework with the environment configuration of Windows 11, Python 3.8, Pytorch 1.12, and Nvidia RTX3070 GPU. In this experiment, the size of BatchSize is set to 16, Momentum is set to 0.937, decay coefficient is set to 0.0005, the number of training rounds of Epoch is set to 300, the initial size of the learning rate is set to 0.01, and finally after completing the 300 rounds of training, we will get the trained model. In this experiment, we use the publicly available datasets NEU-DET [34] and GC10-DET to validate the

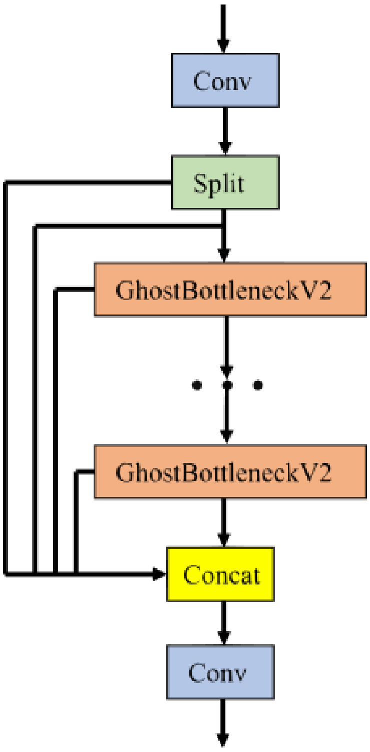


Figure 5. C2f_GhostV2Bottleneck structure diagram

performance of the proposed model. The NEU-DET steel defect detection dataset is published by Northeastern University (NEU). DET steel defect detection dataset is a surface defect database published by Northeastern University (NEU), which collects six typical surface defects of hot rolled steel strip, i.e., rolled oxide skin (RS), plaque (Pa), cracking (Cr), pitting surface (PS), inclusions (In), and scratches (Sc). The dataset consists of 1800 grayscale images: six different types of typical surface defects, each containing 300 samples. GC10-DET is a dataset of surface defects collected in a real industry. A real industry. It contains ten types of surface defects, i.e. Punch (Pu), Weld (Wl), Crescent Gap (Cg), Water Spot. Oil Spot (Os), Silk Spot (Ss), Inclusions (In), Rolled Pits (Rp), Creases (Cr), Waist Folds (Wf). The collected defects are on the surface of the steel plate. The dataset consists of 3570 grayscale images.

4.2. **Evaluation metrics.** The experiments used Size(MB), mAP, P, R, F1 and Parameter count (Param) as evaluation metrics to assess the performance of the model. Size denotes the size of the model and mAP denotes the mean average precision mean, through which five metrics are used to evaluate the model performance.

$$\text{Precision} = \frac{TP}{TP+FP} \tag{15}$$

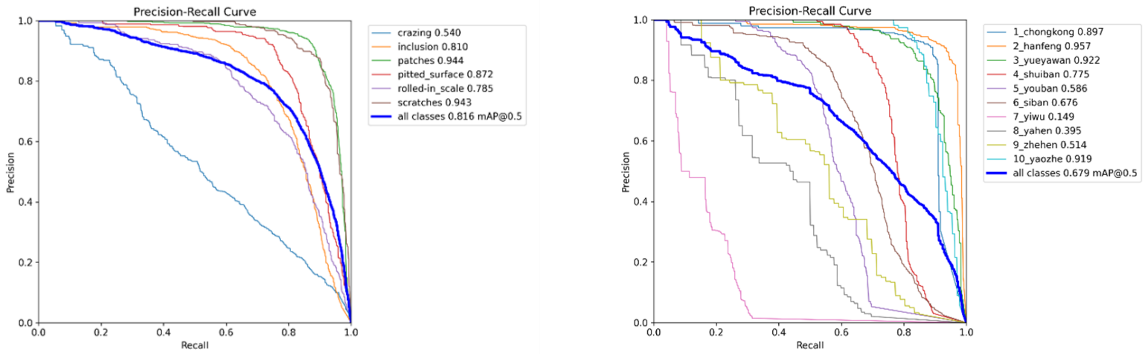
$$\text{Recall} = \frac{TP}{TP+FN} \tag{16}$$

$$\text{mAP} = \frac{1}{c}\sum_{n=1}^{c} AP_i \tag{17}$$

$$F1 = \frac{2TP}{2TP+FN+FP} \tag{18}$$

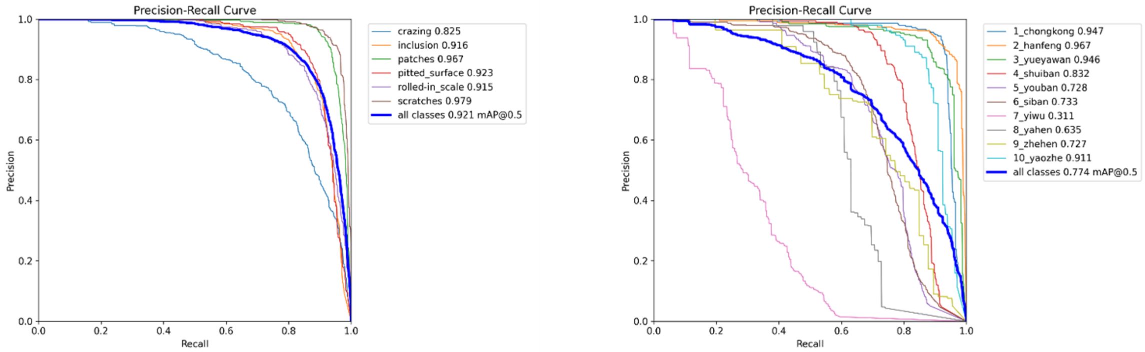
TP denotes correctly predicted defects, FP denotes incorrectly predicted defects, and FN denotes incorrectly predicted non-defects. n is the defect category and c is the number of defect categories.

4.3. **Experimental results.** Trained on NEU-DET and GC10-DET datasets, the training results of YOLOv8 are shown in Figure 6, and the training results of MRS-YOLO are shown in Figure 7. Some of the detection results of MRS-YOLO are shown in Figure 8.



(a) NEU-DET (b) GC10-DET

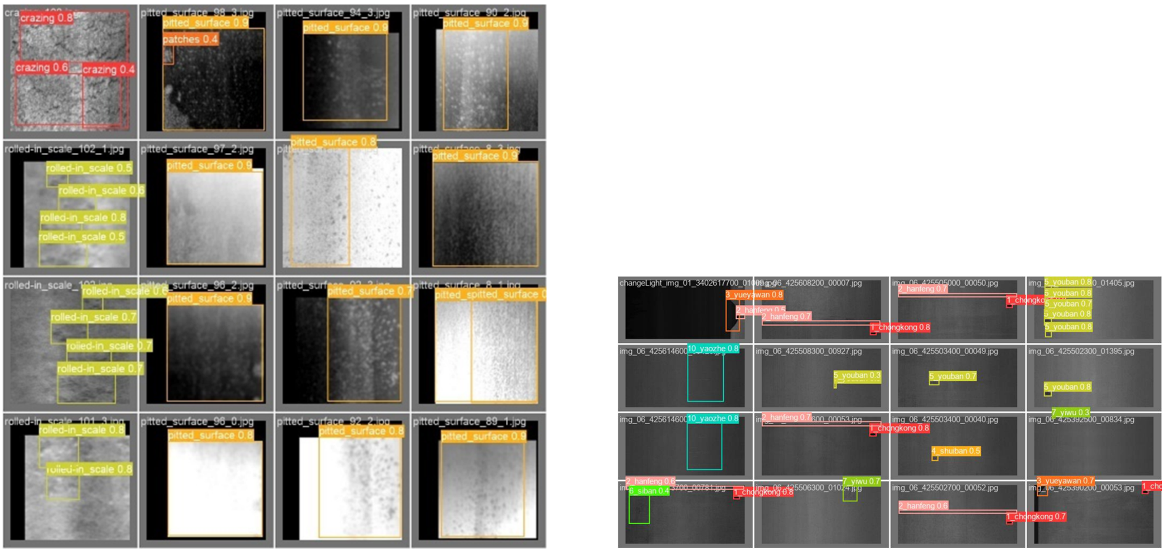
Figure 6. (a) and (b) separately denote the training results of YOLOv8 on NEU-DET and GC10-DET datasets



(a) NEU-DET (b) GC10-DET

Figure 7. (a) and (b) separately denote the training results of MRS-YOLO on NEU-DET and GC10-DET datasets

4.4. **Ablation experiments.** Ablation experiments were performed on the NEU-DET dataset and the GC10-DET dataset to verify the validity of each module added, and the results of the ablation experiments on these two datasets are represented in Table 4 and Table 5, respectively. Experiment 1 denotes Yolov8+ MSCAAttention; Experiment 2 indicates that Yolov8+ MSCAAttention +RFCAConv+SPD. Combining the data in the tables, it can be seen that the model proposed in this study has a great improvement over the original model. The improved convolution can better capture local and global information in the image, help the model to understand the context and relevance of the image more comprehensively, and improve the perception of defects. SPD further improves the model's perceptual ability, which helps the model to capture small targets in the image as well as have good processing ability for low-resolution images. The attentional mechanism increases the nonlinear representation of the model, and different levels of



(a) NEU-DET

(b) GC10-DET

Figure 8. (a) and (b) separately represent the detection results of MRS-YOLO on NEU-DET and GC10-DET datasets

features can be learned at different stages, allowing the model to adapt to features of different scales and complexity. It makes the number of parameters of the model decrease while maintaining the detection accuracy, improves the feature extraction ability of the model, and reduces the computation time of the model.

Table 4. Ablation experiment on NEU-DET

Scheme	F1	P	R	mAP	Size	Param(M)
Yolov8	0.75	0.789	0.73	0.816	6.2	3.01
Experiment 1	0.78	0.818	0.74	0.836	6.4	3.1
Experiment 2	0.83	0.821	0.769	0.853	6.6	3.15
MRS-YOLO	0.86	0.882	0.84	0.921	5.8	2.72

Table 5. Ablation experiment on GC10-DET

Scheme	F1	P	R	mAP	Size	Param(M)
Yolov8	0.66	0.724	0.63	0.679	6.2	3.01
Experiment 1	0.68	0.73	0.622	0.696	6.4	3.1
Experiment 2	0.78	0.77	0.635	0.735	6.6	3.15
MRS-YOLO	0.75	0.808	0.719	0.774	5.8	2.72

4.5. **Comparative experiments.** In order to verify the validity of the model, it was compared with some common detection models YOLOv5, YOLOv7 [35], YOLOv8, YOLOv8-SSDW [36], YOLOv8-DSG [37], SCFNet [38], YOLO-RDP [39], EC-YOLO [40] on the NEU-DET and GC10-DET dataset. The Size, mAP, F1 and the number of parameters of the different models on the NEU-DET dataset and the GC10-DET dataset are shown in Table 6 and Table 7. On the NEU-DET dataset, MRS-YOLO is optimal in all metrics and has the smallest and least number of Size and number of parameters. In order to further verify the robustness of the model, a comparison test is conducted on the GC10-DET dataset, and MRS-YOLO is also optimal in all the metrics, and the Size and the number of parameters are also the smallest and least. From the experimental results, it can be seen that MRS-YOLO has achieved good results in improving the detection accuracy and reducing the model size.

Table 6. Results of different models on NEU-DET

Scheme	F1	P	R	mAP	Size	Param(M)
Yolov5	0.77	0.807	0.739	0.804	14.3	7.03
Yolov7	0.67	0.704	0.664	0.724	12.3	6.03
Yolov8	0.75	0.789	0.73	0.816	6.2	3.01
YOLOv8-SSDW	0.81	0.832	0.789	0.855		8.38
YOLOv8-DSG	0.76	0.77	0.743	0.8		3.18
SCFNet	0.75	0.786	0.715	0.812		2
YOLO-RDP	0.72	0.67	0.779	0.798		3.5
EC-YOLO	0.8	0.814	0.783	0.834		
MRS-YOLO	0.86	0.882	0.84	0.921	5.8	2.72

Table 7. Results of different models on GC10-DET

Scheme	F1	P	R	mAP	Size	Param(M)
Yolov5	0.63	0.672	0.635	0.624	14.3	7.03
Yolov7	0.61	0.695	0.582	0.629	12.3	6.03
Yolov8	0.66	0.724	0.63	0.679	6.2	3.01
EC-YOLO	0.69	0.66	0.718	0.706		
YOLO-RDP	0.76	0.801	0.727	0.764		4.21
MRS-YOLO	0.75	0.808	0.719	0.774	5.8	2.72

5. **Conclusion.** In this study, the MSCAAttention attention module is added to YOLOv8 to help the model pay better attention to the feature information; the convolution module is replaced with RFCAConv to improve the model's ability to perceive defects; a layer of SPD module is added to each layer of RFCAConv, which helps in the detection of small targets; and the C2f module in the feature fusion network is replaced with the C2f_GhostV2Bottleneck module, which helps to reduce the number of model parameters and make the model lightweight. The performance of the model is evaluated on two datasets with different resolutions, and the experimental results show that our proposed model significantly improves the overall accuracy while decreasing the number of parameters, which well mitigates the problems of misdetection and omission. In future research, we still need to face some challenges and directions for improvement. First, a better balance between detection speed and detection accuracy needs to be found to ensure that the model can still remain efficient and accurate in scenarios with high real-time requirements. Second, considering the complexity and diversity of industrial environments, the generalization ability of the model is an important research direction, and we need to further optimize the model so that it can adapt to different industrial scenarios and challenges. In addition, with the development of technology, new algorithms and hardware may provide possibilities for further optimization of the models, so it is also necessary to continuously track the latest research progress and integrate them into the existing models. Finally, the quality and diversity of the dataset is crucial for model training. In future work, we may consider collecting and organizing more challenging industrial scenario data to

further improve the robustness and usefulness of the model. Meanwhile, interdisciplinary cooperation may also bring new perspectives and methods to solve the industrial target detection problem.

Acknowledgment. This work is partially supported by the optimization Study of Electricity Load Forecasting and Supply-Demand Coordination under Demand Side Response.

REFERENCES

[1] R. Mordia and A. Kumar Verma, "Visual Techniques for Defects Detection in Steel Products: A Comparative Study," Engineering Failure Analysis, vol. 134, 245993385, 2022.

[2] S. Kim, W. Kim, Y.K. Noh and F. Park, "Transfer learning for automated optical inspection," 2017 International Joint Conference on Neural Networks (IJCNN), 2017, pp. 2517-2524.

[3] Y.-Z. Xiao, Z.-Q. Tian, J.-C. Yu, X.-G. Lan, "A review of object detection based on deep learning," Multimedia Tools and Applications, vol. 79, pp. 23729-23791, 2020.

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

[5] R.B. Girshick, "Fast R-CNN," Computer Science, 2015, Available: https://doi.org/10.48550/arXiv.1504.08083

[6] S. Ren, K.-M. He, J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, pp. 1137-1149, 2015.

[7] J. Redmon, S. Divvala, R.B. Girshick, A. Farhadi, "You only look once: Unified, real-time object detection," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 779-788.

[8] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," Computer Science, 2018, Available: https://doi.org/10.48550/arXiv.1804.02767

[9] A. Bochkovskiy, C.-Y. Wang, H. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," Computer Science, 2020, Available: https://doi.org/10.48550/arXiv.2004.10934

[10] G. Jocher, A. Stoken, F. Ingham, "ultralytics/yolov5: v5.0 - YOLOv5-P6 1280 models, AWS, Supervise.ly and YouTube integrations," Zenodo, 2021, Available: https://doi.org/10.5281/zenodo.4679653

[11] W. Liu, D. Anguelov, A. Berg, "SSD: Single shot MultiBox detector," European Conference on Computer Vision, 2015, pp. 21–37.

[12] T.-Y. Lin, P. Goyal, R.B. Girshick, K.-M. He and P. Dollár, "Focal loss for dense object detection," 2017 IEEE International Conference on Computer Vision (ICCV), 2017, pp. 2999-3007.

[13] B. Tang, L. Chen, W. Sun and Z.-K. Lin, "Review of surface defect detection of steel products based on machine vision," IET Image Process, vol.17, pp. 303-322, 2022.

[14] T. Ohkubo, N. Terada, Y. Yoshida, "Preliminary scanning fluorescence detection of a minute particle running along a waveguide implemented microfluidic channel using a light switching mechanism," Microsystem Technologies, vol. 22, pp. 1227-1240, 2016.

[15] J. Zhao, J.-F. Huang, R. Wang, H.-R. Peng, W. Hang and S. Ji, "Investigation of the optimal parameters for the surface finish of k9 optical glass using a soft abrasive rotary flow polishing process," Journal of Manufacturing Processes, vol. 49, pp. 26–34, 2020.

[16] S.-W. Liu, Y.-H. Sun, M. Gu, C.-D. Liu, L.-S. He and Y.-H. Kang, "Review and analysis of three representative electromagnetic NDT methods," Insight, vol. 59, no.4, pp. 176-183, 2017.

[17] S. Choudhary, A. Kumar, S. Ganguly, M.r Laru and E. Z. Chacko, "Application of Thermodynamics in Mitigating Wire Rod Chipping During Hot Rolling of Continuously Cast Steel Billets," ISIJ International, vol. 58, no. 10, pp. 1811–1819, 2018.

[18] M. Dombrowski, J. Labelle, D. McGaw and M. Broughton, "An autonomous receiver/digital signal processor applied to ground-based and rocket-borne wave experiments," Journal of Geophysical Research: Space Physics, vol. 121, no.7, pp. 7334 – 7343, 2016.

[19] S. Shu, C.-M. Yu, C. Liu, M.-W. Chen, Y.-Z. Zhang and X. Li, "Improved plasma position detection method in EAST Tokamak using fast CCD camera," Nuclear Science and Techniques, vol. 30, no. 24, pp. 67–76, 2019.

[20] M. Hossam, H. M. Ebeid, M. Abdel-Aziz and M. Tolba, “Accelerated hyperspectral image recursive hierarchical segmentation using GPUs, multicore CPUs, and hybrid CPU/GPU cluster,” Journal of Real-Time Image Processing, vol.14, pp. 413–432, 2019.

[21] X.-F. Lv and J.-H. Xu, “Steel Plate Surface Defect Detection Method - Based on Improved YOLO Algorithm,” 2023 China Automation Congress (CAC), 2023, pp. 1331-1335.

[22] Q.-T. Zeng, D.-B. Wei, X. Zhang, Q. Gan, Q.-H. Wang and M.-H. Zou, “MFAM-Net: A Surface Defect Detection Network for Strip Steel via Multiscale Feature Fusion and Attention Mechanism,” 2023 International Conference on New Trends in Computational Intelligence (NTCI) 1, 2023, pp. 117-121.

[23] S.-Y. Luo, B. Jia, J. Zhao and Q.-S. Niu, “Surface Defect Detection Method for Steel Materials Based on Improved YOLOv7,” 2023 8th International Conference on Image, Vision and Computing (ICIVC), 2023, pp. 194-199.

[24] C.-H. Song, J.-X. Chen, Z. Lu, F. Li and Y.-Y. Liu, “Steel Surface Defect Detection via Deformable Convolution and Background Suppression,” IEEE Transactions on Instrumentation and Measurement, vol.72, pp. 1-9, 2023.

[25] J.-Q. Li, Q.-W. Gu, Y.-D. Chen and D. He, “Steel Surface Defect Detection Method Based on Improved YOLOv4-tiny,” 2023 International Conference on Image Processing, Computer Vision and Machine Learning (ICICML), 2023, pp. 1049-1054.

[26] Y. Yu, S.-X. Chan, T.-L. Tang, X.-L. Zhou, Y. Yao and H.-K. Zhang, “Surface Defect Detection of Hot Rolled Steel Based on Attention Mechanism and Dilated Convolution for Industrial Robots,” Electronics, vol. 12, no. 8, pp. 1856-1869, 2023.

[27] X.-Y. Hao, T. Dong, D.-H. Zhang, “A Highly Efficient Surface Defect Detection Approach for Hot Rolled Strip Steel Based on Deep Learning,” 2021 6th International Conference on Robotics and Automation Engineering (ICRAE), 2021, pp. 318-322.

[28] Q. Sun and B. Sheng, “Research on defect detection algorithm of strip steel based on improved YOLOv4,” International Conference on Artificial Intelligence and Computer Engineering (ICAICE 2022), vol. 12610, 258508531, 2023.

[29] R. Yan, R.-Y. Zhang, J.-Q. Bai, H. Hao, W.-J. Guo, X.-Y. Gu and Q. Liu, “STMS-YOLOv5: A Lightweight Algorithm for Gear Surface Defect Detection,” Sensors (Basel, Switzerland), vol. 23, no. 13, pp. 5992-6009, 2023.

[30] M.-H. Guo, C.-G. Lu, Q.-B. Hou, Z. Liu, M.-M. Cheng and S.-Y. Hu, “SegNeXt: Rethinking Convolutional Attention Design for Semantic Segmentation,” Neural Information Processing Systems, 2022, Available: https://doi.org/10.48550/arXiv.2209.08575

[31] X. Zhang, C. Liu, D.-G. Yang, T.-T. Song, Y.-C. Ye, K. Li and Y. Song, “RFAConv: Innovating Spatial Attention and Standard Convolutional Operation,” 2023, Available: https://doi.org/10.48550/arXiv.2304.03198

[32] R. Sunkara and T. Luo, “No More Strided Convolutions or Pooling: A New CNN Building Block for Low-Resolution Images and Small Objects,” ECML/PKDD, 2022, Available: https://doi.org/10.48550/arXiv.2208.03641

[33] Y.-H. Tang, K. Han, J.-Y. Guo, C. Xu, C.-T. Xu and Y.-H. Wang, “GhostNetV2: Enhance Cheap Operation with Long-Range Attention,” Computer Science, 2022, Available: https://doi.org/10.48550/arXiv.2211.12905

[34] Y. He, K.-C. Song, Q. Meng, Y.-H. Yan, “An End-to-End Steel Surface Defect Detection Approach via Fusing Multiple Hierarchical Features,” IEEE Transactions on Instrumentation and Measurement, vol. 69, no. 4, pp. 1493-1504, 2020.

[35] C-Y. Wang, A. Bochkovskiy and H. Liao, “YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors,” 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022, pp. 7464-7475.

[36] L.-H. Dai, Y.-S. Li and R. Shi, “YOLOv8-SSDW: YOLOv8-based surface defect detection algorithm for strip steel,” Journal of Chongqing Technology and Business University (Natural Science Edition), 2024, Available: http://kns.cnki.net/kcms/detail/50.1155.N.20240204.1531.002.html

[37] Y.-Y. Zou, Y.-F. Cao, X.-Y. Zhang, Z. Li and S.-L. Cui, “Algorithm for steel surface defect detection based on YOLOv8-DSG,” Journal of Jilin University (Information Science Edition), 2024, Available: https://doi.org/10.19292/j.cnki.jdxxp.20240517.010

[38] H.-L. Li, Z.-Q. Yi, L.-Y. Mei, J. Duan, K.-M. Sun, M.-C. Li, W. Yang and Y. Wang, “SCFNet: Lightweight Steel Defect Detection Network Based on Spatial Channel Reorganization and Weighted Jump Fusion,” Processes, vol. 12, no. 5, pp. 931-949, 2024.

[39] G.-H. Zhang, S.-X. Liu, S.-Q. Nie and L.-B. Yun, "YOLO-RDP: Lightweight Steel Defect Detection through Improved YOLOv7-Tiny and Model Pruning," Symmetry, vol.16, no. 4, pp. 458-472, 2024.

[40] Z. Cheng, L.-P. Gao, Y. Wang, Z.-H. Deng, Y. Tao, "EC-YOLO: Effectual Detection Model for Steel Strip Surface Defects Based on YOLO-V5," IEEE Access, vol. 12, pp. 62765-62778, 2024.