Steel Surface Defect Detection Algorithm Based on S-YOLOv8

Xu Zhang, Wenhua Cui, Ye Tao, Tianwei Shi

***Abstract*—Steel, being a widely utilized material in industrial production, holds a pivotal role in ensuring product safety and longevity. Hence, the exploration and implementation of steel surface defect detection technology carry significant importance. This paper introduces a steel surface defect detection algorithm based on S-YOLOv8. The algorithm, rooted in YOLOv8n as a benchmark model, initially incorporates a shift-wise shift operator in the backbone network. This introduction notably enhances accuracy compared to conventional CNN models while markedly reducing computational demands. Furthermore, the utilization of the SF-Neck framework, integrating the scale sequence feature fusion module (SSFF) and triple feature encoder module (TFE) in the head network, enriches the network's multi-scale information extraction capabilities. Subsequently, the adoption of the WIoU loss function enhances the overall detector performance. Lastly, the integration of the SEAM occlusion attention module refines the detection head segment of the YOLOv8 algorithm, effectively addressing defect occlusion challenges. Experiments conducted on the NEU-DET dataset reveal that the mAP value of the S-YOLOv8 model reaches an impressive 84.2%. Comparative analysis with other mainstream algorithms demonstrates a substantial enhancement in detection accuracy, alongside a reduction in instances of leakage and misdetection. Consequently, this study charts a new technical trajectory for quality control within the steel manufacturing industry.**

***Index Terms*—Steel surface Defect detection, S-YOLOv8, SWC2f, WIoU, SEAM.**

I. Introduction

As an essential construction material, Steel is widely used in industrial production and construction. As the global economy advances and industrialization accelerates, the demand for strip steel continues to grow. However, during the production and processing stages, steel often manifests various surface defects such as oxidation, cracks, pits, and bubbles. These flaws not only detract from the visual appeal of the steel but can also compromise its mechanical strength and resistance to corrosion. More critically, these defects could lead to product failure. Therefore, the prompt and accurate identification and evaluation of imperfections on the steel surface are of paramount importance[1].

The detection of defects on the surface of steel commonly employs several methods. Manual Visual Inspection is one such method that relies on an inspector's visual observation during the inspection process. However, this approach is often subject to issues of subjectivity, inefficiency, and a higher likelihood of errors. Another method, Optical Microscope Inspection, magnifies the steel surface using an optical microscope, enabling the identification of small defects. Despite its efficacy, this method necessitates the expertise of skilled professionals to operate the equipment. Magnetic Particle Flaw Detection (MPFD) is a technique that involves applying magnetic powder to the steel surface, followed by a magnetic field to detect defects such as surface cracks. However, its application is restricted to specific types of defects. Ultrasonic Inspection, on the other hand, employs ultrasonic technology to scan the steel surface, detecting both internal and surface defects with high sensitivity and accuracy. This method, while effective, requires skilled operators and comes with high equipment costs. Lastly, Thermal Infrared Imaging involves scanning the steel surface with an infrared thermographic camera to detect surface temperature anomalies and defects. However, the resolution of this technique may be constrained by the equipment, potentially impeding the accurate detection of small defects[2].

With the advancement of computer vision and image processing technologies, automated surface defect detection technology based on image processing and machine learning has emerged as a significant research area. Scholars have been actively exploring defect detection using deep learning-based target detection algorithms[3].

In this field, two prevalent strategies include single-stage and two-stage algorithms. Single-stage algorithms, such as YOLO (You Only Look Once)[4] and SSD (Single Shot MultiBox Detector)[5], make direct predictions about the target category and location from the image. Conversely, two-stage algorithms, including R-CNN[6], Fast R-CNN[7], and Faster R-CNN[8], first generate candidate regions, followed by classification and localization of targets within these regions. For instance, Liu et al[9] presented a multi-scale contextual strip surface defect detection network named MSC-DNET. This network leverages an enhanced inflated convolutional parallel architecture to capture multi-scale contextual information. Moreover, a feature enhancement and selection module is used to strengthen the identification of single-scale features and effectively utilize multi-scale features, thereby preventing information overload and confusion. Yushang Weng et al[10] proposed an enhanced version of the Mask R-CNN algorithm, which includes the k-means II clustering algorithm to improve the RPN anchor frame generation method. They also adjusted the model structure by removing the mask branches to boost detection accuracy. Lu Yao et al[11] introduced a defect detection model based

Manuscript received July 31, 2024; revised January 13, 2025. The work was supported by the Department of Education of Liaoning Province (LJKFZ20220197).

Xu Zhang is a Postgraduate of School of Computer Science and Software Engineering, University of Science and Technology Liaoning, Anshan, China. (e-mail: zhang41864512@163.com).

Wenhua Cui is a Professor of School of Computer Science and Software Engineering, University of Science and Technology Liaoning, Anshan, China. (Corresponding author to provide phone: +86-133-0422-4928; e-mail: taibeijack@126.com).

Ye Tao is an Associate Professor of School of Computer Science and Software Engineering, University of Science and Technology Liaoning, Anshan, China. (e-mail: taibeijack@163.com).

Tianwei Shi is an Associate Professor of School of Computer Science and Software Engineering, University of Science and Technology Liaoning, Anshan, China. (e-mail: tianweiabbcc@163.com).

on Cascade RCNN. This model employs switchable null convolution in place of ordinary convolution, thereby expanding the sensory field of the output unit. Furthermore, the feature pyramid is augmented through top-down connections. The model also incorporates the up-sampling operator CARAFE to enhance localization accuracy and up-sampling precision, resulting in improved detection accuracy. Wu Shan et al[12] proposed an advanced SSD network that constructs bottom-up down-sampling paths and top-down up-sampling paths. This design enhances the semantic information of spatial features and introduces an attention mechanism module to amplify the expressive power of feature fusion. Yanting Ma et al[13] proposed an improved MT-YOLOv5 algorithm. This algorithm integrates a Transformer self-attention mechanism module and a BiFPN network structure to enhance the extraction of image feature information, thereby achieving superior results. Huang et al[14] put forward an improved YOLOv8 algorithm, which replaces the C2F module with the GhostNetv2 module to enhance model representation. They also employ a progressive feature pyramid structure to facilitate more effective feature fusion across non-adjacent levels, thereby enhancing feature extraction capabilities and accelerating training. Kebin Cui et al[15] proposed an MCB-FAH-YOLOv8 algorithm for steel surface defect detection. This algorithm features an improved CBAM attention mechanism and a replaceable four-head ASFF prediction head to enhance detection accuracy. LiMing Liang et al[16] proposed an improved DCD-YOLOv8n algorithm, which enhances network accuracy by using a multi-branch feature aggregation network and a cross-dimension aggregation module. It also adopts a deformable multi-head attention mechanism to effectively handle complex defect features.

To summarize, the two-stage algorithms exhibit high detection accuracy but slower detection speeds, making them less suitable for real-time monitoring. On the other hand, single-stage algorithms offer faster detection but with slightly lower accuracy. To address the need for real-time monitoring while enhancing defect detection accuracy, this paper proposes an algorithm based on the improved YOLOv8 network. The key innovations of this paper are as follows: Introducing the shift-wise shift operator: By integrating this operator into the YOLOv8[17] network backbone, the algorithm significantly enhances the accuracy of regular CNNs while reducing computational requirements. SF-Neck framework: This framework incorporates the Scale Sequence Feature Fusion Module(SSFF) and Triple Feature Encoder Module(TFE) in the header network. It utilizes the Path Aggregation Network(PANet) structure to fuse multi-scale feature mappings extracted from the backbone network, thereby enhancing the network's ability to extract information across multiple scales. Adoption of WIoU loss function: This loss function addresses the issue of imbalanced sample quality in defective samples, thereby enhancing the overall detector performance. SEAM occlusion attention module: This module is employed to enhance the detection head part of the YOLOv8 algorithm, effectively addressing the challenge of defect occlusion and improving detection accuracy in such scenarios.

II. Algorithm Design

YOLOv8 represents the latest iteration in the YOLO (You Only Look Once) series, incorporating a new network architecture and advanced technologies to enhance accuracy and efficiency in target detection tasks. The YOLOv8 network comprises the backbone network, neck, and head network, each playing a distinct role in the detection process. YOLOv8 utilizes the CSPDarkNet backbone network for extracting image features. Notably, it replaces the original C3 module with the C2f module, leading to a significant reduction in parameters.This modification, coupled with improved gradient flow, results in enhanced convergence speed and accuracy. Responsible for predicting target locations and categories, the detection head in YOLOv8 introduces innovations such as SPP (Spatial Pyramid Pooling) and PAN (Path Aggregation Network) modules. These modules enhance the network's ability to perceive targets across different scales and facilitate effective feature fusion. YOLOv8 adopts a decoupled-head structure, separating the classification and detection heads. Additionally, it transitions from Anchor-Based to Anchor-Free methodology, contributing to improved detection performance. YOLOv8 offers models with varying scales (N/S/M/L/X) based on scaling coefficients. In this paper, the YOLOv8n network structure is selected for enhancement, aligning with the specific goals of the research. Overall, YOLOv8's incorporation of novel modules, improved network structures, and anchor-free methodology signifies a significant advancement in target detection capabilities, catering to a wide range of applications requiring high accuracy and efficiency. To enhance the detection accuracy of the steel surface defect detection algorithm, this paper introduces an algorithm specifically designed for detecting small targets, named S-YOLOv8. The structure of the S-YOLOv8 network is illustrated in Figure 1.

A. SWC2f module

Large convolutional kernels can enhance the sensory domain's scope and significantly improve detection accuracy. However, conventional large convolutional kernels are not hardware-friendly operators, leading to compatibility issues with hardware platforms due to their high parameter count and computational complexity. Merely enlarging the convolution kernel size is not advisable. Instead, a small convolution kernel and operation can simulate the effects of a larger kernel, achieving a similar impact with fewer resources. In this study, the SWC2f module introduces the shift-wise shift operator, which acts as a small convolution kernel to mitigate the drawbacks of using large convolution kernels. By incorporating the shift-wise shift operator into the YOLOv8 network's backbone, it replaces the convolution kernel of the Bottleneck in C2f. This strategic use of the shift-wise shift operator optimizes computational efficiency while maintaining detection accuracy.

The large convolution kernel is deconstructed into a series of standard small convolution kernels, with a shift operation applied to each convolution to achieve the equivalent operation of the large convolution kernel. This decomposition through transformations is illustrated in equation (1).

$$\begin{gathered} y(p_{(i,j)}) = \sum_{k=0}^{\lceil \frac{kw}{kh} \rceil} \sum_{m=0}^{kh} \sum_{n=0}^{kh} w(p_{(\Delta m,\Delta n)} + \Delta p) \\ \cdot x(p_{(i,j)} + p_{(\Delta m,\Delta n)} + \Delta p) \\ \Delta m = m - \frac{kw}{2}; \Delta n = n - \frac{kh}{2} \\ \Delta p = kh * k, (k \in \left[0, \lceil \frac{kw}{kh} \rceil\right]) \end{gathered} \tag{1}$$

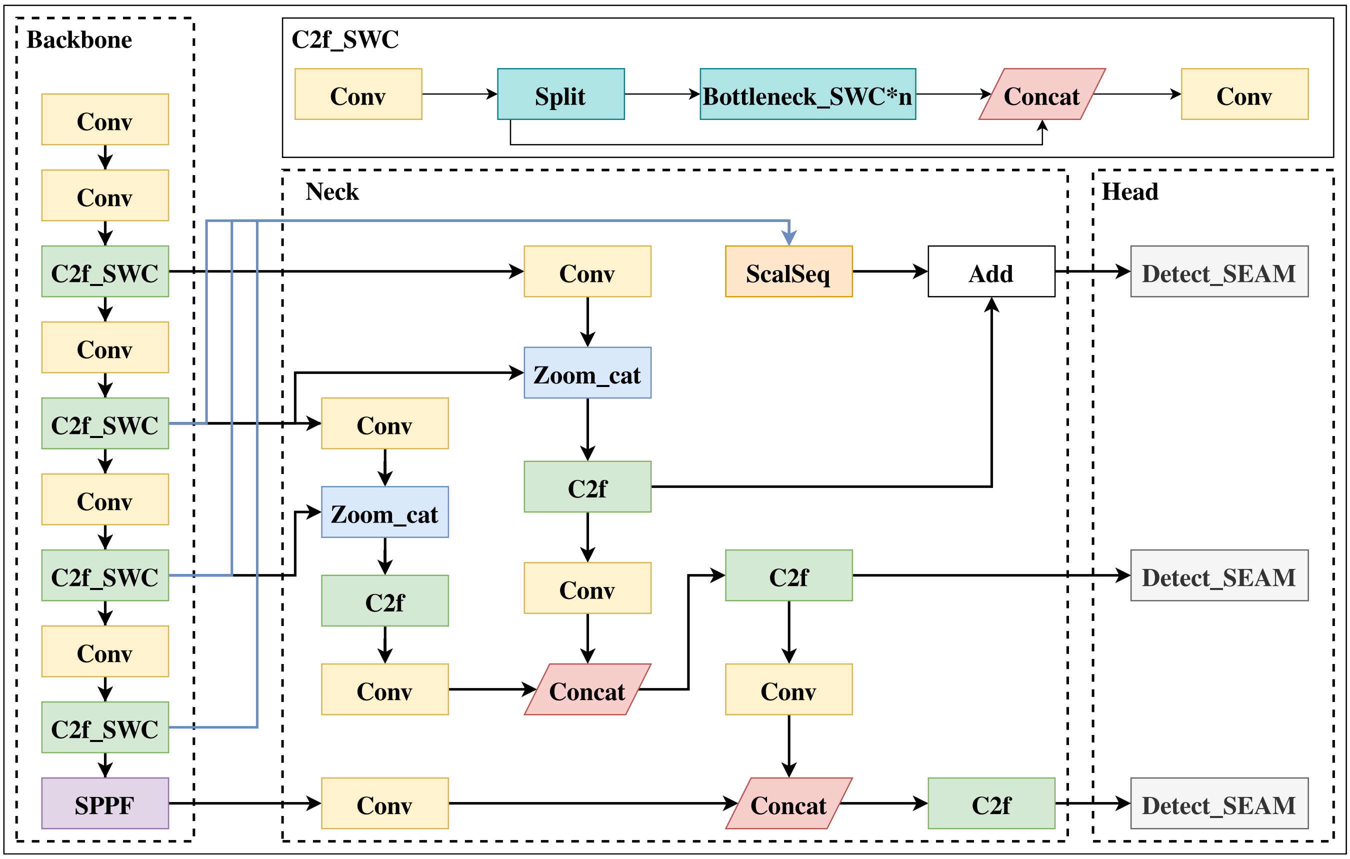

Fig. 1: S-YOLOv8 network structure diagram

In this context, the notation (i,j) indicates the position of the sliding window on the feature map, with kw and kh representing the dimensions of the convolution kernel. The symbol p denotes positional information, while w and × refer to the weight values and feature values at their respective positions. For generalization purposes, it is assumed that kw ¿ kh. A large convolution kernel can be effectively replaced by several standard small convolution kernels, which requires alignment adjustments for parameters such as offset control and padding settings. The module structure is illustrated visually in Fig. 2, where different colored blocks highlight the substitution relationships. For example, a 15 × 3 convolution can be equivalently represented as five 3 × 3 convolutions. After this substitution, a shift operation is necessary for the convolution, which must extend further along one dimension and align with a grid of size kh.

Large convolutional kernels can introduce long-range dependencies in the feature space. However, certain details are frequently neglected, leading to the adoption of pruning during the training process to eliminate certain connections. Through coarse-grained pruning, a sparse group convolution is achieved. Addition operations are then employed to maintain a constant total output across the module's channels.

The continuous optimization of dependencies within the data flow, while preserving the overall network architecture, is encapsulated in the concept of the group shift operation. The holistic structure of the shift-wise shift operator is illustrated in Fig. 3.

Initially, multiple output branches are created by performing group shift operations on the same inputs, simulating different convolution kernel sizes. Next, a single channel is sampled from each group to form an identity branch. Ultimately, all output branches are merged into a single unified branch. By introducing new concepts of focus length

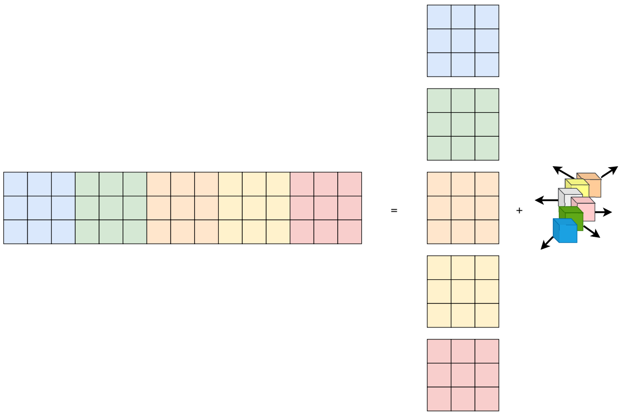

Fig. 2: Structure of the large convolution kernel decomposition

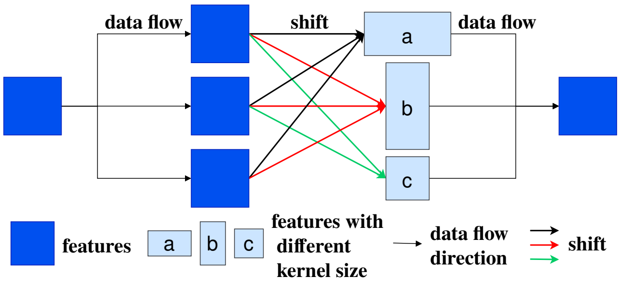

Fig. 3: Overall structure of shift-wise shift operator

and focus width, which arise from offset and sparsity considerations, the shift-wise shift operator method delineates a rectangular area defined by the specified focus length and focus width. It then selects convolution kernels that are equal to or smaller than the focus width, modifying the arrangement of these convolutional groups to enhance information fusion. The focus length is required to be at least as large as the current feature map's size. Following these modifications, Equation (2) is derived from Equation (1).

$$\begin{gathered}
\mathrm{y}(p_{(i,j)}) = \prod_{k=0}^{f(kw,kh,A)} \sum_{m=0}^{A} \sum_{n=0}^{A} w(p_{(\Delta m,\Delta n)} + \Delta p) \\
\cdot x(p(i,j) + p_{(\Delta m,\Delta n)} + \Delta p) \\
\Delta m = m - \tfrac{A}{2}; \Delta n = n - \tfrac{A}{2} \\
\Delta p = g(kh,k), (k \in [0, f(kw,kh,A)])
\end{gathered} \tag{2}$$

In this context, A denoting the size of the minor convolution kernel. The terms kw and kh refer to the focal length and focal width, respectively, while $f(kw,kh,A)$ signifies the function associated with (kw,kh,A). Similarly, the weights and feature offsets p are functions linked to (kh,k), denoted as g.

To streamline computational efforts during training, The 'ghost' and reparameterization (REP) techniques are incorporated into the shift-wise module. This module comprises convolution operations with k sets of kernel sizes n × n, alongside corresponding shift-wise operations, where k is an integer part of m/n. Assuming the input feature sizes are denoted as B, C, H, and W, representing batch, channel, height, and width, respectively, the computational complexity of the shift-wise operation is outlined in Equation (3).

$$\begin{gathered}
n * n * H' * W' * k * C * B \\
+add(H * W * k * C * 2) \\
C_{shift} = (k * n) * n * H * W * C * B + \delta
\end{gathered} \tag{3}$$

In the output of the group convolution, the unit branch randomly selects C feature samples, resulting in a total of kC channels in the group convolution. Two branches then reorganize the features within the group convolution output. The two branches with larger convolution kernels perform a shift operation followed by feature summation. All C channels undergo the identical shift operation. The "add" function defines the required shift and addition processes, with the cost of the addition operation represented as δ. This spatially sparse dependency reduces the time expense, and as a result, the shift-wise shift module considerably lowers both the number of parameters and the computational complexity of the convolution operation.

B. ST-Neck Framework

To improve the network's ability to extract multi-scale feature information and enhance the model's performance in detecting small targets, the Scale Sequence Feature Fusion (SSFF) module and the Triple Feature Encoder (TFE) module are incorporated into the Neck section of the YOLOv8 model. The SSFF module merges the semantics of images at different scales by normalizing, up-sampling, and feeding the multi-scale sequence features into a 3D convolutional information unit. The TFE module comprises three different sizes of feature maps to better capture fine-grained object information across various scales, with a focus on leveraging the smaller feature maps.

1) SSFF Module: To address the multi-scale challenge in steel surface images, the SSFF scale sequence feature fusion module is employed. This module effectively combines high-level information from deep feature maps with semantic details from shallow feature maps. thereby boosting the neural network's capability to extract features across various scales. The scale space is established along the scale axis of the image, capturing both the scale specifics of the image and hinting at the potential scale range of an object. Even for blurred images where fine details may be obscured, the fundamental feature structure of the image remains intact. Sequential representations of multi-scale feature maps (such as P4, P6, and P8) derived from the backbone are assembled, each encapsulating distinct scale information of the image contents. The scaled image, which serves as the input to SSFF, is depicted in equation (4).

$$\begin{gathered}
F_\sigma(w,h) = G_\sigma(w,h) \times f(w,h) \\
G_\sigma(w,h) = \tfrac{1}{2\pi\sigma^2} e^{-(w^2+h^2)/2\sigma^2}
\end{gathered} \tag{4}$$

In this context, $f(w,h)$ denotes a two-dimensional feature map characterized by a width of w and a height of h. $F_\sigma(w,h)$ is produced through a sequence of convolutional smoothing operations utilizing a two-dimensional Gaussian filter $G_\sigma(w,h)$, where σ acts as the scaling factor for the standard deviation of the two-dimensional Gaussian filter used in the convolution operation.

The resulting images have varying resolutions and scales. Feature maps with different scale sizes are treated as their respective scale spaces, and the effective feature maps with varying resolutions are standardized to a uniform resolution for alignment. The feature maps at different scales are aligned horizontally, and their scale sequence features are extracted using 3D convolution. The output feature maps display varying resolutions due to Gaussian smoothing. The high-resolution feature map at the P4 level retains most of the essential information for detecting small targets; thus, the SSFF module is designed based on the P4 level, as shown in Fig. 4. The proposed SSFF module consists of the following five components.

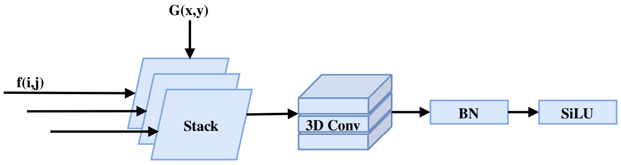

Fig. 4: Structure of SSFF module

Adjusted the channel count of the P6 and P8 layer feature levels to 256 using 1×1 convolution. Resized the feature maps of the P6 and P8 layers to align with the dimensions of the P4 level using nearest neighbor interpolation[18]. Employed the unsqueeze method to increase the dimensions of each feature map, converting them from a three-dimensional tensor [height, width, channels] to a four-dimensional tensor [depth, height, width, channels]. The resulting 4D feature maps were then concatenated along the depth axis to form 3D feature maps for subsequent convolution operations. Finally, scale sequence feature extraction was performed using 3D convolution, 3D batch normalization, and the SiLU[19] activation function.

2) TFE structure: Since steel surface defects primarily consist of small-sized targets that are densely overlapped, the changes in shape and appearance across different sizes can be compared by zooming into the image. Considering the varying dimensions of feature layers within the backbone network, the conventional FPN fusion approach primarily upsamples the smaller feature maps and subsequently merges or adds them to the previous feature layer. This process neglects the valuable information present in the larger feature layers. To remedy this, the TFE module has been introduced to categorize features into large, medium, and small groups, consolidate the large feature maps, and utilize a feature enhancement technique to enrich the detailed feature information. The design of the TFE module is illustrated in Fig. 5.

The structural diagram of the TFE module features three distinct sizes of input feature maps, with C representing the number of channels and S signifying the size of the feature maps. Prior to feature encoding, the channel count is adjusted to align with the primary scale features. The large-scale feature map (Large) is processed using a convolution module, which modifies its channel count to 1C. Subsequently, a hybrid approach that integrates max pooling and average pooling is employed for downsampling, effectively reducing spatial dimensions and bolstering the network's robustness to spatial variations and translations in the image.

Likewise, the small-scale feature map (Small) undergoes processing through a convolution module to adjust the channel count, followed by upsampling using nearest neighbor interpolation. This method helps preserve local features and mitigate the loss of information related to small targets.

Finally, three feature maps of different sizes with the same dimensions undergo convolution once, after which the features are merged using the Concat operation to generate a comprehensive feature representation. This process is illustrated in equation (5).

$$F_{TFE} = Concat(F_l, F_m, F_s) \tag{5}$$

In this context, FTFE signifies the feature mapping output produced by the TFE module, while Fl, Fm, and Fs denote the feature mappings of large, medium, and small sizes, respectively. FTFE is created by concatenating Fl, Fm, and Fs, ensuring it retains the same resolution as Fm, but with three times the number of channels compared to Fm.

C. WIoU Loss Function

Target detection, as the fundamental issue in computer vision, relies heavily on the design of the loss function for its detection performance. This function serves to evaluate the detection efficacy between predicted and actual detection frames. YOLOv8 employs CIoU as its regression loss function, which considers the overlap area and aspect ratio of target frames. It introduces a correction factor to enhance the accuracy of similarity assessment between these frames. However, the description of aspect ratio is somewhat ambiguous, and the issue of directional mismatch between real and predicted frames is overlooked. This omission results in slow model convergence and reduced prediction accuracy.

Due to the quality imbalance in steel surface defect samples, this study employs the WIoU loss function as the regression loss criterion. The WIoU is designed based on the bounding box loss framework of the dynamic non-monotonic focusing mechanism. It incorporates the idea of ”outlier degree” to assess the quality of anchor frames and introduces an improved gradient gain distribution strategy. These improvements significantly boost model detection accuracy.

WIoU constructs a distance attention mechanism grounded in metric distance and yields WIoUv1 with a two-layer attention mechanism, as illustrated in Equation (6).

$$\begin{gathered} L_{WIoUv1} = R_{WIoU} L_{IoU} \\ R_{WIoU} = exp(\frac{(x-x_{gt})^2+(y-y_{gt})^2}{(W_g{}^2+H_g{}^2)^*}) \end{gathered} \tag{6}$$

Where L_{IoU} represents the ratio occupied by the intersection range of the prediction and target frames, R_{WIoU} will significantly amplify the L_{IoU} of normal quality anchor frames. L_{WIoU} significantly reduces the L_{WIoU} loss of high-quality anchor frames and focuses on the center point when the anchor and target frames are well overlapped. W_g and H_g denote the width and length sizes of the minimum closure frames, (x, y) indicate the relationship between each point of the anchor frame and the target frame corresponding to the position (x_{gt}, y_{gt}), and * denotes the separation operation. Moreover, to prevent large harmful gradients that may lead to low sample quality, small gradient gains are allocated to the large outliers. The -constructed non-monotonic aggregation coefficient is utilized to formulate WIoUv3, and the WIoUv3 formula is presented in Equation (7).

$$L_{WIoUv3} = rL_{WIoUv1}, r = \frac{\beta}{\delta\alpha^{\beta-\delta}} \tag{7}$$

In the case where (r=1) and (β=δ), the gradient gain of the anchor frame reaches its maximum when the outlier condition for that frame satisfies (β=C), with C being a constant value. WIoUv3 provides a reduced gradient gain for low-quality anchor frames, thereby effectively minimizing harmful gradients and enhancing the performance of the target detection model. This approach improves the model's generalization ability and accelerates its convergence speed through a weighting mechanism and dynamic adjustment of the weights.

D. SEAMHead

Defect occlusions on steel surfaces can result in alignment errors, local aliasing, and missing features. In this study, we introduce the SEAM attention module, integrated into the detection head, to address the diminished response of occluded defects. The module aims to enhance the response of unoccluded defects. The SEAM module primarily comprises a fusion of depth-separable convolution and residual concatenation. Depth-separable convolution operates on a channel-by-channel basis, this approach focuses on understanding the importance of different channels while minimizing the number of parameters. However, it fails to consider the relationships between channels. To remedy this, the outputs from various depth-separable convolutions are merged using pointwise (1×1) convolution. Following this, a two-layer fully connected network integrates information from each channel to enhance connections across all channels. The structure of the SEAM module is illustrated in Fig. 6.

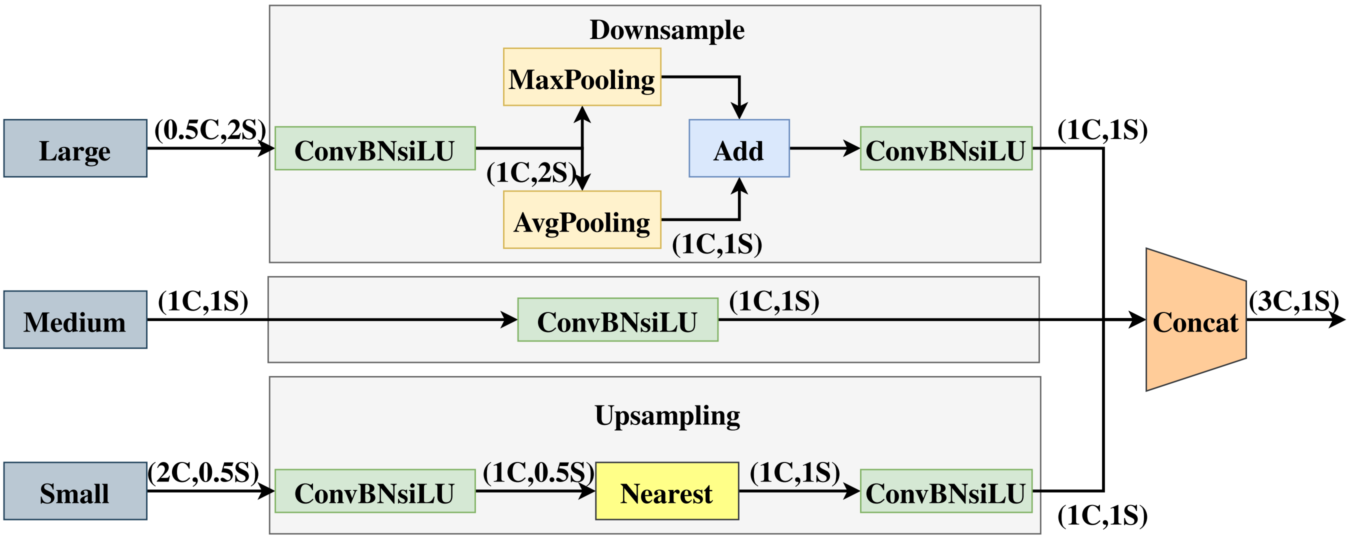

Fig. 5: Structure of TFE module

III. Experiments

A. Experimental Dataset

The NEU-DET dataset is a compilation of steel surface defect images developed by Northeastern University. It includes six prevalent types of steel surface defects: cracking (Cr), inclusions (In), patches (Pa), pitting surfaces (PS), rolled oxide (RS), and scratches (Sc). The dataset contains a total of 1,800 grayscale images, with 300 samples for each defect category. Each image is annotated to indicate both the type of defect and its location. An example from the dataset is shown in Fig. 7.

B. Experimental Environment

In this study, the experimental configuration utilized the Windows 10 operating system, an Intel(R) Core (TM) i7-10700F CPU operating at 2.90GHz, and an NVIDIA GeForce RTX 3070 graphics card featuring 8GB of video memory. The coding was carried out using the PyCharm integrated development environment, with PyTorch version 1.12.1 serving as the deep learning framework. The development environment was based on Python 3.8, and graphics acceleration was facilitated by CUDA version 11.6. The experimental parameters were set as follows: a learning rate of 0.01, a batch size of 16, and a total of 150 iterations. During model training, the Mosaic data augmentation technique was employed on the input data to process the images. This technique involved scaling and merging four random images to enhance the model's capability to detect small targets, thereby boosting the performance and robustness of the network model. All experiments in this research were conducted within the same environment to train the model, compare performance metrics, and validate the effectiveness of the model enhancements.

C. Evaluation Indicators

To more effectively evaluate the performance of the enhanced model, this paper utilizes key metrics including mAP (mean Average Precision), number of parameters (Params), computational load (GFLOPs), and floating-point operations as criteria for assessment. mAP is an essential metric that reflects the average precision for detecting all target categories within the dataset. It integrates precision (P) and recall (R) to provide a comprehensive evaluation of the model's effectiveness. Precision (P) denotes the fraction of samples predicted as positive by the model that are actually positive, while recall (R) indicates the proportion of all true positive samples that are correctly identified. Average precision (AP) computes the average precision value for each category of defective targets. The formula for Average Precision (AP) is as follows:

$$P = \frac{TP}{TP + FP} \tag{8}$$

$$R = \frac{TP}{TP + FN} \tag{9}$$

$$f_{AP} = \int_0^1 p(R)dR \tag{10}$$

$$mAP = \frac{1}{n}\sum_{i=1}^{n} AP_i \tag{11}$$

In this context, TP refers to the count of actual positive samples that are accurately classified as positive, while FP indicates the number of actual negative samples that are mistakenly classified as positive. FN signifies the count of actual positive samples that are incorrectly labeled as negative. Here, n represents the total number of defect categories, and i denotes the number of detections.

D. Experimental Results and Analysis

To assess the effectiveness of the proposed algorithm and the influence of each enhancement on model performance, this paper conducts four sets of ablation experiments using YOLOv8 as the baseline model. The four experimental groups are designated as YOLOv8-1, YOLOv8-2, YOLOv8-3, and YOLOv8-4. YOLOv8-1 incorporates the SWC2f module. YOLOv8-2 combines the SWC2f module with the ST-Neck structure. YOLOv8-3 integrates the SWC2f module, the ST-Neck structure, and the WIoU loss function. Finally,

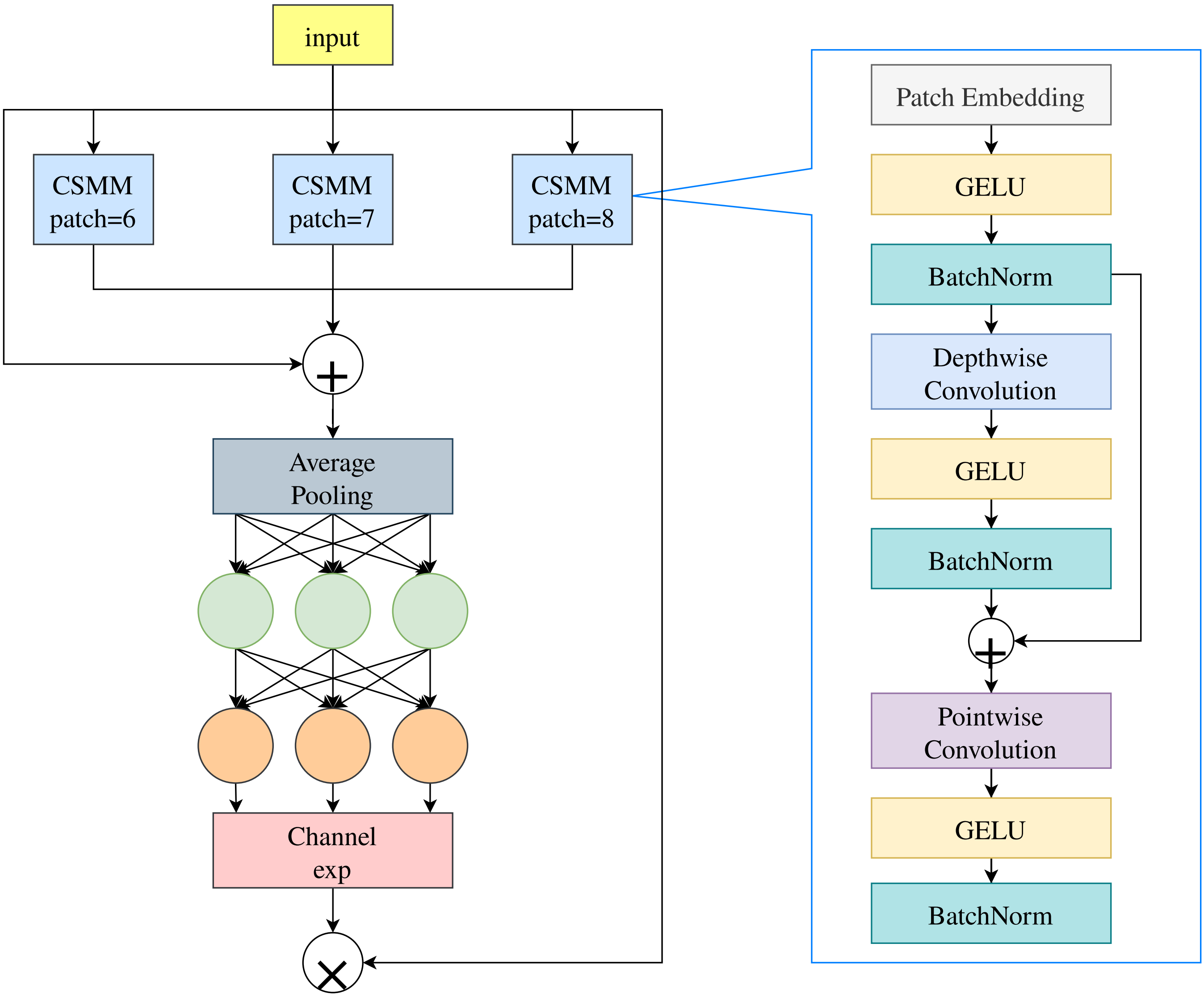

Fig. 6: SEAM module structure

YOLOv8-4 includes the SWC2f module, the ST-Neck structure, the WIoU loss function, and the addition of the SEAM attention mechanism in the Head. The experimental findings are summarized in Table I.

TABLE I: Results of ablation experiments

Models	SWC2f	ST-Neck	WIoU	SEAMHead	mAP	Params
YOLOv8n	×	×	×	×	79.5	3.15
YOLOv8-1	✓	×	×	×	82.2	3.18
YOLOv8-2	✓	✓	×	×	83.1	3.22
YOLOv8-3	✓	✓	✓	×	83.5	3.22
YOLOv8-4	✓	✓	✓	✓	84.2	3.03

As depicted in Table 1, the original YOLOv8n model exhibits a mAP value of 79.5% with a model parameter count of 3.15M. Introducing the SWC2f module in YOLOv8-1 yields a mAP value of 82.2% compared to the YOLOv8n algorithm, marking a 2.7% enhancement over the original model, and a parameter count of 3.18M. This underscores the positive impact of the SWC2f module in bolstering the algorithm's accuracy. In the case of YOLOv8-2, which incorporates both the SWC2f module and the ST-Neck structure simultaneously, achieves a mAP value of 83.1% compared to YOLOv8n, representing a 3.6% increase over the original model, with a parameter count of 3.22M. This indicates that the fusion of these two structures leads to a more pronounced enhancement in the algorithm's detection efficiency, further elevating the detection accuracy. Moving on to YOLOv8-3, which introduces the SWC2f module, ST-Neck structure, and WIoU loss function concurrently, attains a mAP value of 83.5% compared to YOLOv8n, a 4% improvement over the original model, with a parameter count of 3.22M. This demonstrates that replacing the CIoU loss function with the WIoU loss function alongside integrating the SWC2f module and ST-Neck structure will significantly boost the algorithm's detection accuracy. Lastly, by jointly introducing the SWC2f module, ST-Neck structure, WIoU loss function, and integrating the SEAM attention mechanism into the Head, the enhanced algorithm achieves a mAP value of 84.2%, a 4.7% increase over the original model, with a parameter count of 3.03M. This highlights that the SEAM attention mechanism can replace the CIoU loss function when introduced alongside the SWC2f module and ST-Neck structure, further enhancing the algorithm's detection accuracy. Integrating the SEAM attention mechanism into the Head section of the detection head further improves the

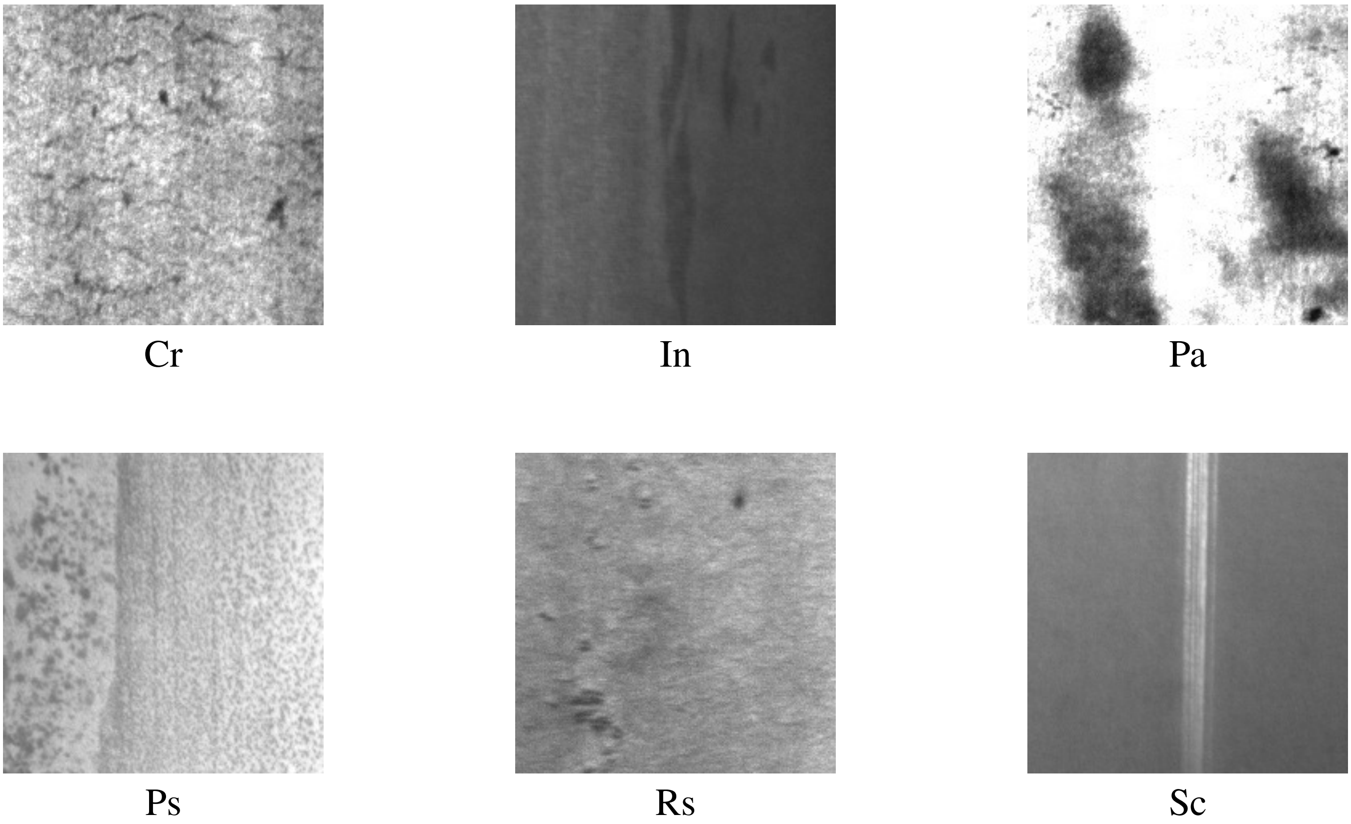

Fig. 7: Sample plot of the dataset

algorithm's detection accuracy. In conclusion, the algorithm proposed in this paper significantly enhances the defect detection accuracy of the model, particularly enhancing the feature extraction capability for small target defects and improving the detection of defective features in small targets.

A visual comparison of the experimental results between the enhanced S-YOLOv8 model and the original YOLOv8 model is presented in Fig. 8. In the figure, (a) represents the outcomes obtained using the original YOLOv8n algorithm for detection, while (b) illustrates the results obtained using the enhanced S-YOLOv8 algorithm proposed in this paper. Above each marked box in the figure, the defect category and the corresponding confidence level are labeled. The experiments reveal that while the original model can accurately identify defect categories, it often exhibits low confidence levels and missed detections. In contrast, the improved algorithm demonstrates higher detection accuracy in identifying defective targets. Additionally, the enhanced model displays an additional marked box for detecting cracked defects, indicating that the improved algorithm also exhibits performance enhancements in leakage detection.

The table depicts the average accuracy of each type of defect before and after the algorithm improvement (Table II). Specifically, the average accuracy of cracking-type defects has increased by 22.1%, inclusions-type defects by 0.4%, plaque-type defects by 2.4%, pitting surface-type defects by 4%, rolling oxidized skin-type defects remain almost unchanged, and scratches-type defects have seen an improvement of 3.5%.

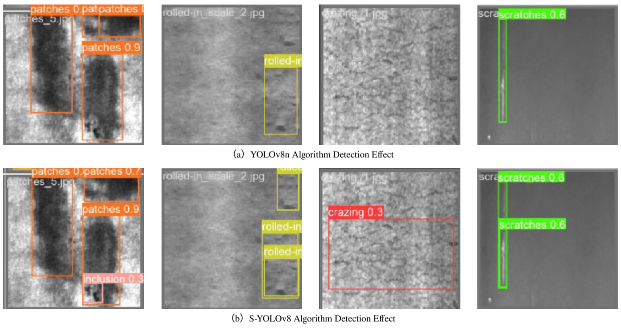

Fig. 8: Comparison chart for visualization of experimental results

After comparison, the S-YOLOv8 algorithm proposed in this paper effectively enhances the average detection accuracy of various types of defects, with the most notable improvement observed in the cracking class of defects. This highlights the efficacy of the enhanced algorithm in detecting defects in small targets.

The algorithm proposed in this paper has been compared with current mainstream models such as SSD, YOLOv5n, YOLOv7, CenterNet[20], YOLOv8n, MT-YOLOv5, DCN-YOLOv5, MCB-FAH-YOLOv8[15]. In comparative experiments as depicted in Table 3. Following the analysis, the mAP value of the algorithm in this paper exhibited an 11% improvement over the SSD algorithm, a 13% enhancement over the YOLOv5n algorithm, an 8.7% increase compared to the YOLOv7 algorithm, a 9.7% rise compared to the CenterNet algorithm, a 4.7% boost compared to the YOLOv8n algorithm, a 2.4% advancement compared to the MCB-FAH-YOLOv8 algorithm, a 1.8% progress compared to the MT-YOLOv5 algorithm, and a 7.1% progress compared to the DCN-YOLOv8n algorithm. In comparison to other mainstream algorithms, S-YOLOv8 displayed a significant en-

TABLE II: Average accuracy of each type of defect before and after algorithm improvement

Defect type	YOLOv8n	S-YOLOv8
Cr	63.3	81.3
In	68.4	68.8
Pa	95.6	98
Ps	95.5	99.5
Rs	71.7	71.4
Sc	82.5	86

hancement in detection accuracy, accompanied by a notable reduction in parameter quantity. In conclusion, S-YOLOv8 showcases superior detection performance when contrasted with other models.

TABLE III: Comparison of algorithms

Models	mAP	FPS	Params
SSD	73.2	47.8	26.29
YOLOv5n	71.2	163.9	1.77
YOLOv7	75.5	42.4	36.51
CenterNet	74.5	37.4	32.67
YOLOv8n	79.5	137	3.15
MCB-FAH-YOLOv8	81.8	101	6.06
MT-YOLOv5	82.4	65.4	29.7
DCD-YOLOv8n	77.1	188	2.5
S-YOLOv8	84.2	138	3.03

IV. Conclusion

A novel approach based on S-YOLOv8 is introduced to tackle the issue of steel surface defect detection. This method builds upon the original YOLOv8n model, incorporating a shift-wise operator in the backbone network, which significantly decreases both the parameter count and the computational complexity of convolution operations. Additionally, the SF-Neck framework integrates the Scale Sequence Feature Fusion (SSFF) module and the Triple Feature Encoder (TFE) module within the head network. The multiscale feature mapping extracted from the backbone is combined in the Path-Aggregation Network (PANet) structure, enhancing the model's capability to extract multiscale information. The WIoU loss function is utilized to accelerate convergence and improve accuracy. Moreover, the SEAM occlusion attention module is implemented to bolster the detection head of the YOLOv8 algorithm, effectively addressing the challenge of defect occlusion. The NEU-DET strip surface defect dataset is employed for ablation experiments and comparative analyses. Experimental results reveal that the mAP value of the S-YOLOv8 algorithm reaches 84.2%, representing a 4.7% improvement over the original YOLOv8n algorithm. The effectiveness and feasibility of the proposed algorithm are confirmed, leading to a reduction in instances of leakage and misdetection during defect identification. Future research may focus on further optimizing the model to minimize the number of parameters, enhance detection speed, and maintain accuracy.

References

[1] Zhang Yan, Feng Feng. Exploration of surface defect detection technology of strip steel[J]. Information and Computer: Theoretical Edition, 2021, 33(11): 19-22.

[2] SONG Yubin, KONG Weibin, CHEN Xi, et al. A review of research on steel surface defect detection[J]. Software Guide,2024,23(03):203-211.

[3] Yongzhong Fu, Liang Qiu, Xiao Kong, and Haifu Xu, "Deep Learning-Based Online Surface Defect Detection Method for Door Trim Panel," Engineering Letters, vol. 32, no. 5, pp939-948, 2024

[4] REDMON J, DIVVALA S, GIRSHICK R, et al. You only look once: unified, real-time object detection[C]// Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Piscataway, NJ: IEEE, 2016:779–788.

[5] LIU W, ANGUELOV D, ERHAN D, et al. SSD: single shot multibox detector[C]//Proceedings of the 14th European Conference on Computer Vision Amsterdam. The Netherlands: Springer, 2016: 21-37.

[6] GIRSHICK R, DONAHUE J, DARRELL T, et al. Rich feature hierarchies for accurate object detection and semantic segmentation[C]// Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Piscataway, NJ: IEEE, 2014:580–587.

[7] R. Girshick, Fast R-CNN, 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 2015, pp. 1440-1448.

[8] REN S, HE K, GIRSHICK R, et al. Faster R-CNN: towards real-time object detection with region proposal networks[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017, 39(6): 1137-1149.

[9] Liu R Q, Huang M, Gao Z M, et al. MSC-DNet: an efficient detector with multi-scale context for defect detection on strip steel surface[J]. Measurement, 2023, 209: 112467.

[10] WENG Yushang, XIAO Jinqiu, XIA Yu. Strip surface defect detection based on improved Mask R-CNN algorithm [J]. Computer Engineering and Applications, 2021,57(19):235-242.

[11] LU Yao, XUE Lin, WANG Yun-sen, et al. Surface defect detection of hot-rolled strip based on Cascade RCNN[J]. Instrumentation Technology and Sensors, 2023(08): 101-106+126.

[12] Wu Shan,Zhou Feng.Small target detection based on improved SSD algorithm[J].Computer Engineering, 2023,49(7):179-188,195.

[13] Ma Yan-ting, Zhao Hong-dong, Yan Chao, et al. Strip steel surface defect detection method by improved YOLOv5 network[J]. Journal of Electronic Measurement and Instrumentation, 2022,36(08):150- 157.

[14] Huang M, Cai Z. Steel surface defect detection based on improved YOLOv8[C]//International Conference on Algorithms, High Performance Computing, and Artificial Intelligence (AHPCAI 2023). SPIE, 2023, 12941: 1356-1360.

[15] K. B. Cui, J. Y. Jiao. Steel surface defect detection algorithm based on MCB-FAH-YOLOv8[J/OL]. Journal of Graphics.

[16] LiMing Liang, Kangquan Chen, Yi Zhong, Pengwei Long, Yao Feng. DCD-YOLOv8n: An efficient algorithm for surface defect detection in steel [J]. Computer Engineering and Applications, 1-12.

[17] Ziqiang Lin, Lijun Zhu, Jinyu Zhang, Yuanhang Zhang, and Xudong Liu, "Research on Improving YOLOv5s Algorithm for Defect Detection in Cylindrical Coated Lithium-ion Batteries," Engineering Letters, vol. 32, no. 7, pp1521-1528, 2024.

[18] D.J. Lowe, Distinctive image features from scale-invariant keypoints, Int. J. Comput. Vis. 60 (2004) 91110.

[19] S. Elfwing, E. Uchibe, K. Doya, Sigmoid-weighted linear units for neural network function approximation in reinforcement learning, Neural Netw. 107 (2018) 3–11.

[20] K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang and Q. Tian, CenterNet: Keypoint Triplets for Object Detection, 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea (South), 2019, pp. 6568-6577.