applied sciences

MDPI

Article

Steel Surface Defect Detection Based on YOLOv8-TLC

Chunling Liu and Hui Cheng *

College of Information Engineering, Dalian University, Dalian 116000, China; liuchunling@dlu.edu.cn
* Correspondence: chenghui@s.dlu.edu.cn; Tel.: +86-139-0968-2478

Abstract: To address the issues of low accuracy and efficiency in traditional image processing algorithms for steel surface defect detection, a novel steel surface defect detection algorithm based on YOLOv8-TLC is proposed. To more accurately detect defect targets in images that are missed due to their large size, an additional scale detection layer is introduced. Meanwhile, the Large Selective Kernel (LSK) attention mechanism is incorporated to deeply explore spatial structural information that is highly relevant to the steel surface defect targets, further enhancing the model's spatial feature extraction capabilities. A triple spatial pyramid module is also constructed to address the problem of redundant feature extraction. Additionally, the C2f-DS module is designed to ensure the acquisition of richer gradient flow information without increasing the number of parameters. Experimental results on the NEU-DET dataset show that the YOLOv8-TLC algorithm achieves a mean average precision (mAP) of 79.8%, improving the mAP by 3.2% while enhancing detection speed.

Keywords: defect detection; steel; YOLOv8; Triple Space Pyramid; C2f-DS module

check for updates

Citation: Liu, C.; Cheng, H. Steel Surface Defect Detection Based on YOLOv8-TLC. *Appl. Sci.* **2024**, *14*, 9708. https://doi.org/10.3390/app14219708

Academic Editor: Douglas O'Shaughnessy

Received: 30 September 2024
Revised: 13 October 2024
Accepted: 18 October 2024
Published: 24 October 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

In recent years, many researchers have used machine vision and image processing technologies to detect objects in images. By identifying the types of objects and determining their locations within the image, they achieve classification and localization of various targets. However, traditional object detection methods are easily influenced by the detection environment. While they can perform well under specific conditions, in real production environments, factors like lighting and object orientation can affect the accuracy and speed of detection. Moreover, traditional methods rely on manually designed features, requiring extensive prior knowledge and cumbersome procedures. This process is not only complex but also computationally heavy, making it unsuitable for real-time detection. Since metal surface defects are usually detected on production lines to manage defective products, there is a high demand for real-time detection, which traditional object detection algorithms struggle to meet.

With the rapid development of deep learning, object detection technology has made significant progress. Deep learning, by simulating the way the human brain processes information, can automatically learn and extract features from images, enabling efficient object recognition and localization. This approach not only enhances the accuracy and robustness of algorithms but also simplifies the detection process, reducing reliance on prior knowledge. Using deep learning for object detection offers numerous advantages. In the early days, Krizhevsky et al. proposed AlexNet [1] and achieved remarkable success in the ImageNet (ILSVRC) competition, marking a major milestone for deep learning in computer vision tasks and shifting the focus of object detection towards deep learning techniques. Object detection methods based on deep learning are mainly divided into two branches: two-stage detectors and one-stage detectors. Two-stage detectors first generate candidate object regions through a region proposal network, followed by fine classification and location adjustments of these regions. This approach achieves high accuracy but at the cost of speed. In contrast, one-stage detectors adopt an end-to-end approach, directly

predicting object categories and locations across the entire image, offering superior speed, and making them more suitable for real-time object detection.

By integrating deep learning with region proposal techniques, RCNN [2,3] emerged as a significant approach. This method converts the object detection task into a region-based classification problem, offering simplicity and scalability. However, due to frequent convolution operations, it suffers from substantial redundant computations, limiting its performance. Girshick et al. further improved RCNN by introducing Fast RCNN [4], which enhanced detection accuracy and sped up the detection process, though the time-consuming region extraction issue remained unsolved. Building upon this, Ren et al. introduced Faster RCNN [5], significantly improving model accuracy, but the issue of inconsistent region proposal box sizes in the region proposal network still persists.

Single-stage object detectors directly use deep convolutional neural networks on the raw image to predict the locations and classes of objects, eliminating the need for a region proposal step, which significantly increases detection efficiency. To address the high computational complexity, slow speed, and difficulties in handling different scales and targets associated with traditional object detection methods, Liu et al. introduced the SSD [6] model, which employs multi-scale features and data augmentation techniques to better adapt to objects of varying sizes, further enhancing the generalization capability of object detection. To continue improving the efficiency of object detection tasks, Redmon et al. proposed YOLO [7,8] at CVPR 2016. Subsequently, they introduced Anchor Boxes and additional convolutional layers, resulting in YOLOv2 and YOLO9000 [9]. YOLOv3 [10] added context information modules and feature pyramid networks, proposing a new backbone network called Darknet53, which further improved the detection accuracy of multi-scale objects. To further enhance detection efficiency and accuracy, Alexey et al. proposed YOLOv4 [11] using CSPDarknet53 [12] and PAN [13], while Ultralytics introduced YOLOv5 [14], which operates at an inference speed of 140 FPS, enabling faster and more accurate model deployment. To simplify the model design and training process, Facebook was the first to propose DETR [15]. The team later combined deformable convolutional networks with Transformer networks to propose RT-DETR [16]. Addressing the issue of poor model generalization caused by manual setting of positive and negative samples in the YOLO series, Zheng et al. introduced YOLOX [17]. To meet the needs of industrial applications, Meituan integrated many advanced object detection designs from both industry and academia, resulting in YOLOv6 [18]. Subsequently, Ultralytics released a model named YOLOv8, which further enhanced the model's performance and flexibility to meet diverse market demands.

2. Related Works

Steel strips play an important role in various fields of industrial production, and their quality is crucial for industrial development. However, due to limitations in production processes and technology, defects such as cracks, inclusions, and spots often appear on the surface of steel. As a result, surface defect detection is particularly important. Currently, steel surface defect detection technology has evolved from traditional manual and machine-vision-based methods to deep learning-based techniques. Traditional manual inspection methods suffer from low accuracy, missed detections, and false positives, while machine-vision-based detection methods typically only extract shallow features of surface defects and are easily influenced by environmental factors, leading to unstable detection performance. Therefore, it is essential to employ deep learning-based object detection algorithms for detecting surface defects in steel strips. In 2022, Guo Zexuan et al. proposed the MSFT-YOLO [19] algorithm, which introduces the TRANS module based on YOLOv5 and combines a multi-scale feature fusion structure to achieve the integration of features at different scales, enhancing the detector's ability to dynamically adjust to targets of varying sizes.

In 2023, Zhao Zemin [20] introduced the Self Notice Ratio (SNR) deep network attention model, combining it with YOLOv6 for surface defect detection in rolled steel.

Their method inputs images processed by YOLOv6 into a U-Net convolutional model to confirm the presence of defects and accurately detect their positions. The final output is a binary mask that overlays the original image, achieving the detection goal. However, the detection time is relatively long, requiring further lightweight optimization. Xie et al. [21] did not directly use a lightweight convolutional network but instead designed the DSConv lightweight convolutional module to optimize the backbone of YOLOv4. Li Fengrun [22] proposed a novel approach for model lightweighting—a dual-stream network structure—dividing a single path into deep and shallow branches. The optimized YOLOv4 model had a parameter count reduced to 18.4 M, significantly lower than SSD and YOLOv4, successfully achieving model lightweighting. However, the model requires high image resolution, and detection speed may decrease at lower resolutions. Li Shaoxiong [23] proposed an improved method based on YOLOv5m, significantly enhancing the precision of steel surface defect detection, though the model's parameter count is relatively large. Wang et al. developed an improved YOLOv5-based algorithm, creating a multi-scale block module [24] that effectively explores steel surface defects at different resolutions.

By 2024, Ren et al. [25] optimized YOLOv5 by introducing the ECA mechanism, which focuses on channel information while neglecting spatial information, thus being suboptimal for detection tasks. Their model achieved a mAP of 78.8% in steel surface defect detection, but the accuracy for detecting less visible defects was relatively low. Li et al. [26], based on the introduction of the spatial attention (SA) mechanism, integrated the CSPCrossLayer module into the YOLOX backbone, establishing cross-layer connections. They also introduced the PSblock convolutional module to replace the CSPLayer structure in the feature fusion network. Experiments on the railway track dataset showed that this algorithm improved detection speed while maintaining accuracy, though the performance for small defect detection in the NEU-DET dataset was suboptimal. Gao Chunyan et al. [27], to save computational resources, abandoned the FPN optimization of YOLOv7 and proposed a high-precision real-time defect detection algorithm, CDN-YOLOv7. This algorithm introduced the CARAFE lightweight upsampling operator to improve feature fusion ability, though CARAFE is more suited to scenarios requiring high-quality upsampling and devices with limited resources. Fan et al. [28] designed a new anchor point optimization algorithm, GA-K-Means, based on the K-means algorithm, and proposed the ACD-YOLO model. By using a Genetic Algorithm (GA) to adjust the anchor point boundaries, they reduced dependencies. Although the model achieved 72 FPS in experiments, its detection speed still lagged behind YOLOv7 and YOLOv8, requiring further optimization. To address the problem of aluminum surface defect detection, Kong et al. [29] introduced the WIoU loss function based on YOLOv8, focusing on anchor boxes of normal quality to reduce harmful gradients generated by extreme samples. By using a convolutional residual module to enhance feature extraction capabilities, the model's mAP improved by 3.4% compared to the original YOLOv8, reaching 74.9%, although the computational load increased by about 20%. Zhao Baiting [30] proposed the ECC-YOLO algorithm, which is based on YOLOv7 and incorporates the MPCE module, improving detection accuracy for small steel surface defects. Dai Linhua's C2f-DD algorithm [31] expanded the receptive field, effectively enhancing the network's feature extraction ability. Wang Mengyu [32] proposed an adaptive weighted downsampling method based on YOLOv8, which combined different downsampled feature maps to improve the model's focus on defect information.

The above examples provide valuable insights for subsequent research, but they generally suffer from insufficient detection accuracy. In response to these shortcomings and the requirements of defect detection, this paper selects YOLOv8n, the model with the smallest weights among the five versions of YOLOv8, as the base model and proposes an improved YOLOv8-TLC (YOLOv8-Triple Space Pyramid module and Large Selective Kernel attention and C2f-DS module) algorithm, which effectively enhances the accuracy of steel surface defect detection.

YOLOv8n Algorithm

YOLOv8n consists of three main components: a feature extraction network, a feature fusion network, and a detection head, as shown in Figure 1. This model is an optimized version of YOLOv5, not only improving detection accuracy but also achieving greater lightweight efficiency. YOLOv8n replaces the C3 module in YOLOv5 with a more lightweight C2f module. Additionally, the detection hea YOLOv8n adopts an Anchor-Free structure and utilizes a decoupled head design that separates classification and regression tasks, enhancing both the model's accuracy and convergence speed [33].

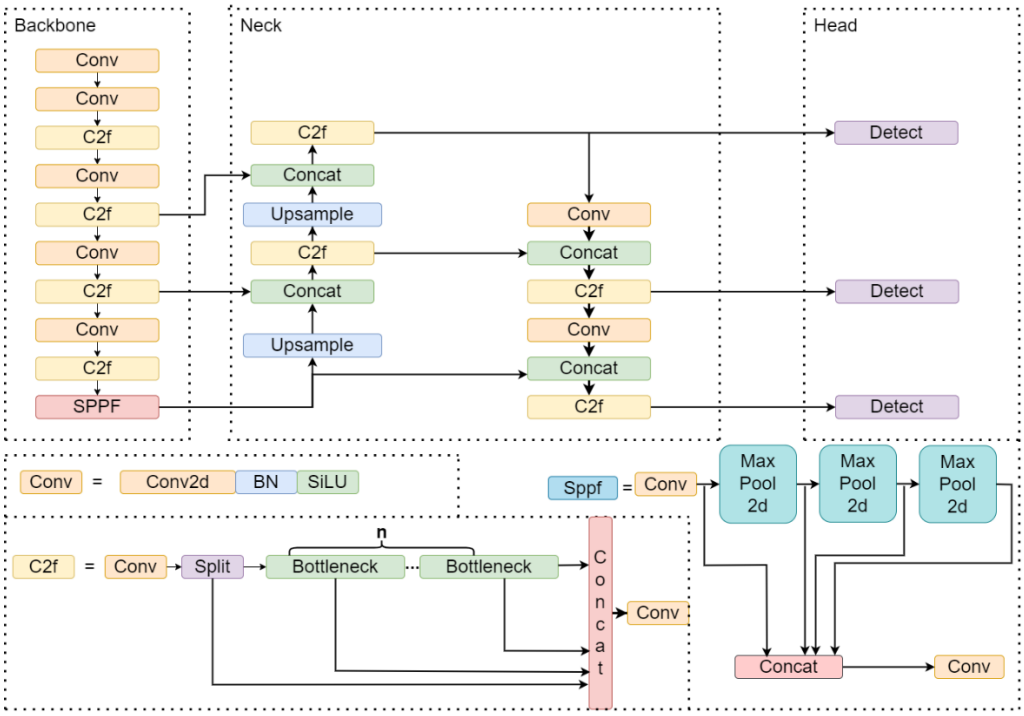


Figure 1. YOLOv8n model structure.

3. YOLOv8-TLC Algorithm

To address the issue of low detection accuracy in YOLOv8n for steel surface defect detection, this paper proposes the YOLOv8-TLC algorithm. To reduce the missed detection rate of steel surface defects, the YOLOv8-TLC algorithm adds a new scale detection head. It also incorporates the Large Selective Kernel (LSK) attention mechanism to enhance the focus on spatial regions most relevant to steel surface defects. Additionally, a Triple Space Pyramid (TSP) module is introduced, along with the design of a C2f-DS module, which improves the feature extraction capabilities of the backbone network and enhances the model's detection efficiency. The structure of the improved algorithm is shown in Figure 2.

3.1. New Detection Layer

To address the issue of missed detections caused by certain defects occupying a large proportion of the image in steel datasets, this paper introduces a new large-scale detection head into the network. The original YOLOv8n detection head includes three different detection layers with scales of 80 × 80, 40 × 40, and 20 × 20. The YOLO model partitions targets based on receptive field size: large-scale feature maps are used for detecting small objects, while small-scale feature maps are used for detecting large objects [34]. Since the sizes of various defects on steel surfaces differ, their scales when mapped onto images also vary. To improve YOLOv8n's detection accuracy for steel surface defects, this paper adds an extra-large scale detection layer, specifically targeting defects that occupy a large proportion of the image. This modification aims to reduce missed detections while enhancing the network's generalization ability. The size of the new detection layer is shown in Table 1.

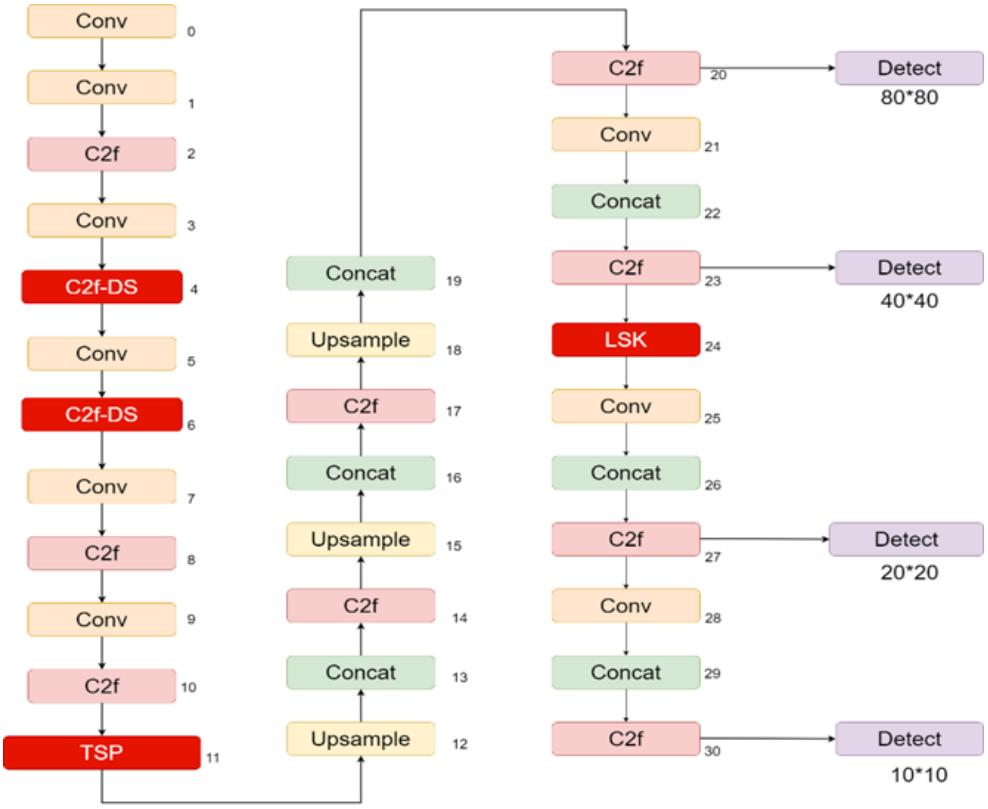


Figure 2. Structure diagram of YOLOv8-TLC model.

Table 1. Added dimensions of the detection layer.

Detecting Feature Map Size	**80 × 80**	**40 × 40**	**20 × 20**	**10 × 10 (Additional)**
Receptive fields	Small	Medium	Large	Extra Large

3.2. Large Selective Kernal-Attention

For steel surface defect images, the dataset consists of grayscale images, which results in unclear boundaries between defects and the background, causing them to blend together. This makes it difficult to distinguish between defect targets and the background, while also presenting a significant size variation among the targets. To address these issues, we introduce LSK-Attention into the feature fusion network to fully explore and integrate the spatial structural information needed for defect detection, further enhancing feature fusion.

Traditional spatial attention mechanisms have a relatively narrow range of background information when extracting spatial features, which does not guarantee the extraction of ideal spatial characteristics. The adaptive attention mechanism helps the model focus on important areas, thereby improving the accuracy and robustness of target detection. The LSK model adapts to the varying requirements for background information based on different types of targets by transforming larger convolution kernels into a sequence of different kernels, dynamically selecting relevant feature maps for spatial combination. This not only reduces the computational load but also enhances the model's adaptability to targets of different sizes, making it more efficient. The main idea of LSK-Attention is to effectively weigh and spatially fuse features across different receptive fields by utilizing spatial information at various scales, integrating both global and local features, thus better consolidating information and improving feature representation capability. The structure of LSK-Attention is shown in Figure 3.

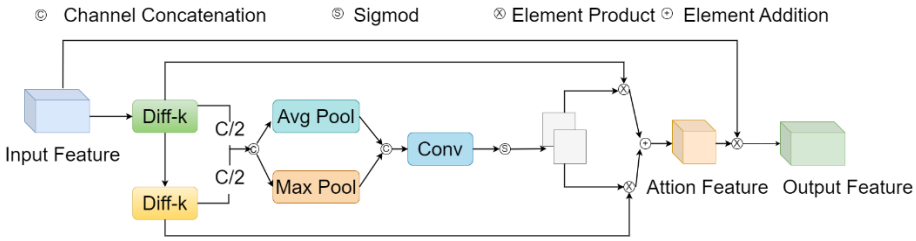


Figure 3. LSK-Attention structure diagram.

The main process of LSK-Attention [35] is as follows: it utilizes convolution kernels with varying receptive fields to extract rich background features from different regions of the input data X.

$$U_0 = X, U_1 = F^{5*5}(U_0), U_2 = F^{7*7}(U_1) \tag{1}$$

In this context, F^{5*5} represents the depth convolution operation with a kernel size of 5 and a dilation rate of 1, while F^{7*7} denotes the depth convolution operation with a kernel size of 7 and a dilation rate of 3. To accommodate the contextual information of different target types, LSK-Attention dynamically adjusts the receptive field to better match various targets and backgrounds. Additionally, it dynamically selects suitable convolution kernels based on the local information of the input feature maps. After each convolution operation, a 1×1 convolution layer is used to achieve channel fusion of the spatial feature vectors.

$$\widetilde{U}_i = F_i^{1*1}(U_i) \tag{2}$$

Then, the feature maps from different receptive fields are concatenated using Formula (3).

$$\widetilde{U} = [\widetilde{U}_1; \ldots\ldots; \widetilde{U}_i] \tag{3}$$

Next, channel-level average pooling P_{avg} and maximum pooling P_{max} are employed to extract spatial relationships.

$$SA_{avg} = P_{avg}(\widetilde{U}), SA_{max} = P_{max}(\widetilde{U}) \tag{4}$$

Here, SA_{avg} and SA_{max} represent the spatial feature descriptors obtained from average pooling and maximum pooling, respectively. Subsequently, convolution layer $F^{2\to N}$ is used to transform these descriptors into a spatial attention map, facilitating information interaction between the spatial descriptors. The mathematical expression is as follows:

$$\widehat{SA} = F^{2\to N}([SA_{avg}; SA_{max}]) \tag{5}$$

Next, the sigmoid activation function is applied to each spatial attention feature map $\widehat{SA}_i$ to obtain the spatial selection weights corresponding to each decoupled large convolution kernel.

$$\widetilde{SA}_i = \sigma(\hat{SA}_i) \tag{6}$$

The weighted deep convolution feature maps are then combined through element-wise weighted multiplication with the corresponding deep convolutions. Finally, a convolution layer is used to fuse these feature maps and produce the final attention features. The mathematical expressions are given in Formulas (7) and (8).

$$S = F(\sum_{i=1}^{N} (\widetilde{SA}_i \cdot \overline{U}_i)) \tag{7}$$

$$Y = X \cdot S \tag{8}$$

Here, F represents the convolution layer, and S represents the attention features.

3.3. Building the TSP Module

3.3.1. Triplet Attention

To better capture the semantic information of defects on the surface of steel and to balance detection accuracy with computational cost, we adopted the Triplet Attention (TA) mechanism proposed by Misra et al. [36] to extract more semantic information about defect features. The TA mechanism identifies important regions in the image through spatial attention, emphasizes specific features with channel attention, and focuses on information at different levels with depth attention. This allows the model to have a more comprehensive understanding of the image and to establish dependencies across different dimensions through rotation and residual operations, thereby more accurately identifying key features and enhancing the detection capability of steel surface defects. Additionally, the TA mechanism has a smaller number of parameters and supports cross-dimensional interaction, significantly improving training and inference speed. The implementation process of the TA attention mechanism can be represented by the following formula [37]:

$$y_1 = \hat{\chi_1}\sigma(\Psi_1(\hat{\chi_1^*})) \tag{9}$$

$$y_2 = \hat{\chi_2}\sigma(\Psi_2(\hat{\chi_2^*})) \tag{10}$$

$$y_3 = \chi\sigma(\Psi_3(\hat{\chi_3})) \tag{11}$$

$$\overline{y} = \frac{1}{3}(y_1 + y_2 + y_3) \tag{12}$$

In this process, σ represents the activation function, while Ψ_1, Ψ_2 and Ψ_3 denote the convolution operations in the three branches. The input feature map χ undergoes rotation operations in the top and middle branches, resulting in feature maps $\hat{\chi_1}$ and $\hat{\chi_2}$. These are then reduced in dimensionality through Z-Pool, yielding feature maps $\hat{\chi_1^*}$ and $\hat{\chi_2^*}$. In the bottom branch, the input feature map χ is also reduced in dimensionality via Z-Pool, producing feature map $\hat{\chi_3}$. Finally, the overall attention weight $\overline{y}$ is obtained by averaging the attention weights y_1 from the top branch, y_2 from the middle branch, and y_3 from the bottom branch.

3.3.2. TSP Module

In order to adapt to the scenario of steel surface defect detection, and to solve the problem of repeated feature extraction and the degradation of detection accuracy of traditional convolutional neural networks under the operations of cropping, flipping, and scaling that may be caused by the input of fixed-size images, this paper proposes a TSP module. Spatial Pyramid Pooling (SPP) [38], which was proposed by He et al., provides the basis for this research. On this basis, in order to further reduce the computation of the model and speed it up, Spatial Pyramid Pooling-Fast (SPPF) is proposed, which replaces the parallel pooling operation with different window sizes as the serial pooling operation with the same window size and improves the activation function. Cross-Stage Partial Connection (CSPC) [12] is a technique designed to enhance feature transfer and network efficiency. Introducing partial connections across different stages in the network accelerates feature propagation and better utilizes the information between low-level and high-level features. Inspired by SPPCSPC [39], the Triple-Space Pyramid module is proposed by combining the ability of TA to enhance important features for steel surface defects and the advantages of SPPF and CSPC. This module effectively addresses issues related to image distortion caused by cropping and scaling operations, as well as the problem of redundant feature extraction. Its structural diagram is shown in Figure 4.

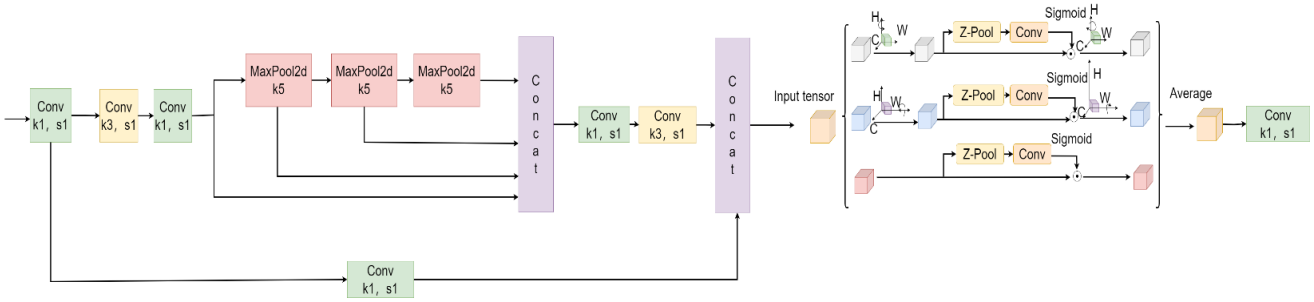


Figure 4. TSP structure.

3.4. C2f-DS Module

As the depth of the network increases and the number of parameters expands, the detection speed of steel surface defect models gradually slows down. To address this issue, the C2f-DS module is proposed, which achieves high computational speed with low storage requirements, thereby improving the overall efficiency of the defect detection model. In addition by incorporating a distributed shift mechanism, Distributed Shift Convolution (DSConv) [40] can model complex local feature distributions more accurately, thus significantly improving the feature representation capability of the model. DSConv is an efficient quantized convolution operator that decomposes the traditional convolution kernel into a Variable Quantized Kernel (VQK) and distributed shift components, aiming to reduce computational and storage overhead. The distributed shift components consist of two tensors: Kernel Distribution Shifter (KDS) and Channel Distribution Shifter (CDS). The framework is shown in Figure 5.

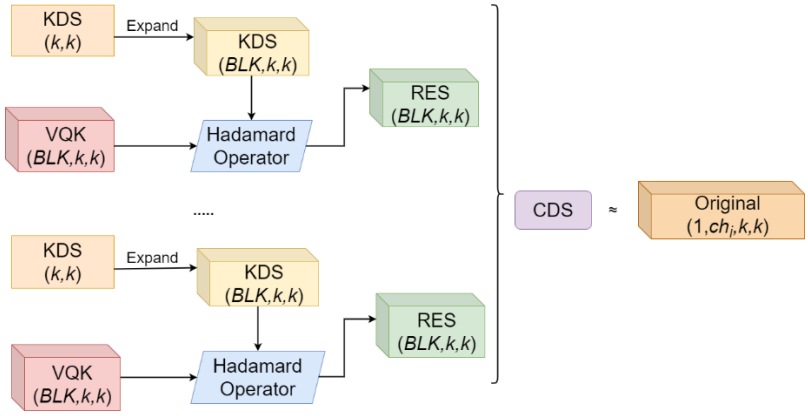


Figure 5. DSConv frame diagram.

During network pretraining, the weight tensor filters are divided by depth into blocks of length B, with each block sharing a single floating-point value. All these blocks are then quantized. The memory saved for each tensor using this method can be expressed by the following formula:

$$p = \frac{b}{32} + [\frac{C_i}{B}]/C_i \tag{13}$$

In the equation, C_i represents the number of channels, while b refers to the chosen hyperparameter setting.

Leveraging the advantages of DSConv in low storage and high computational speed, we designed the C2f-DS module, as shown in Figure 6. This module adjusts the spatial positions of the convolution kernels without increasing the number of parameters, which reduces the neglect of important information during the convolution process and enables more effective extraction of local features. This enhancement not only improves the utilization of computational resources in steel surface defect detection but also elevates the overall performance of the model.

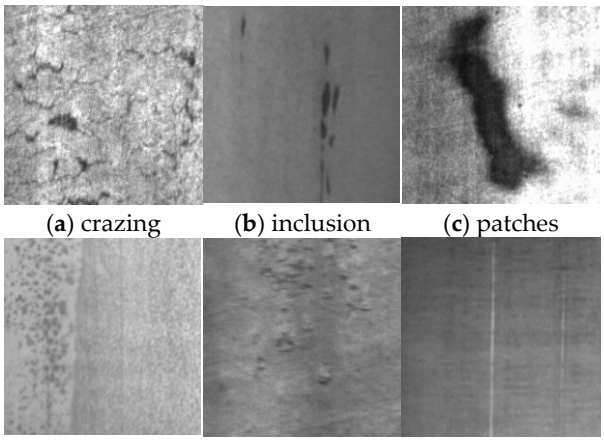


Figure 6. C2f-DS structure diagram.

4. Experimental Results and Analysis

4.1. Experimental Environment and Dataset

This study utilizes the NEU-DET [41] dataset provided by Northeastern University, which contains 1800 grayscale images covering various types of defects, including crazing (Cr), inclusion (In), patches (Pa), pitted surfaces (Ps), scratches (Sc), and rolled-in scale (Rs), with 300 samples for each type. Some examples of steel surface defects are shown in Figure 7. The dataset is divided into training, validation, and test sets in an 8:1:1 ratio.



(**a**) crazing (**b**) inclusion (**c**) patches

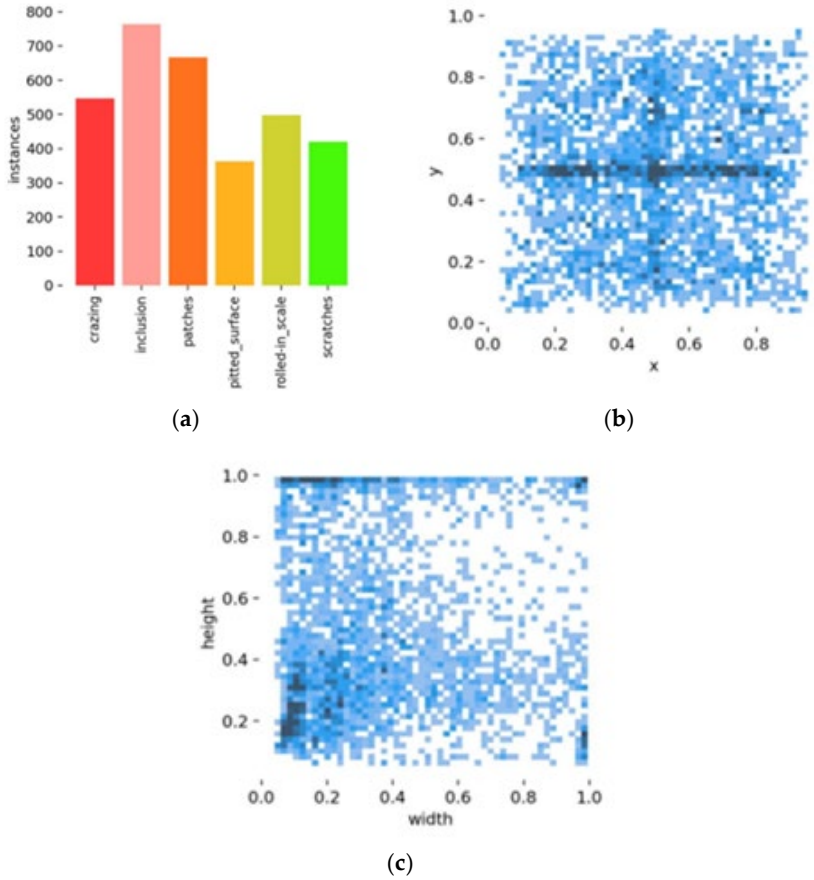
(**d**) pitted_surface (**e**) rolled-in_scale (**f**) scratches

Figure 7. Sample diagram of various types of defects.

To detect defects of different scales and various sizes, this paper analyzes the proportion of defect samples, the distribution of defect center coordinates within the images, and the aspect ratio distribution of the defects. As shown in Figure 8, the bar chart in Figure 8a demonstrates that the distribution of defect sample quantities is uneven, with significant disparities. To improve data quality and obtain a better-trained model, the mosaic data augmentation method was applied during training. This method randomly combines images from different categories, allowing multiple categories of objects to appear in a single training sample, thus making the dataset more balanced and mitigating the data imbalance problem to some extent. Figure 8b shows the distribution of defect center coordinates in the images, highlighting that defects are mainly concentrated in the central region of the images. Figure 8c illustrates the aspect ratio distribution of the defects, indicating that most defects have aspect ratios concentrated within a smaller range, although there are some defects with aspect ratios distributed over a larger range.

The configuration of the experimental environment in this paper is shown in Table 2.

The experiments in this paper used the default hyperparameter settings recommended by YOLOv8. To ensure the fairness of the experimental results, commonly used configurations from numerous related studies were referenced, and through experimental validation, the number of training epochs was set to 300. Based on the hardware capabilities, the batch size was set to 16. The specific hyperparameters are shown in Table 3.



(**a**) (**b**)

(**c**)

Figure 8. Dataset characteristics. (**a**) The distribution of defect sample quantities. (**b**) The distribution of defect center coordinates. (**c**) The distribution of defect aspect ratios.

Table 2. Experimental environment configuration.

Name	Parameters
GPU	Quadro RTX 8000, 48 GB of memory (NVIDIA, Santa Clara, CA, USA)
CPU	40 Intel(R) Xeon(R) Silver 4210R CPU @ 2.40 GHz (Intel, Santa Clara, CA, USA)
Operating System	Ubuntu 11.3.0
Programming Language	Python 3.10.9
Deep Learning Framework	Pytorch 2.0.1 + CUDA 12.2

Table 3. Experimental parameter settings.

Parameter Name	Parameter Value
Optimizer	SGD
Learning rate	0.01
Momentum	0.937
Weight decay coefficient	0.0005
Batch size	16
Number of training sessions	300

4.2. Performance Evaluation Indicators

To accurately validate the superiority of the proposed algorithm in this paper, the evaluation metrics used in the experiments include precision (P), measuring the probability that the positive samples identified by the classifier are actually positive; recall (R), indicating the classifier's ability to identify all true positive samples; average precision (AP),

the average precision for each category; and mean average precision (mAP), the average detection accuracy across all categories. Additionally, the number of parameters is used to objectively assess the model's performance, and frames per second (FPS) is used to evaluate the model's detection efficiency. The specific formulas for these metrics are provided in Equations (14)–(17) as shown below:

$$P = \frac{TP}{TP + FP} \tag{14}$$

$$R = \frac{TP}{TP + FN} \tag{15}$$

$$AP = \int_0^1 P(R)dR \tag{16}$$

$$mAP = \frac{1}{n}\sum_{i=0}^{n} AP(i) \tag{17}$$

In the above equations, TP is the true positive, FP is the true negative, and FN is the false negative.

4.3. Analysis of Realization Results

This paper conducts ablation studies and comparative experiments to visually demonstrate the impact of model structure on detection results. Additionally, the proposed YOLOv8-TLC model is compared with other detection algorithms to validate its effectiveness in detecting surface defects in steel. The training loss over the course of training is also presented.

4.3.1. Training Loss

The loss function of YOLOv8 consists of three components: classification loss (cls_loss), bounding box regression loss (box_loss), and distribution focal loss (dfl_loss). The classification loss measures the discrepancy between the predicted object classes and the true classes; the bounding box regression loss evaluates the similarity between the predicted bounding boxes and the ground truth boxes, improving the model's localization accuracy and enabling the predicted boxes to better encompass the target objects; the distribution focal loss enhances the accuracy of bounding box regression by modeling the distribution of each coordinate.

As shown in Figure 9, after approximately 50 epochs of training, significant performance improvements were achieved through substantial weight updates. Subsequently, the rate of decline for each loss gradually slowed and converged, making the detection task more accurate and efficient. The sudden drop in loss during the final 10 epochs was due to the disabling of the mosaic method at this stage.

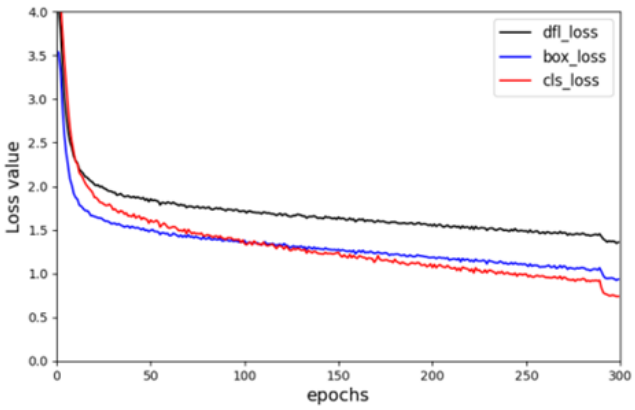


Figure 9. Training loss.

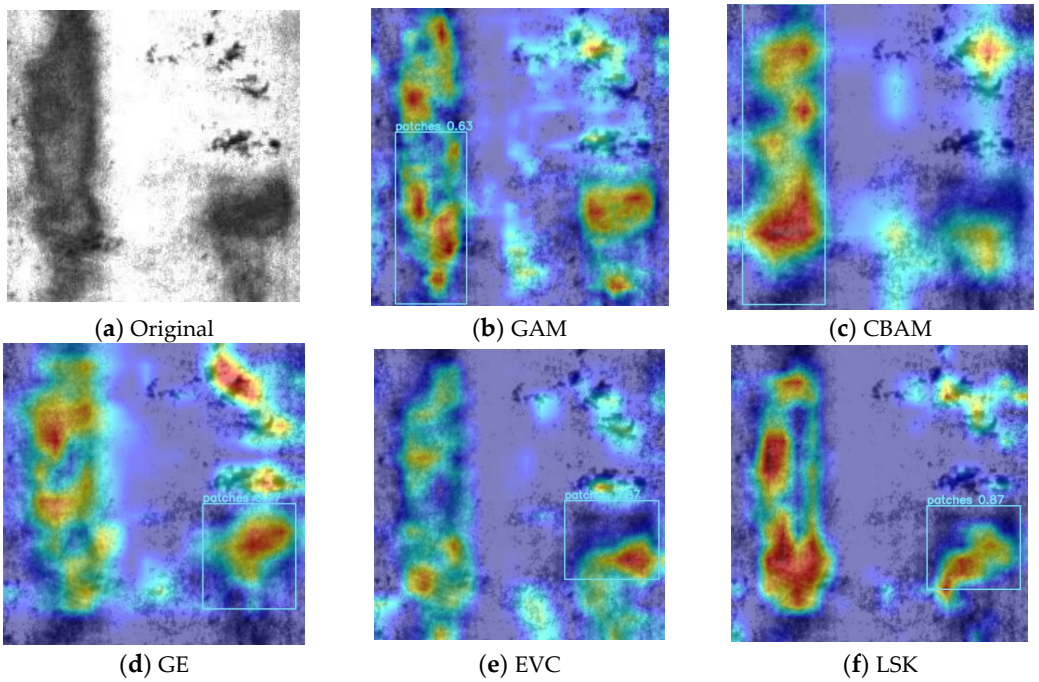
4.3.2. Horizontal Comparison Experiment

To validate the effectiveness of the TSP module constructed in this paper, we designed a horizontal comparison experiment with similar modules, SPPF and SPPCSPC [39]. As shown in Table 4, the TSP module can effectively enhance the detection accuracy of the model. The best results of each experimental indicator are displayed in bold in the table.

Table 4. Comparison of similar modules.

Model	Parameters/10^6	mAP/(%)	AP/(%)					
			Cr	In	Pa	Ps	Rs	Sc
YOLOv8 + BIG + SPPF	**4.87**	77.5	43.2	80.6	91.1	**85.0**	**70.3**	95.0
YOLOv8 + BIG + SPPCSPC	5.21	77.8	40.7	83.3	**93.6**	83.5	69.5	96.0
YOLOv8 + BIG + TSP (ours)	5.21	**78.5**	**43.7**	**84.6**	93.2	83.8	68.4	**97.2**

To better demonstrate the advantages of the LSK-Attention introduced in this paper, we conducted experiments by embedding different attention mechanisms in the same location, including Global Attention Mechanism (GAM) [42], Convolutional Block Attention Module (CBAM) [43], Gather–Excite (GE) [44], Explicit Visual Center (EVC) [45], and LSK-Attention. Heat maps of patch defects generated by each attention mechanism were compared, as shown in Figure 10. The darker the red color in the heatmap, the greater the contribution of the region to the final detection result, and the higher the attention to this part. The green and yellow parts have the second highest attention, and the blue part indicates that the feature has a smaller impact on object detection and recognition. Table 5 illustrates that incorporating the LSK module enables the model to more accurately identify various defects and effectively extract defect features on the steel surface. To visually showcase the superiority of the LSK module, we compared the defect heat maps generated by each attention mechanism. As shown in Figure 10, the LSK module better focuses on defect information and effectively extracts steel surface defect features.



(**a**) Original (**b**) GAM (**c**) CBAM

(**d**) GE (**e**) EVC (**f**) LSK

Figure 10. Heat map for each attention mechanism.

Table 5. Comparison of different attention mechanisms.

Model	mAP/(%)	AP/(%)					
		Cr	In	Pa	Ps	Rs	Sc
YOLOv8 + BIG + TSP + GAM	78.2	45.3	77.3	92.5	91.5	69.1	93.5
YOLOv8 + BIG + TSP + CBAM	78.1	**47.3**	81.3	92.9	91.6	64.9	90.3
YOLOv8 + BIG + TSP + GE	77.7	47.0	80.6	88.9	94.9	66.3	88.5
YOLOv8 + BIG + TSP + EVC	78.2	44.6	80.1	92.5	**98.0**	64.5	89.2
YOLOv8 + BIG + TSP + LSK (ours)	**79.5**	43.2	**82.7**	**95.3**	83.2	**78.2**	**94.3**

To validate the effectiveness of introducing DSConv to improve the C2f module, this experiment compared several other convolution modules, including Receptive-Field Attention Convolution (RFAConv) [46], Omni-Dimensional Dynamic Convolution (ODConv) [47], Dynamic Snake Convolution (DYSnake) [48], and Deformable Convolution v2 (DCNV2) [49]. As shown in Table 6, only the C2f-DS module improved with DSConv was able to enhance the model's feature representation capability while maintaining low computational complexity. It did not significantly increase the computational cost of the model, thus improving performance while preserving efficiency.

Table 6. Comparison of different convolution modules.

Model	Parameters/10^6	mAP/(%)	AP/(%)					
			Cr	In	Pa	Ps	Rs	Sc
YOLOv8+ BIG + TSP + LSK + C2f-rfaconv	5.34	78.1	42.5	83.7	**95.1**	81.6	71.8	93.9
YOLOv8 + BIG + TSP + LSK + C2f-odconv	5.58	79.1	43.3	84.4	93.5	**86.4**	71.4	95.4
YOLOv8 + BIG + TSP + LSK + C2f-dysnake	5.45	78.5	42.7	**86.4**	93.9	81.7	70.8	95.4
YOLOv8 + BIG + TSP + LSK + C2f-dcnv2	5.24	78.2	45.2	81.7	93.3	80.7	**74.5**	93.7
YOLOv8 + BIG + TSP + LSK + C2f-DS (ours)	**5.20**	**79.8**	**50.1**	82.6	93.4	84.8	72.3	**95.4**

4.3.3. Ablation Experiment

To validate the effectiveness of the newly added detection head (A), the constructed Triplet Spatial Pyramid Module (B), the introduced attention mechanism (C), and the C2f-DS module (D) in the detection of surface defects in steel, six sets of ablation experiments were designed.

The effectiveness of the improvements made to the YOLOv8-TLC algorithm is validated in Table 7, with the P-R curve of the original YOLOv8n algorithm shown in Figure 11. Building on this, the new detection head, TSP, LSK-Attention, and C2f-DS modules were sequentially integrated, resulting in the P-R curve for YOLOv8-TLC, depicted in Figure 12. Visual representations of some results are shown in Figure 13. The newly added detection head successfully identified targets that were previously unrecognized, leading to a 0.7% increase in mean average precision (mAP). Additionally, the proposed TSP structure enhanced the network's ability to learn important features, resulting in a 1.1% improvement in mAP and an increase in processing speed to 25.1 frames per second. Following this, the LSK-Attention mechanism was introduced to fully exploit the required spatial structural information, further enhancing mAP by 1%. Finally, optimization of the C2f module through DSConv not only improved detection accuracy but also achieved high computational speed with low storage requirements, thereby enhancing the overall efficiency of the defect detection model. The comparison of detection results for the improved algorithm is illustrated in Figure 14, where, aside from a slight decrease in the detection capability for scratch defects, all other defect categories exhibited improved detection performance.

Table 7. Ablation experiment.

Model	Improvement Strategies				Paramerters/ 10^6	FPS/ (Frame/s)	mAP/%	Ap/%					
	A	B	C	D				Cr	In	Pa	Ps	RS	Sc
YOLOv8+					**3.01**	157.9	76.6	40.7	79.8	92.3	80.5	70.0	96.0
YOLOv8+	√				4.87	181.8	77.4	43.4	80.6	91.1	**85.1**	69.3	94.9
YOLOv8+		√			3.35	**225.0**	77.8	44.2	**85.0**	92.1	83.9	65.3	96.1
YOLOv8+	√	√			5.21	206.9	78.5	43.7	84.6	93.2	83.8	68.4	**97.2**
YOLOv8+	√	√	√		5.28	178.2	79.5	43.3	82.7	**95.3**	83.2	**78.2**	94.2
YOLOv8+ (ours)	√	√	√	√	5.20	219.5	**79.8**	**50.1**	82.6	93.4	84.8	72.3	95.4

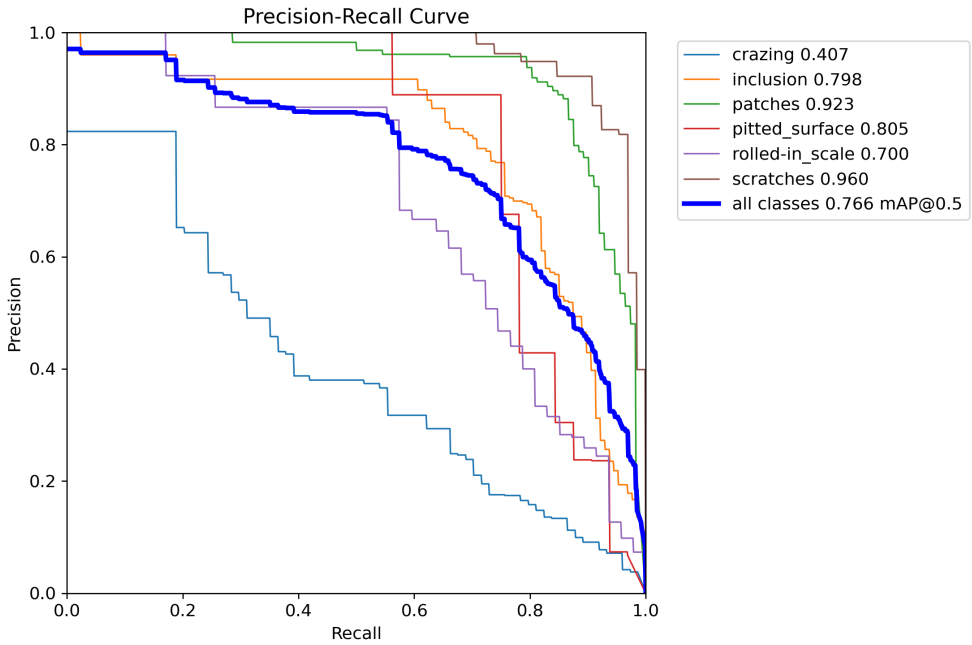


Figure 11. P-R curve of YOLOv8n algorithm.

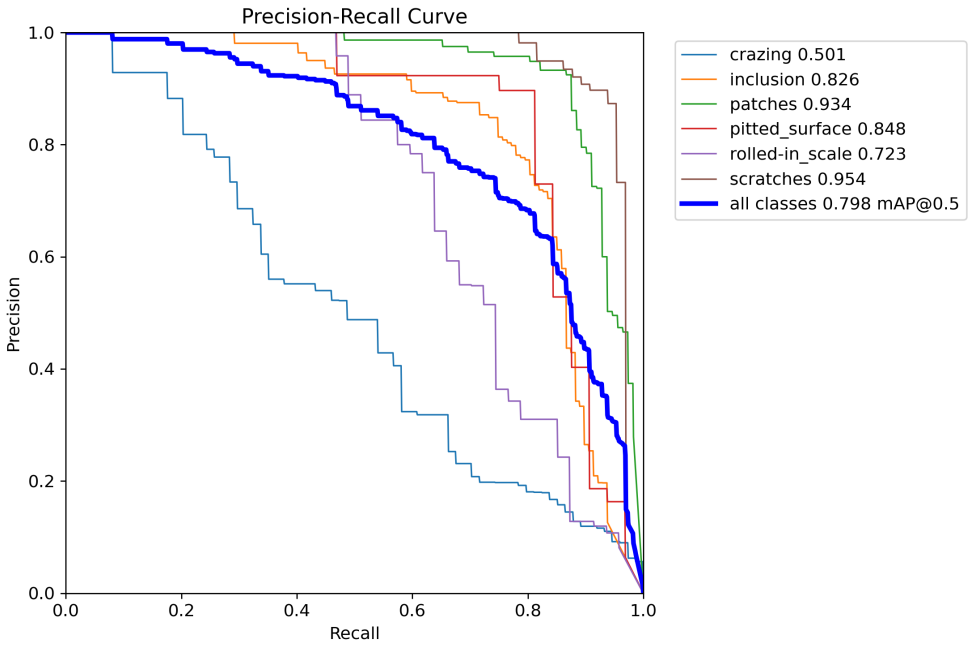
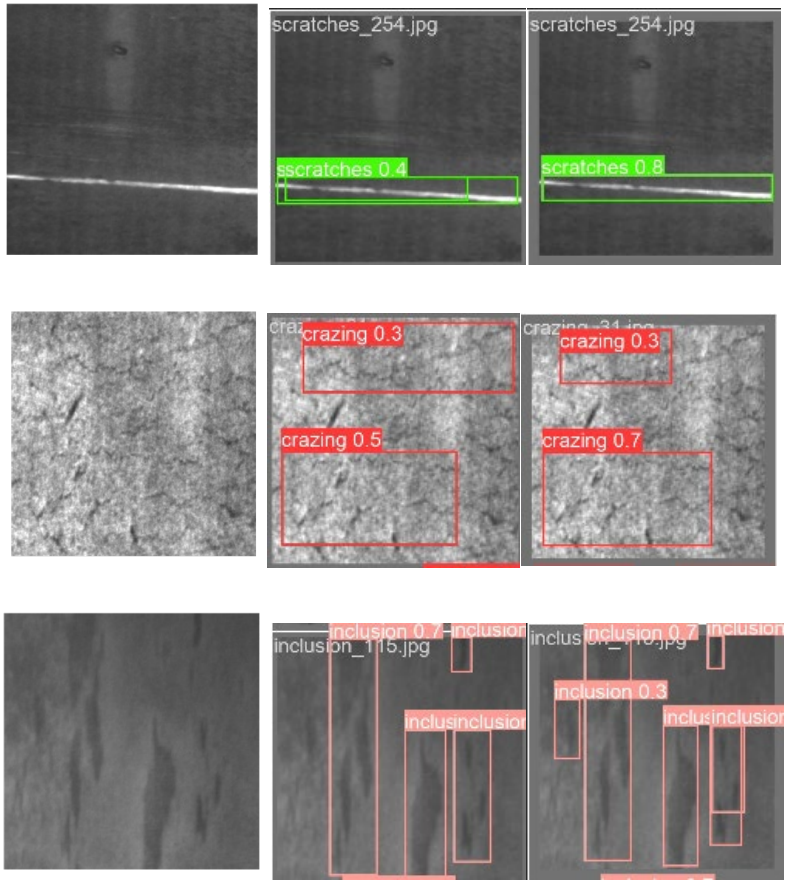


Figure 12. P-R curve of YOLOv8-TLC algorithm.



(**a**) (**b**) (**c**)

Figure 13. Part of the visualization results. (**a**) Original Figure, (**b**) YOLOv8, (**c**) YOLOv8-TLC.

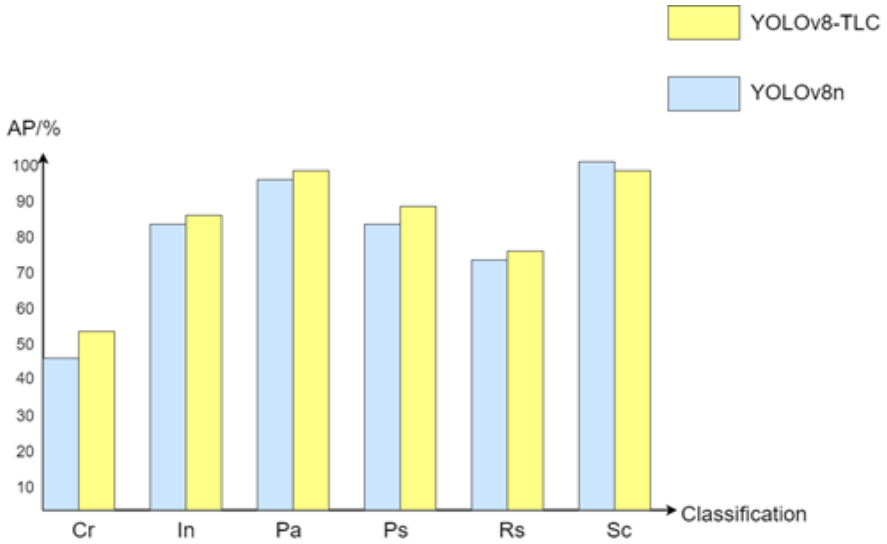


Figure 14. Comparison chart.

The comparison between Figures 15 and 16 shows that after adding the large detection head, the model can extract features from a broader field of view. Except for patches, the missed detection rate of other defects is reduced, effectively demonstrating that the large detection head can lower the missed detection rate.

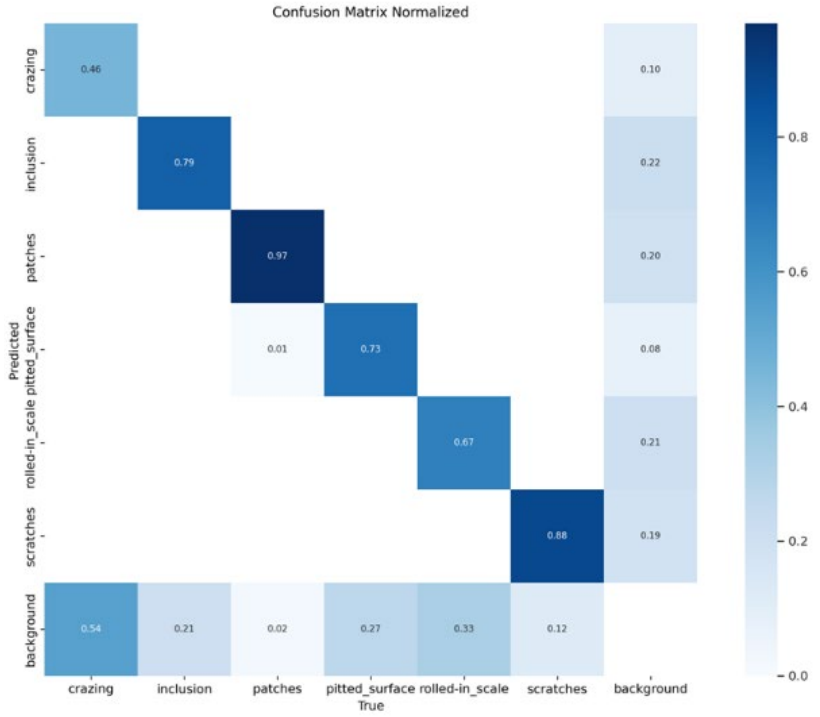


Figure 15. Normalized confusion matrix of YOLOV8.

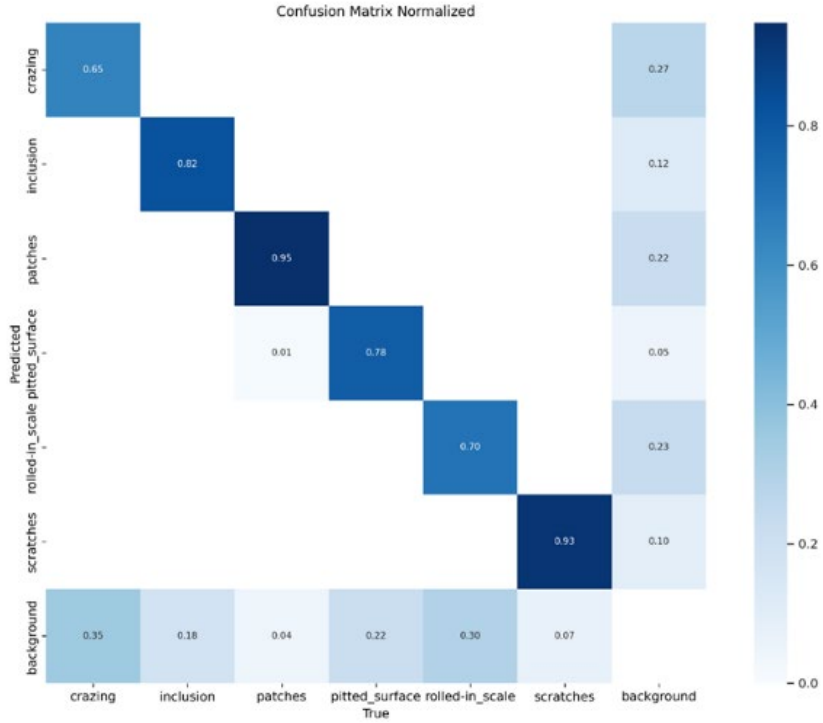


Figure 16. Normalized confusion matrix of YOLOv8-TLC.

4.3.4. Comparative Experiments with Detection Algorithms

To verify the superiority of the proposed algorithm, we selected several detection algorithms for comparison, including YOLOv3, YOLOv5n, YOLOv5s, YOLOv6n, YOLOv6s, YOLOv9s, YOLOv10n, YOLOv11n, as well as methods from references [24,30], and also included YOLOv8n. The comparison results are presented in Table 8.

Table 8. Comparison experiments of detection algorithms.

Method	AP/%						mAP/%
	Cr	In	Pa	Ps	Rs	Sc	
YOLOv3	45.0	77.0	93.6	71.7	73.9	86.9	74.7
YOLOV5n	37.8	79.5	95.6	77.6	78.6	88.2	76.2
YOLOv5s	38.2	79.9	95.8	79.5	76.3	89.6	76.6
YOLOv6n	42.0	76.6	**96.2**	77.1	73.5	92.4	76.3
YOLOv6s	44.2	76.6	93.8	76.3	78.7	92.6	77.0
Algorithm1 [24]	34.5	76.8	89.8	82.5	61.6	86.8	72.0
Algorithm2 [30]	43.4	**89.3**	84.3	76.6	74.2	95.3	77.2
YOLOv8n	42.7	80.3	90.3	**93.7**	65.4	90.6	77.1
YOLOv9s	49.1	77.9	95.8	74.4	77.0	87.5	77.0
YOLOv10n	42.4	78.4	93.5	75.0	77.4	88.6	75.9
YOLOv11n	44.2	76.3	95.9	75.7	**81.6**	90.6	77.4
YOLOv8-TLC (ours)	**50.1**	82.6	93.4	84.8	72.3	**95.4**	**79.8**

From Table 8, it can be seen that the improved model has the highest mAP value compared to other models. Although the latest YOLOv11n achieves higher accuracy than the baseline YOLOv8n model, the improvement is minimal and does not surpass the enhancements made in this study. Furthermore, the improved algorithm meets the requirements for real-time detection, balancing both detection speed and accuracy, demonstrating superiority in steel surface defect detection.

5. Conclusions

To address the low recognition rate and efficiency of traditional image processing algorithms in detecting surface defects on steel, this paper proposes the YOLOv8-TLC algorithm. By introducing detection heads of different sizes, the algorithm reduces missed detections, while incorporating the LSK-Attention mechanism enhances the network's ability to recognize surface defects on steel. Additionally, the TSP module improves the extraction of redundant features, and the C2f-DS module replaces the traditional C2f module, accelerating the model's detection speed. Testing results on the NEU-DET dataset show that YOLOv8-TLC not only surpasses YOLOv8n in detection accuracy but also significantly improves detection speed. The algorithm meets industrial production requirements for high detection accuracy and real-time performance. However, the detection performance for crack-like defects with similar background colors and textures is still not ideal. Although this paper has made some improvements in this area, the issue remains unresolved. Future research will focus on enhancing crack defect detection and further lightening the model.

Author Contributions: Writing—review and editing, H.C.; supervision, C.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Liaoning Provincial Department of Education's General Fund Project (LJKZ1184) under the number 296.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Publicly available datasets were analyzed in this study. This data can be found here: https://drive.google.com/file/d/1qrdZlaDi272eA79b0uCwwqPrm2Q_WI3k/view (accessed on 8 September 2021).

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

YOLOv8-TLC	YOLOv8-Triple Space Pyramid and Large Selective Kernel and C2f-DS
LSK	Large Selective Kernel
TSP Module	Triple Spatial Pyramid Module
SPP	Spatial Pyramid Pooling
SPPF	Spatial Pyramid Pooling-Fast
CSPC	Cross-Stage Partial Connection
TA	Triplet Attention
DSConv	Distributed Shift Convolution
VQK	Variable Quantized Kernel
KDS	Kernel Distribution Shifter
CDS	Channel Distribution Shifter
C2f-DS	C2f-Distributed Shift Convolution
Cr	crazing
In	inclusion
Pa	patches
Ps	pitted surfaces
Sc	scratches
Rs	rolled-in scale
P	precision
R	recall
AP	Average Precision
mAP	mean Average Precision
FPS	Frames Per Second
TP	the true example,
FP	true counterexample
FN	false counterexample.
GAM	Global Attention Mechanism
CBAM	Convolutional Block Attention Module
GE	Gather–Excite
EVC	Explicit Visual Center
RFAConv	Receptive-Field Attention Convolution
ODConv	Omni-Dimensional Dynamic Convolution
DYSnake	Dynamic Snake Convolution
DCNV2	Deformable Convolution v2

References

1. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet classification with deep convolutional neural networks. *Commun. ACM* **2017**, *60*, 84–90. [CrossRef]
2. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
3. Zhou, Q.; Yu, C. Point RCNN: An angle-free framework for rotated object detection. *Remote Sens.* **2022**, *14*, 2605. [CrossRef]
4. Girshick, R. Fast R-CNN. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
5. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *39*, 1137–1149. [CrossRef] [PubMed]
6. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.-Y.; Berg, A.C. Ssd: Single shot multibox detector. In Proceedings of the Computer Vision–ECCV 2016, 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016; Proceedings, Part I 14; Springer International Publishing: Berlin/Heidelberg, Germany, 2016; pp. 21–37.
7. Redmon, J. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016.
8. Zhou, Y. A YOLO-NL object detector for real-time detection. *Expert Syst. Appl.* **2024**, *238*, 122256. [CrossRef]
9. Redmon, J.; Farhadi, A. YOLO9000, better, faster, stronger. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 7263–7271.
10. Redmon, J. Yolov3, An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
11. Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. Yolov4, Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934.

12. Wang, C.-Y.; Liao, H.-Y.M.; Yeh, I.-H.; Wu, Y.-H.; Chen, P.-Y.; Hsieh, J.-W. CSPNet: A new backbone that can enhance learning capability of CNN. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Seattle, WA, USA, 14–19 June 2020; pp. 390–391.
13. Liu, S.; Qi, L.; Qin, H.; Shi, J.; Jia, J. Path aggregation network for instance segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 8759–8768.
14. Zaidi, S.S.A.; Ansari, M.S.; Aslam, A.; Kanwal, N.; Asghar, M.; Lee, B. A survey of modern deep learning based object detection models. *Digit. Signal Process.* **2022**, *126*, 103514. [CrossRef]
15. Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; Zagoruyko, S. End-to-end object detection with transformers. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; Springer International Publishing: Cham, Switzerland, 2020; pp. 213–229.
16. Zhao, Y.; Lv, W.; Xu, S.; Wei, J.; Wang, G.; Dang, Q.; Liu, Y.; Chen, J. Detrs beat yolos on real-time object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 16–22 June 2024; pp. 16965–16974.
17. Ge, Z. Yolox: Exceeding yolo series in 2021. *arXiv* **2021**, arXiv:2107.08430.
18. Li, C.; Li, L.; Jiang, H.; Weng, K.; Geng, Y.; Li, L.; Ke, Z.; Li, Q.; Cheng, M.; Nie, W.; et al. YOLOv6, A single-stage object detection framework for industrial applications. *arXiv* **2022**, arXiv:2209.02976.
19. Guo, Z.; Wang, C.; Yang, G.; Huang, Z.; Li, G. MSFT-YOLO: Improved YOLOv5 Based on Transformer for Detecting Defects of Steel Surface. *Sensors* **2022**, *22*, 3467. [CrossRef]
20. Zhao, Z. Research and Application of Small Defect Detection on Rolled Steel Surface Based on Improved YOLOv6. Master's Thesis, North University of China, Taiyuan, China, 2023.
21. Xie, Y.; Hu, W.; Xie, S. Surface defect detection algorithm based on feature-enhanced YOLO. *Cogn. Comput.* **2023**, *15*, 565–579. [CrossRef]
22. Li, F. Research and Implementation of Metal Surface Defect Detection Based on Deep Learning. Master's Thesis, South China University of Technology, Guangzhou, China, 2023.
23. Li, S.; Shi, Z.; Kong, F.; Wang, R.; Luo, T. An improved YOLOv5 algorithm for steel surface defect detection. *Adv. Laser Optoelectron.* **2023**, *60*, 192–200.
24. Wang, L.; Liu, X.; Ma, J.; Su, W.; Li, H. Real-time steel surface defect detection with improved multi-scale YOLO-v5. *Processes* **2023**, *11*, 1357. [CrossRef]
25. Ren, F.; Fei, J.; Li, H.; Doma, B.T. Steel Surface Defect Detection Using Improved Deep Learning Algorithm: ECA-SimSPPF-SIoU-Yolov5. *IEEE Access* **2024**, *12*, 32545–32553. [CrossRef]
26. Li, C.; Xu, A.; Zhang, Q.; Cai, Y. Steel Surface Defect Detection Method Based on Improved YOLOX. *IEEE Access* **2024**, *12*, 37643–37652. [CrossRef]
27. Gao, C.; Qin, S.; Li, M.; Lv, X. Research on Steel Surface Defect Detection Using an Improved YOLOv7 Algorithm. *Comput. Eng. Appl.* **2024**, *60*, 282–291.
28. Fan, J.; Wang, M.; Li, B.; Liu, M.; Shen, D. ACD-YOLO: Improved YOLOv5-based method for steel surface defects detection. *IET Image Process.* **2024**, *18*, 761–771. [CrossRef]
29. Kong, X.; Fan, W. Surface Defect Detection of Steel Strips based on an Improved YOLOv8. In Proceedings of the 2024 5th International Seminar on Artificial Intelligence, Networking and Information Technology (AINIT), Nanjing, China, 22–24 March 2024; pp. 2325–2328.
30. Zhao, B.; Zhang, C.; Jia, X. ECC-YOLO: An improved method for detecting surface defects on steel. *J. Electron. Meas. Instrum.* **2024**, *38*, 108–116.
31. Dai, L.; Li, Y.; Shi, R. Strip Steel Surface Defect Detection Algorithm Based on Improved YOLOv8. *Manuf. Technol. Mach. Tools* **2024**, 1–16. Available online: http://kns.cnki.net/kcms/detail/11.3398.TH.20240918.1520.006.html (accessed on 29 September 2024).
32. Wang, M.; Liu, Z. Steel Surface Defect Detection Based on Improved YOLOv8 Algorithm. *Mech. Sci. Technol.* **2024**, 1–11. [CrossRef]
33. Wang, L.; Zhang, B.; Wu, Q.-H. RCSA-YOLO: Improved SAR ship instance segmentation with YOLOv8. *Comput. Eng. Appl.* **2024**, *60*, 103. [CrossRef]
34. Wang, S.; Xu, H.; Zhu, X.; Song, J.; Li, Y. Lightweight small target detection algorithm based on improved YOLOv8n aerial photography: PECS-YOLO. *Comput. Eng.* **2024**, 1–16. [CrossRef]
35. Li, Y.; Hou, Q.; Zheng, Z.; Cheng, M.-M.; Yang, J.; Li, X. Large selective kernel network for remote sensing object detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Paris, France, 4–6 October 2023; pp. 16794–16805.
36. Misra, D.; Nalamada, T.; Arasanipalai, A.U.; Hou, Q. Rotate to attend: Convolutional triplet attention module. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Waikoloa, HI, USA, 3–7 January 2021; pp. 3139–3148.
37. Zhang, L.; Tian, Y. Improved multi-scale lightweight vehicle target detection algorithm for YOLOv8. *Comput. Eng. Appl.* **2024**, *60*, 129–137.
38. He, K.; Zhang, X.; Ren, S.; Sun, J. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 1904–1916. [CrossRef]
39. Wang, C.Y.; Bochkovskiy, A.; Liao, H.Y.M. YOLOv7, Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 7464–7475.

40. Nascimento, M.G.; Fawcett, R.; Prisacariu, V.A. Dsconv: Efficient convolution operator. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 5148–5157.
41. Song, K.; Yan, Y. A noise robust method based on completed local binary patterns for hot-rolled steel strip surface defects. *Appl. Surf. Sci.* **2013**, *285*, 85. [CrossRef]
42. Liu, Y.; Shao, Z.; Hoffmann, N. Global attention mechanism: Retain information to enhance channel-spatial interactions. *arXiv* **2021**, arXiv:2112.05561.
43. Woo, S.; Park, J.; Lee, J.-Y.; Kweon, I.S. Cbam: Convolutional block attention module. In Proceedings of the European Conference on Computer Vision (ECCV), Vision, 8–14 September 2018; pp. 3–19.
44. Hu, J.; Shen, L.; Albanie, S.; Sun, G.; Vedaldi, A. Gather-excite: Exploiting feature context in convolutional neural networks. *Adv. Neural Inf. Process. Syst.* **2018**, *31*, 9423–9433.
45. Quan, Y.; Zhang, D.; Zhang, L.; Tang, J. Centralized feature pyramid for object detection. *IEEE Trans. Image Process.* **2023**, *32*, 4341–4354. [CrossRef]
46. Zhang, X.; Liu, C.; Yang, D.; Song, T.; Ye, Y.; Li, K.; Song, Y. RFAConv: Innovating spatial attention and standard convolutional operation. *arXiv* **2023**, arXiv:2304.03198.
47. Li, C.; Zhou, A.; Yao, A. Omni-dimensional dynamic convolution. *arXiv* **2022**, arXiv:2209.07947.
48. Qi, Y.; He, Y.; Qi, X.; Zhang, Y.; Yang, G. Dynamic snake convolution based on topological geometric constraints for tubular structure segmentation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Paris, France, 4–6 October 2023; pp. 6070–6079.
49. Zhu, X.; Hu, H.; Lin, S.; Dai, J. Deformable convnets v2, More deformable, better results. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 9308–9316.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.