Arabian Journal for Science and Engineering (2023) 48:10213–10225
https://doi.org/10.1007/s13369-022-07567-x

RESEARCH ARTICLE-COMPUTER ENGINEERING AND COMPUTER SCIENCE

Automatic Detection and Quantification of Hot-Rolled Steel Surface Defects Using Deep Learning

Zongchao Liu[1] · Zeyuan Zeng[1] · Junhui Li[1] · Shuai Teng[2]

Received: 9 September 2022 / Accepted: 14 December 2022 / Published online: 30 December 2022

Abstract

Timely defect detection plays a positive role in preventing the evaluation of steel derivative damage. As a state-of-the-art image recognition technology, pixel-level segmentation technology can obtain the pixel distribution and generate the shape of the objects accurately, which provides a potential detection method for steel surface defects. Therefore, this paper employed the well-known pixel-level segmentation CNN (DeepLab_v3+) to detect three defect categories (inclusions, patches and scratches) on the steel surface, and the '*regionprops*' function was used to quantify the defect features (length, average width, maximum width, area and ratio). The results show that ResNet50, as the backbone network of DeepLab_v3+, has the highest detection precision for steel surface defects, and its accuracy is more competitive than that of other algorithms (FCN, SegNet, U-Net and PAG-Net). The proposed quantitative method also achieved encouraging results (the average relative error (ARE) of the evaluation indicator, 10%, 18%, 17%, 23% and 23%, respectively), and the precision was higher than that of the other methods. This demonstrated that the proposed method can greatly benefit steel surface defect detection and evaluation of defect levels.

Keywords Steel surface defects · Detection · Quantification · Deep learning · DeepLab_v3+ · Pixel-level segmentation

1 Introduction

Steel surface defect detection (SSDD) is the key to controlling the quality of industrial products. Steel is the most fundamental material of infrastructures (e.g., construction, machinery, aerospace), and considerable economic losses and reputation losses can be avoided by evaluating steel safety [1]. Visual observation is the most effective method for steel surface defect detection. However, the manual implementation of this task is time-consuming, inefficient and subjective, and an automatic visual observation (AVO) instrument is necessary. However, the AVO system faces the following main challenges [2]: (1) low contrast: the presence of dust and the change in light intensity lead to low contrast between the defect and background shown in images; (2) intraclass difference: the shape of the defects in industrial production is irregular, and the size of the same defect ranges widely; and (3) similarity between categories: due to the uncertainty of the production process, different defect categories have little difference, their textures and gray information are close.

The development of computer vision (CV) technology provides a new solution to these challenges and has become a hot topic. In recent decades, some traditional methods have been widely used, including statistical-based, spectral-based and model-based methods. (1) Statistical-based methods: The distribution of pixel values was measured by statistical methods, including clustering [3], edge-based [4], gray-level [5], cooccurrence matrix [6] and local binary pattern [7] for surface defect detection. (2) Spectral-based methods, which use a set of filters to describe the texture of the image in the transform domain, are widely used in texture analysis. Related research includes Fourier transforms [8], Gabor filters [9], and wavelet transforms [10]. (3) Model-based methods, which need to establish a mathematical model to connect the original texture and defects, require high computational complexity, including the Markov random field

✉ Shuai Teng
tengs93@foxmail.com

[1] School of Civil Engineering, Guangzhou University, Guangzhou 510006, China

[2] School of Civil and Transportation Engineering, Guangdong University of Technology, Guangzhou 510006, China

model [11] and Weibull model [12]. Although these techniques have achieved good results in defect detection, these methods still face major challenges, e.g., noise interference, inaccurate locating and high computational costs.

The emergence of machine learning (ML) technology provides a more advanced algorithm for SSDD. The essence of ML is to analyze and learn the training data and then make accurate decisions or/and predictions for further operations. As an important branch of the model-based method, ML has been widely used in SSDD. The most common methods include support vector machines (SVMs) and neural networks (NNs). As a binary classifier, the SVM is often used to detect defect and non-defect regions [13, 14]. A study also employed the SVM to classify hot-rolled flat steel into defect and normal images [15]. The NN can extract the defect features from the training data and classify the testing data. A study used a two-layer feed forward neural network (FFNN) to detect the image pixels in defect and non-defect regions and achieved encouraging accuracy [16]. Another study used the back-propagation neural network (BPNN) to classify the defects on the steel plate surface [17]. Other methods include distance function, sparse representation, and multi-classifier fusion [18]. However, a large number of parameters in the NN lead to considerable computational complexity and easily lead to over-fitting, which weakens the generalization ability of the SVM and/or traditional NN for the new data (testing data).

A deep convolutional neural network (DCNN) provides a faster and more accurate method for SSDD. Its weight sharing and sparse connection significantly reduce the computational parameters and effectively prevent over-fitting. DCNNs have been widely used in feature extraction and surface defect detection. The defect detection methods based on DCNN include classification-based, region-based and pixel-level segmentation-based methods. (1) Classification-based methods: a semi-supervised [19] and a supervised DCNN [20], were used for the feature extraction and classification of steel surface defects, and the results showed that the DCNN is powerful and robust for classification tasks. However, these methods cannot give accurate defect locations. Additionally, when there are many defects in an image, the accuracy of this method is reduced. (2) Relevant researchers have employed Faster-RCNN to detect a variety of defects and have achieved high locating accuracy (an average precision (AP) value higher than 0.8) [21]. To improve the accuracy of real-time location, the more advanced YOLO series algorithm was used to detect steel surface defects, and the accuracy and speed were 99% and 83 FPS [22], respectively. (3) Pixel-level segmentation-based methods: a fully convolutional network (FCN) model has achieved effective results in the surface detection field [23]. Then, many researchers proposed more pixel-level segmentation models for surface defect detection (including pavement cracks [24], steel surface defects, welds and wood [25]). As an auxiliary tool, GAN technology [26] is also applied to the defect detection task of steel surfaces [27]. In the latest research, a pyramid network model (namely, PGA-Net [2]) was used to achieve multi-scale feature fusion, and competitive accuracy was achieved in the SSDD.

Pixel-level segmentation technology can effectively and accurately obtain the distribution path and shape of defects and provide more accurate information for further quantitative analysis of defects (e.g., length, width, area). With the continuous development of the pixel-level segmentation algorithm, the state-of-the-art DeepLab_v3+ algorithm [28] has achieved more accurate segmentation accuracy. It uses the atrous spatial pyramid pooling (ASSP) sub-module to obtain the multi-scale features of defects, and the encoder–decoder structure can better capture the boundary of the objects [29]. In previous studies, it has been applied to crack detection of asphalt pavement [29] and walls [30]. In this paper, the novel DeepLab_v3+ is applied to the complex SSDD task to evaluate the performance of the model. Additionally, the detection results of pixel-level segmentation are used for subsequent defect quantification tasks, so the main contribution of this paper is to establish an integrated defect detection and quantification method, which provides the main decision-making basis for industrial generation. This article specifically includes the following contents: (1) Compare different backbone network models to obtain the best DeepLab_v3+ detection model; (2) compare different pixel-level segmentation models, and the latest research results confirm the reliability of the proposed method; and (3) implement defect quantification tasks and compare them with other quantification methods.

2 Methods

In this paper, an automatic steel surface defect detection and quantification method is established. In the detection stage, the well-known DeepLab_v3+ algorithm is employed to obtain the accurate defect pixel distribution. In the quantification stage, the reliable region extraction algorithm is used to calculate the corresponding physical parameters of defects. In both stages, it compares with some other popular technologies. The specific workflow is shown in Fig. 1.

2.1 DeepLab_v3+

The overall architecture of the DeepLab_v3+ model is shown in Fig. 2, with an encoder–decoder structure. The main body of the encoder is a classification network (a DCNN) with strong feature extraction ability and an atrous spatial pyramid pooling (ASSP) sub-module with atrous convolution and pooling layers. It uses the DCNN to extract the features of the object and uses the ASSP sub-module to obtain the

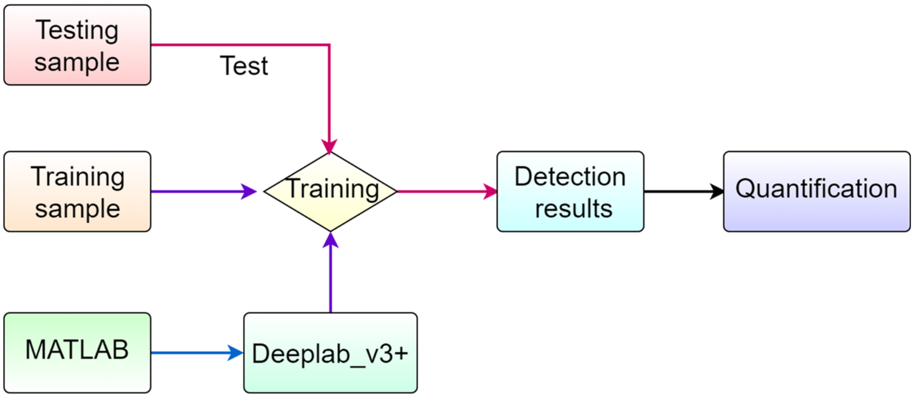

Fig. 1 Specific workflow of the proposed method

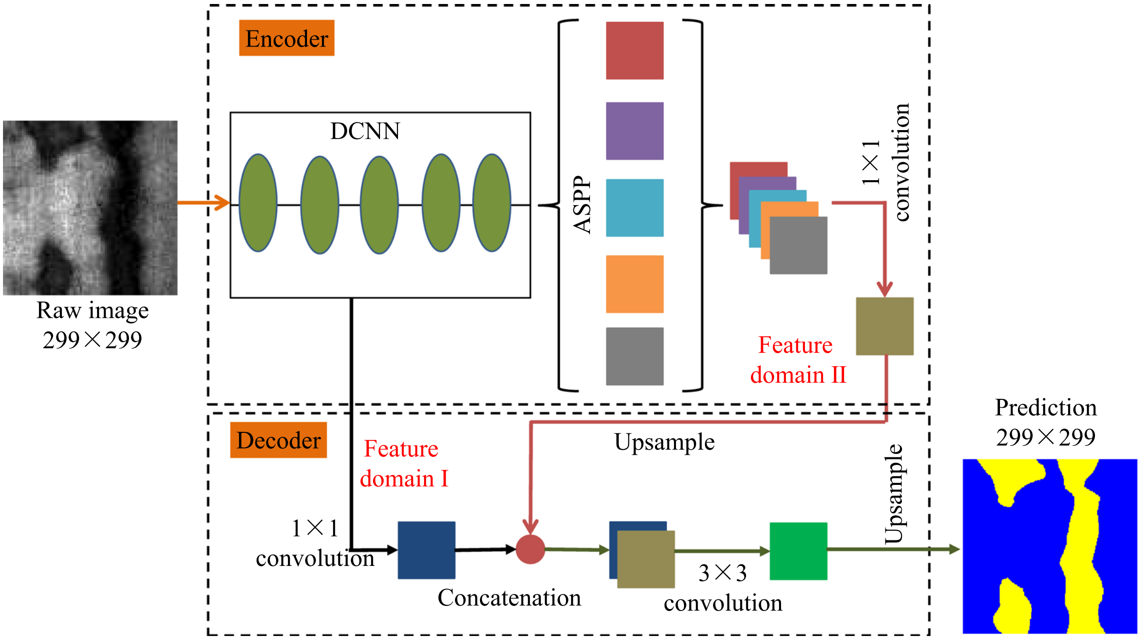

Fig. 2 The DeepLab_v3+ architecture

multi-scale information of the object. In the decoder module, the low-level features (feature domain I, primary feature extraction of the DCNN (backbone network)) and high-level features (feature domain II, advanced feature of the ASPP sub-module) are further fused to improve the accuracy of the segmentation boundary.

The atrous convolution is one of the key operations of the DeepLab_v3+ model. It can control the receptive field of convolution kernels without changing the size of feature images, which is helpful for extracting multi-scale information. The atrous convolution is shown in Fig. 3, where the rate controls the size of the receptive field, and the larger the rate is, the larger the receptive field.

In DeepLab_v3+, the ASSP module is used to further extract the multi-scale information. Here, the atrous convolution of different rates is used to achieve this. The ASPP sub-module includes the following parts: (1) one standard convolution layer (its rate is 1) and three atrous convolutions (its rate is 6, 12 and 18), (2) an average pooling layer is used to obtain image-level features, and then it is sent to a convolution layer (1 × 1 convolution kernel), and (3) the five different scale features obtained from (1) and (2) are concatenated in the channel dimension and then sent to the convolution layer (1 × 1 convolution kernel) for fusion to obtain new features.

Subsequently, in the decoder module, for the first branch input (Feature domain II), the transposed convolution is used to upsample the input (extending the dimension of the feature map). The second branch input (Feature domain I) of the decoder comes from the backbone network and uses the special convolution operation (1 × 1 convolution kernel). After the concatenation, convolution (3 × 3 convolution kernel) and upsampling, the feature maps are gradually restored to their original spatial dimensions, and the output layer outputs each pixel classification of the raw image, that is, marks

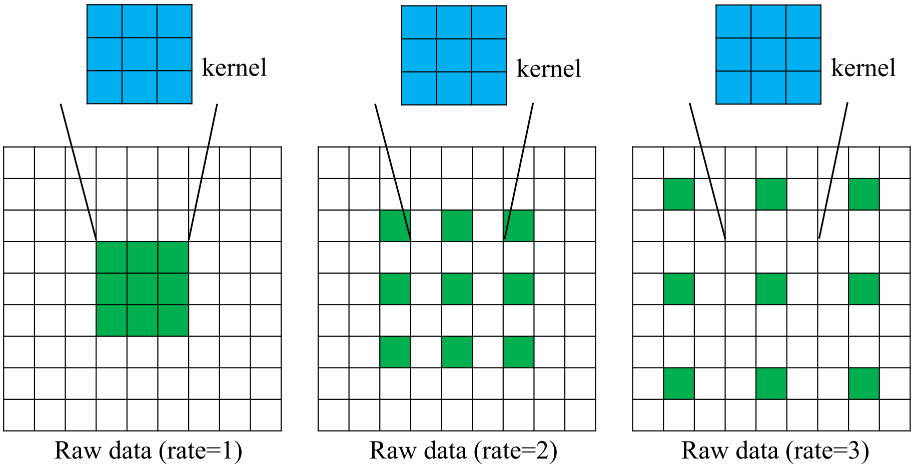

Fig. 3 Atrous convolution

the cracks, thereby achieving pixel-level segmentation of the object region.

In the DeepLab_v3+ model, a DCNN is used as a feature extractor, which provides the primary features for subsequent processing. To compare the impact of different DCNNs on the results, this paper employs five well-known DCNN models as the backbone network of DeepLab_v3+, which are ResNet18, ResNet50, MobileNetv2, Xception, InceptionResNetv2, EfficientNet-b0 and Place365GoogLeNet. Compared with the other popular pixel-level segmentation algorithms (FCN, SegNet and U-Net).

2.2 Defect quantification

Defect quantification is the focus of further research on SSDD. Quantitative analysis of the steel surface defects can effectively evaluate the damage level of the steel and/or steel plate and determine whether it needs to be repaired. The quantitative parameters (physical properties) of steel surface defects usually include length, width, area and ratio. A quantitative method of the asphalt pavement [29] provides the most advanced automatic evaluation inspiration for the surface defects: using the most advanced pixel-level segmentation technology to segment the surface defects, then employing the skeleton extraction algorithm to extract the bones of segmented defects, and calculating the length of defects according to the extracted bones of the defect (Fig. 4). The calculation method is defined as follows:

$$L = \sum_{i=1}^{n} \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2} \tag{1}$$

where $(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2$ is the distance between adjacent points.

However, the quantization results will be affected by the skeleton extraction results (that is, poor skeleton extraction results will reduce the quantization accuracy). Therefore, on this basis, this paper proposes a new defect quantification method (removing the skeleton extraction), that is, employing a function ('*regionprops*' of the image processing toolbox) of MATLAB, which can generate multiple ellipses in the length direction, and the long axis of each ellipse is the length of the subregion (Fig. 5). Therefore, the total length of the defects is defined as:

$$L_{\text{total}} = \sum_{i=1}^{n} L_i \tag{2}$$

where L_i is the length of the major axis of an ellipse.

The area of the defect is the number of pixels of all defects, and the average width of the defect is calculated according to the length of the defect. The ratio of defect pixels is calculated according to the number of pixels in the entire image.

$$W_{\text{Average}} = \frac{\sum_i^N \frac{\text{Area}_i}{L_i}}{N} \tag{3}$$

$$\text{Ratio} = \frac{\text{Area}}{I_{\text{Area}}} \tag{4}$$

where Area_i is the defect area of length L_i and I_{Area} is the number of all pixels in the entire image.

To evaluate the proposed quantization algorithm ('*regionprops*' function of MATLAB), the average relative error (*ARE*) is used to evaluate the effectiveness of the quantization algorithm, which is defined as follows:

$$\text{RE} = \frac{R_{\text{p}} - R_{\text{r}}}{R_{\text{r}}} \times 100\% \tag{5}$$

$$\text{ARE} = \sum_{j=1}^{M} \frac{R_{\text{p}} - R_{\text{r}}}{R_{\text{r}}} \times 100\% \tag{6}$$

where R_{p} and R_{r} are the prediction results and real labels, respectively. M is the number of detection samples.

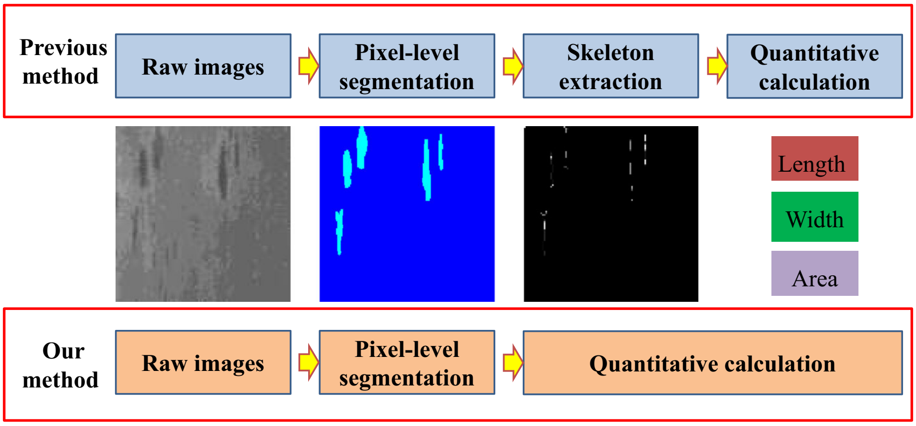

Fig. 4 Previous methods and our methods

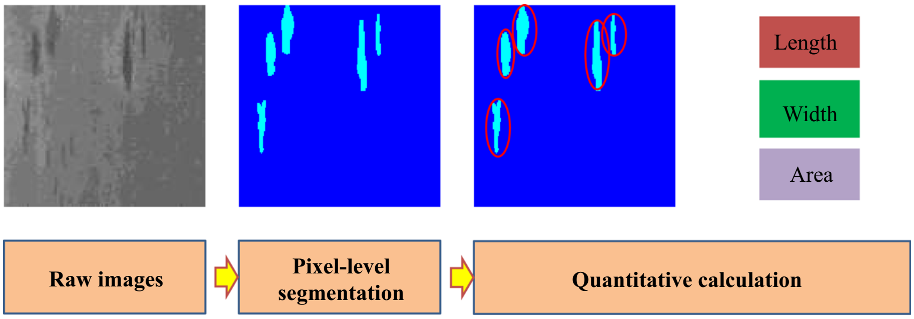

Fig. 5 The acquisition method of defect length

2.3 Experimental Setup

The dataset contains 900 images of steel surface defects (NEU dataset [31], including 300 images of inclusions, 300 images of patches and 300 images of scratches) with 200 × 200 pixels. It is collected through relevant literature and can be obtained on GitHub. The image is resized to 299 × 299 by using the 'imresize' function, and the image is annotated by using the 'ImageLabeler' toolbox of MATLAB. Twenty percent of them were used as the validation data (10%) and testing data (10%). Several labeled examples are shown in Fig. 6.

The operating platform is on a computer with an Intel (R) Core(TM) i7-10700/10700F CPU and an NVIDIA GeForce RTX2060, 6 GB GPU. Software configuration: Windows 10, MATLAB 2020a. The pixel-level segmentation algorithm DeepLab_v3+ is established by the 'deep learning toolbox' and 'computer vision toolbox.' The quantitative analysis is implemented by the 'image processing toolbox.'

The precision of the pixel-level segmentation algorithm is evaluated using the following indicators:

Intersection over union (IoU) is an indicator for evaluating the overlap between the predicted (A_p) and real (A_r) object pixels. IoU is defined as:

$$\mathrm{IoU} = \frac{\mathrm{area}(A_\mathrm{p} \cap A_\mathrm{r})}{\mathrm{area}(A_\mathrm{p} \cup A_\mathrm{r})} \tag{7}$$

The mean IoU of all classes is defined as:

$$\mathrm{MIoU} = \frac{\mathrm{IoU}}{N} \tag{8}$$

where N is the number of classes (in this paper, $N = 2$, i.e., the crack and noncrack).

The accuracy and F-score were used to evaluate the classification effect (crack and noncrack):

$$\mathrm{Accuracy} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{FP} + \mathrm{FN} + \mathrm{TN}} \tag{9}$$

$$\mathrm{Precision} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{10}$$

$$\mathrm{Recall} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{11}$$

$$F\text{ - score} = 2 \times \frac{\mathrm{Precision} + \mathrm{Recall}}{\mathrm{Precision} \times \mathrm{Recall}} \tag{12}$$

where true positive (TP) denotes that a real defect pixel is predicted correctly. False positive (FP) denotes that a real non-defect pixel is predicted as a defect pixel. False negative (FN) denotes that a real defect pixel is predicted as a non-defect pixel. True negative (TN) denotes that a real non-defect pixel is predicted correctly.

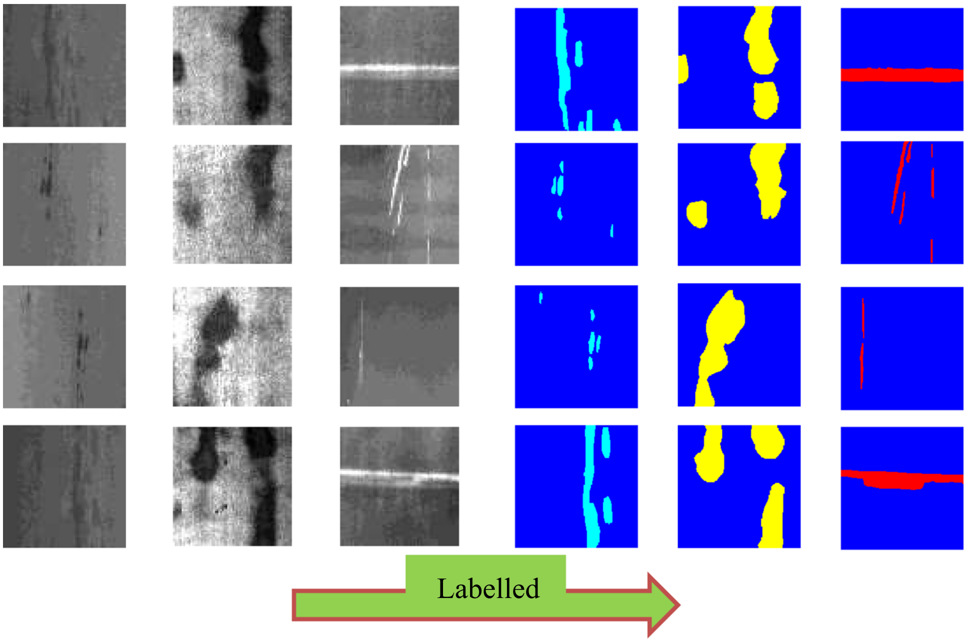

Fig. 6 Labeled examples of some defect images

3 Results and Discussion

3.1 Training Process and Testing Results

To identify a high-precision pixel-level segmentation model, the training data were used to train DeepLab_v3+ of different backbone networks. The training process is shown in Fig. 7 (360 iterations). Among them, ResNet50 as a backbone network has achieved the highest training accuracy and the lowest loss, and Xception has the worst performance. The testing data were used to assess the generalization capability of the network. The testing results are shown in Table 1. ResNet50 has the highest detection precision (with MIoU = 0.81, accuracy = 97% and F-score = 0.81, some detection examples in Fig. 8, the detection results were basically consistent with the real labels). ResNet18, MobileNetv2, InceptionResNetv2, EfficientNet-b0 and Place365GoogLeNet have similar detection precision, and the precision of Xception is the lowest. ResNet50 has a detection speed comparable to those of other models.

Table 2 shows the detailed testing results of DeepLab_v3+ based on the ResNet50 backbone network. The results show that different defect categories have different performances in evaluation indices, the highest MIoU value (0.85) was obtained for patches, the highest accuracy (99%) was obtained for scratches, and the highest F-score value (0.78) was obtained for inclusions. This means that for the overall evaluation of the model, multiple evaluation indicators are necessary. For small defects, the F-score and accuracy can be employed for evaluation; for block defects, the MIoU and accuracy can be employed for evaluation.

Then, an ablation experiment was carried out in DeepLab_v3+, and the backbone network ResNet50 of DeepLab_v3+ was frozen (the weight did not change during network training). Table 3 shows the test results of various defects. The results show that the detection effect of DeepLab_v3+ of the frozen backbone network for various defects has decreased, especially for the scratches (MIoU decreased by 17.6%, accuracy decreased by 13%, and F-score decreased by 20%).

3.2 Comparative Study

To further demonstrate the effectiveness of the proposed method, the results of Sect. 3.1 are compared with those of other popular pixel-level segmentation algorithms. The testing results are shown in Table 4. The results show that the detection precision of the FCN, SegNet and U-Net (MIoU = 0.23, accuracy = 25%, F-score = 0.6 to 0.78) was lower than that of DeepLab_v3+. Therefore, the results confirm that DeepLab_v3+ was the best algorithm for SSDD.

Subsequently, we compared our method with the relevant steel surface defect segmentation algorithm [2]. Figure 9 shows some detection examples. The comparison shows that our method was more competitive than the previous method in the detection of steel surface defects. The previous method has an ideal detection effect for clear defects (there was a clear boundary with the background), but for some fuzzy defects, the method is invalid (marked by the red box in Fig. 8).

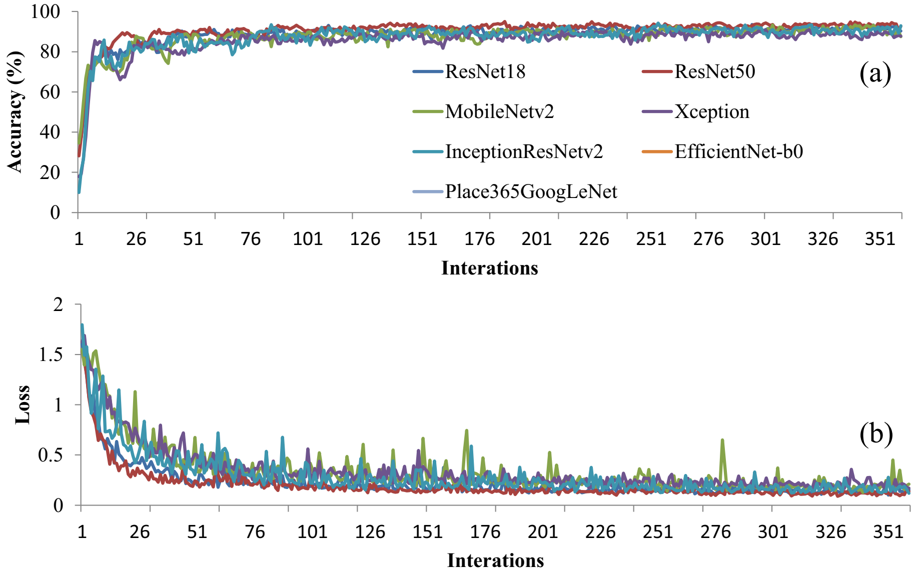

Fig. 7 Training process of the DeepLab_v3+ using different backbone networks

Table 1 Segmentation results of DeepLab_v3+ using different backbone networks

Evaluation indicators	Backbone network						
	ResRet18	ResRet50	MobileNetv2	Xception	InceptionResnetv2	EfficientNet-b0	Place365GoogLeNet
MIoU	0.74	0.81	0.74	0.71	0.76	0.75	0.73
Accuracy	97%	97%	96%	94%	97%	97%	95%
F-score	0.72	0.81	0.71	0.67	0.72	0.75	0.76
Detection time (per image)	0.6 s	0.7 s	0.3 s	0.4 s	0.4 s	0.8 s	1.6 s

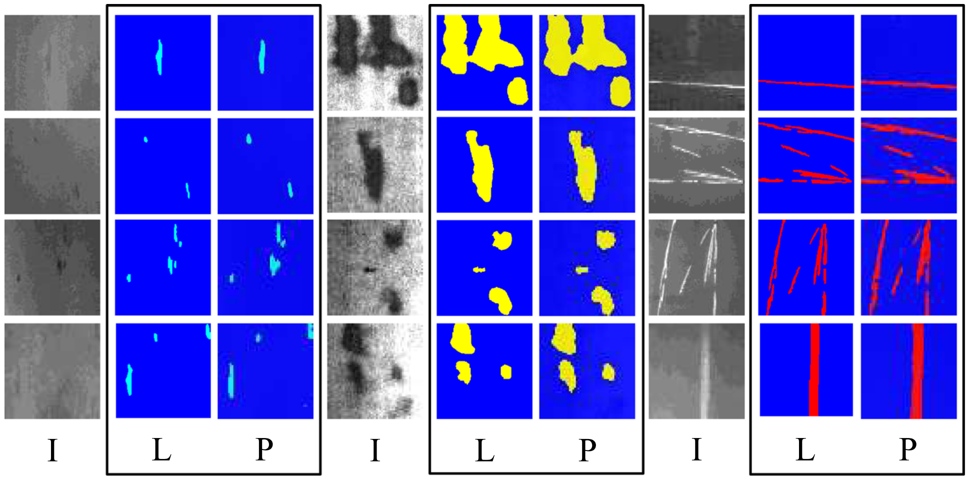

Fig. 8 Testing results of the DeepLab_v3+. I: Raw image, L: Label, P: Prediction

Table 2 Detailed testing results of different defects

Evaluation indicators	Defect category			
	Background	Inclusion	Patches	Scratches
MIoU	0.97	0.68	0.85	0.74
Accuracy	98%	95%	97%	99%
F-score	0.89	0.78	0.63	0.75

Table 3 Detailed testing results of the ablation experiment

Evaluation indicators	Defect category			
	Background	Inclusion	Patches	Scratches
MIoU	0.93	0.53	0.82	0.61
Accuracy	92%	92%	95%	86%
F-score	0.82	0.73	0.59	0.60

Table 4 Detection results of popular pixel-level segmentation networks

Evaluation indicators	Popular pixel-level segmentation network			
	FCN	SegNet	U-Net	DeepLab_v3+
MIoU	0.23	0.23	0.23	0.84
Accuracy	25%	25%	25%	97%
F-score	0.78	0.60	0.78	0.81
Detection time (per image)	0.9 s	0.6 s	0.7 s	0.7 s

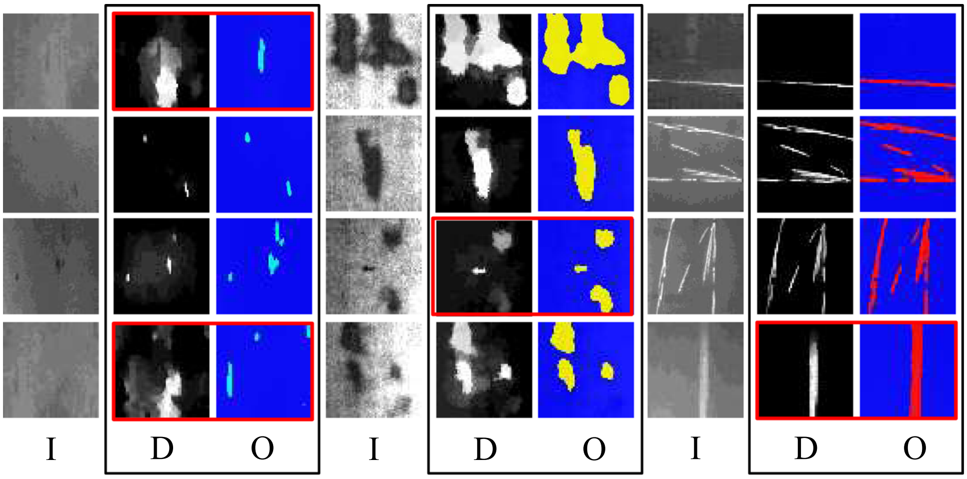

Fig. 9 Comparison with related steel surface defect segmentation technology. I: Raw image; D: Dong et al. method [2]; O: Our method

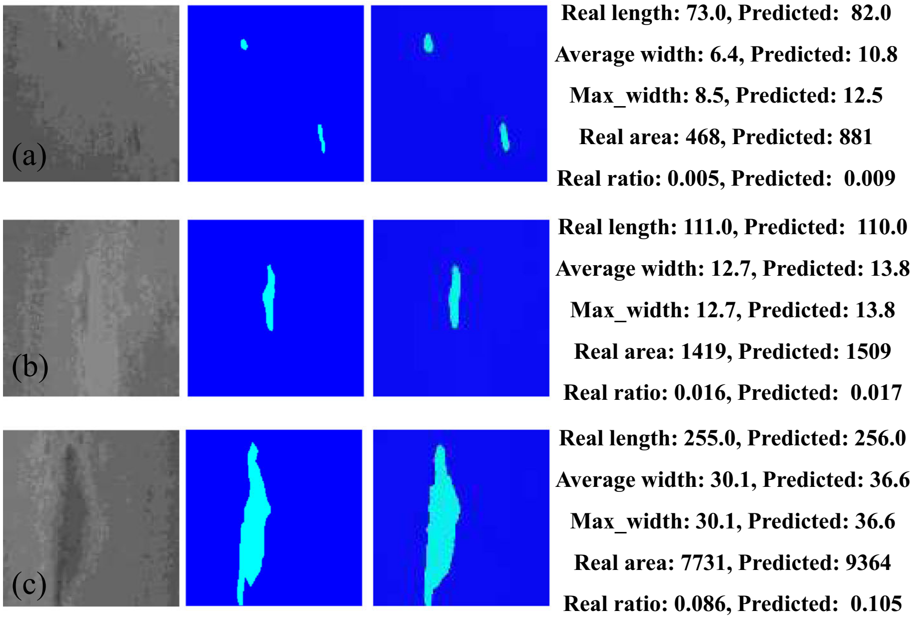
(a)
Real length: 73.0, Predicted: 82.0
Average width: 6.4, Predicted: 10.8
Max_width: 8.5, Predicted: 12.5
Real area: 468, Predicted: 881
Real ratio: 0.005, Predicted: 0.009
(b)
Real length: 111.0, Predicted: 110.0
Average width: 12.7, Predicted: 13.8
Max_width: 12.7, Predicted: 13.8
Real area: 1419, Predicted: 1509
Real ratio: 0.016, Predicted: 0.017
(c)
Real length: 255.0, Predicted: 256.0
Average width: 30.1, Predicted: 36.6
Max_width: 30.1, Predicted: 36.6
Real area: 7731, Predicted: 9364
Real ratio: 0.086, Predicted: 0.105

Fig. 10 Quantitative results of the inclusions

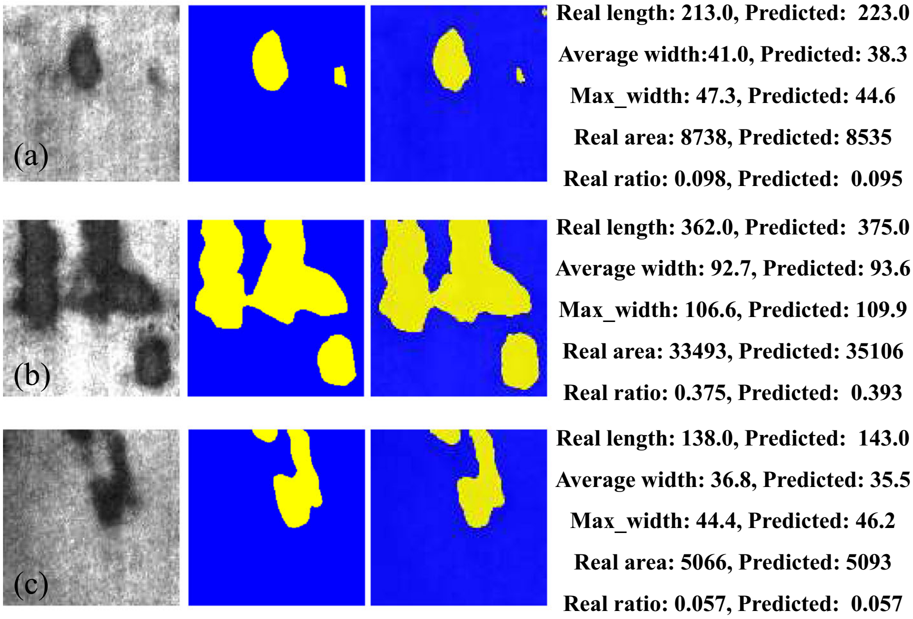
(a)
Real length: 213.0, Predicted: 223.0
Average width:41.0, Predicted: 38.3
Max_width: 47.3, Predicted: 44.6
Real area: 8738, Predicted: 8535
Real ratio: 0.098, Predicted: 0.095
(b)
Real length: 362.0, Predicted: 375.0
Average width: 92.7, Predicted: 93.6
Max_width: 106.6, Predicted: 109.9
Real area: 33493, Predicted: 35106
Real ratio: 0.375, Predicted: 0.393
(c)
Real length: 138.0, Predicted: 143.0
Average width: 36.8, Predicted: 35.5
Max_width: 44.4, Predicted: 46.2
Real area: 5066, Predicted: 5093
Real ratio: 0.057, Predicted: 0.057

Fig. 11 Quantitative results of the patches

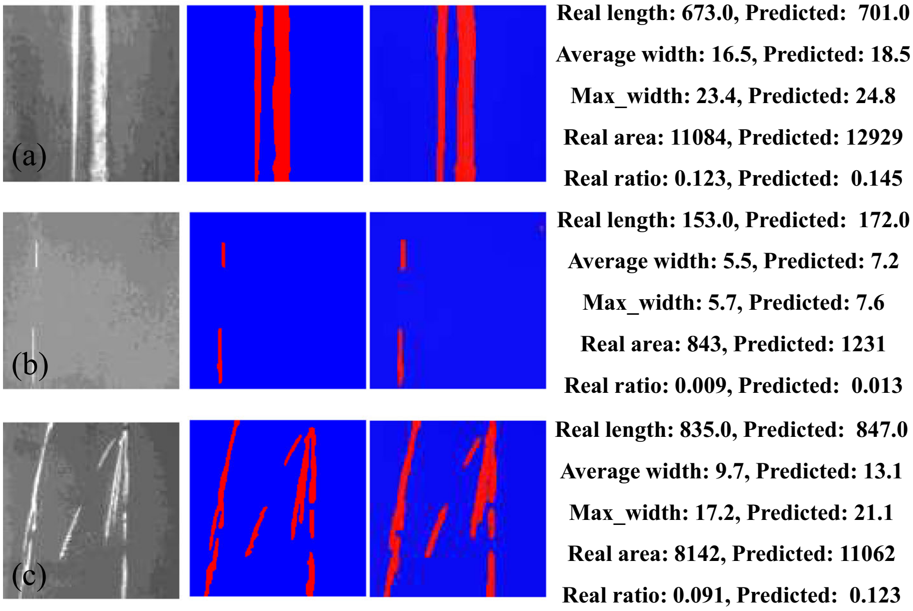
(a)
Real length: 673.0, Predicted: 701.0
Average width: 16.5, Predicted: 18.5
Max_width: 23.4, Predicted: 24.8
Real area: 11084, Predicted: 12929
Real ratio: 0.123, Predicted: 0.145
(b)
Real length: 153.0, Predicted: 172.0
Average width: 5.5, Predicted: 7.2
Max_width: 5.7, Predicted: 7.6
Real area: 843, Predicted: 1231
Real ratio: 0.009, Predicted: 0.013
(c)
Real length: 835.0, Predicted: 847.0
Average width: 9.7, Predicted: 13.1
Max_width: 17.2, Predicted: 21.1
Real area: 8142, Predicted: 11062
Real ratio: 0.091, Predicted: 0.123

Fig. 12 Quantitative results of the scratches

3.3 Quantification and Comparison

The high-precision defect segmentation results obtained in Sect. 3.1 were applied to the quantitative task of defects, and the length, width, area and ratio of various defects were extracted by using the function '*regionprops*' of MATLAB. Some detection examples of inclusions, patches and scratches are shown in Figs. 10, 11 and 12, respectively. All results of the testing data are shown in Fig. 13 (including predefined labels and predicted results). The average relative errors of the length, average width, maximum width, area and ratio were 10%, 18%, 17%, 23%, and 23%, respectively.

Figure 10a shows the detection results of small inclusions, with low length measurement error and high width measurement error (average relative error of approximately 47%), which results in high area and ratio errors. Figure 10b shows the detection results of fuzzy inclusions, and all the indices achieved excellent accuracy. Figure 10c shows the detection results of large inclusions, with correct length measurement and poor measurement accuracy of width (average relative error of approximately 20%). Figure 11a shows the detection results of different sizes of patches, and all the indicators achieved excellent accuracy. Figure 11b shows the detection results of large patches, and all the indices achieved

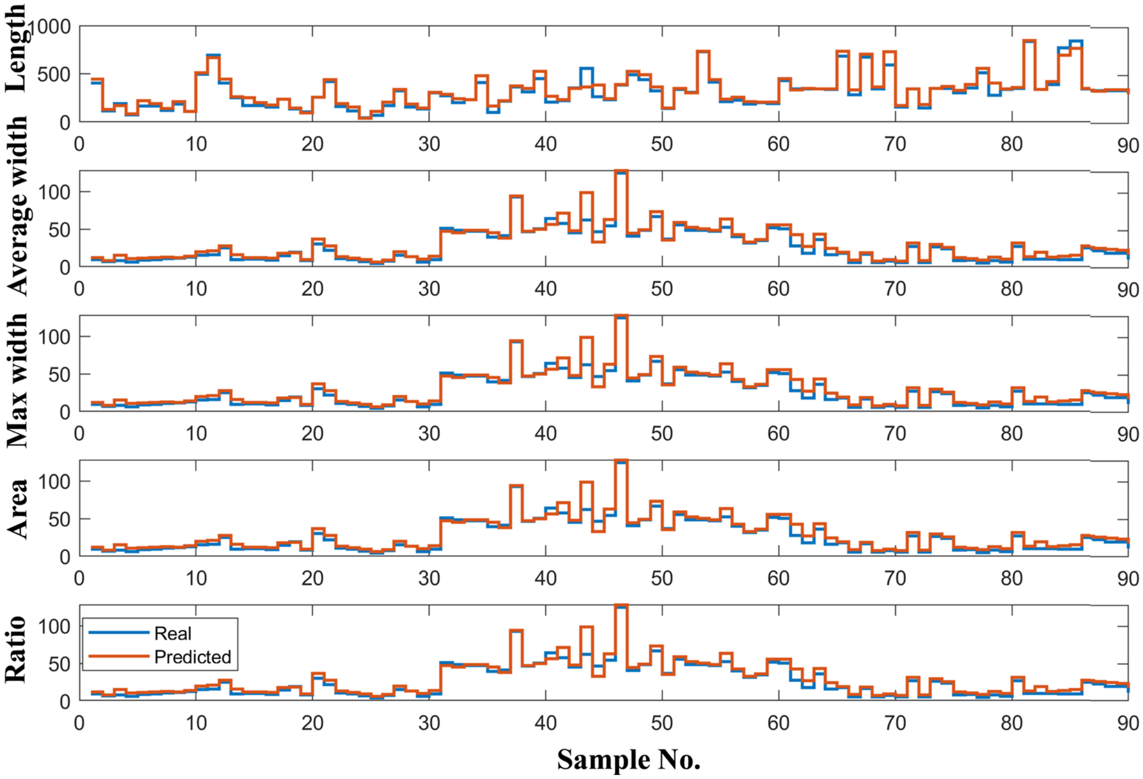

Fig. 13 Comparison of real results and predicted results

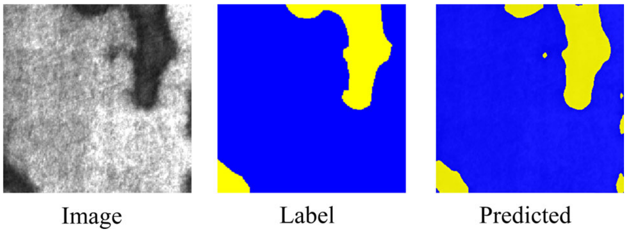

Fig. 14 The label and predicted result of Sample 44

excellent accuracy. Figure 11c shows the detection results of curved patches, and all the indices also obtain excellent accuracy. Figure 12a shows the results of the different scratch detection widths, and all the indicators achieved excellent accuracy. Figure 12b shows the detection results of intermittent scratches, and all the indicators achieved excellent accuracy. Figure 12c shows the detection results of complex scratches, the length measurement is correct, and the width detection has a high relative error (35%). Therefore, the proposed method has the highest accuracy for patch detection and cannot detect small inclusions and scratches accurately.

The statistical results in Fig. 13 show that the relative error of Sample 44 was the largest. The raw image (Sample 44), label and prediction results are shown in Fig. 14. It is found that the artificially labeled defects were incomplete (in the label image, the small defects in the edge area are not labeled), and the predicted results are more consistent with the defect distribution of the raw image. Therefore, it has a great influence on the quantitative calculation results of labels and prediction results (large relative error), which is due to manual error labeling, so the quantitative calculation of prediction results was more reliable.

To further examine the performance of this method, this paper employs the previous method [29] (skeleton extraction in Sect. 2.2) to quantitatively analyze the hot-rolled steel surface defects. The average relative errors (Table 5) of defect length, average width, maximum width, area and ratio were 12%, 22%, 22%, 23% and 23%, respectively, and bold represents excellent detection effect. The average relative errors of our method (10%, 18%, 17%, 23%, and 23%) were 20%, 22%, 29%, 0% and 0% lower than those of previous methods, respectively.

Subsequently, the error of the previous method was analyzed. Figure 15 shows some extraction results of the skeleton extraction algorithm for defects. It was found that the skeleton extraction algorithm has high precision for some small defects (inclusions), but the extraction effect is poor for blocky (patches) and/or wide (scratches) defects, which forms many branches, resulting in the predicted results being higher than the real results, thus affecting the positive effect of quantification precision. Therefore, for the previous defect quantification method based on the skeleton extraction algorithm, there are some limitations. For small defects,

Table 5 Detection results of the previous method and our method

	Previous method					Our method				
	L (%)	AW (%)	MW (%)	*A* (%)	*R* (%)	*L* (%)	AW (%)	MW (%)	*A* (%)	*R* (%)
I	14	27	21	29	29	**11**	**21**	22	29	29
P	10	12	17	12	12	11	**9**	**7**	12	12
S	12	29	28	28	28	**7**	**25**	**24**	28	28
Total	12	22	22	23	23	**10**	**18**	**17**	23	23

L, AW, MW, *A* and *R* are the length, average width, max width, area and ratio, respectively. *I*, *P* and *S* are the inclusion, patches and scratches, respectively

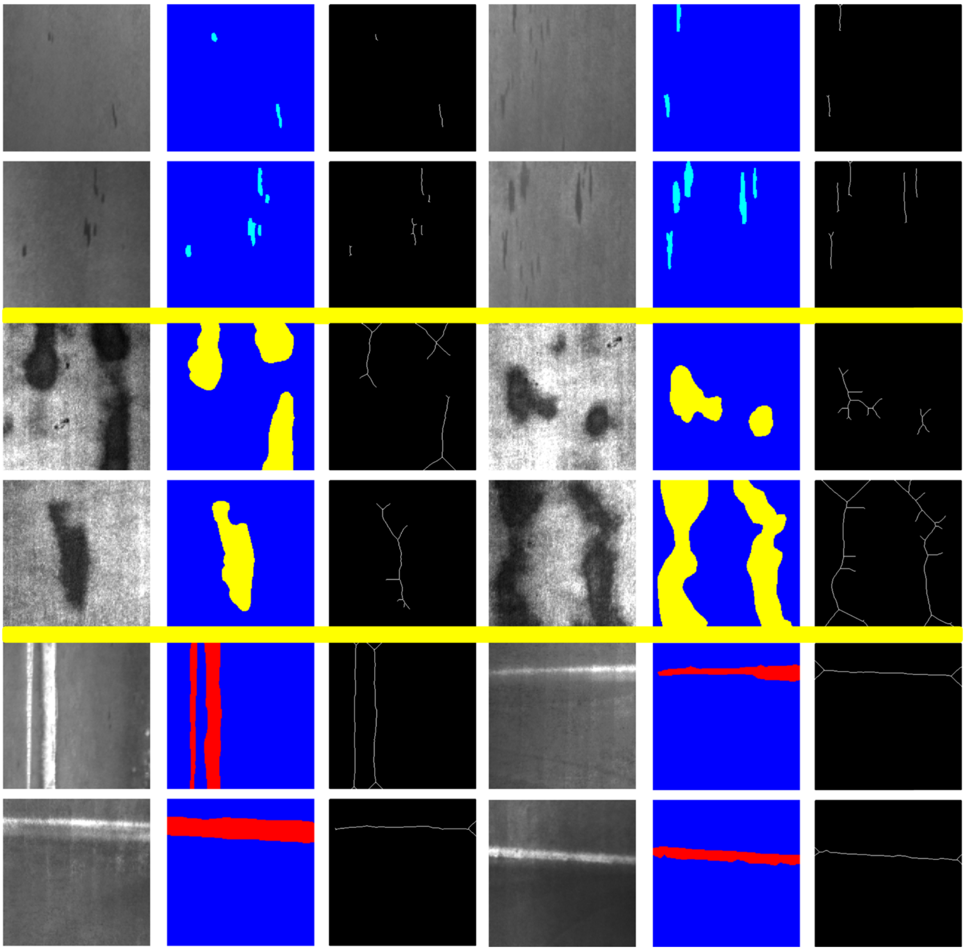

Fig. 15 Skeleton extraction results of steel surface defects

the skeleton extraction algorithm will be a highly accurate method, while for some coarse defects, this method has some disadvantages.

4 Conclusions

This paper presents an automatic method for detecting and quantifying steel surface defects. In the detection stage, a well-known pixel-level segmentation model DeepLab_v3+ is employed to detect the defect images, and an optimal DeepLab_v3+ model is identified. Compared with other popular pixel-level segmentation algorithms (FCN, SegNet, U-Net and PGA-Net), its detection precision (with MIoU = 0.81, accuracy = 97% and *F*-score = 0.81) is competitive. In the quantization stage, the reliable region quantization function (‘*regionprops*’) of MATLAB is employed to quantify the defects in the image. The relative errors of the length, average width, maximum width, area and ratio are 10%, 18%, 17%, 23% and 23%, respectively. Moreover, our method is more competitive than the latest quantization algorithms proposed in a related reference (12%, 22%, 22%, 23% and 23%).

Based on the above results, the following conclusions are drawn:

(1) For steel surface defect detection, ResNet50 is the best backbone network of DeepLab_v3+.
(2) DeepLab_v3+ is more suitable for detecting steel surface defects than other algorithms (FCN, SegNet, U-Net and PGA-Net).
(3) For the quantification of steel surface defects, the '*regionprops*' function of MATLAB is more effective than the related quantification algorithm.

Acknowledgements This research was supported by the project (No. 52208309) of the National Natural Science Foundation of China.

Author Contributions Conceptualization, ST; Data curation, XL and ZL; Formal analysis, JL; Funding acquisition, JL; Investigation, ST and XL; Methodology, ST and ZL; Project administration, ST; Software, ST; Writing – original draft, ST; Writing – review & editing, ZL.

Declarations

Conflict of interest The authors declare no conflict of interest.

References

1. Luo, Q.; Fang, X.; Liu, L.; Yang, C.; Sun, Y.: Automated visual defect detection for flat steel surface: a survey. IEEE Trans. Instrum. Meas. **69**, 626–644 (2020)
2. Dong, H.; Song, K.; He, Y.; Xu, J.; Meng, Q.: PGA-Net: pyramid feature fusion and global context attention network for automated surface defect detection. IEEE Trans. Ind. Inform. **16**, 7448–7458 (2019)
3. Wang, J.; Li, Q.; Gan, J.; Yu, H.; Yang, X.: Surface defect detection via entity sparsity pursuit with intrinsic priors. IEEE Trans. Ind. Inform. **16**, 141–150 (2020)
4. Shi, T.; Kong, J.; Wang, X.; Liu, Z.; Zheng, G.: Improved Sobel algorithm for defect detection of rail surfaces with enhanced efficiency and accuracy. J. Cent. South Univ. **23**, 2867–2875 (2016)
5. Choi, J.; Kim, C: Unsupervised detection of surface defects: a two-step approach. In: IEEE International Conference on Image Processing, Orlando, FL, USA, pp. 1037–1040 (2012)
6. Xing, Z.; Jia, H.: Multilevel color image segmentation based on GLCM and improved salp swarm algorithm. IEEE Access **7**, 37672–37690 (2019)
7. Luo, Q.; Fang, X.; Sun, Y.; Liu, L.; Ai, J.; Yang, C.; Simpson, O.: Surface defect classification for hot-rolled steel strips by selectively dominant local binary patterns. IEEE Access **7**, 23488–23499 (2019)
8. Yong-Hao, A.I.; Ke, X.U.: Surface detection of continuous casting slabs based on curvelet transform and kernel locality preserving projections. J. Iron Steel Res. Int. **20**, 80–86 (2013)
9. Kim, J.; Um, S.; Min, D.: Fast 2-D complex Gabor filter with kernel decomposition. IEEE Trans. Image Process. **27**, 1731–1822 (2018)
10. Liu, W.; Yan, Y.: Automated surface defect detection for cold-rolled steel strip based on wavelet anisotropic diffusion method. Int. J. Ind. Syst. Eng. IJISE **17**, 224–239 (2014)
11. Ke, X.U.: Application of hidden Markov tree model to on-line detection of surface defects for steel strips. J. Mech. Eng. **49**, 34 (2013)
12. Liu, K.; Wang, H.; Chen, H.; Qu, E.; Ying, T.; Sun, H.: Steel surface defect detection using a new Haar–Weibull-variance model in unsupervised manner. IEEE Trans. Instrum. Meas. **66**, 2585–2596 (2017)
13. Choi, D.C.; Jeon, Y.J.; Lee, S.J.; Yun, J.P.; Kim, S.W.: Algorithm for detecting seam cracks in steel plates using a Gabor filter combination method. Appl. Opt. **53**, 4865–4872 (2014)
14. Ashour, M.W.; Khalid, F.; Halin, A.A.; Abdullah, L.N.; Darwish, S.H.: Surface defects classification of hot-rolled steel strips using multi-directional shearlet features. Arab. J. Sci. Eng. **44**, 2925–2932 (2019)
15. Ghorai, S.; Mukherjee, A.; Gangadaran, M.; Dutta, P.K.: Automatic defect detection on hot-rolled flat steel products. IEEE Trans. Instrum. Meas. **62**, 612–621 (2013)
16. Kang, G.W.; Liu, H.B.: Surface defects inspection of cold rolled strips based on neural network. In: Machine Learning and Cybernetics, 2005. Proceedings of 2005 International Conference on, Guangzhou, China, pp. 5034–5037 (2005)
17. Zhao, X.Y.; Lai, K.S.; Dai, D.M.: An improved BP algorithm and its application in classification of surface defects of steel plate. J. Iron Steel Res. Int. **14**, 52–55 (2007)
18. Luo, Q.; Fang, X.; Su, J.; Zhou, J.; Zhou, B.; Yang, C.; Liu, L.; Gui, W.; Tian, L.: Automated visual defect classification for flat steel surface: a survey. IEEE Trans. Instrum. Meas. **69**, 9329–9349 (2020)
19. Song, K.; Dong, H.; Yan, Y.: Semi-supervised defect classification of steel surface based on multi-training and generative adversarial network. Opt. Lasers Eng. **122**, 294–302 (2019)
20. Masci, J.; Meier, U.; Fricout, G.; Schmidhuber, J.: Multi-scale pyramidal pooling network for generic steel defect classification. In: The 2013 International Joint Conference on Neural Networks (IJCNN), Dallas, TX, USA, pp. 1–8 (2013)
21. Cha, Y.J.; Choi, W.; Suh, G.; Mahmoudkhani, S.; Büyükztürk, O.: Autonomous structural visual inspection using region-based deep learning for detecting multiple damage types. Comput. Aided Civ. Infrastruct. Eng. **33**, 1–17 (2018)
22. Li, J.; Su, Z.; Geng, J.; Yin, Y.: Real-time detection of steel strip surface defects based on improved YOLO detection network. IFAC-PapersOnLine **51**, 76–81 (2018)
23. Long, J.; Shelhamer, E.; Darrell, T.: Fully convolutional networks for semantic segmentation. IEEE Trans. Pattern Anal. Mach. Intell. **39**, 640–651 (2015)
24. Yang, F.; Zhang, L.; Yu, S.; Prokhorov, D.; Mei, X.; Ling, H.: Feature pyramid and hierarchical boosting network for pavement crack detection. IEEE Trans. Intell. Transp. Syst. **3**, 1525–1535 (2019)
25. Ren, R.; Hung, T.; TanKay, C.: A generic deep-learning-based approach for automated surface inspection. IEEE Trans. Cybern. **48**, 929–940 (2018)
26. Li, X.; Du, Z.; Huang, Y.; Tan, Z.: A deep translation (GAN) based change detection network for optical and SAR remote sensing images. ISPRS J. Photogramm. Remote Sens. **179**, 14–34 (2021)
27. Liu, K.; Li, A.; Wen, X.; Chen, H.; Yang, P: Steel surface defect detection using GAN and one-class classifier. In: 2019 25th International Conference on Automation and Computing (ICAC) (2019)
28. Chen, L.C.; Zhu, Y.; Papandreou, G.; Schroff, F.; Adam, H: Encoder–decoder with atrous separable convolution for semantic image segmentation. In: Computer Vision—ECCV, vol. 11211, pp. 833–851 (2018)

29. Ji, A.; Xue, X.; Wang, Y.; Luo, X.; Xue, W.: An integrated approach to automatic pixel-level crack detection and quantification of asphalt pavement. Autom. Constr. **114**, 103176 (2020)
30. Dais, D.; Bal, İE.; Smyrou, E.; Sarhosis, V.: Automatic crack classification and segmentation on masonry surfaces using convolutional neural networks and transfer learning. Autom. Constr. **125**, 103606 (2021)
31. Song, G.; Song, K.; Yan, Y.: EDRNet: encoder–decoder residual network for salient object detection of strip steel surface defects. IEEE Trans. Instrum. Meas. **69**, 9709–9719 (2020)