

Article

YOLO-ADS: An Improved YOLOv8 Algorithm for Metal Surface Defect Detection

Zili Gui  and Jianping Geng *

School of Electronic Engineering and Automation, Guilin University of Electronic Technology, Guilin 541004, China; ziligui085400@mails.guet.edu.cn

* Correspondence: jianpinggeng@guet.edu.cn

Abstract: Addressing issues such as susceptibility to background interference and variability in feature scales of fine-grained defects on metal surfaces, as well as the relatively poor versatility of the baseline model YOLOv8n, this study proposes a YOLO-ADS algorithm for metal surface defect detection. Firstly, a novel CSPNet with Average SPP-Fast Block (ASPPFCSPC) module is proposed to enhance the model's fusion and representation ability between local features and global background information. Secondly, the newly improved module C2f_SimDCNv2 is utilized to improve the ability of the model to extract multi-scale features. Finally, the Space-to-Depth (SPD) layer is introduced to prevent the loss of fine-grained information from small target features and reduce the redundancy between convolution operations. Experimental results demonstrate that the mean Average Precision (mAP) and Precision of the YOLO-ADS algorithm on the steel strip surface defect dataset NEU-DET reach 81.4% and 79.7%, which are severally increased by 3.5% and 6.1%, and the Frames Per Second (FPS) reaches 140.4. Meanwhile, the versatility and robustness of the model are verified on the industrial steel surface defect dataset GC10-DET, the industrial aluminum surface defect dataset APSPC and even the larger public benchmark dataset VOC2012, the mAP is respectively increased by 3.7%, 3.4% and 4.3%. Compared with the mainstream detection algorithms, YOLO-ADS algorithm is ahead of a certain advanced level in detection accuracy while maintaining a good real-time performance, which provides an efficient and feasible solution for the field of metal surface defect detection.

Keywords: metal surface defect detection; YOLOv8; ASPPFCSPC; C2f_SimDCNv2



Citation: Gui, Z.; Geng, J. YOLO-ADS: An Improved YOLOv8 Algorithm for Metal Surface Defect Detection. *Electronics* **2024**, *13*, 3129. <https://doi.org/10.3390/electronics13163129>

Academic Editor: Chiman Kwan

Received: 8 July 2024

Revised: 3 August 2024

Accepted: 5 August 2024

Published: 7 August 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

Metal surface defect detection has always been one of the most important links in the industrial field [1]. With the development of modern industry, metal plays an indispensable role in construction [2], machinery manufacturing, transportation and other industries, and its quality is directly related to the performance and safety of products. However, the metal surface is inevitably prone to defects such as scratches, cracks, and pits during the process of automating large productivity [3]. These defects will not only impact the aesthetic appearance of products, but also significantly compromise the intrinsic properties of the metal, such as reducing the fatigue strength, wear resistance and corrosion resistance of the metal [4]. In addition, it will cause serious problems such as metal belt break and accumulation, which will increase material loss and reduce production efficiency [5]. Therefore, the industry attaches great importance to the research of metal surface defect detection technology, which is regarded as the key to improve product quality [6]. The utilization of precise and effective algorithms for the detection of defects in metal materials prior to their practical application holds immense significance.

Machine vision technology is divided into methods based on machine learning and deep learning. In recent years, it has quickly gained the favor of researchers at home and abroad due to its safety and reliability, wide application range and high detection

efficiency [7]. Optical devices are used to collect images, image processing algorithms are used to extract image information, and defects can be identified according to the obtained feature information. Machine learning technology mainly relies on a mass of prior knowledge and design experience of experts [8]. Special rule algorithms are designed for specific types of defects to extract features, which are suitable for simple scenarios and difficult to capture high-level semantic features. Yet, the algorithm based on deep learning can automatically integrate pattern recognition into the process of building the model, so as to verify and optimize the model while learning features, thereby weakening the incompleteness caused by artificial design features, and has strong generalization and robustness in complex environments [9]. Due to its unique advantages, deep learning methods have increasingly taken the place of traditional machine learning techniques in industrial fields such as metal surface defect detection [10].

At present, the mainstream object detection algorithms consist of two-stage Faster R-CNN [11] as a representative, single-stage SSD [12] and YOLO [13–23] series, as well as the Transformer-based RT-DETR [24]. In the two-stage object detection algorithm, candidate boxes are initially generated, followed by feature extraction from the content within these boxes. Subsequently, object regression is performed on the region content [25]. Generally, this kind of algorithm has high detection accuracy, but it will sacrifice a portion of the detection speed because the candidate boxes need to be screened first. While the single-stage object detection algorithm is a regression-based algorithm that combines the localization and classification tasks [26], so it shows a faster detection speed and has a huge advantage in real-time detection. Single-stage has higher detection efficiency than two-stage in defect detection tasks in industrial scenarios, so many domestic and foreign researchers have carried out a lot of research on single-stage algorithm. For example, Liu et al. [27] combined attention mechanism with multi-feature fusion network, and proposed RAF-SSD model for steel surface defect detection task. The detection mAP is 1.6% and 1.8% higher than that of SSD and YOLOv3, respectively. Aiming at the complexity and diversity of steel surface defects, Li et al. [28] proposed a deep learning model based on Multiscale Feature Extraction (MSFE) module and Efficient Feature Fusion (EFF) module. Compared with the baseline model YOLOv5s, the mAP is improved by 2.2% on the premise of only increasing the number of parameters by 2.5 M. To enhance the attention to defect areas, Liang et al. [29] combined the C3 module with the Efficient Channel Attention (ECA) mechanism, and they introduced the CSPNet with SPP Block (SPPCSPC) module of YOLOv7 to extract multi-scale features of aluminum surface defects. The detection accuracy of the proposed YOLOv5-ESP algorithm is better than that of classical algorithms such as YOLOv5. Gao et al. [30] proposed the SRN-YOLO model to solve the defects of tiny and fuzzy steel surface. The segmentation residual convolutional network was used to capture the gradient feature information, and the feature fusion pyramid network was constructed to minimize the feature loss. The detection accuracy of the baseline model YOLOv7 is greatly improved, but the sacrificed model parameters make it unsuitable for industrial deployment. Yuan et al. [31] improved YOLOv8n, combined adaptive receptive field and Channel Attention (CA) mechanism to reduce the extraction of redundant features, and used Ghost convolution to reduce model parameters. Compared with the baseline model, the improved model GDCP-YOLO increased the mAP by 1.4% while reducing the number of parameters by 0.2 M. Huang et al. [32] proposed the WFRE-YOLOv8s model out of the consideration that the traditional method is hard to meet the high detection accuracy and efficiency of the steel industry. The original C2f module in the backbone network was replaced with the newly designed CFN structure to reduce model parameters, and Efficient Multi-scale Attention (EMA) mechanism was added to enhance valuable feature extraction. The mAP is 4.7% higher than that of YOLOv8s.

Defects in industrial production often have two characteristics, one is the large difference in shape between the same type of defects, and the other is the similar feature information between different types of defects [33], that is, fine-grained defects. Most of the above models are designed to alleviate the issue of low detection accuracy resulting from

this factor, and they are used to adding attention mechanisms to each part of the model. While the design concept of our model is more inclined to focus on rectifying inherent deficiencies in the backbone network of the baseline model, thereby augmenting its capability to extract, integrate, and represent fine-grained features for improved detection accuracy and versatility. To this end, this study proposes an improved YOLO8 algorithm, which aims to improve the accuracy of metal surface defect detection while ensuring real-time performance. The main contributions of our work are as follows:

1. We propose a novel spatial pyramid pooling module ASPPFCSPC, which enables the model to process local features, surrounding context and global context at the same time, and effectively fuse this information to enhance the model's ability to represent fine-grained features under complex backgrounds on metal surfaces, thereby improving the detection accuracy and versatility of the model. This improvement significantly improves mAP by about 1.5% on NEU-DET dataset.
2. We newly improve the C2f module, which is combined with the Deformable Convnets v2 (DCNv2) to enhance the adaptability to target shapes or local structures. The expanded receptive field can strengthen the ability of the underlying network to extract multi-scale defect features on metal surfaces, thereby improving the detection accuracy and robustness of the model. Building on the previous work, the mAP again grows by approximately 0.9%.
3. We add an SPD layer after each Conv in the backbone network, so that the feature representation can be fine-tuned by learnable parameters, which avoids the loss of fine-grained information of small target features on metal surfaces during down-sampling, and reduces the dependence on nonlinear activation, thereby improving the detection accuracy and speed of the model. The mAP gains another 1.1%, for a total increase of 3.5%. And the FPS reaches up to 140.4.
4. We not only conduct numerous experiments on the steel strip surface defect dataset NEU-DET, but also verify the metal surface defect detection algorithm YOLO-ADS on the industrial steel surface defect GC10-DET, the industrial aluminum surface defect dataset APSPC and even the larger public benchmark dataset VOC2012. Its detection accuracy is not only ahead of a certain advanced level, but the model also has good versatility and robustness. The performance of mAP on different metal surface defect datasets is improved by 3.7%, 3.4% and 4.3%, respectively.

The remainder of this paper is organized as follows. Section 2 introduces the related work of this study. Section 3 details YOLOv8 and its improved algorithm, introducing the novel proposed spatial pyramid pooling module ASPPFCSPC, the newly improved feature extraction module C2f_SimDCNv2, and the freshly introduced SPD layer. Section 4 systematically analyzes the primary dataset to clarify the categories of target defects and the directions for algorithmic improvement, and it conducts a comprehensive benchmark evaluation of YOLO-ADS. Section 5 discusses the study's findings, limitations, and future work. Finally, Section 6 concludes this study.

2. Related Work

Current methods for detecting defects on metal surfaces in the industry are primarily divided into some outstanding traditional machine learning techniques and State-Of-The-Art (SOTA) deep learning algorithms.

2.1. Traditional Machine Learning Techniques

Traditional machine learning techniques mainly rely on a mass of prior knowledge and design experience of experts, there are still many units use some excellent techniques in metal surface defect detection tasks.

Xu et al. [34] applied a two-dimensional Haar wavelet transform to all blocks in the model to identify surface defects in steel strips. The proposed method outperformed the other three comparable models by 1%. Zhao et al. [35] utilized the vector-valued regularized kernel function approximation and the support vector machine to localize

defects, effectively extracting wrinkle features from images of steel strips. Zhang et al. [36] posited that the background of steel strips follows a Gaussian distribution and defined a membership function to evaluate the gray level of each defect. The detection rate of this fuzzy sets-based method was up to 96.8%. Ji et al. [37] integrated the genetic algorithm with the generalized linear regression, enabling the proposed data-driven algorithm to balance prediction accuracy and the interpretability of the resulting model for hot-rolled steel strips.

However, conditional machine learning techniques impose higher demands on external conditions and require timely targeted retuning or redesign when the characteristics of defects change significantly. Consequently, they may only be suitable for certain specific stable scenarios.

2.2. Deep Learning Algorithms

It is well known that the single-stage object detection algorithm is lighter and faster than the two-stage algorithm. Many studies have attempted to apply the current SOTA algorithm YOLOv8 to the industrial field. The YOLOv8 exists three main flaws in detecting metal surface defects, there have been many researchers proposed improved methods and made great contributions.

Firstly, the Spatial Pyramid Pooling-Fast (SPPF) module in YOLOv8 has a limited receptive field due to the blind use of Max pooling operations, which leads to poor ability to perceive complex backgrounds and fuse multi-scale features. To overcome this limitation, Zhang et al. [38] proposed a variant of YOLOv8n for defect detection on steel strip surfaces, they suggested the integration of the Large Separable Kernel Attention (LSKA) design and the SPPF module to pool feature maps at different scales. This step improved the detection accuracy by 2.3%. Kong et al. [39] fused the Convolutional Block Attention Module (CBAM) mechanism into the SPPF module of YOLOv8x to avoid the loss of key information caused by a single pooling layer. This resulted in a 3.0% improvement on the steel defect dataset and 2.6% improvement on the aluminum surface industrial defect detection dataset through this step.

Secondly, the original C2f module in YOLOv8 has poor performance in extracting multi-scale defect features. As to this problem, Zhang et al. [40] added the Dilation-wise Residual (DWR) block to the C2f module of YOLOv8n, which decomposed the method of extracting multi-scale context information into a two-step method to reduce the difficulty of multi-scale defect extraction on steel surfaces. This step improved the detection accuracy by 2.5%. Xie et al. [41] proposed a multi-scale feature extraction block based on group convolution to improve the detection performance of multi-scale defects by replacing the bottleneck of the C2f module. Compared to the baseline algorithm YOLOv8l, this step improved the mAP by 1.2% and 2.8% on two steel strip datasets, respectively.

Additionally, it is very easy for YOLOv8 to lose fine-grained features during down-sampling. To address this issue, Yuan et al. [42] replaced the original neck network of YOLOv8s with a simplified version of Bidirectional Feature Pyramid Network (BiFPN) to improve the sensitivity to fuse minor features and enhance the ability to recognize fine-grained target defects on steel surfaces. This step improved the detection accuracy by 1.1%. Wang et al. [43] proposed a YOLOv8n-DSDM by adding a small target detection layer in the current neck network, which improved the mAP by 2.3% on an industrial aluminum sheet surface defect dataset through this step.

In response to the above problems in YOLOv8, it is obvious that most related work do not directly solve them fundamentally. They are used to adding attention mechanisms or enhancing the fusion ability of the neck network. Although object detection networks have become increasingly robust in terms of architectural design and training strategies, there seems to be no specific evolution in the detection strategies for objects with extreme scales. And recent studies still rely on superior neck network designs, but this cannot radically address the inadequacies in feature extraction capabilities of the underlying network.

Inspired by the inherent deficiencies of YOLOv8, this study proposes a novel metal surface defect detection algorithm, YOLO-ADS, which is based on a deep and large backbone network. The network is designed to capture and integrate dense information across various spatial scales and levels of potential semantics. This is all attributed to three targeted strategies for the backbone network in this study: introducing the average pooling operation to the spatial pyramid pooling module, which can instantly expand the receptive field for fusion; combining the original C2f module with the deformable network, which can markedly enhance adaptability to varying scales; and adding an SPD layer before the convolution operation, which can directly shorten the stride, thereby reducing the loss of fine-grained information during down-sampling.

Compared to YOLOv8n, YOLO-ADS significantly improves detection accuracy and identifies a wider range of metal surface defects, offering a new paradigm of the heavy-backbone design in the field of defect detection.

3. Methodology

3.1. YOLOv8 Algorithm

YOLOv8 is an upgraded version based on the previous generation of YOLO series, comprising primarily of three components: Backbone, Neck and Head. The network architecture of YOLOv8 is illustrated in Figure 1.

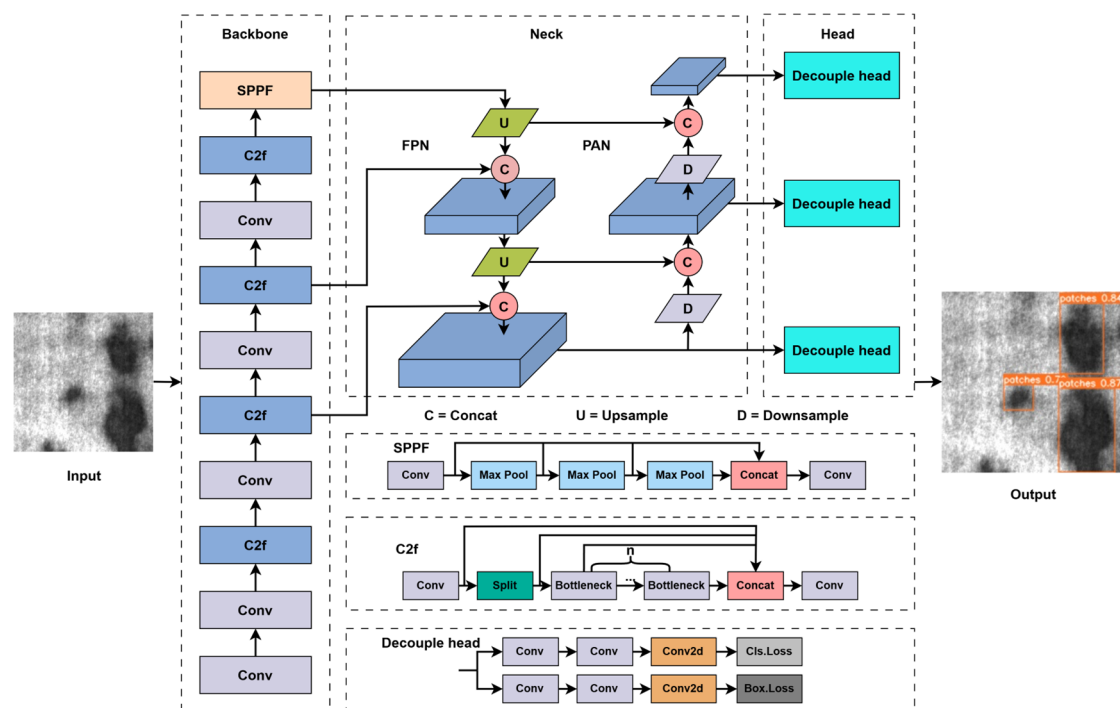


Figure 1. Network architecture of YOLOv8 algorithm.

The Backbone part is mainly used for feature extraction. Building upon the C3 module from YOLOv5, YOLOv8 combines the idea of Efficient Layer Aggregation Network (ELAN) module in YOLOv7 algorithm to improve C2f module. The new module has more residual connections, and according to the scaling coefficient, the number of channels is changed by splitting and splicing operations, so as to achieve lightweight and obtain more rich gradient flow information. In the tail of Backbone, the serial SPPF module in YOLOv5 is used to obtain faster speed gain while ensuring the same receptive field.

The Neck part is mainly used for feature fusion, which is inspired by Path Aggregation Network (PAN) [44] and adopts Feature Pyramid Network (FPN) [45] (top-down) + PAN (bottom-up) dual feature pyramid structure to help the shallow information output from three different levels of Backbone to better aggregate to deep features.

The Head part is mainly used for feature classification and localization, and the popular decoupled head structure is used to separate the classification and detection heads to alleviate the conflicts existing between tasks. The Anchor-Free method with fully autonomous learning avoids the negative impact of excessive dependence on the prior frame on the detection accuracy, which offers significant advantages in object detection tasks involving objects with irregular length and width. Distribution Focal Loss [46] combined with Complete Intersection over Union (CIoU) Loss is introduced as the Loss function of regression branch, and Varifocal Loss [47] is used as the loss function of classification branch. At the same time, the dynamic Task-Aligned Assigner sample allocation strategy is adopted to achieve a high degree of alignment consistency between classification and regression tasks.

However, the application scenarios of general object detection algorithms are usually regular natural images. While fine-grained defects on metal surfaces in industrial production environments are often susceptible to complex backgrounds and variable scales. Therefore, simply using basic object detection technology is not suitable for defect detection, and it is necessary to improve and optimize the model structure.

3.2. Improved YOLOv8 Algorithm

YOLOv8 provides multiple versions of the choice, divided into n, s, m, l, x, five different versions, and the number of model parameters increases in turn. Considering that the metal surface defect detection task in industrial scenarios needs to take into account both detection accuracy and real-time performance, the smallest version YOLOv8n is selected for improvement in this study.

Aiming at the fine-grained defects on metal surface which are susceptible to background interference, we improve SPPCSPC module to replace SPPF module and add SPD layers. For the ASPPFCSPC module, we mainly introduce an adaptive average pool layer in parallel with the Max pool layer to enhance the fusion and representation ability between local features and global background information. At the same time, we reference the idea of the serial Max pooling operation in the SPPF module and use a simpler activation function Rectified Linear Unit (ReLU) to improve the inference speed of the model.

In view of the variable scale of metal surface defects, we simply combine the first Conv in the bottleneck with the deformable convolutional network, and then replace the last three C2f modules in the Backbone part with the improved module C2f_SimDCNv2 to improve the multi-scale feature extraction ability of the model.

By improving the shortcomings of YOLOv8n, the detection accuracy and versatility of the model are significantly improved. The network architecture of the improved algorithm YOLO-ADS is illustrated in Figure 2.

3.2.1. A Novel Spatial Pyramid Pooling Module ASPPFCSPC

The Spatial Pyramid Pooling (SPP) [48] module was proposed by K. He in 2015, which mainly solves two key problems: one is to effectively avoid image distortion caused by image region cropping and scaling operations, and the other is to solve the issue of extracting relevant repetitive features by convolutional neural networks.

The SPPCSPC module in the original YOLOv7 algorithm is one of the latest versions of the variants of the improved SPP structure. It not only inherits the two advantages of SPP, but also increases the receptive field, enabling adaptability to images with varying resolutions. As shown in the SPPCSPC module structure in Figure 3, the features extracted through Backbone undergo Max pooling operations of different sizes of 5×5 , 9×9 and 13×13 in parallel to obtain receptive fields of different sizes. This enables the algorithm to effectively discriminate between large and small targets. However, at the same time, it relatively limits the attention to global background information, and the amount of calculation also increases a lot, which leads to the detection accuracy and speed fail to achieve the optimal effect.

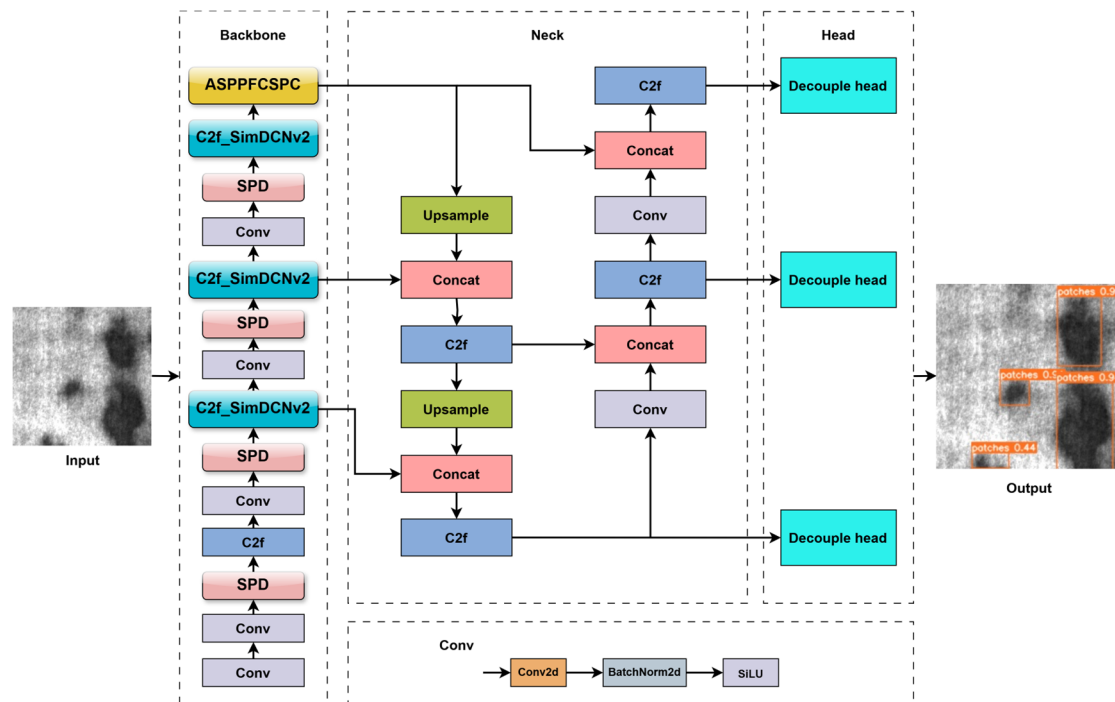


Figure 2. Network architecture of the improved algorithm YOLO-ADS.

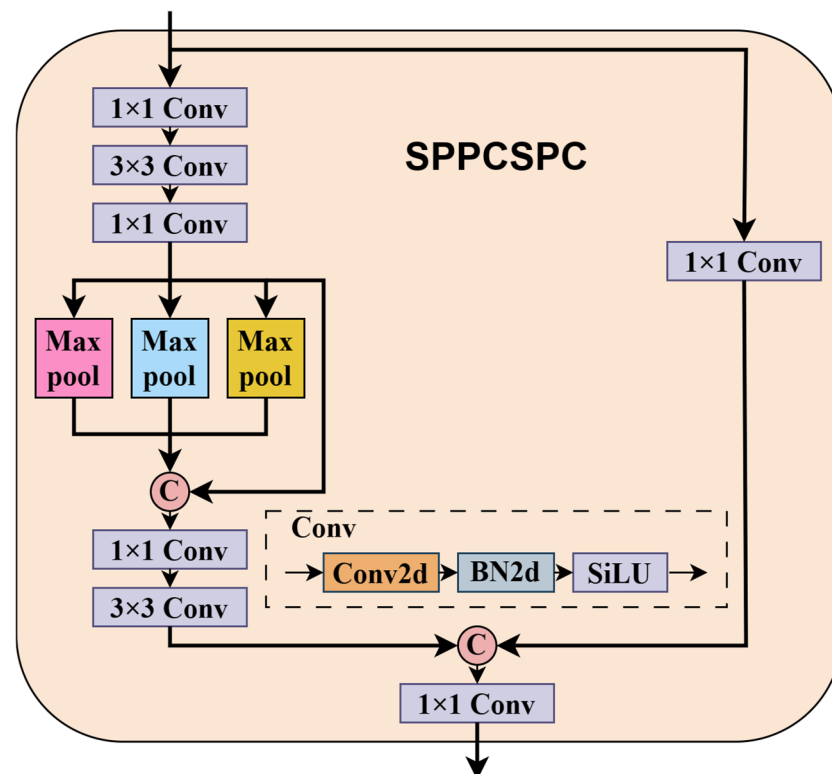


Figure 3. Structure of SPPCSPC module.

To solve these problems, this study redesigns the SPPCSPC module and proposes a novel spatial pyramid pooling module ASPPFCSPC. The structure is shown in Figure 4.

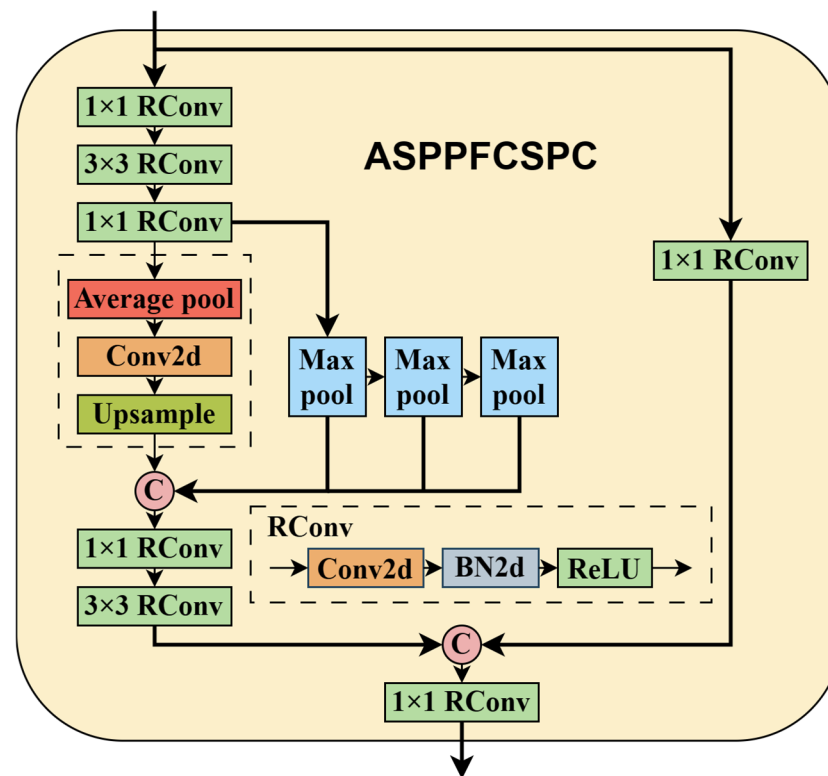


Figure 4. Structure of ASPPFCSPC module.

SPPF is based on SPP, which is much faster than SPP. Therefore, we first learn from the idea of SPPF, and the output of the upper branch is obtained by three 5×5 Max pooling operations on the features in order. At the same time, the receptive fields that pass through each Max pool layer are concatenated with them, which can effectively reduce the calculation cost of the pooling scale. The model's detection speed has been enhanced while maintaining the same receptive field as before improvement. Secondly, to fully reflect the semantic relationship between global and local information, we introduce the adaptive Average pool layer in the lower branch and parallel it with the Max pool layers. The Max pooling operation can extract the feature texture, while the Avg pooling operation can preserve the background information. The resulting scheme enables us to easily obtain local features, surrounding contexts, and global background information at the image level, making fine-grained feature extraction and fusion, and representation smoother and more accurate. Finally, for the activation function of RConv in ASPPFCSPC, we abandon the complex mathematical operation and use ReLU to optimize computational efficiency and accelerate model detection. ReLU has its own gating mechanism, which only performs nonlinear transformation on critical inputs, and the operation of linear processing of positive values can accelerate the convergence speed of Stochastic Gradient Descent (SGD), so that our neural network model is trained more efficiently. Therefore, the redesigned module ASPPFCSPC can effectively leverage the hardware's computing power, significantly reduce the delay and enhance the model's ability to extract, fuse and represent fine-grained features, so as to improve the detection accuracy and versatility of the model greatly.

3.2.2. Newly Improved Module C2f_SimDCNv2

In the original YOLOv8 model, the network uses the traditional conventional convolution, and the conventional convolution calculation formula is as follows:

$$y(p_0) = \sum_{p_n \in R} W(p_n) \cdot X(p_0 + p_n) \quad (1)$$

Here, R represents the size of the partial feature map corresponding to the convolution kernel; $W(p_n)$ represents the weight corresponding to the N TH sampling point in the sampling area, and $X(p_0 + p_n)$ represents the pixel value of n sampling points. The size of the convolution kernel in the traditional convolution layer is fixed, whereas the size of the object to be detected varies. The fixed size convolution kernel is difficult to accurately match the shape and size of the object to be detected, which will affect the accuracy of extracting the boundary information of the object to be detected, which will lead to defects such as poor robustness and poor universality.

While the difference between deformable convolution [49] and traditional convolution lies in that convolution is not fixed and can be adjusted adaptively. It incorporates trainable partial parameter Δp_n derived from conventional convolution, and its output formula is formulated as follows:

$$y(p_0) = \sum_{p_n \in R} W(p_n) \cdot X(p_0 + p_n + \Delta p_n) \quad (2)$$

Since $p_0 + p_n + \Delta p_n$ is generally a fraction, we can use bilinear interpolation to compute $X(p_0 + p_n + \Delta p_n)$:

$$y(p_0) = \sum_{q \in R} G(q, p) \cdot X(q) \quad (3)$$

where p represents any position in the region, q is the pixel value corresponding to the sampling point, and $G(q, p)$ represents the two-dimensional bilinear interpolation kernel.

Deformable convolution can strengthen the feature extraction ability of the underlying network for the target due to its unique adaptive ability, but the introduction of offset also leads to the possibility of covering irrelevant areas in the process of scale adjustment, thus interfering with the overall performance of the model. To solve this problem, an upgraded version of DCNv2 [50] is used in this study. The DCNv2 network architecture is shown in Figure 5. DCNv2 introduces a modulation mechanism to learn the weights of sampling points by introducing a modulation parameter $\Delta m_k \in [0, 1]$. For regions of no interest, the weight coefficient Δm_k is assigned a small value, so the above problem of interference can be avoided. The output formula of DCNv2 is as follows:

$$y(p_0) = \sum_{p_n \in R} W(p_n) \cdot X(p_0 + p_n + \Delta p_n) \cdot \Delta m_k \quad (4)$$

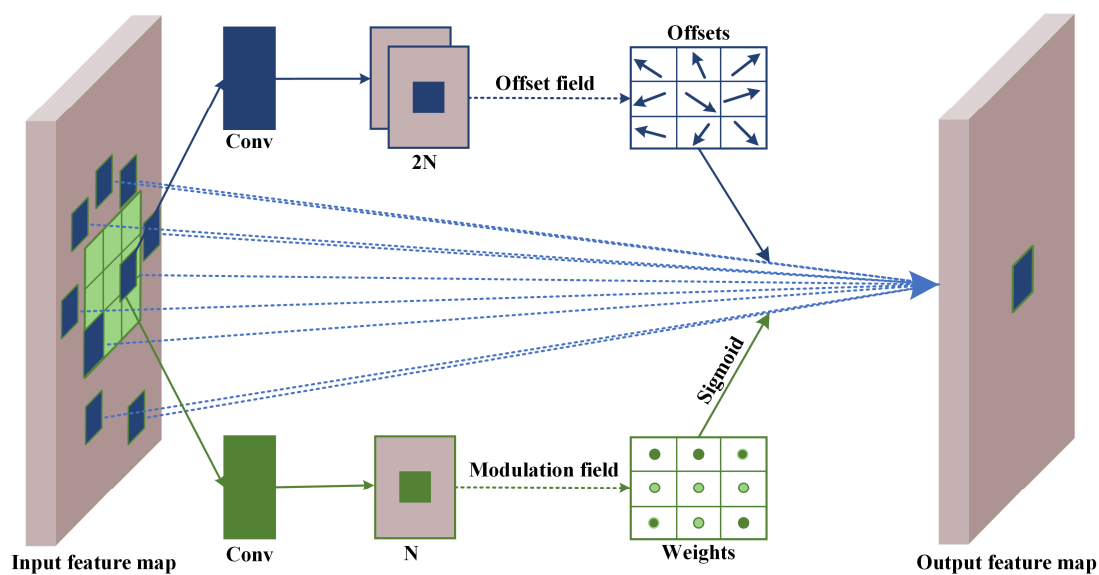


Figure 5. Network architecture of DCNv2.

In this study, the concept of DCNv2 is integrated into the C2f module of YOLOv8, which simply replaces the first Conv in the bottleneck with DCNv2, thus forming C2F_SimDCNv2. The improved Bottleneck_DCnV2 makes C2f inherit the advantages of deformable convnets. The primary responsibility of the backbone network is to extract shallow features and global information from the original image, which are essential for comprehending the overall context of the entire image. Hence, in this study, we replace the last three C2f modules in the original YOLOv8 backbone network with C2f_SimDCNv2 modules. In the following verification experiments of the improved strategy, the results show that compared with all replacement and last replacement, only replacing the first Conv of the bottleneck in C2f module with DCNv2 can make the performance of the model play the best. The introduction of DCNv2 significantly expands the receptive field of the C2f module, enabling the model to sample more abundant and effective gradient flow information. Consequently, the model can more accurately capture boundary and detail information pertaining to target defects amidst complex backgrounds. The improved module C2f_SimDCNv2 further strengthens the model's detection accuracy and robustness with its superior adaptive ability, which is especially suitable for the metal surface defect detection task with variable target shapes in this study. The improved module C2f_SimDCNv2 structure is shown in Figure 6.

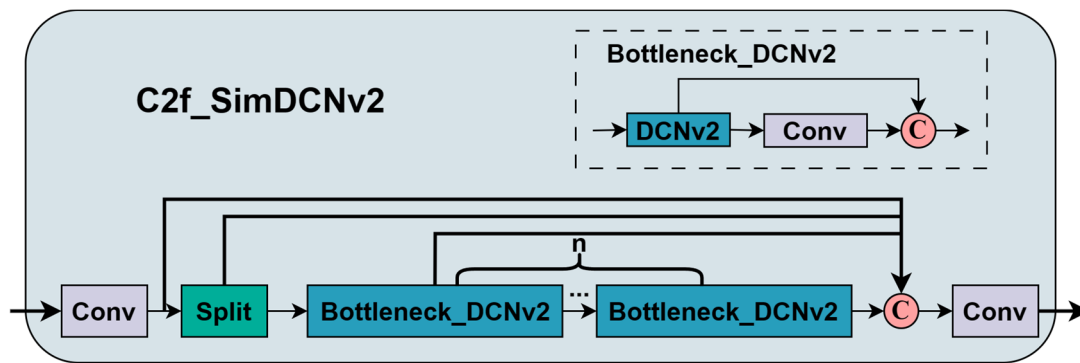


Figure 6. Structure of the improved module C2f_SimDCNv2.

3.2.3. The Small Target Killer: SPD Layer

To further improve the detection performance of the model for fine-grained defects on metal surfaces, this study also introduces the SPD layer of SPD-Conv [51] into the underlying backbone network. The SPD layer adopts the frame-cycle super-resolution conversion technique [52] to down-sample the feature map on the premise of preserving the feature information as much as possible, so it is very suitable for small targets and low-resolution images. Figure 7 gives an example when $scale = 2$.

Given a feature map of size $W \times H \times C$, the SPD layer will divide it into sub-feature maps in Equation (5) by sampling every pixel in every row and column:

$$\begin{aligned} f_{i,j} &= X[i : W : scale, j : H : scale] \\ i &\in [0, scale - 1], j \in [0, scale - 1] \end{aligned} \quad (5)$$

In Equation (5), X is the input feature map, f is the sub-feature map, and $scale$ is the scale factor of the sub-feature map. After obtaining these new sub-feature maps, they are concatenated along the channel dimension to reduce the spatial dimension and increase the channel dimension, that is, the output size of the feature map with the input size of (W, H, C) is $(\frac{W}{scale}, \frac{H}{scale}, scale^2 \times C)$. As you can see from the figure, the SPD layer is equivalent to performing a factor of 2 down-sampling to obtain four sub-feature map maps $f_{0,0}, f_{1,0}, f_{0,1}, f_{1,1}$ of size $(\frac{W}{2}, \frac{H}{2}, C)$, which are then concatenated to obtain $(\frac{W}{2}, \frac{H}{2}, 4 \times C)$. This design integrates the feature information on W and H into the channel dimension, expands the number of channels by four times, and the down-sampling method without

information loss enables the network to retain more feature information during feature extraction. The above work makes the small target features of fine-grained defects be retained as much as possible, and compared with the ordinary convolution alone, the parametrically free SPD layer recombines the spatial information and depth information, which can reduce the redundancy between convolution operations, thereby reducing the dependence on nonlinear activation, and speeding up the gradient inversion and reasoning speed of the model to a certain extent. Meanwhile, it also expands the receptive field and improves the overall detection performance of the model.

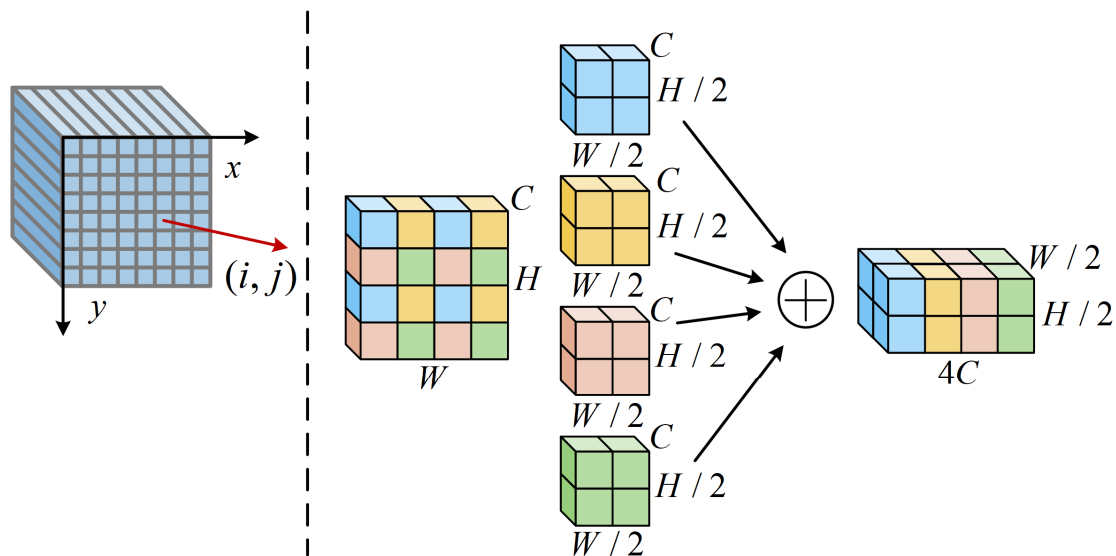


Figure 7. SPD layer down-sampling process at $scale = 2$.

4. Experiments and Analysis

4.1. Experimental Dataset

4.1.1. Causes of Defects

This study mainly focuses on hot rolled steel strips, including the following six common metal surface defects:

Crazing [53]. During hot rolling, internal stresses may form in the strip due to temperature gradients, deformation stresses and other factors. When the internal stress reaches the bearing limit of the material, it will lead to the formation of crazing. Surface crazing usually appears as linear or depressed defects extending along the surface of steel strips.

Patches [53]. The raw material of the steel strip may contain impurities such as non-metallic sulfides, which are prone to form patches during the hot rolling process. In addition, when the steel strip is cooled, the internal tissue structure is unstable due to the uneven local temperature gradient, which also forms patches. Patches usually appear as small round or irregular dark patches on the surface of steel strips, which vary in size and distribution density from tiny point to large sheet.

Rolled-in scale [53]. The steel strip is in contact with air and reacts with oxygen at high temperatures. Contaminants may be present on the steel strip surface if it is not cleaned adequately before hot rolling. These pollutants can promote oxidation reactions at high temperatures, leading to the formation of the rolled-in scale. The rolled-in scale usually appears as a rust-colored or dark brown covering, which makes the surface rough and uneven.

Scratches [53]. In the process of production, transportation and storage, if it comes into contact with hard or sharp objects, it may lead to the generation of scratches. As well as in the processing process, if the operator uses improper tools or methods, it may also cause the formation of scratches. Scratches vary in shape and size and usually appear on the surface of steel strips and can be directly and visibly observed.

Inclusion [53]. In the steelmaking process, if non-metallic substances such as impurities, oxides and sulfides are present in the molten steel, these substances may be carried to the billet during the hot rolling process, forming the inclusion. At the same time, chemical reactions in the smelting process produce solid substances, such as carbides and nitrides. If these reaction products are not completely removed or processed, they may also become the inclusion. Inclusion appears in different shapes, sizes, such as spherical, sheet, and filamentous, and is usually unevenly distributed.

Pitting surface [53]. In the hot rolling process, when liquid metal or impurities exist on the surface of the billet, the impact force at high temperature will make the droplets splash. These droplets will solidify and deposit on the surface of steel strips after cooling, forming the pitting surface. If impurities are present in the molten steel, they may also be carried to the surface, forming the pitting surface during hot rolling. Pitting surface usually appears as small round or irregularly shaped dark spots on the surface of steel strips, which vary in size and distribution density.

All of the above defects have a negative impact on the aesthetics and functionality. Due to the different causes, the same type of defects shows different characteristics, and the distinction between different types of defects is not obvious enough, so it is extremely difficult to identify metal surface defects accurately.

4.1.2. Defect Dataset

In the actual steel strip production line, the new production line has a good environment, defects do not occur often, and there is a certain randomness. It often takes a lot of time to collect enough samples. Due to the different production line environments of different steel mills, the images vary greatly, and the quality of the collected images is uneven. However, the model trained by a small number of samples will have poor generalization ability and insufficient feature representation ability. Therefore, this study mainly focuses on the steel strip surface defect dataset (NEU-DET) provided by Song's team in Northeastern University (NEU) [54,55].

This dataset collects six typical surface defects of hot rolled steel strips, these are crazing, patches, rolled-in scale, scratches, inclusion, and pitted surface. The dataset includes 1800 grayscale images, each with a resolution of 200×200 pixels, and 300 samples for each defect. NEU-DET dataset can cover different types and degrees of steel strip surface defects, which can help evaluate and compare the performance of different defect detection algorithms in practical applications. NEU-DET dataset has become one of the widely used datasets in the field of metal surface defect detection. Typical samples of NEU-DET dataset are shown in Figure 8.

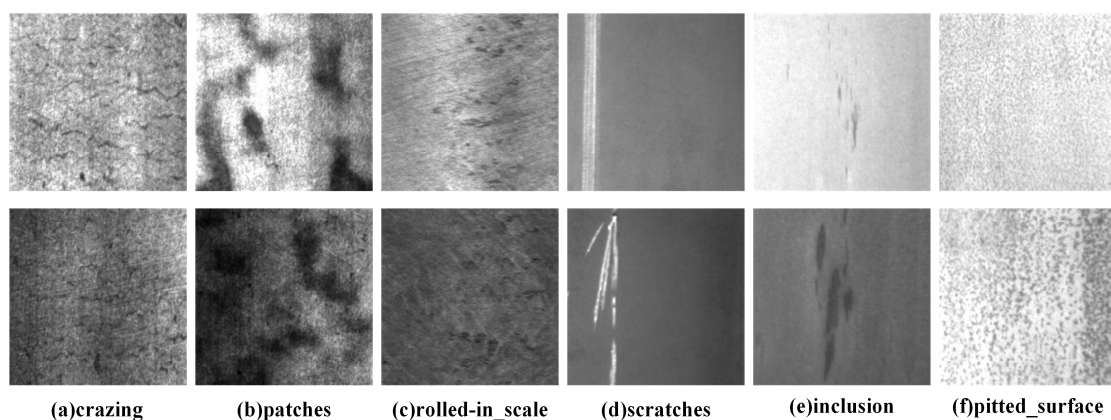


Figure 8. Six typical samples of NEU-DET dataset.

It is worth noting that there are two problems in this defect dataset. First, fine-grained defects (crazing, patches, and rolled-in scale) are easily affected by the background. Due to the influence of illumination and material changes, the gray value of the defect image

within the class will change, and the dust in the environment will lead to low contrast with the background, which reduces the representation ability of the defect. Secondly, the scale of fine-grained defects is variable. For example, in the intra-class defect image, the crazing defect may be a connected middle object or a small object, and the small object itself has low resolution, so the context information for the model to learn is limited. Therefore, the detection task of fine-grained defects in this study is more challenging.

4.1.3. Statistics and Analysis

As shown in some samples in the NEU-DET dataset above, the number of features, defect size and background in each defect sample image are quite different, and these factors will affect the selection of the detection model and the corresponding improvement strategy. To make the detection model achieve better detection results, this study conducted detailed statistics and analysis of defect samples in NEU-DET dataset.

Figure 9 shows the distribution of the number of defect categories in the dataset. The inclusion category has the largest number of defects, more than 1000, followed by the patches category with about 900, the crazing category with about 700, and the least defect category pitted surface with only 400. The number of various types of defects is quite different.

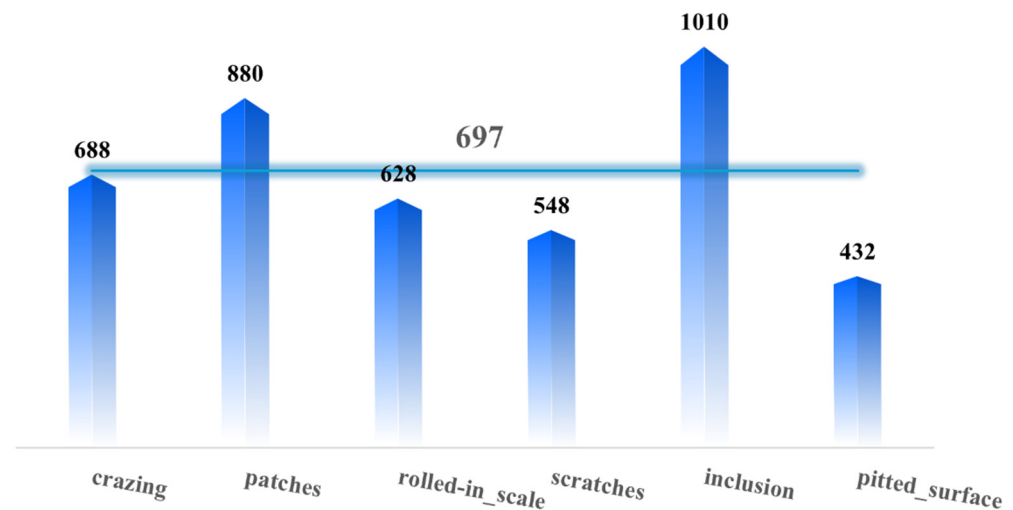


Figure 9. Distribution of the number of defect categories.

Among them, for defects with relatively obvious characteristics such as scratches and pitting surface, only four or five hundred sample images are enough to support the training of the detection model. While for fine-grained defects such as crazing and rolled-in scale, which are easily affected by the background and have variable feature scales within the class, six or seven hundred training samples are still far from enough. Therefore, it is necessary to improve the model for such fine-grained defects specifically.

Figure 10 shows the distribution of the proportion of defect parts area in the dataset. Firstly, by comparison, it is found that 44.8% of the defect parts are less than 1/10 of the sample image area, 26.2% of the defect parts are between 1/10 and 2/10, and 71% of the defects are between 0–2/10. It shows that the small-sized and medium-sized defect parts are the majority, and the design and analysis of the model should be inclined to small and medium targets. Secondly, the defects in pitted surface category in 0–2/10 area proportion only account for 1.4%, which can be temporarily ignored in this study. Finally, it is obvious from Figure 8 that the defect image background of scratches and inclusion classes is relatively plain and the defect features are comparatively obvious. Moreover, this research plan solves the problem that the fine-grained defect on the metal surface is susceptible to background interference and its feature scale is variable. Therefore, the

design and analysis of the model in this study mainly focus on the remaining three types of defects: crazing, patches, and rolled-in scale.

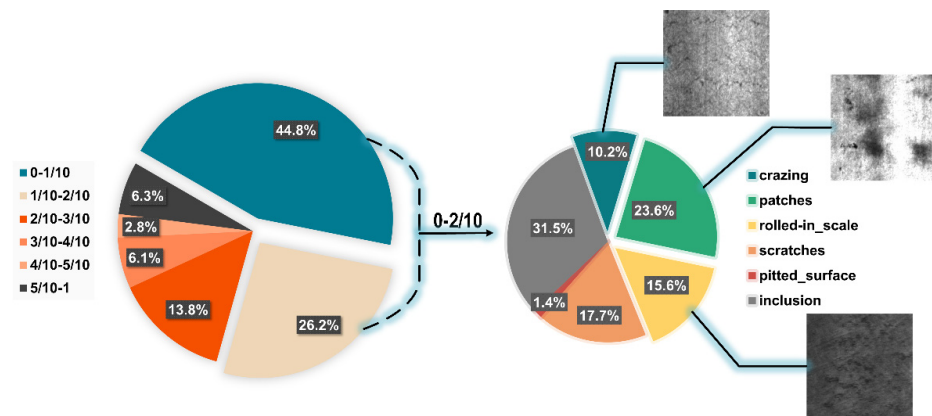


Figure 10. Distribution of the proportion of defect parts area.

In summary, through the statistics and analysis of strip surface defects, the target task of this study is clarified, which is to improve the model structure for fine-grained defects such as crazing, patches, and rolled-in scale that are susceptible to background influence and variability in feature scales, so as to improve the detection accuracy of such defects and the versatility of the model for metal surface defect detection tasks.

4.2. Experimental Settings

The experimental environment utilized in this study employed the Windows 11 operating system with a memory capacity of 32 GB. In the training phase, a single GPU (NVIDIA GeForce RTX 4060Ti 16GB) is used, and the compilation environment is Python3.8.19 + torch2.0.0 + CUDA11.7. The image input size is set to 640×640 , with data augmentation performed using the Mosaic method. The initial learning rate is established at 0.01, while Batchsize is configured as 16. The SGD optimizer has been chosen, and the number of training iterations has been set to 150 rounds. Finally, default parameters are employed for all remaining hyperparameters, and the above experimental setup is used for all experiments in this paper unless otherwise stated.

In this study, the steel strip surface defect dataset NEU-DET analyzed in Section 4.1 was primarily used for training and testing. It was partitioned into a training set, validation set, and test set with an 8:1:1 ratio respectively. Consequently, the training set consisted of 1440 images, while the validation and test sets contained 180 images each.

4.3. Evaluation Metrics

In this study, mAP, Precision (P), Recall (R) and FPS are mainly used as evaluation criteria to evaluate each model. The formula for P and R is as follows:

$$P = \frac{TP}{TP + FP} \quad (6)$$

$$R = \frac{TP}{TP + FN} \quad (7)$$

TP (True Positive), FP (False Positive), and FN (False Negative) respectively represent the number of defect targets correctly identified, incorrectly identified, and missed. In the context of metal surface defect detection, Precision and Recall play a crucial role and typically exhibit an inverse relationship.

mAP refers to the mean Average Precision (AP) of all categories of defects, AP refers to the area of the curve below the Precision-Recall (P-R) curve, the formula of AP and

mAP is as follows, the higher the value of mAP, the better the comprehensive detection performance of the model in all categories, the specific formula is as follows:

$$AP = \int_0^1 p(r) dr \quad (8)$$

$$mAP = \frac{1}{N} \sum_{i=1}^N AP_i \quad (9)$$

IoU represents the ratio of intersection and union between the bounding box of the original label and the predicted bounding box by the model. All the mAP in this study indicates the average accuracy of all detected defect targets in the test set, considering correct localization when IoU exceeds 0.5, and vice versa. A higher mAP signifies improved detection performance across different object categories.

FPS reflects the number of images processed by the model per unit time, and a higher FPS value indicates a better real-time performance of the model, the specific formula is as follows:

$$FPS = \frac{frameNum}{elapsedTime} \quad (10)$$

where frameNum is the total number of frames processed and elapsedTime is the time used.

In the improved strategies verification experiments and the comparison experiment with mainstream object detection algorithm, Parameter (Param) is used as a key indicator to reflect the number of parameters occupied by the model memory. The fewer Param, the lighter the model. It is verified that the number of model parameters of the proposed algorithm YOLO-ADS has certain advantages, which meets the requirements of memory consumption of industrial deployment devices.

4.4. Improved Strategies Verification Experiments

To verify that the strategies of improving ASPPFCSPC module and C2f_SimDCNv2 module is correct and effective, two sets of comparative experiments are set up on NEU-DET dataset, and the experimental results are shown in Tables 1 and 2.

Table 1. Comparison of verification experiments of ASPPFCSPC.

Model	mAP/%	P/%	R/%	Param/M	FPS
SPPF(YOLOv8n)	77.9	73.6	73.7	3.0	169.5
SPPCSPC	78.5	76.0	72.8	4.6	149.9
ASPPFCSPC(Ours)	79.4	80.8	71.2	4.7	157.7

Table 2. Comparison of verification experiments of C2f_SimDCNv2.

Model	mAP/%	P/%	R/%	Param/M	FPS
C2f(YOLOv8n)	77.9	73.6	73.7	3.0	169.5
C2f_DCNv2_all	79.6	77.3	74.2	3.2	119.8
C2f_DCNv2_2nd	78.5	76.2	74.8	3.1	138.2
C2f_SimDCNv2(Ours)	80.1	79.8	71.9	3.1	145.7

Table 1 shows that the mAP of the model using SPPCSPC module is 0.6% higher than that of the baseline model using SPPF module. This is because SPPCSPC module integrates the Cross Stage Partial Network (CSPNet) structure, which can realize the channel splicing between the output of the previous layer of SPP and the output of the small residual structure after passing through the convolution module. Thus, more useful feature information can be extracted. However, the number of parameters of the model using SPPCSPC module is 1.5 times that of the baseline model, which also shows that CSPNet is the main reason for the increase of parameters and the decrease of FPS of the improved model. And the mAP of the model using our novel proposed spatial

pyramid module ASPPFCSPC further improves by 0.9% to reach the highest, which is 1.5% higher than the baseline model using SPPF module, and the Precision is improved by 7.2%, which is also the highest. This is because we introduce an adaptive Average pool layer to enhance the semantic relationship between local features, surrounding contexts and global background information. It makes multi-scale feature extraction and fusion, and representation smoother and more accurate. At the same time, we draw on the idea of Max pool layer concatenation in SPPF to reduce the calculation cost of pool scale, and use the simpler activation function ReLU with faster response, which effectively alleviates the problem of slow detection speed caused by the increase of model parameters. Therefore, while the number of model parameters and the detection speed FPS meet the real-time requirements of industrial deployment, the ASPPFCSPC module improves the mAP and Precision of the model, which verifies that the strategy to improve ASPPFCSPC is correct and effective.

From Table 2, it is obvious that compared with the C2f_DCNv2_all scheme, which replaces all the Conv bits in the bottleneck with DCNv2, and the C2f_DCNv2_2nd scheme, which replaces only the last Conv bits in the bottleneck with DCNv2, The C2f_SimDCNv2 scheme only replaces the first Conv in the bottleneck with DCNv2 and performs the best in the improved comparison experiment. The mAP of the model using improved module C2f_SimDCNv2 reaches the highest, which is 2.2% higher than that of the baseline model using C2f module, and the Precision is increased by 6.2%, which is also the highest. While the number of model parameters and the detection speed FPS meet the real-time requirements of industrial deployment, the C2f_SimDCNv2 module improves the mAP and Precision of the model, which verifies that the strategy of improving C2f_SimDCNv2 is correct and optimal.

4.5. Ablation Experiments

In this study, YOLOv8n is used as the baseline to improve the backbone network of the model, and the spatial pyramid pooling module uses the novel proposed ASPPFCSPC module to fully reflect the semantic relationship between local features and global background information to enhance the fusion and representation ability of the model for fine-grained defects, so as to improve the detection accuracy and versatility of the model. Meanwhile, the DCNv2 is used to improve the first convolution in the bottleneck to improve the model's ability to extract multi-scale features. The C2f_SimDCNv2 module enhances the detection accuracy and robustness of the model by virtue of its superior adaptive ability. In addition, to further improve the detection performance of the model for small target defects, the SPD layer is introduced to expand the receptive field, so that the network significantly reduces the loss of fine-grained features and effectively improves the model's detection accuracy. The parameter-free SPD layer mitigates the impact on nonlinear activation, enhances both inference speed and gradient inversion speed of the model to a certain extent, thereby enhancing the overall detection performance of the model. To validate the effectiveness of this study's improvement over the baseline model, ablation experiments are conducted on NEU-DET dataset from two perspectives:

1. Based on the baseline model, only one improvement module is introduced at a time to verify the impact of individual modules on mAP, Precision, Recall and detection speed.
2. Only one improvement module was eliminated at a time on the final model (YOLO-ADS) (excluding the elimination of ASPPFCSPC module) to verify the impact of individual improvement modules on the performance of the final model. The experimental results are shown in Table 3.

According to the ablation experimental results in Table 3, for model 1, we use the new ASPPFCSPC module proposed in this study in the backbone network of the baseline model YOLOv8n, and both mAP and Precision are significantly improved, mAP is increased by 1.5%, and the Precision is greatly increased by 7.2%. The AP of the target defect categories of this study, crazing, patches and rolled-in scale has been improved by 2.5%, 2.7% and

4.9%, respectively. The ASPPFCSPC module introduces an average pooling layer in the lower branch of the spatial pyramid pooling structure, which fully reflects the semantic relationship between local features and global background information, and makes the extraction, fusion and representation of fine-grained defects more fluent. Therefore, the ASPPFCSPC module as the feature splicing fusion end in the backbone network can effectively improve the detection accuracy of the model. For model 2, the improved C2f_SimDCNv2 module still has good performance by virtue of the superior adaptive ability of deformable convolution to multi-scale features, mAP is increased by 2.2%, and the Precision is greatly improved by 6.2%. The AP of the target defect categories of crazing, patches and rolled-in scale in this study was greatly improved by 5.3%, 5.5% and 3.8%, respectively, which undoubtedly verified the effectiveness of C2f_SimDCNv2 module for multi-scale feature extraction. For model 3, after introducing the SPD layer alone, a double down-sampling feature map without feature information loss is obtained by expanding the number of channels, which improves the receptive field. In this way, the dependence on nonlinear activation is reduced, and the inference speed and gradient inversion speed of the model are improved. The improved model mAP was consistent with the baseline model, but the Precision was increased by 4.3%. The detection accuracy of the target defect categories of crazing, patches and rolled-in scale defects in this study was improved, and the FPS reached the highest of 177.9. For model 4, after introducing the ASPPFCSPC module and SPD layer at the same time, mAP is increased by 1.3%, the Precision is greatly improved to the highest of 82.0%, and the Recall is increased by 2.5% compared with model 3 using SPD layer alone. The detection accuracy of the target defect categories of this study, crazing, patches and rolled-in scale has been improved, and the AP of rolled-in scale defects has been greatly improved by 7.2%. For model 5, when ASPPFCSPC module and C2f_SimDCNv2 module are introduced at the same time, mAP innovation reaches 80.3% and Recall reaches 76.0%. However, the Precision of model 1 is poor when ASPPFCSPC module is introduced alone, which is only 0.1% higher than the baseline model. The detection accuracy of the target defect categories of this study, crazing, patches and rolled-in scale has been improved, and the AP of rolled-in scale defect has been greatly increased by 11.5%, reaching the highest of 73.0%. The final model proposed in this paper introduces the ASPPFCSPC module, C2f_SimDCNv2 module and SPD layer at the same time, and the mAP of the model is up to the highest of 81.4%. The AP of the target defect categories of crazing, patches and rolled-in scale in this study was greatly improved by 5.5%, 7.2% and 10.3%, respectively, and the AP of crazing and patches defects reached the highest of 59.6% and 94.9%, respectively. In addition, Precision and Recall are usually in a trade-off, and the final model YOLO-ADS proposed in this paper significantly improves the Precision by 6.1% to 79.7% on the premise that the Recall is 0.8% higher than that of the baseline model. At the same time, the FPS finally reaches 140.4, which fully meets the real-time requirements of industrial deployment.

Table 3. Ablation experiments on NEU-DET dataset.

Model	ASPPFCSPC	C2f_SimDCNv2	SPD	AP/%						mAP/%	P/%	R/%	FPS
				Cr	Pa	Rs	Sc	In	Ps				
YOLOv8n	-	-	-	54.1	87.7	61.5	95.0	85.2	83.8	77.9	73.6	73.7	169.5
1	✓			56.6	90.4	66.4	95.5	85.6	81.7	79.4	80.8	71.2	157.7
2		✓		59.4	93.2	65.3	93.6	86.7	82.3	80.1	79.8	71.9	145.6
3			✓	52.9	91.7	66.2	94.1	81.3	81.3	77.9	77.9	70.0	177.9
4	✓		✓	54.8	91.5	68.7	95.4	84.2	80.5	79.2	82.0	72.5	165.9
5	✓	✓		55.6	90.8	73.0	92.1	88.0	82.2	80.3	73.7	76.0	132.7
YOLO-ADS	✓	✓	✓	59.6	94.9	71.8	95.3	84.5	82.4	81.4	79.7	74.5	140.4

Note: Cr, Pa, Rs, Sc, In and Ps respectively denote crazing, patches, rolled-in scale, scratches, inclusion and pitted surface. And ✓ means to introduce the corresponding improvement module.

In summary, the metal surface defect detection algorithm YOLO-ADS proposed in this study can effectively solve the problem that fine-grained defects on the metal surface are susceptible to background interference and feature scale variation. It not only improves

the mAP of the baseline model YOLOv8n, but also maintains good Precision, Recall and real-time performance.

4.6. Visual Qualitative Analysis Experiments

The P-R curves in Figures 11 and 12 show the experimental results of the baseline model YOLOv8n and the improved YOLO-ADS on the test set under the same experimental Settings in Section 4.2. The figure shows the defects of each category and the overall mAP, so it is not difficult to see the optimization trend. The improved YOLO-ADS algorithm improves mAP from 77.9% to 81.4%, a significant increase of 3.5%. It is worth noting that in the detection of the YOLOv8n algorithm, the AP of the target defect categories of this study, crazing, patches and rolled-in scale, are only 54.1%, 87.7% and 61.5%, respectively, while in the detection of the improved YOLO-ADS algorithm, they reach 59.6%, 94.9% and 71.8%. Compared with the baseline algorithm, the performance improvement of the model for fine-grained defect detection is very significant.

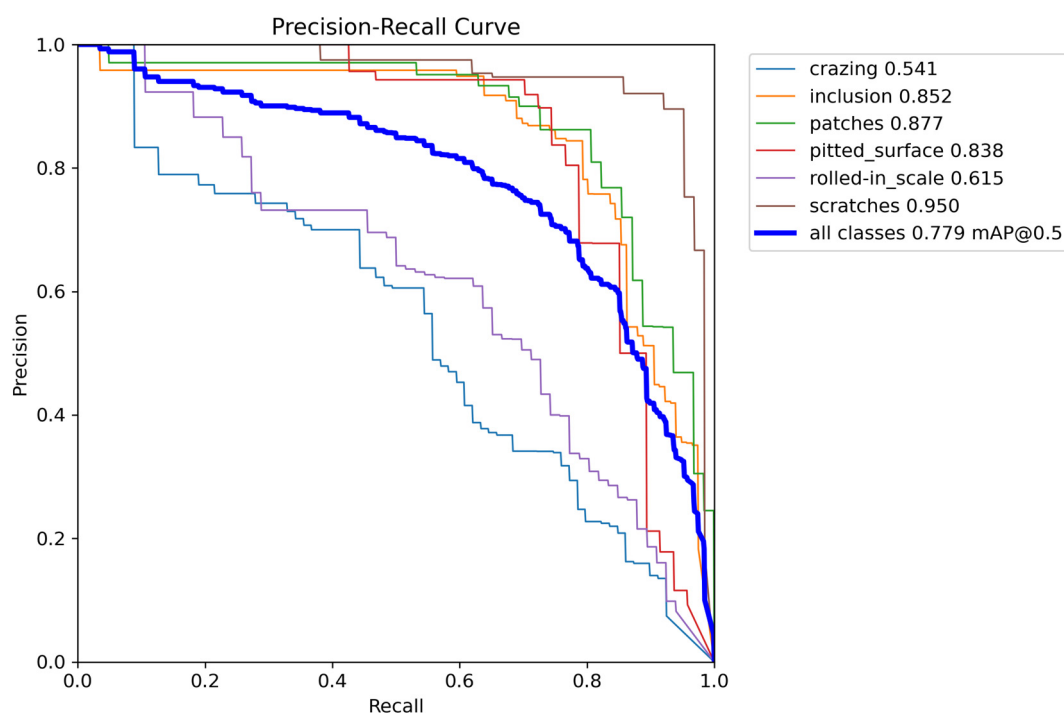


Figure 11. P-R curve of YOLOv8n algorithm.

To more intuitively compare the detection performance of the proposed YOLO-ADS with the baseline algorithm in detecting metal surface defects, YOLOv8n and YOLO-ADS are respectively used for further qualitative analysis on the test set of NEU-DET dataset, and the qualitative results are shown in Figure 13. The first row shows the detection results of the baseline model, and the second row shows the detection results of YOLO-ADS. Comparing the detection results of each column shows that the proposed YOLO-ADS detects the fine-grained defects with varying scales missed by the baseline model, and it also captures the missed fine-grained defects for large target defects such as pitted surface. In addition, it is hearty to find that although YOLO-ADS performs poorly on the evaluation metrics in training for non-target defect inclusion and pitted surface in this study, in the actual detection, the confidence scores are also all higher than the baseline algorithm.

In summary, the results of qualitative analysis experiments on NEU-DET dataset show that the proposed YOLO-ADS has obvious advantages over the baseline algorithm YOLOv8n in the detection of fine-grained metal surface defects with varying scales under such complex backgrounds.

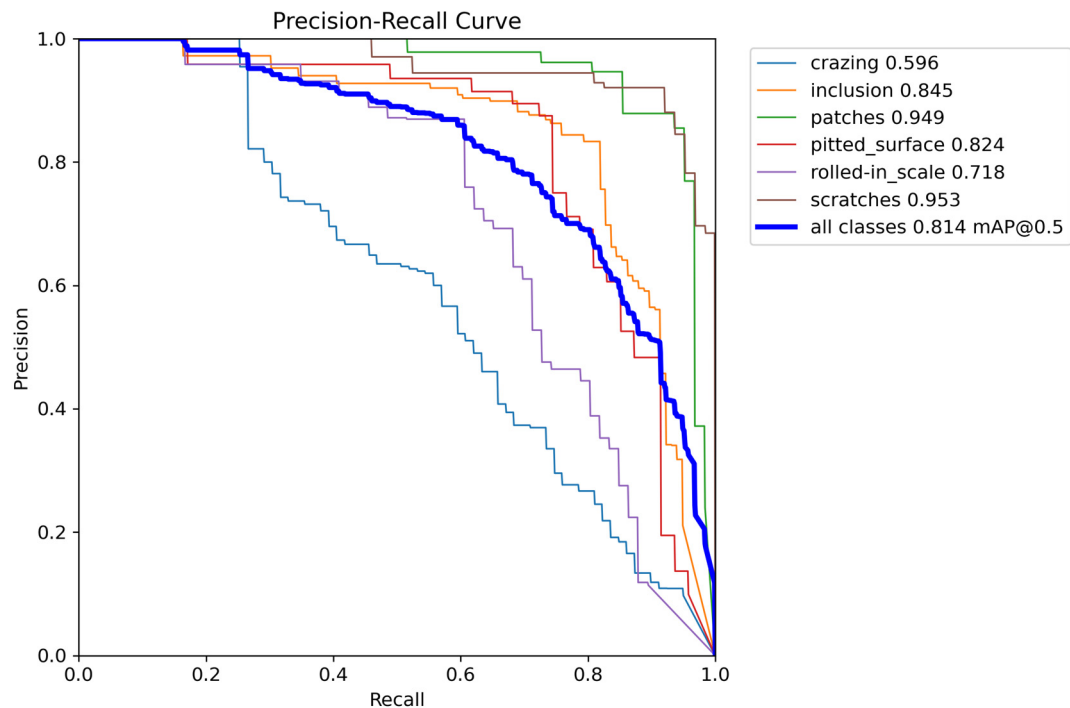


Figure 12. P-R curve of the improved algorithm YOLO-ADS.

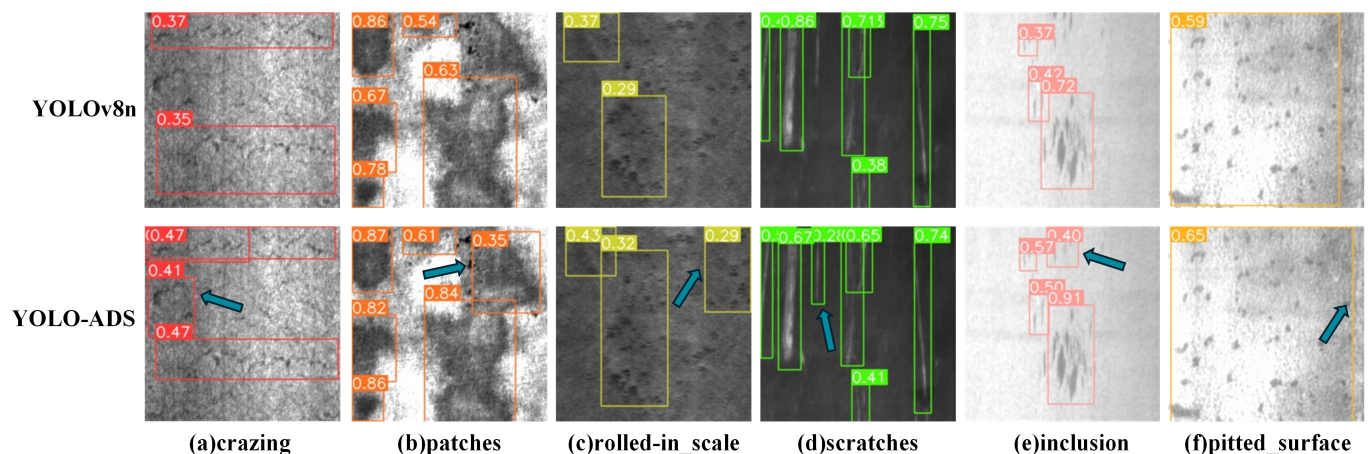


Figure 13. Qualitative analysis on NEU-DET dataset.

4.7. Versatility and Robustness Verification Experiments

To further verify the versatility and robustness of the proposed algorithm YOLO-ADS, the baseline algorithm and YOLO-ADS are tested on the public GC10-DET [56], APSPC and VOC2012 datasets [57]. GC10-DET dataset collected 2294 real industrial steel surface defect images, including 10 defect types, namely, punching, crescent gap, waist folding, rolled pit, water spot, silk spot, inclusion, oil spot, crease and weld line. The APSPC dataset collects 1885 real industrial aluminum surface defect images, which contain 10 types of defect categories, including non-conductive, scratches, corner leakage, orange peel, leakage, jet, paint blister, pit, variegated color and dirty spots. VOC2012 dataset serves as a publicly available benchmark, offering a standardized evaluation system for detection algorithms and learning performance. It encompasses 17,125 images featuring 20 distinct object categories. All experimental Settings remain the same as in Section 4.2. The experimental results are shown in Tables 4–6.

Table 4. Versatility and robustness verification experiments on GC10-DET dataset.

Model	mAP/%	P/%	R/%	FPS
YOLOv8n	64.3	64.3	64.8	173.0
YOLO-ADS	68.0	67.2	65.7	138.7

Table 5. Versatility and robustness verification experiments on APSPC dataset.

Model	mAP/%	P/%	R/%	FPS
YOLOv8n	60.7	72.2	55.6	97.2
YOLO-ADS	64.1	75.0	59.1	113.1

Table 6. Versatility and robustness verification experiments on VOC2012 dataset.

Model	mAP/%	P/%	R/%	FPS
YOLOv8n	60.6	66.5	54.5	169.4
YOLO-ADS	64.9	69.1	58.3	135.3

According to the experimental results in Tables 4 and 5, the proposed algorithm YOLO-ADS shows obvious advantages in steel and aluminum surface defect detection tasks. Compared with the baseline model, mAP is increased by 3.7% and 3.4%, the Precision is increased by 2.9% and 2.8%, and the Recall is increased by 0.9% and 3.5%, respectively. It means that the proposed algorithm YOLO-ADS can be applied to different metal surface defect types and has strong versatility and robustness. It is worth noting that the performance of YOLO-ADS on the two datasets meets the real-time requirements for industrial deployment. Meanwhile, the FPS of YOLO-ADS algorithm on the aluminum surface defect detection task reaches 113.1, which is higher than the baseline model 15.9, which further verifies that YOLO-ADS is may more suitable for aluminum such metal surface defects, and thus shows higher detection speed.

According to the experimental results in Table 6, on the larger and more diverse data set VOC2012, YOLO-ADS proposed in this study has 4.3% improvement on mAP, 2.6% improvement on Precision, and 3.8% improvement on Recall compared with the baseline model. This shows that the YOLO-ADS algorithm can better learn and capture the characteristics of the target, has strong robustness, and further verifies its good versatility. At the same time, the FPS reaches 135.3, which also meets the real-time requirements of side device deployment.

In summary, the experimental results on the industrial steel dataset GC10-DET, the industrial aluminum dataset APSPC and the larger public benchmark dataset VOC2012 show that the YOLO-ADS algorithm proposed in this study has strong versatility and robustness.

4.8. Comparison Experiments with Current Mainstream Algorithms

Finally, in order to prove the advancement of the proposed algorithm in metal surface defect detection, this study selects four mainstream target detection algorithms Faster R-CNN, RT-DETR-r18, SSD300 and YOLO series representative algorithms to compare with YOLO-ADS algorithm on NEU-DET dataset. The results are shown in Tables 7 and 8.

Table 7 shows that, compared with RT-DETR-r18, SSD300 and YOLO series representative algorithms, Fast-RCNN has the highest accuracy in strip surface defect detection with the advantages of two-stage detection network, but its shortcomings are also obvious: the number of model parameters is large, and the FPS performance is poor, only 25.6. It is not conducive to the real-time demand of industrial deployment. In addition, it can be clearly seen from the table that YOLO-ADS proposed in this study improves both mAP and Precision to the highest among the current mainstream algorithms, with 81.4 and 79.7%, respectively, and the recall rate is second only to YOLOv9tiny with 74.5%. At the same time, the number of model parameters is only 5.0 M, and the FPS performance reaches 140.4,

and the real-time performance is at the upper middle level among the current mainstream algorithms. Compared with RT-DETR-r18, the mAP_{0.5} is increased by 8.4%, and the number of model parameters is much lower than RT-DETR-r18. YOLO-ADS's mAP is 3.2%, 8.0% and 3.5% higher than the newly released YOLOv9s and YOLOv10s and the baseline model YOLOv8n that balances accuracy and real-time performance, and the Precision is also 7.8%, 8.5% and 6.1% higher than them, respectively.

Table 7. Comparison with mainstream object detection algorithms on NEU-DET dataset.

Model	mAP/%	P/%	R/%	Param (M)	FPS
Faster R-CNN	78.4	76.8	61.0	137.1	25.6
RT-DETR-r18	73.0	76.1	66.9	19.9	64.9
SSD300	73.7	69.3	48.6	24.3	127.1
YOLOv5s	75.0	72.1	69.3	7.0	166.2
YOLOv7tiny	71.1	64.1	70.8	6.0	164.0
YOLOv8n	77.9	73.6	73.7	3.0	169.5
YOLOv9tiny	77.8	69.9	75.7	2.6	61.0
YOLOv9s	78.2	71.9	74.0	9.6	56.5
YOLOv10n	69.0	63.4	64.2	2.7	101.9
YOLOv10s	73.4	71.2	68.9	8.0	92.0
YOLO-ADS	81.4	79.7	74.5	5.0	140.4

Table 8. AP of six types of defects.

Model	AP/%					
	Cr	Pa	Rs	Sc	In	Ps
Faster R-CNN	45.3	92.4	63.5	95.1	86.1	88.0
RT-DETR-r18	35.8	89.1	65.7	93.5	78.5	75.6
SSD300	38.9	91.7	62.5	83.3	81.9	83.9
YOLOv5s	48.5	91.3	56.6	88.8	83.2	81.7
YOLOv7tiny	40.5	88.4	51.8	88.3	77.4	80.0
YOLOv8n	54.1	87.7	61.5	95.0	85.2	83.8
YOLOv9tiny	41.4	93.9	63.7	91.9	85.2	90.4
YOLOv9s	44.9	91.4	61.2	90.4	88.8	92.5
YOLOv10n	41.8	88.2	61.2	78.9	74.2	69.8
YOLOv10s	43.3	85.9	63.2	91.5	78.7	77.6
YOLO-ADS	59.6	94.9	71.8	95.3	84.5	82.4

More closely, it can be seen from Table 8 that the proposed algorithm YOLO-ADS has the best performance in the mainstream target detection algorithms for the target defect categories of crazing, patches and rolled-in scale, which are greatly increased by 17.8%, 6.7% and 10.6% respectively compared with the newly released lightweight model YOLOv10n. This fully reflects the detection task of this study, that is, YOLO-ADS solves the problem such as susceptibility to background interference and variability in feature scales of fine-grained defects on metal surfaces, has a good performance.

In summary, the YOLO-ADS proposed in this study not only maintains the advanced level of detection accuracy, but also has a small amount of storage parameters, and FPS meets the real-time requirements of industrial deployment, so that the detection task can be accurately and quickly completed in a resource-limited environment, which provides a new alternative scheme for metal surface defect detection.

Finally, it should be recalled that in the experimental settings of Section 4.2, this study utilized the default YOLOv8n hyperparameters without retuning, implying that the model is likely to exhibit improved performance following specialized hyperparameter tuning. Additionally, it is widely acknowledged that all non-YOLO baselines underwent training using transfer learning techniques and thus benefited from high-quality images, whereas YOLO-ADS algorithm did not.

5. Discussion

5.1. Findings

Firstly, as evident from Table 2, simply replacing the first Conv in the bottleneck with DCNv2 yields better performance compared to replacing only the last Conv, and significantly outperforms replacing both of them. This typically illustrates that strategic, moderate improvements in the right locations are crucial for the key to model enhancements, rather than indiscriminately adding State-Of-The-Art (SOTA) work points.

Secondly, this study finds that the design of the backbone network is vital, which not only focuses on the design of feature extraction, but also pays attention to the feature fusion effect of the model. As can be seen from Table 3, the backbone network of YOLO-ADS achieves much higher detection accuracy than the baseline model YOLOv8n after reconstruction with ASPPFCSPC, C2f_SimDCNv2 and SPD modules. In addition, it can be found that the models using the ASPPFCSPC module can show good detection accuracy, which is all thanks to the novel proposed spatial feature pyramid pooling module ASPPFCSPC, which can better fuse the semantic relationship between local features and global background information, and also pay attention to the fine-grained features that may be missed in complex backgrounds. It is not difficult to see from the improvement effect of AP of the target defect categories crazing, patches and rolled-in scale to 56.6%, 90.4% and 66.4% after only adding the novel proposed ASPPFCSPC module. Thus, the overall detection level of the model is established, and the mAP is maintained above 79.4%.

Besides, in terms of datasets, it can be seen from Figure 9 and Table 3 that the current dataset is far from enough, which is one of the main reasons why the improved model still has low detection accuracy for crazing and rolled-in scale defects. Therefore, it is necessary to increase the expansion of the dataset and improve the sample quality of the dataset.

Additionally, from Table 5, it is gratifying to observe that YOLO-ADS, despite having a larger number of model parameters, surprisingly exhibits better detection speed on the APSPC dataset than YOLOv8n. This preliminarily demonstrates that there is no absolute direct correlation between FPS and the number of parameters, and it may also depend on the complexity of the object detection task.

Finally, according to the detection effect of the target defect categories of this study, it is very effective to design a unique structure to focus on those features that are easy to be ignored. Of course, this also needs to keep the efficiency of the detection model as much as possible, instead of improving the accuracy of the model by blindly designing more redundant parameter-heavy structures.

5.2. Limitation and Future Works

Compared with the most mainstream methods in the field of metal surface defect detection, YOLO-ADS satisfies the comprehensive considerations of detection accuracy, real-time detection, and model light-weighting, exhibiting a notable degree of competitiveness. And it also performs well in terms of universality and robustness, but there is still a lot of room for improvement in detection accuracy and model parameter quantity.

It is worth mentioning that all the datasets used in this study are public datasets commonly used in the industry, and the improvement of their detection accuracy is due to the improvement of the algorithm. However, in the actual metal surface defect detection task, the detection accuracy is mainly affected by the quantity and quality of datasets, which are largely limited by the objective conditions to be selected, such as lighting, viewing angle, etc. In the future deployment work, we will focus on further research. Meanwhile, in terms of the number of datasets, Generative Adversarial Network (GAN) and diffusion models can also be used to obtain more natural defect samples. In terms of the quality of datasets, specific image preprocessing techniques can be used for the sample, super-resolution algorithms such as Real-ESRGAN [58] can also be used to enhance the model's ability to detect low-resolution defects, and label information can be further optimized to locate defect information more accurately.

In addition, Table 7 shows that although YOLO-ADS has a lower number of model parameters than most mainstream object detection algorithms such as YOLOv9s and YOLOv10s, since metal surface defect detection tasks are mostly performed in resource-constrained scenarios such as industrial production, the YOLO-ADS model still needs to be compressed to achieve deployment on edge terminal devices with lower arithmetic capabilities, thereby reducing the investment cost of enterprises in computing power equipment. In future work, the optimization can be further improved by using lightweight modules, model pruning, and knowledge distillation.

6. Conclusions

The YOLO-ADS algorithm proposed in this study has shown excellent performance and wide applicability in metal surface defect detection tasks. By focusing on improving the underlying backbone network of the baseline model YOLOv8n, the introduction of ASPPFCSPC, C2f_SimDCNv2 and SPD modules enhances the model's multi-feature fusion and representation, multi-scale feature extraction and small target feature learning capabilities for fine-grained defects in complex backgrounds, respectively. The ablation experiments confirmed the effectiveness of these improvements, reflected in a significant increase of 3.5% in mAP and 6.1% in Precision for the improved model. And through visual qualitative experiments, we can intuitively see that the improved model in the actual detection, confidence and missed detection also show significant improvement.

In addition, not only on the steel strip dataset NEU-DET, but also on the industrial steel dataset GC10-DET, industrial aluminum dataset APSPC and even the larger public benchmark dataset VOC2012, the improved model shows excellent versatility and robustness. This is demonstrated by substantial increases in mAP performance, with improvements of 3.7%, 3.4%, and 4.3%, respectively. At the same time, in the comparison experiments with the current mainstream algorithms, the detection accuracy, the number of model storage parameters and the detection speed also show a good level.

In summary, the YOLO-ADS algorithm proposed in this study has achieved significant performance improvement in metal surface defect detection tasks. It not only possesses excellent accuracy and versatility, but also meets the practical requirements for parameter volume in the deployment of deep learning edge devices. Moreover, its detection speed reaches up to 140.4 FPS, which is suitable for real-time industrial scenarios with limited resources. Thus, it provides an efficient and feasible solution for the field of metal surface defect detection.

For YOLO-ADS, fine-grained defects still present a challenge with lower detection accuracy compared to other categories. Additionally, there is still potential for reducing the number of model parameters. Moving forward, our advancements will focus on the following pivotal aspects:

1. Explore diffusion models to generate natural and realistic metal surface defect training samples, thereby improving the detection accuracy and generalization ability of the model to all categories of defects.
2. Investigate incremental learning methods to enable the model to automatically continue learning from new data samples without having to retrain the entire model.
3. Combine the knowledge distillation with model pruning techniques to avoid the loss of accuracy after further lightweight of the model. Moreover, deploy the model locally to deep learning edge devices such as Jetson Nano for practical application in industrial scenarios with limited computing resources.

Author Contributions: Conceptualization, J.G. and Z.G.; methodology, Z.G.; software, Z.G.; validation, Z.G.; formal analysis, Z.G.; investigation, Z.G.; writing—original draft preparation, Z.G.; writing—review and editing, J.G. and Z.G.; visualization, Z.G.; supervision, J.G.; funding acquisition, Z.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Innovation Project of GUET Graduate Education, grant number 2024YCXS122.

Data Availability Statement: Data are contained within the article.

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

Abbreviations

The following abbreviations are used in this manuscript:

ASPPFCSPC	CSPNet with Average SPP-Fast Block
SPD	Space-to-Depth
mAP	mean Average Precision
FPS	Frames Per Second
MSFE	Multiscale Feature Extraction
EFF	Efficient Feature Fusion
ECA	Efficient Channel Attention
SPPCSPC	CSPNet with SPP Block
Channel Attention	Channel Attention
EMA	Efficient Multi-scale Attention
SOTA	State-Of-The-Art
SPPF	Spatial Pyramid Pooling-Fast
LSKA	Large Separable Kernel Attention
CBAM	Convolutional Block Attention Module
DWR	Dilation-wise Residual
BiFPN	Bidirectional Feature Pyramid Network
DCNv2	Deformable Convnets v2
ELAN	Efficient Layer Aggregation Network
PAN	Path Aggregation Network
FPN	Feature Pyramid Network
CIOU	Complete Intersection over Union
ReLU	Rectified Linear Unit
SPP	Spatial Pyramid Pooling
SGD	Stochastic Gradient Descent
NEU	Northeastern University
P	Precision
R	Recall
TP	True Positive
FP	False Positive
FN	False Negative
AP	Average Precision
P-R	Precision-Recall
Param	Parameter
CSPNet	Cross Stage Partial Network
GAN	Generative Adversarial Network

References

1. Dong, H.; Liu, Y.; Wang, L.; Li, X.; Tian, Z.; Huang, Y.; McDonald, C. Roadmap of China steel industry in the past 70 years. *Ironmak. Steelmak.* **2019**, *46*, 922–927. [\[CrossRef\]](#)
2. Wang, X.; Wang, Z.; Guo, C.; Han, Y.; Zhao, J.; Lu, N.; Tang, H. Application and prospect of new steel corrugated plate technology in infrastructure fields. *IOP Conf. Ser. Mater. Sci. Eng.* **2020**, *741*, 012099. [\[CrossRef\]](#)
3. Yu, J.; Cheng, X.; Li, Q. Surface defect detection of steel strips based on anchor-free network with channel attention and bidirectional feature fusion. *IEEE Trans. Instrum. Meas.* **2021**, *71*, 5000710. [\[CrossRef\]](#)
4. Li, Z.F.; Cao, G.M.; Wang, F.X. Review of characterization techniques on oxide scale microstructure of hot rolled steels. *Steel Roll.* **2017**, *34*, 56–60.
5. Guo, L.; Zhang, L.; Andersson, J.; Ojo, O. Additive manufacturing of 18% nickel maraging steels: Defect, structure and mechanical properties: A review. *J. Mater. Sci. Technol.* **2022**, *120*, 227–252. [\[CrossRef\]](#)
6. Luo, Q.; Li, B.; Su, J.; Yang, C.; Gui, W.; Silvén, O.; Liu, L. CDDNet: Camouflaged Defect Detection Network for Steel Surface. *IEEE Trans. Instrum. Meas.* **2023**, *73*, 5000313. [\[CrossRef\]](#)
7. Liu, L.; Ouyang, W.; Wang, X.; Fieguth, P.; Chen, J.; Liu, X.; Pietikäinen, M. Deep learning for generic object detection: A survey. *Int. J. Comput. Vis.* **2020**, *128*, 261–318. [\[CrossRef\]](#)

8. Gao, Y.; Li, X.; Wang, X.V.; Wang, L.; Gao, L. A review on recent advances in vision-based defect recognition towards industrial intelligence. *J. Manuf. Syst.* **2022**, *62*, 753–766. [\[CrossRef\]](#)
9. Murthy, C.B.; Hashmi, M.F.; Bokde, N.D.; Geem, Z.W. Investigations of object detection in images/videos using various deep learning techniques and embedded platforms—A comprehensive review. *Appl. Sci.* **2020**, *10*, 3280. [\[CrossRef\]](#)
10. Usamentiaga, R.; Lema, D.G.; Pedrayes, O.D.; Garcia, D.F. Automated surface defect detection in metals: A comparative review of object detection and semantic segmentation using deep learning. *IEEE Trans. Ind. Appl.* **2022**, *58*, 4203–4213. [\[CrossRef\]](#)
11. 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. [\[CrossRef\]](#) [\[PubMed\]](#)
12. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.-Y.; Berg, A.C. Ssd: Single shot multibox detector. In *Computer Vision—ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*; Springer International Publishing: Cham, Switzerland, 2016; pp. 21–37.
13. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016*; pp. 779–788.
14. 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.
15. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
16. Bochkovskiy, A.; Wang, C.-Y.; Liao, H.-Y.M. Yolov4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934.
17. Jocher, G.; Chaurasia, A.; Stoken, A.; Borovec, J.; Kwon, Y.; Michael, K.; Fang, J.; Yifu, Z.; Wong, C.; Montes, D.; et al. YOLOv5 by Ultralytics, Version 7.0. 2020. Available online: <https://github.com/ultralytics/yolov5> (accessed on 18 April 2024).
18. Ge, Z.; Liu, S.; Wang, F.; Li, Z.; Sun, J. Yolox: Exceeding yolo series in 2021. *arXiv* **2021**, arXiv:2107.08430.
19. 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.
20. 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*; IEEE: Piscataway, NJ, USA, 2023; pp. 7464–7475.
21. Jocher, G.; Chaurasia, A.; Qiu, J. Ultralytics YOLO, Version 8.0.0. 2023. Available online: <https://github.com/ultralytics/ultralytics> (accessed on 5 April 2024).
22. Wang, C.Y.; Yeh, I.H.; Liao HY, M. Yolov9: Learning what you want to learn using programmable gradient information. *arXiv* **2024**, arXiv:2402.13616.
23. Wang, A.; Chen, H.; Liu, L.; Chen, K.; Lin, Z.; Han, J.; Ding, G. Yolov10: Real-time end-to-end object detection. *arXiv* **2024**, arXiv:2405.14458.
24. Zhao, Y.; Lv, W.; Xu, S.; Wei, J.; Wang, G.; Dang, Q.; Liu, Y.; Chen, J. Detrs beat yolos on real-time object detection. *arXiv* **2023**, arXiv:2304.08069.
25. Du, L.; Zhang, R.; Wang, X. Overview of two-stage object detection algorithms. *J. Phys. Conf. Ser.* **2020**, *1544*, 012033. [\[CrossRef\]](#)
26. Sultana, F.; Sufian, A.; Dutta, P. A review of object detection models based on convolutional neural network. In *Intelligent Computing: Image Processing Based Applications*; Springer: Singapore, 2020; pp. 1–16.
27. Liu, X.; Gao, J. Surface defect detection method of hot rolling strip based on improved SSD model. In *Database Systems for Advanced Applications, Proceedings of the DASFAA 2021 International Workshops: BDQM, GDMA, MLDLDSA, MobiSocial, and MUST, Taipei, Taiwan, April 11–14, 2021, Proceedings 26*; Springer International Publishing: Cham, Switzerland, 2021; pp. 209–222.
28. Li, Z.; Wei, X.; Hassaballah, M.; Li, Y.; Jiang, X. A deep learning model for steel surface defect detection. *Complex Intell. Syst.* **2024**, *10*, 885–897. [\[CrossRef\]](#)
29. Liang, J.; Kong, R.; Ma, R.; Zhang, J.; Bian, X. Aluminum Surface Defect Detection Algorithm Based on Improved YOLOv5. *Adv. Theory Simul.* **2024**, *7*, 2300695. [\[CrossRef\]](#)
30. Gao, S.; Chu, M.; Zhang, L. A detection network for small defects of steel surface based on YOLOv7. *Digit. Signal Process.* **2024**, *149*, 104484. [\[CrossRef\]](#)
31. Yuan, Z.; Ning, H.; Tang, X.; Yang, Z. GDCP-YOLO: Enhancing Steel Surface Defect Detection Using Lightweight Machine Learning Approach. *Electronics* **2024**, *13*, 1388. [\[CrossRef\]](#)
32. Huang, Y.; Tan, W.; Li, L.; Wu, L. WFRE-YOLOv8s: A New Type of Defect Detector for Steel Surfaces. *Coatings* **2023**, *13*, 2011. [\[CrossRef\]](#)
33. Dong, H.; Song, K.; He, Y.; Xu, J.; Yan, Y.; Meng, Q. PGA-Net: Pyramid feature fusion and global context attention network for automated surface defect detection. *IEEE Trans. Ind. Inform.* **2019**, *16*, 7448–7458. [\[CrossRef\]](#)
34. Xu, K.; Xu, Y.; Zhou, P.; Wang, L. Application of RNAMlet to surface defect identification of steels. *Opt. Lasers Eng.* **2018**, *105*, 110–117. [\[CrossRef\]](#)
35. Zhao, T.; Chen, X.; Yang, L. IPCA-SVM based real-time wrinkling detection approaches for strip steel production process. *Int. J. Wirel. Mob. Comput.* **2019**, *16*, 160–165. [\[CrossRef\]](#)
36. Zhang, J.; Wang, H.; Tian, Y.; Liu, K. An accurate fuzzy measure-based detection method for various types of defects on strip steel surfaces. *Comput. Ind.* **2020**, *122*, 103231. [\[CrossRef\]](#)
37. Ji, Y.; Liu, S.; Zhou, M.; Zhao, Z.; Guo, X.; Qi, L. A machine learning and genetic algorithm-based method for predicting width deviation of hot-rolled strip in steel production systems. *Inf. Sci.* **2022**, *589*, 360–375. [\[CrossRef\]](#)

38. Zhang, X.; Wang, Y.; Fang, H. Steel surface defect detection algorithm based on ESI-YOLOv8. *Mater. Res. Express* **2024**, *11*, 056509. [[CrossRef](#)]
39. Kong, H.; You, C. Improved steel surface defect detection algorithm based on YOLOv8. *IEEE Access* **2024**, *12*, 99570–99577.
40. Zhang, T.; Pan, P.; Zhang, J.; Zhang, X. Steel Surface Defect Detection Algorithm Based on Improved YOLOv8n. *Appl. Sci.* **2024**, *14*, 5325. [[CrossRef](#)]
41. Xie, W.; Sun, X.; Ma, W. A light weight multi-scale feature fusion steel surface defect detection model based on YOLOv8. *Meas. Sci. Technol.* **2024**, *35*, 055017. [[CrossRef](#)]
42. Yuan, J.; Wan, F.; Lei, G.; Xu, L. FEB-YOLOv8: A Steel Surface Defect Detection Algorithm Based on Improved YOLOv8s. In Proceedings of the 2023 3rd International Conference on Computer Science, Electronic Information Engineering and Intelligent Control Technology (CEI), Wuhan, China, 15–17 December 2023; IEEE: Piscataway, NJ, USA, 2023; pp. 629–633.
43. Wang, L.; Zhang, G.; Wang, W.; Chen, J.; Jiang, X.; Yuan, H.; Huang, Z. A defect detection method for industrial aluminum sheet surface based on improved YOLOv8 algorithm. *Front. Phys.* **2024**, *12*, 1419998. [[CrossRef](#)]
44. 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.
45. Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature pyramid networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2117–2125.
46. Li, X.; Wang, W.; Wu, L.; Chen, S.; Hu, X.; Li, J.; Tang, J.; Yang, J. Generalized focal loss: Learning qualified and distributed bounding boxes for dense object detection. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 21002–21012.
47. Zhang, H.; Wang, Y.; Dayoub, F.; Sunderhauf, N. Varifocalnet: An iou-aware dense object detector. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 8514–8523.
48. 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](#)] [[PubMed](#)]
49. Dai, J.; Qi, H.; Xiong, Y.; Li, Y.; Zhang, G.; Hu, H.; Wei, Y. Deformable convolutional networks. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 764–773.
50. 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.
51. Sunkara, R.; Luo, T. No more strided convolutions or pooling: A new CNN building block for low-resolution images and small objects. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Proceedings of the European Conference, ECML 2022, Grenoble, France, 19–23 September 2022, Proceedings, Part III*; Springer Nature: Cham, Switzerland, 2023; pp. 443–459.
52. Sajjadi, M.S.; Vemulapalli, R.; Brown, M. Frame-recurrent video super-resolution. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 6626–6634.
53. Su, J.J.; Luo, Q.W. A review of feature representation methods for surface defects of hot-rolled strip steel. *Metall. Ind. Autom.* **2022**, *46*, 1–19. (In Chinese)
54. Bao, Y.; Song, K.; Liu, J.; Wang, Y.; Yan, Y.; Yu, H.; Li, X. Triplet-graph reasoning network for few-shot metal generic surface defect segmentation. *IEEE Trans. Instrum. Meas.* **2021**, *70*, 50111111. [[CrossRef](#)]
55. He, Y.; Song, K.; Meng, Q.; Yan, Y. An end-to-end steel surface defect detection approach via fusing multiple hierarchical features. *IEEE Trans. Instrum. Meas.* **2019**, *69*, 1493–1504. [[CrossRef](#)]
56. Lv, X.; Duan, F.; Jiang, J.J.; Fu, X.; Gan, L. Deep metallic surface defect detection: The new benchmark and detection network. *Sensors* **2020**, *20*, 1562. [[CrossRef](#)]
57. Everingham, M.; Eslami, S.A.; Van Gool, L.; Williams, C.K.; Winn, J.; Zisserman, A. The pascal visual object classes challenge: A retrospective. *Int. J. Comput. Vis.* **2015**, *111*, 98–136. [[CrossRef](#)]
58. Wang, X.; Xie, L.; Dong, C.; Shan, Y. Real-esrgan: Training real-world blind super-resolution with pure synthetic data. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 1905–1914.

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.