Research Article

A New Steel Defect Detection Algorithm Based on Deep Learning

Weidong Zhao, Feng Chen ⓘ, Hancheng Huang, Dan Li, and Wei Cheng

College of Electrical and Information Engineering, Anhui University of Technology, Maanshan 243000, China

Correspondence should be addressed to Feng Chen; hellochenf@163.com

In recent years, more and more scholars devoted themselves to the research of the target detection algorithm due to the continuous development of deep learning. Among them, the detection and recognition of small and complex targets are still a problem to be solved. The authors of this article have understood the shortcomings of the deep learning detection algorithm in detecting small and complex defect targets and would like to share a new improved target detection algorithm in steel surface defect detection. The steel surface defects will affect the quality of steel seriously. We find that most of the current detection algorithms for NEU-DET dataset detection accuracy are low, so we choose to verify a steel surface defect detection algorithm based on machine vision on this dataset for the problem of defect detection in steel production. A series of improvement measures are carried out in the traditional Faster R-CNN algorithm, such as reconstructing the network structure of Faster R-CNN. Based on the small features of the target, we train the network with multiscale fusion. For the complex features of the target, we replace part of the conventional convolution network with a deformable convolution network. The experimental results show that the deep learning network model trained by the proposed method has good detection performance, and the mean average precision is 0.752, which is 0.128 higher than the original algorithm. Among them, the average precision of crazing, inclusion, patches, pitted surface, rolled in scale and scratches is 0.501, 0.791, 0.792, 0.874, 0.649, and 0.905, respectively. The detection method is able to identify small target defects on the steel surface effectively, which can provide a reference for the automatic detection of steel defects.

1. Introduction

It will produce a variety of defects in the steel rolling process due to the environment, production process, and other restrictions, which have a certain influence on the wear resistance and toughness of steel. For example, it will produce crack defects during steel heating due to improper processing or the quality of the steel. It may lead to inclusion defects on the steel surface if some nonmetallic inclusions are rolled into the steel surface or the rolling mill environment is not clean. It may produce plaque defects if the steel surface has corrosion, emulsification, and other conditions. Due to the high temperature and long time, it will lead to surface defects pitted surface, that is, coarse grains and other phenomena. There will be an oxide layer and rust on the surface when the steel is exposed to air for a long time, and then the steel will contact with the air at a high temperature to form the rolling scale [1] in the rolling process. The steel surface may show a scratch phenomenon because

the steel oxide scale or other foreign matters contact with the high temperature rolled piece and scratch. The images of several common defects on the steel surface are shown in Figure 1, including crazing, inclusions, patches, and pitted surface, rolled in scale and scratches.

The traditional steel surface defect detection [2–4] is completed by manual visual inspection combined with traditional machine vision [5–7]. There are some shortcomings in manual testing, such as low confidence and high labor intensity. The traditional target detection [8] selects candidate regions on a given image; then, the features are extracted manually and the trained classifier is used for classification. This method has high time complexity and low precision and is difficult to meet the actual production needs of the steel industry. With the continuous development of the convolutional neural network [9], target detection based on deep learning has become the mainstream surface defect detection method. Qu et al. [10] designed an improved Gabor filter
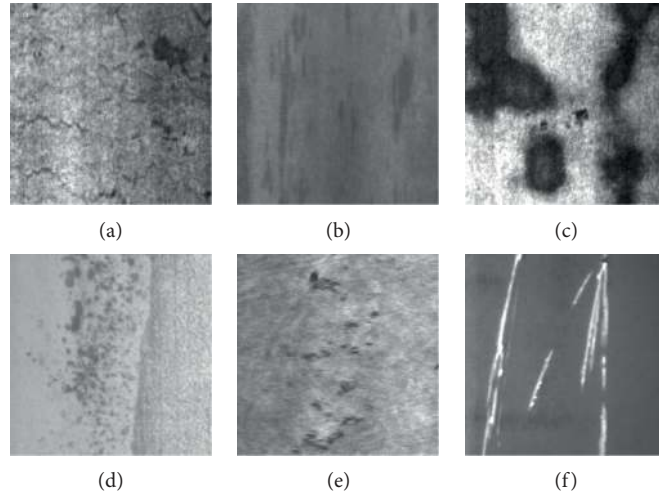
FIGURE 1: Steel surface defect image. (a) Crazing, (b) inclusions, (c) patches, (d) pitted surface, (e) rolled in scale, and (f) scratches.

optimization method to complete the defect detection of strip steel surface by comparing the shortcomings of the traditional gray projection model and HTM model. Xu et al. [11] proposed a rapid detection method for visual, significant defects based on spectral residual, which realized the accurate detection and classification of various defects on the strip surface. Versaci et al. [12] was committed to the development of ultrasonic nondestructive testing and classification technology based on the continuous wave and proposed a fuzzy similarity based method. The above detection and classification methods for steel surface defects are mainly to extract the shallow features of the target, so the adaptability of this method in the practical application of steel surface defect detection is poor because the defects on the steel surface are complex and uncertain. In contrast, deep learning has multilayer perceptron with multiple hidden layers, which can form more abstract category features by combining low-level features. Therefore, the steel surface defect detection method based on deep learning is widely used in steel production, and more scholars begin to improve and perfect it [13, 14]. Fu et al. [15] proposed a convolutional neural network model which emphasizes the training of the underlying features and realized the rapid and accurate classification of steel surface defects by combining with multiple receptive fields. Lv et al. [16] proposed a new end-to-end defect detection network (EDDN) to deal with different types of steel surface defects. Marco et al. [17] discussed the improved methods of steel surface defect detection and classification by comparing the traditional machine learning model and deep learning model in steel defect classification. Song et al. [18] proposed a new saliency detection method based on an encoder-decoder residual network (EDR-Net). In the coding stage, the full-convolution neural network is used to extract defect features, and an attention mechanism is integrated to accelerate the convergence of the model. As the mainstream target detection framework, Faster R-CNN is used in surface defect detection of steel and

metal widely. Wang et al. [19] proposed a Faster R-CNN algorithm integrating multilevel features, which solved the problem of detection of diverse and random defects on the surface of metal plate and strip. Dai et al. [20] designed a defect detection algorithm based on improved Faster R-CNN in order to solve the problems of limitations and low precision of workpiece surface defect detection, which improves the detection performance of defects compared with traditional methods. The research of automatic detection of steel surface defects is focused on in this paper. At the first, the ResNet-50 [21–24] network is reconstructed by deformable convolution as the prefeature extraction network of Faster R-CNN [25–27]. Then, the fixed convolution layer and pooling layer are replaced with deformable convolution layer and deformable pooling layer. Finally, the FPN [28–30] network is used for multifeature fusion, and the soft non-maximum suppression (soft NMS) [31] is used to reduce the confidence of the detection frame larger than the threshold, so as to alleviate the situation of the target missing detection. According to the test results on the open dataset NEU-DET, the proposed algorithm can effectively detect a variety of defects on steel surface, which is higher than the ordinary steel surface detection algorithms in accuracy.

2. Steel Defect Detection Method

2.1. Faster R-CNN Model. Faster R-CNN is the mainstream two-stage target detection algorithm, which is used in face detection and defect detection widely. The detection accuracy of Faster R-CNN is higher than SSD [32, 33], Fast R-CNN [34], and other algorithms verified on multiple datasets. Faster R-CNN algorithm structure includes feature extraction network, region proposal network (RPN), and regional convolution neural network mainly. The algorithm structure is shown in Figure 2. Firstly, the features of the input image are extracted by the feature extraction network; then the extracted features are shared to the RPN network
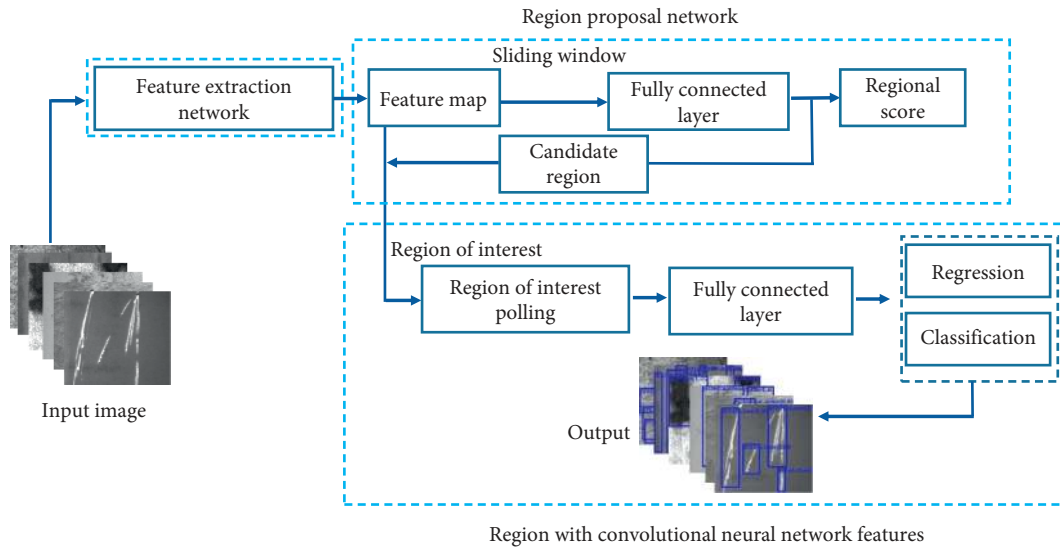
Figure 2: The structure diagram of Faster R-CNN.

and R-CNN network. Next, the region of interest (ROI) of the image is extracted. Finally, the detection results are output through ROI pooling and fully connected layer.

2.2. Steel Defect Detection Algorithm Based on Improved Faster R-CNN. In this paper, a steel defect detection algorithm based on the deformable network [35] and multiscale feature fusion is proposed. Faster R-CNN is used as the basic framework, which is composed of feature extraction network, regional recommendation network, and detection network. The improved method is shown in Figure 3, in which ResNet-50 is used as a feature extraction network. Firstly, the deformable convolution is used to reconstruct the ResNet-50 network. Secondly, the feature pyramid network FPN is used to fuse the multiscale features, and the fixed region of interest (ROI) pooling layer is replaced by the variable pooling layer. Finally, the soft nonmaximum value suppression algorithm is used to suppress the detection frame with obvious overlap with the highest score detection frame.

The conventional convolution network or fully connected network will lead to the loss of partial information in information transmission, which can not train a deep network. However, the ResNet transmits the input information to the output directly, which can protect the integrity of information and reduce the learning difficulty. Therefore, ResNet-50 is selected as the feature extraction network in this paper. A bottleneck module is used in ResNet-50, as shown in Figure 4. It reduces the amount of computation and speeds up the operation by 1×1 convolution. Firstly, a 1×1 convolution is used to reduce the input dimension characteristic matrix channel from 256 to 64. In the middle, a 3×3 convolution is used to deepen the network. Finally, the dimension is restored by the 1×1 convolution. Aiming at the small and complex defects on the steel surface, convolution layers of stages 2 and stage 4 (as shown in Figure 5) are reconstructed by deformable convolution to improve the ability to extract features.

2.3. Deformable Network. The block convolution kernels of fixed size are often used in the conventional convolution neural network while building model transformation, which is usually limited to fixed geometry structure. The convolution unit is weak in feature extraction of convolution layer with fixed sampling points, which makes the network difficult to adapt to geometric transformation. To solve the above problems, Dai et al. [36] introduced deformable convolution and deformable pooling into convolution neural network to enhance the modeling ability of the network. The end-to-end training is carried out through standard backpropagation to generate a deformable convolution network. There is no fixed geometric structure due to the different shapes of steel surface defects. Therefore, the idea of deformable convolution is introduced to reconstruct ResNet-50 for the poor adaptability to the ability of unknown changes and weak generalization, so as to improve the recognition ability of the neural network for irregular targets.

The offset variable is added to each element of the convolution kernel in deformable convolution, which is calculated by standard convolution unit. Therefore, the range in the training process can be expanded by the convolution kernel. In addition, the size and position can be adjusted dynamically to adapt to the geometric deformation of different objects in accordance with the information of the image recognized. A convolution layer is added to the input feature map extracted by the conventional convolution kernel to obtain the deformation offset of deformable convolution. The convolution kernel is used to generate feature map and offset to realize synchronous learning. The various forms of the conventional convolution kernel and the deformable convolution kernel are shown in Figure 6, where Figure 6(a) represents the conventional convolution kernel sampling point, Figure 6(b) represents the deformable convolution kernel sampling point after adding offset variables,
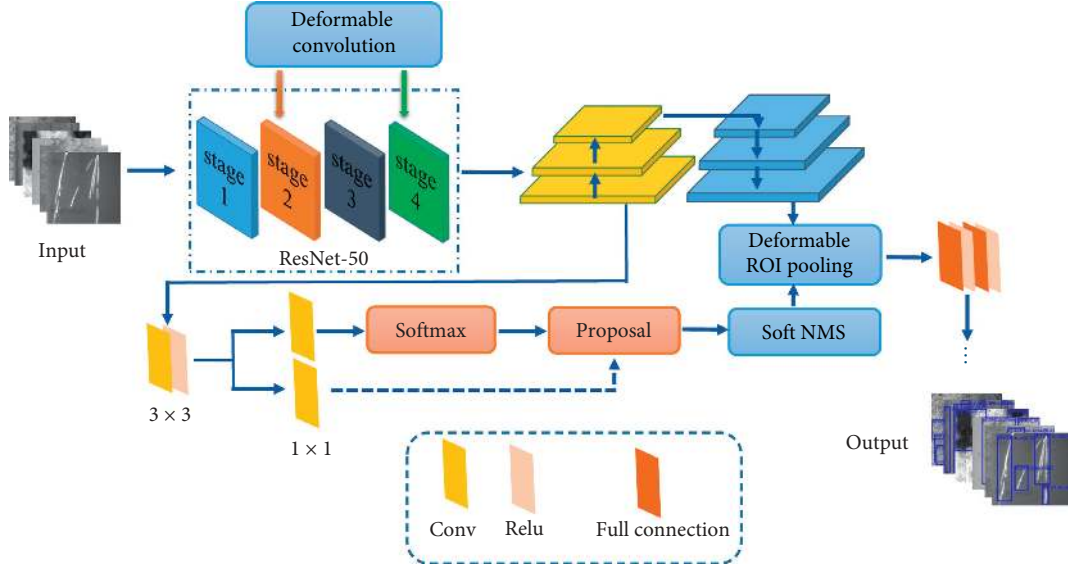
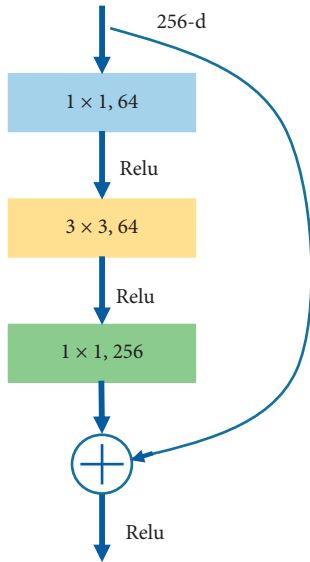FIGURE 3: The improved method of Faster R-CNN.

FIGURE 4: The bottleneck structure of ResNet-50 network.
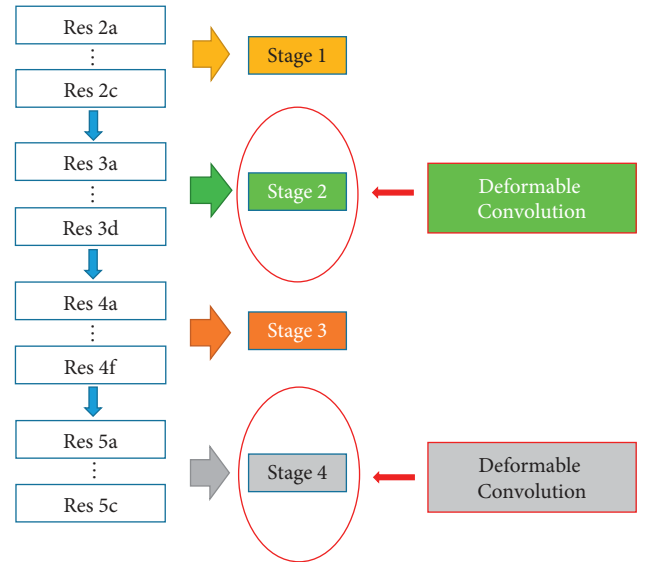

FIGURE 5: Reconstruction of ResNet-50 using variable convolution.

Figures 6(c) and 6(d) are special cases of deformable convolution kernel sampling.

In the conventional convolution kernel, the convolution of each pixel in the input image at position of p_0 is expressed as follows:

$$y(p_0) = \sum_{p_n \in R} w(p_n) x(p_0 + p_n).$$ (1)

An offset variable $\{\Delta p_n \mid n = 1, 2, \ldots, N\}$ is introduced into the deformable convolution; the deformation convolution of each pixel in the input image is expressed as follows:

$$y(p_0) = \sum_{p_n \in R} w(p_n) x(p_0 + p_n + \Delta p_n).$$ (2)

In formulas (1) and (2) [36], y is the output feature map, x is the input feature map, p_0 is the position of pixel, w is the parameter of weight, p_n is any pixel in convolution, and Δp_n is the offset value. The bias domain of deformable convolution points to the sampling point with a strong purpose and outputs more feature information because the bias matrix makes the sampling position of convolution transform freely, so when the effect of deformable convolution is stacked, the feature extraction ability is greatly improved. Taking a 3×3 convolution as an example, the author uses Figure 7 to show the difference between conventional convolution and deformable convolution. The sampling position of the conventional convolution on the target is fixed, as shown in Figure 7(a), while the receptive field can be learned adaptively by the deformable convolution during calculation in Figure 7(b), which has a strong adaptive
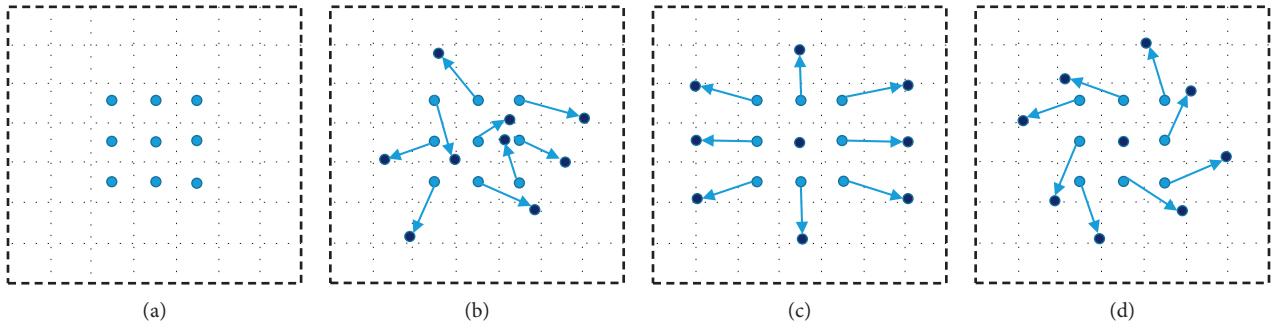
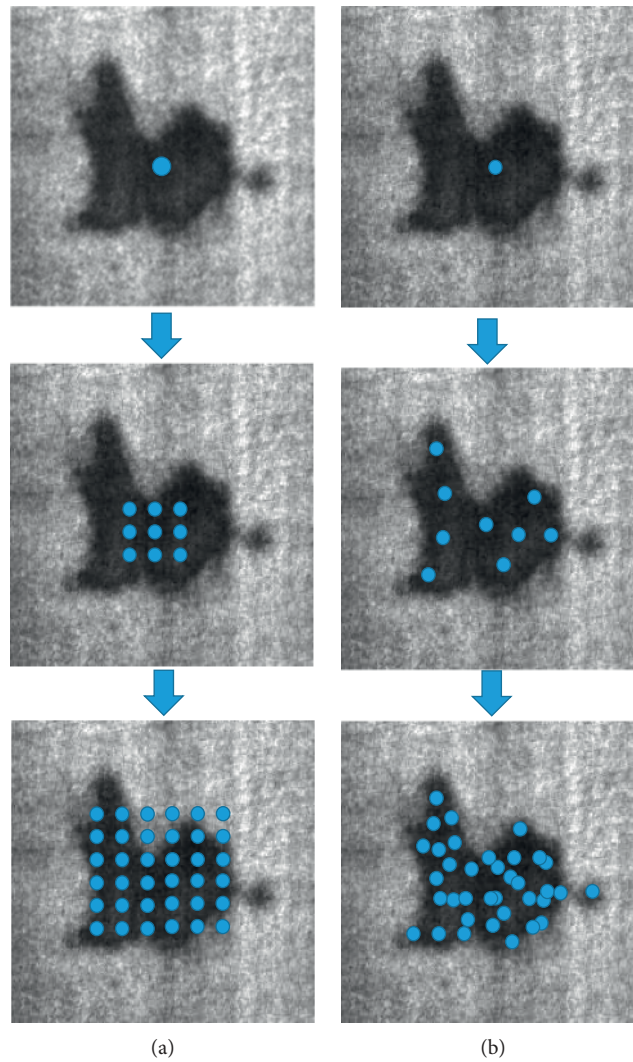FIGURE 6: Four sampling methods of convolution.

FIGURE 7: The calculation process of conventional and deformable convolution.

extraction ability for complex and irregular targets. It can be adjusted adaptively under the shape and size of the target. Therefore, this adaptive learning receptive field is very necessary in steel defect detection.

ROI pooling module is introduced to maximize the pool of the proposed regions after segmentation in order to map the input regions of different sizes into feature vectors of the same length and output a fixed size feature map. The fixed ROI pooling module is replaced with the deformable ROI pooling module in the proposed method in order to enhance the modeling ability of the network for geometric transformation and increase the pool area of defects to obtain the

location accurately of complex defects on the steel surface. The network structure of deformable ROI pooling is shown in Figure 8.

The regions of interest are divided into blocks of $k \times k$ by ROI pooling, and the size of each area is $w \times h$. The output of normal ROI pooling [36] is as follows:

$$y(i, j) = \sum_{p \in \text{bin}(i,j)} \frac{x(p_0 + p)}{n_{i,j}}. \tag{3}$$

As can be seen from Figure 7, in the deformable ROI pooling, ROI pooling generates convolution feature graphs and then generates regularized offsets with the size of $2 \times k \cdot k$ at each position through the full-connection layer. Finally, the region with enhanced offset is pooled to generate the output feature map. Referring to the idea of deformable convolution, the expression of deformable ROI pooling [36] is as follows:

$$y(i, j) = \sum_{p \in \text{bin}(i,j)} \frac{x(p_0 + p + \Delta p_{i,j})}{n_{i,j}}, \tag{4}$$

where $y(i, j)$ is the output characteristic graph after pooling, x is the input feature map, p_0 is the upper left corner pixel of ROI, p is a pixel at any position, $\text{bin}(i, j)$ is the set of horizontal and vertical coordinates of the pixel, $n_{i,j}$ is the pixel value of the grid, and $\Delta p_{i,j}$ is the offset at each location, where $0 \le i, \ j < k$.

2.4. Multiscale Feature Fusion.

The features of the last layer are used in RPN because the top-level features of the network have the strongest semantic information in Faster R-CNN, but this idea is not suitable for small target detection. The feature pyramid network FPN is introduced into Faster R-CNN to perform degree scale fusion operation on the feature map in order to improve the ability of the network to detect small area defects on the steel surface. The structure is shown in Figure 9.

The characteristics of all layers are integrated into FPN and the feature activated output of the last residual structure of conv2_x, conv3_x, conv4_x, and conv5_x phases in ResNet-50 network is used as input. The channel number of C2-C5 is reduced to 256 by the 1×1 convolution, and M2-M5 are obtained. The output P2-P5 of FPN is obtained by adding the same size of shallow feature map and deep feature map by upsampling, and then the 3×3 convolution is used. The P6 is obtained by downsampling with the two largest pools on P5, and the multiscale fusion feature combination is output. The target candidate frame of interest is generated by using the fused features in RPN, and the detection results are obtained by classification.

2.5. Soft Nonmaximum Suppression.

Nonmaximum suppression (NMS) is used as a postprocessing method commonly to eliminate duplicate frames and reduce the false detection rate in the target detection. However, other similar targets attached to the box with higher prediction score are removed by the NMS, which will lead to missed detection and false detection of similar targets with close proximity and overlapping. The reduction range of the prediction accuracy of dense targets nearby is determined by soft NMS in light of the intersection over union (IOU) [36] of the highest score box, so as to improve the prediction accuracy of dense targets. For example, the IOU of the detection frames A and B [35] is the following:

$$\text{IOU} = \frac{A \cap B}{A \cup B} = \frac{S_U}{S_A + S_B - S_U}, \tag{5}$$

where S_A, S_B are the areas of the detection frame A and B, respectively, and S_U is the overlapping area of the detection frames A and B. The expressions of traditional NMS and Soft NMS algorithms are as follows [35]:

$$S_i = \begin{cases} S_i, \ \text{IOU}(M, b_i) < N_t, \\ 0, \ \text{IOU}(M, b_i) \ge N_t, \end{cases} \tag{6}$$

$$S_i = \begin{cases} S_i, \ \text{IOU}(M, b_i) < N_t, \\ S_i(1 - \text{IOU}(M, b_i)), \ \text{IOU}(M, b_i) \ge N_t, \end{cases} \tag{7}$$

where S_i is the score of the detection box i, M is the maximum score box, b_i is the collection of detection frames i, N_t is the threshold set. The NMS sets the score of the detection frame whose IOU is greater than the threshold value to 0, while the soft NMS attenuates the score of the detection frame whose IOU is greater than the threshold value, which can alleviate the problems of missed detection and false detection of targets. The IOU threshold of soft NMS is 0.5 and the minimum score is 0.05.

2.6. Experimental Results and Analysis

2.6.1. Experimental Environment and Parameter Setting.

The experimental platform is shown in Table 1 and the initialization model parameters of the experiment are shown in Table 2.

The average precision (AP) is used as the evaluation index of each defect detection in this experiment after the model training is completed, and the mean average precision (mAP) is used as the evaluation index of the whole model performance. The definitions of precision (P), AP, and mAP [37] are shown as follows:

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

$$\text{AP} = \frac{1}{11} \sum_{r\{0, 0.1, 0.2, \cdots, 1\}} \rho_{\text{interp}(r)}, \tag{9}$$

$$\text{mAP} = \frac{1}{n_j \sum_{j=1}^{n_j} \text{AP}_j}, \tag{10}$$

where TP is the number of samples with correct detection, FP is the number of negative samples with detection, and r is the value of recall (R). The definition of recall rate is shown in formula (10), $\rho_{\text{interp}(r)}$ can be expressed as formula (11) [37],
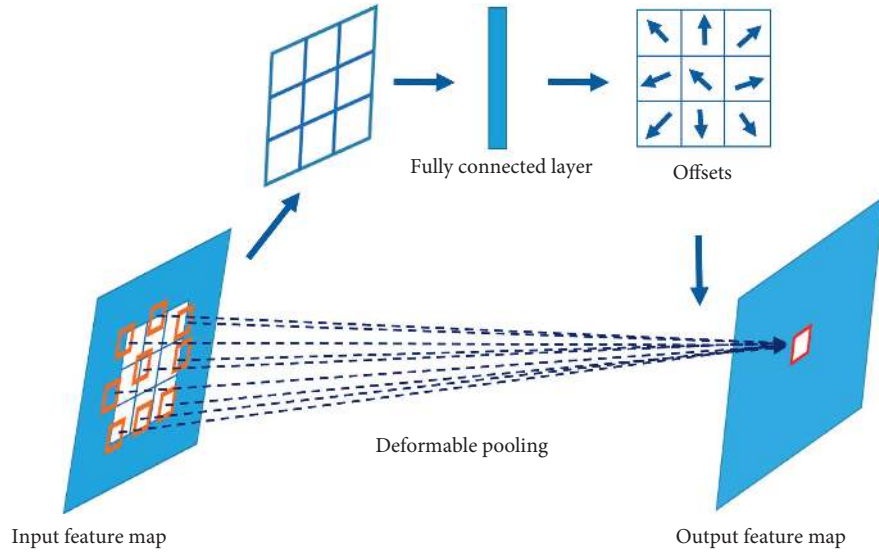
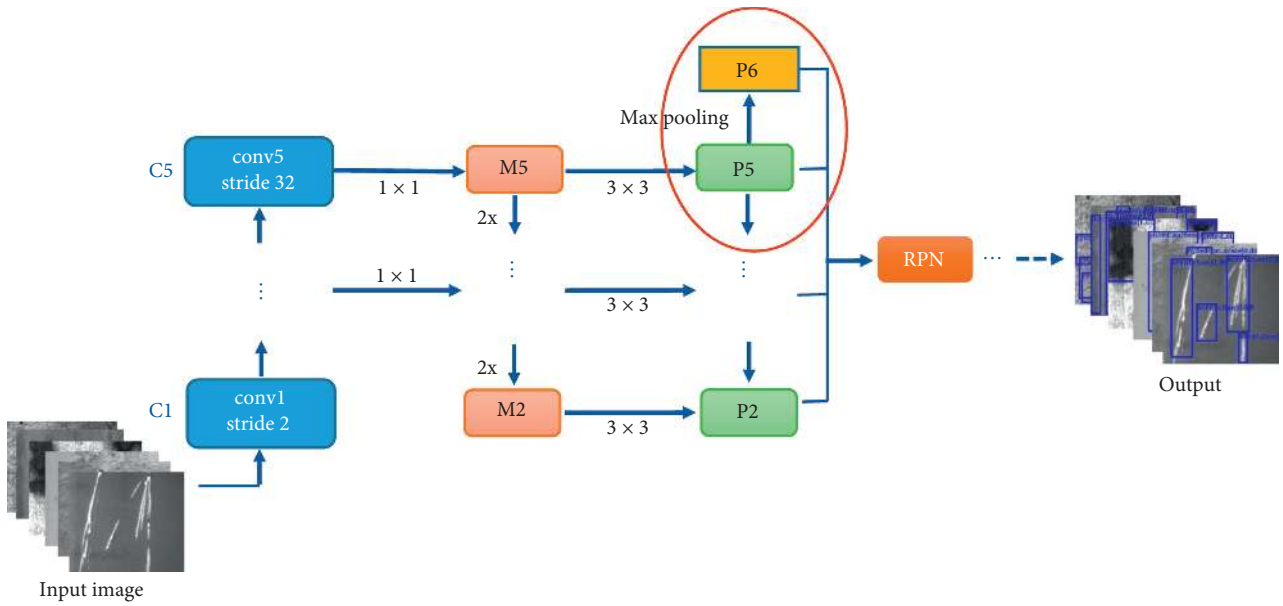FIGURE 8: The network structure of deformable pooling.

FIGURE 9: The network structure diagram of multiscale feature fusion.

j is the category index, n_j is the total number of categories, and AP_j is the average precision of each category:

$$R = \frac{TP}{TP + FN},\tag{11}$$

$$\rho_{\mathrm{interp}(r)} = \frac{\max \rho(\tilde{r})}{\tilde{r}: \tilde{r} \geq r},\tag{12}$$

where FN is the number of samples with error detection and $\rho(\tilde{r})$ is the recall rate of $\tilde{r}$.

The results and discussion may be presented separately, or in one combined section, and may optionally be divided into headed subsections.

2.7. Analysis of Experimental Results. The proposed algorithm is verified on the open dataset of steel defects, and the experimental results are shown in Table 3. The test dataset is NEU-DET, which is the open dataset of Northeast University. There are 1800 pictures in total. Among them, there are 300 pictures of six kinds of defects. The six defects are marked as "crazing," "inclusion," "patches," "pitted_ surface," "rolled-in_scale," and "scratches." The test results are shown in Figure 10, and four different pictures are listed for each defect result.

It is not hard to see that the average accuracy of "scratch" is the highest and the "pitted_ surface" is the second in Table 3, which can reach 90.5% and 87.4%, respectively. The

TABLE 1: Hardware and software parameters of the experimental platform.

Name	Parameter
CPU	Intel core i7-6800 @ 3.4 GHz
GPU	NVIDIA GTX 1080Ti
Operating system	Linux Ubuntu 16.04
Deep learning framework	PyTorch 1.3.1
Language	Python 3.7.5

TABLE 2: Setting of the training parameter value.

Parameters	Learning rate	Weight decay	Momentum	Epochs	Pictures/GPU	Threads/GPU
Settings	0.02	0.0001	0.9	20	2	2

TABLE 3: The experimental data of the algorithm on all defects.

Defects	Crazing	Inclusion	Patches	Pitted_surface	Rolled-in_scale	Scratches
AP	0.501	0.791	0.792	0.874	0.649	0.905
mAP	**0.752**					

average accuracy of "crashing" is the lowest, which is 50.1%, and the overall mean average accuracy is 75.2%. As shown in Figure 10, the defect of "inclusion," "patches," "pitted_surface," "rolled-in_scale," and "scratches" can be detected and located accurately. The proposed algorithm still has the possibility of optimizing the detection of small targets due to the fuzzy and small size of crazing defects and a small number of missed detection. It can be seen from Figures 10(a)–10(c) that the defect target can also be detected in the case of light or dark, which indicates that the algorithm proposed in this paper is also suitable for defect detection on steel surface under complex background. Generally speaking, the proposed algorithm has high test effectiveness on the NEU-DET dataset.

2.8. Ablation Experiments.

The proposed scheme is compared with different improved schemes on the same dataset in order to verify the effectiveness of it. The specific improvement schemes are shown in Table 4.

The five specific schemes in Table 4 are as follows:

(1) Scheme 1 is the original Faster R-CNN model.

(2) Scheme 2 uses ResNet-50 as the feature extraction network of Faster R-CNN.

(3) Scheme 3 introduces a deformable network on the basis of ResNet-50 as the feature extraction network of Faster R-CNN.

(4) Scheme 4 adds the feature pyramid network on the basis of Scheme 5.

(5) Scheme 5 replaces nonmaximum suppression scheme with a soft nonmaximum on the basis of Scheme 4.

The final test results of the four schemes are shown in Figure 11.

The experimental results show that Schemes 1, 2, and 3 have missed detection for six kinds of defects, and the detection accuracy is low. Scheme 2 has overlapping detection frames for "inclusion" and "pitted_surface" defects, but the score of detection frames on these two defects is higher than that of Scheme 1. Scheme 3 has obvious overlapping detection frames for "patches" and "pitted_surface", but the overall detection effect is better than Schemes 1 and 2. The defects of "inclusion," "roll-in_scale," and "scratches" are all detected in Scheme 4, but the "crashing" is still missed. The NMS algorithm was replaced by soft NMS on the basis of Scheme 4 in Scheme 5 to attenuate the nonmaximum detection frame score, rather than removing it completely by NMS. Therefore, the nonmaximum score detection frame is less suppressed compared with Schemes 1–3, and the detection of crazing defects is more accurate. The specific test results are shown in Table 5.

In Table 5, a is "crazing," b is "inclusion," c is "patches," d is "pitted_surface," e is "rolled-in_scale," f is "scratches." When the original Faster R-CNN model (Scheme 1) is used, the overall mean average accuracy is 62.4%. After replacing the original feature extraction network with ResNet-50, the average overall accuracy of Scheme 2 is 1.9% higher than that of Scheme 1, in which the average accuracy of "crashing," "roll-in_scale," and "scratches" is not improved significantly, but the accuracy of the pitted surface is improved to a certain extent. The mean average accuracy of Scheme 3 is 9% higher than that of the original Faster R-CNN, and 7.1% higher than that of Scheme 2, in which the average accuracy of "crashing" is the highest, which is 20.6%. The mean average accuracy of Scheme 4 is 12.1% higher than that of the original Faster R-CNN model, 3.1% higher than that of Scheme 3, and the average accuracy of "pitted_surface" is increased by 7.2%. The proposed algorithm (Scheme 5) is better than the original Faster R-CNN model and the average accuracy of "crashing" is reduced by 1.5% compared with Scheme 4, but the average accuracy of other defects is improved. The mean average accuracy is increased by 10.9%, 3.8%, and, 0.7%, respectively, compared with Schemes 2–4.
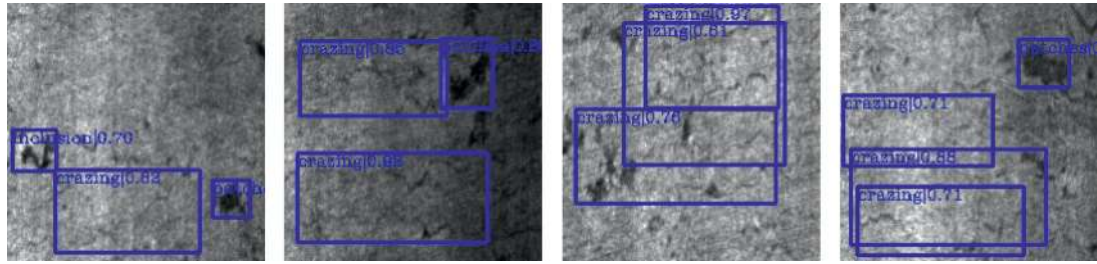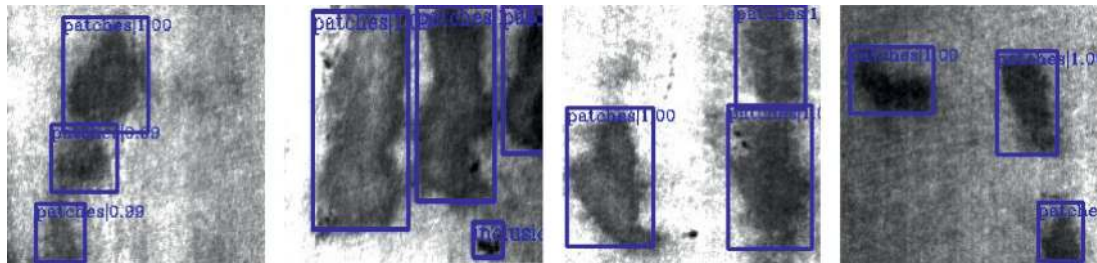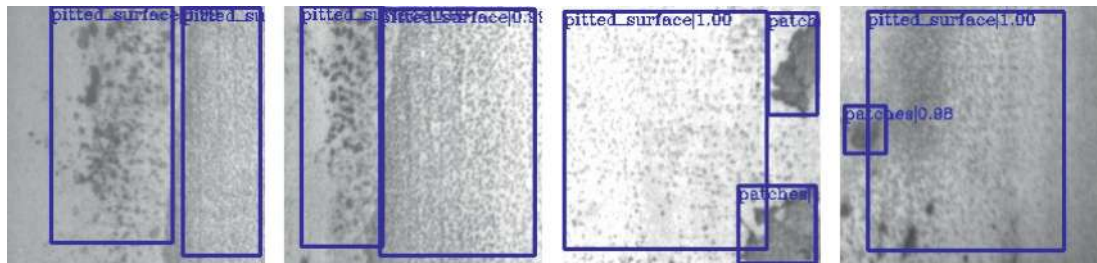
(a)
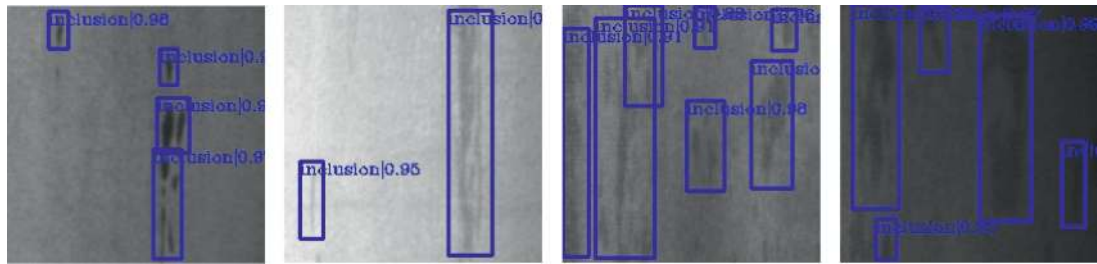
(b)
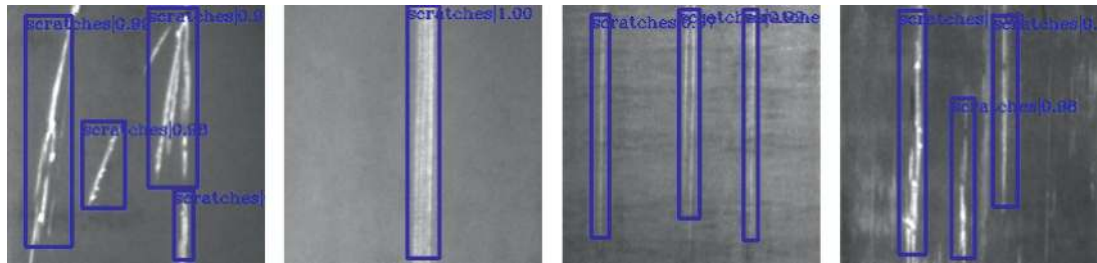
(c)

(d)

(e)

Figure 10: Continued.

(f)
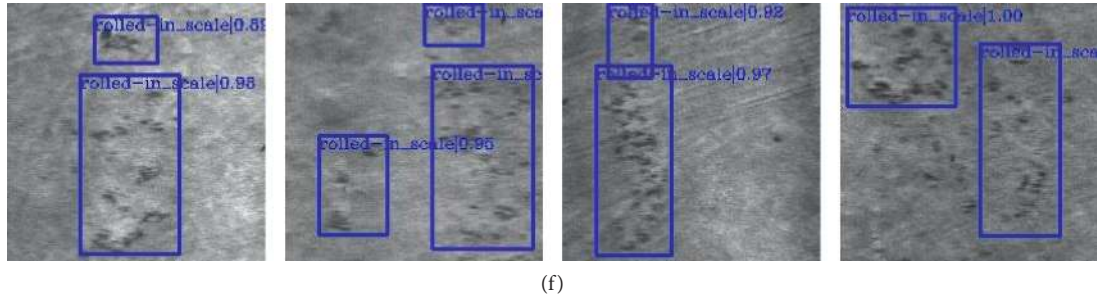
FIGURE 10: The detection results of the proposed algorithm on all defects. (a) Crazing. (b) Inclusion. (c) Patches. (d) Pitted_surface. (e) Rolled-in_scale. (f) Scratches.

TABLE 4: Five schemes based on different improvement methods.

Schemes	ResNet-50	Deformable networks	FPN	Soft NMS
1	×	×	×	×
2	√	×	×	×
3	√	√	×	×
4	√	√	√	×
5	√	√	√	√

FIGURE 11: The results of different schemes.

TABLE 5: The data results comparison of five schemes.

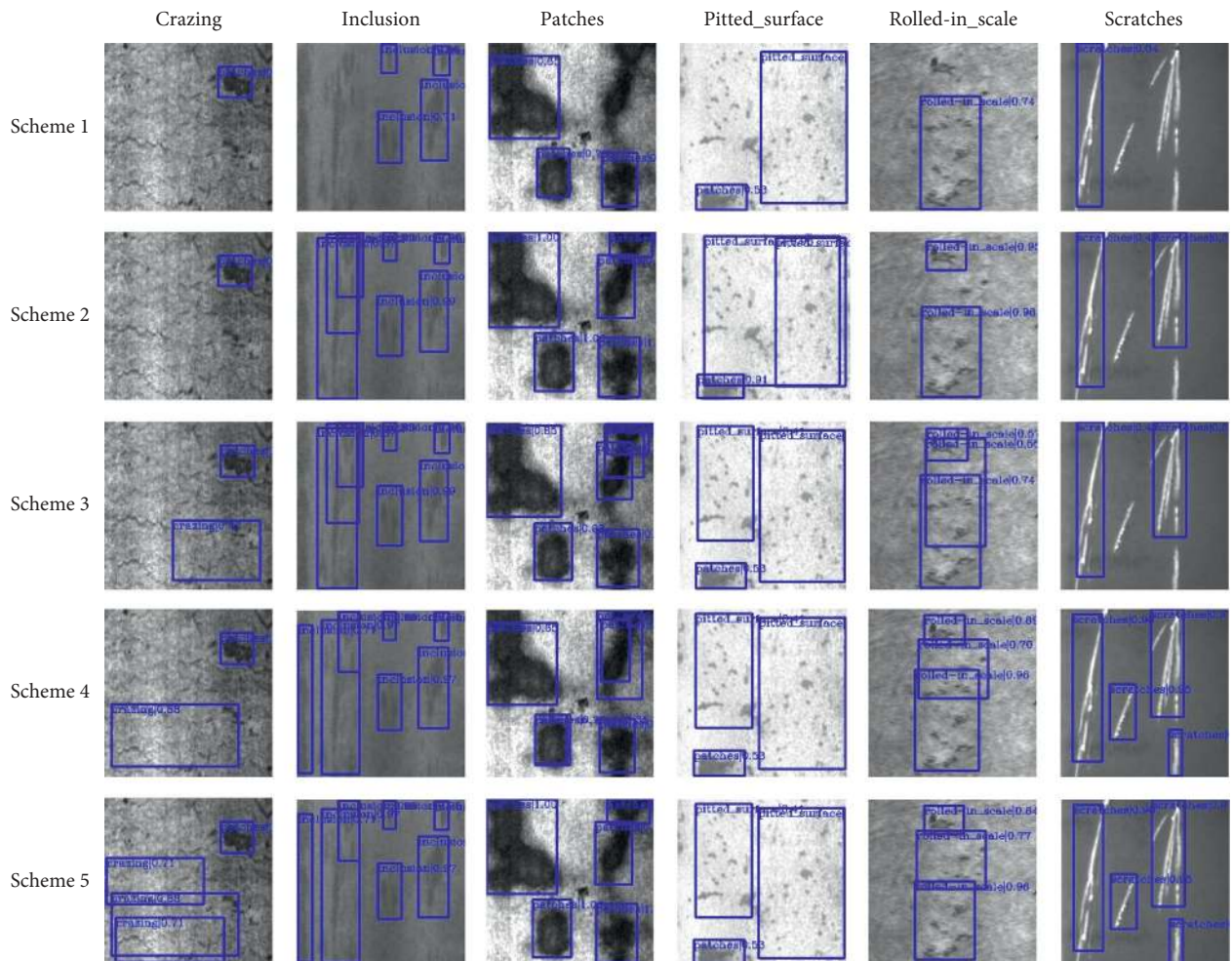| Schemes	AP						mAP	mAP increasing
	a	b	c	d	e	f		
1	0.250	0.652	0.752	0.735	0.545	0.811	0.624	−
2	0.252	0.677	0.772	0.789	0.555	0.816	0.643	+0.019
3	0.458	0.738	0.841	0.782	0.583	0.881	0.714	+0.090
4	0.516	0.754	0.768	0.854	0.610	0.902	0.745	+0.121
5	**0.501**	**0.791**	**0.792**	**0.874**	**0.649**	**0.905**	**0.752**	**+ 0.128**

TABLE 6: The algorithms of SSD, RetinaNet, and Cascade R-CNN are compared with the experimental results of the proposed algorithm.

Algorithms	AP						mAP	mAP increasing
	a	b	c	d	e	f		
SSD	0.302	0.522	0.620	0.390	0.709	0.515	0.510	+0.242
RetinaNet	0.391	0.676	0.772	0.711	0.546	0.542	0.606	+0146
Cascade R-CNN	0.321	0601	0.794	0.723	0.509	0.805	0.626	+0.126
This paper	**0.501**	**0.791**	**0.792**	**0.874**	**0.649**	**0.905**	**0.752**	−

We can see that the introduction of deformable convolution and deformable ROI pooling has a great impact on the network under the data in Table 5, which can improve the network's ability to extract complex features. It can be seen that the idea of fusing multiscale features with FPN has a certain positive impact on the network from the data of Scheme 4, and the effect is better than that of the model obviously with only deformable network. The scheme of replacing NMS with soft NMS reduces the average accuracy of crazing defects but improves the average accuracy on the whole. It also has a certain attenuation on the suppression of nonhighest detection frame and reduces the situation that defects with the large overlapping area are missed. Therefore, the combination of the deformable network and multiscale feature fusion scheme can improve the detection performance of the model.

2.9. Algorithm Comparison. The proposed algorithm is compared with other mainstream target detection algorithms in order to further verify the advantages of the proposed algorithm, as shown in Table 6. The comparison algorithms include single-stage target detection algorithms SSD and RetinaNet [38, 39] and double-stage target detection algorithm Cascade R-CNN [40].

In Table 6, *a* is "crazing," *b* is "inclusion," *c* is "patches," *d* is "pitted_surface," *e* is "rolled-in_scale," *f* is "scratches." It can be seen that the overall mean average accuracy of defects in SSD algorithm is only 51.0% from the data in Table 6, which cannot identify multiple defects effectively. The overall mean average accuracies of RetinaNet and Cascade R-CNN algorithms are all about 10% higher than that of SSD algorithm. The proposed algorithm has a great improvement in the detection accuracy of "crashing," "inclusion," "pitted_surface," and "scratches" compared with other mainstream algorithms. The mean average accuracy of the proposed algorithm is 24.2%, 14.6%, and 12.6% higher than that of SSD, RetinaNet, and Cascade R-CNN, respectively. It

shows that the proposed algorithm has a certain breakthrough in steel surface defect detection.

3. Conclusion

The surface defects of steel are taken as the research object in this paper. A new defect detection algorithm based on a deformable network combined with multiscale feature fusion algorithm is proposed in this paper in order to solve the problem of small size and complex shape of steel defect. (1) The deformable convolution is used to reconstruct the feature extraction network in order to enhance the network's ability to extract features. (2) Feature pyramid network is introduced to fuse the multiscale feature graph output to extract the deep semantic features of defect features. (3) The pooling of ROI is replaced by deformable pooling of ROI in order to improve the perception ability of the network to target defect location information. (4) The soft NMS method is used to alleviate the false detection and missing detection of the target. The proposed algorithm and other improved methods are tested on the NEU-DET dataset, and the detection effect is better than other improved methods. At the same time, the superiority of the proposed algorithm is verified by comparing with other mainstream target detection algorithms. The processed images in this paper could be affected by uncertainty. Therefore, it is necessary to use soft computing technology to improve image quality [41] in the future research. In addition, the detection time of this algorithm is long, so we need to further optimize the algorithm in the future to reduce the detection time. Based on the optimization of the algorithm, the fuzzy and similar defects will be deeply studied in order to improve the accuracy of defect detection further.

Data Availability

The program and image data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

References

[1] B. Sun, Z. Liu, and G. Wang, "Classification and formation mechanism of typical press in scale of hot rolled steel plate," *Journal of Northeastern University (Natural Science)*, vol. 31, no. 10, pp. 1417–1420, 2010.

[2] Y.-J. Jeon, D.-C. Choi, S. J. Lee, J. P. Yun, and S. W. Kim, "Steel-surface defect detection using a switching-lighting scheme," *Applied Optics*, vol. 55, no. 1, pp. 47–57, 2016.

[3] C. Mei, Y. Yu, and J. Yu, "Microstructure evolution and corrosion behavior of dissimilar 304/430 stainless steel welded joints," *Journal of Manufacturing Processes*, vol. 50, pp. 183–191, 2020.

[4] T. Takino, "Shaping of steel mold surface of lens array by electrical discharge machining with single rod electrode," *Applied Optics*, vol. 53, no. 33, pp. 8002–8005, 2014.

[5] Y. He, K. Song, Q. Meng, and Y. Yan, "An end-to-end steel surface defect detection approach via fusing multiple hierarchical features," *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 4, pp. 1493–1504, 2020.

[6] Y. Qing, F. Jin, and T. Jian, "Development of an automatic monitoring system for rice light-trap pests based on machine vision," *Journal of Integrative Agriculture*, vol. 19, no. 10, pp. 2500–2513, 2020.

[7] J. Xi, L. Shentu, J. Hu, and M. Li, "Automated surface inspection for steel products using computer vision approach," *Applied Optics*, vol. 56, no. 2, p. 184, 2017.

[8] J. Lu, S. Tang, and J. Wang, "A review on object detection based on deep convoltional neural networks for autonomous driving," *2019 Chinese Control and Decision Conference (CCDC)*, vol. 2019, pp. 1338–1345, 2019.

[9] H.-J. Yoo, "Deep convolution neural networks in computer vision: a review," *IEIE Transactions on Smart Processing and Computing*, vol. 4, no. 1, pp. 35–43, 2015.

[10] E. Qu, Y. Cui, and S. Xu, "Improved Gabor filter for detection of strip surface defects," *Journal of Huazhong University of Science and Technology (Natural Science Edition)*, vol. 45, no. 10, pp. 12–17, 2017.

[11] S. Xu, E. Qu, and H. Chen, "Design of surface defect detection system for cold rolled strip," *Instrument Technique and Sensor*, vol. 1, pp. 65–69, 2017.

[12] M. Versaci, S. Calcagno, M. Cacciola, F. C. Morabito, I. Palamara, and D. Pellicanò, "Innovative fuzzy techniques for characterizing defects in ultrasonic nondestructive evaluation," *Ultrasonic Nondestructive Evaluation Systems*, vol. 2014, pp. 201–232, 2014.

[13] H. Dong, K. Song, Y. He, J. Xu, Y. Yan, and Q. Meng, "PGA-net: pyramid feature fusion and global context attention network for automated surface defect detection," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 12, pp. 7448–7458, 2020.

[14] Q. Luo, X. Fang, J. Su et al., "Automated visual defect classification for flat steel surface: a survey," *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 12, pp. 9329–9349, 2020.

[15] G. Fu, P. Sun, W. Zhu et al., "A deep-learning-based approach for fast and robust steel surface defects classification," *Optics and Lasers in Engineering*, vol. 121, pp. 397–405, 2019.

[16] X. Lv, F. Duan, and J. Jiang, "Deep metallic surface defect detection: the new benchmark and detection network," *Sensors (Basel, Switzerland)*, vol. 20, no. 6, 2020.

[17] M. Vannocci, A. Ritacco, and A. Castellano, "Flatness defect detection and classification in hot rolled steel strips using convolutional neural networks," 2019.

[18] G. Song, K. Song, and Y. Yan, "EDRNet: Encoder-decoder residual network for salient object detection of strip steel surface defects," *IEEE Transactions on Instrumentation and Measurement*, vol. 99, p. 1, 2020.

[19] H. Wang, J. Wang, and F. Luo, "Research on surface defect detection of metal sheet and strip based on multi-level feature Faster R-CNN," *Mechanical Science and Technology for Aerospace Engineering*, vol. 2020, 2020.

[20] X. Dai, H. Chen, and C. Zhu, "Research on surface defect detection and implementation of metal workpiece based on improved Faster R-CNN," *Surface Technology*, vol. 49, no. 10, pp. 362–371, 2020.

[21] K. He, X. Zhang, and S. Ren, "Deep residual learning for image recognition," *IEEE Computer Society*, vol. 2016, 2016.

[22] Y. Tian, Z. Zheng, and G. Li, "Research on the technology of searching for fashion trend image based on ResNet50 model," *Journal of Physics: Conference Series*, vol. 1642, no. 1, 2020.

[23] J. Yuan, Y. Fan, and X. Lv, "Research on the practical classification and privacy protection of CT images of parotid tumors based on ResNet50 model," *Journal of Physics Conference Series*, vol. 1576, 2020.

[24] D. Theckedath and R. R. Sedamkar, "Detecting affect states using VGG16, ResNet50 and SE-ResNet50 networks," *SN Computer Ence*, vol. 1, no. 2, pp. 18–37, 2020.

[25] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.

[26] J. Peng, D. Wang, X. Liao et al., "Wild animal survey using UAS imagery and deep learning: modified Faster R-CNN for kiang detection in Tibetan Plateau," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 169, pp. 364–376, 2020.

[27] C. Han, G. Li, and Y. Ding, "Chimney detection based on faster R-CNN and spatial analysis methods in high Resoluti0on remote sensing images," *Sensors*, vol. 20, no. 16, 2020.

[28] T. Lin, P. . 0 Dollár, and R. Girshick, "Feature pyramid networks for object detection," 2016.

[29] T. Feng, J. Liu, X. Fang, J. Wang, and L. Zhou, "A double-branch surface detection system for armatures in vibration motors with miniature volume based on ResNet-101 and FPN," *Sensors*, vol. 20, no. 8, pp. 2360–2376, 2020.

[30] N. E. Ocer, G. Kaplan, F. Erdem, D. Kucuk Matci, and U. Avdan, "Tree extraction from multi-scale UAV images using Mask R-CNN with FPN," *Remote Sensing Letters*, vol. 11, no. 9, pp. 847–856, 2020.

[31] N. Bodla, B. Singh, and R. Chellappa, "Soft-NMS-improving object detection with one line of code," 2017.

[32] W. Liu, D. Angomir, and D. Erhan, *SSD: Single Shot Multi Box Detector*, Springer, Berlin, Germany, 2016.

[33] L. Fan, J. Luo, H. Liu, and X. Geng, "Data security concurrent with homogeneous by AES algorithm in SSD controller," *IEICE Electronics Express*, vol. 11, no. 13, 2014.

[34] X. Wang, J. Wang, and P. Tang, "Weakly- and semi-supervised fast region-based CNN for object detection," *IEICE Electronics Express*, vol. 34, no. 7, pp. 1269–1278, 2019.

[35] J. Yan, H. Wang, and M. Yan, "IoU-adaptive deformable R-CNN: make full use of IoU for multi-class object detection in remote sensing imagery," *Remote Sensing*, vol. 11, no. 3, 2019.

[36] J. Dai and Y. Xiong, "Deformable convolutional networks," in *Proceedings of the 2017 IEEE International Conference on Computer Vision*Venice, Italy, 2017.

[37] W. Zhao, H. Huang, D. Li, F. Chen, and W. Cheng, "Pointer defect detection based on transfer learning and improved cascade-RCNN," *Sensors*, vol. 20, no. 17, pp. 4939–4957, 2020.

[38] T.-Y. Lin, P. Goyal, K. He, and P. Dollar, "Focal loss for dense object detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 2, pp. 318–327, 2020.

[39] M. Afif, R. Ayachi, and Y. Said, "An evaluation of RetinaNet on indoor object detection for blind and visually impaired persons assistance navigation," *Neural Processing Letters*, vol. 51, no. 10, pp. 1–15, 2020.

[40] Z. Liu, C. Xie, and J. Li, "Application of cascade R-CNN and YOLOv3 in missile target recognition," *Journal of Image and Signal Processing*, vol. 9, no. 2, pp. 102–110, 2020.

[41] M. Versaci, F. C. Morabito, and G. Angiulli, "Adaptive image contrast enhancement by computing distances into a 4-dimensional fuzzy unit hypercube," *IEEE Access*, vol. 1, 2017.