ISSN: 2688-8645

Faster RCNN based on Feature Fusion and its Application in Steel Surface Defect Detection

Shiwei Xu[1] and Zhangyi Shao[2,*]

[1]School of Artificial Intelligence, Wenzhou Polytechnic, Wenzhou 325000, China

[2]Department of medical engineering, The Second Affiliated Hospital and Yuying Children's Hospital of Wenzhou Medical University, Wenzhou 325000, China

Abstract

Surface defects are an inevitable problem in steel production and processing. To solve the problems of error detection, leakage detection and low accuracy of traditional manual methods for steel surface defect detection, this paper proposes a steel surface defect detection algorithm based on Faster RCNN, a classical model in the field of computer vision object detection. The proposed model uses ResNet50 as its backbone and introduces FPN to fuse the multi-layer feature maps of the backbone to improve the detection capability for defects of different scales. The experimental results on NEU-DET dataset show that the proposed model in this paper has different detection accuracies for six different types of defects in the dataset, and the overall mAP reaches 0.708. In addition, the model execution speed reaches 43.3 frame/s, which meets the applications in industrial scenarios.

Keywords

Surface Defect Detection; Faster RCNN; ResNet; Deep Learning.

1. Introduction

Metal workpiece is an important part of some products in the process of machining, and the quality of metal workpiece directly affects the market competitiveness of enterprise products. With the rapid development of China's manufacturing industry, the demand for the quantity and variety of metal workpiece products is increasing day by day[1]. Consumers and manufacturers are putting forward higher requirements for the quality of metal workpieces, which need to meet the normal performance of use, but also good surface quality. In the production process of the steel industry, the presence of defects such as patches, cracks and scratches on the steel surface can be caused by factors such as production machines and production processes[2]. These defects will not only affect the corrosion resistance and wear resistance of the strip, but even lead to safety accidents in serious cases. Therefore, it is very important to detect surface defects on metal material workpieces during machining, and it is of great practical importance to study an algorithm that can accurately and efficiently detect surface defects on steel.

The metal material workpiece manufacturing process is simple, but the structure is complex and the inspection technology is demanding, the traditional inspection methods are mainly penetration detection technology, eddy current detection technology, magnetic particle detection technology, visual inspection technology and image detection technology[3], these inspection technologies mainly have the following problems: first, the detection speed is slow, resulting in low detection efficiency; second, the detection process generates environmental pollution; third, the detection process is cumbersome and cannot achieve automated detection and identification. How to improve the detection of surface defects on metal workpieces is a problem worthy of in-depth study in today's industrial production process. This paper studies

ISSN: 2688-8645

the metal material workpiece is mainly limited to the production and processing of automotive parts generated by surface defects.

With the development of image processing technology, machine vision-based surface defect detection methods have gradually replaced manual inspection methods and are practiced in the industrial production inspection process. Machine vision inspection technology is a non-contact automatic inspection technology, with the advantages of safety and reliability, high detection accuracy, can operate in complex production environment for a long time, is an effective way to achieve factory production automation and intelligence.

There are two research methods to apply machine vision technology to metal workpiece surface defect detection: traditional machine vision and deep learning vision. Traditional machine vision is based on digital signal analysis and processing theory, and then machine learning methods are used to get the desired results. With the improvement of computer hardware performance and breakthroughs in artificial intelligence algorithms, the research hotspot has changed to a deep learning based approach. Compared with traditional techniques, deep learning vision can achieve higher accuracy in tasks such as image classification, semantic segmentation, and target detection[4]. Although the research hotspots in the field of machine vision in recent years are in the direction of deep learning, traditional machine vision methods for some specific types of surface defects, there is no disadvantage in terms of detection speed and accuracy, but rather some ideas and techniques are worth learning from, so the research on deep learning vision detection has a reference value.

Traditional machine vision-based surface defect detection is mainly divided into two parts: image pre-processing and defect detection. Image pre-processing includes algorithms such as image denoising and image segmentation, which is the preliminary work of defect detection. The defect detection part mainly uses the image feature extraction algorithm to complete the detection of defects, and its algorithm process can be summarized as follows: 1) select the region of interest and select the region that may contain objects; 2) feature extraction of the region that may contain objects; 3) detection and classification of the extracted features. However, the traditional target detection algorithm based on manually extracted features has the following three main drawbacks: 1) the recognition effect is not good enough, the accuracy rate is not high, and may produce multiple correctly identified results; 2) the computation is large and the operation speed is slow; 3) the extracted features are often effective only for a specific defect type, so it is only suitable for products with clear contours and single defects, and not for products with complex backgrounds.

In recent years, with the development of deep learning and its wide application in various scenarios, deep learning-based target detection algorithms have been gradually applied to surface defect detection tasks[5] to improve production and detection efficiency. Current target detection algorithms automatically extract high-level features mainly by introducing convolutional neural networks[6]. According to whether the detection process contains candidate region recommendations, it is mainly divided into two stage algorithms based on candidate regions[7] and one stage algorithms based on regression ideas[8]. The typical algorithms of the former include R-CNN series, which has the advantage of higher detection accuracy; the typical algorithms of the latter include YOLO series, SSD, etc., which has the advantage of faster detection speed. Faster RCNN is one of the optimal detection algorithms with a good balance of detection accuracy and detection speed. Therefore, the Faster RCNN is selected as the base network for steel surface defect detection.

However, the Faster RCNN-based steel surface defect detection has problems such as small defects are difficult to detect and small defect information is lost during feature extraction. In order to detect steel surface defect more accurately and quickly, this paper makes the following improvements on the basis of Faster RCNN algorithm: Add multi-scale feature fusion module to fuse high and low-level features to improve the detection of small targets.

ISSN: 2688-8645

2. Model Design

2.1. Faster RCNN Basics

The core idea of the Faster RCNN [7] algorithm is to abandon the time-consuming selective search algorithm and use the Region Proposal Network(RPN) to obtain the target candidate frames automatically. The feature extraction, candidate region generation, border regression, and classification are all integrated in one network, which is greatly improved in terms of accuracy and speed, and its overall structure is shown in Figure 1.

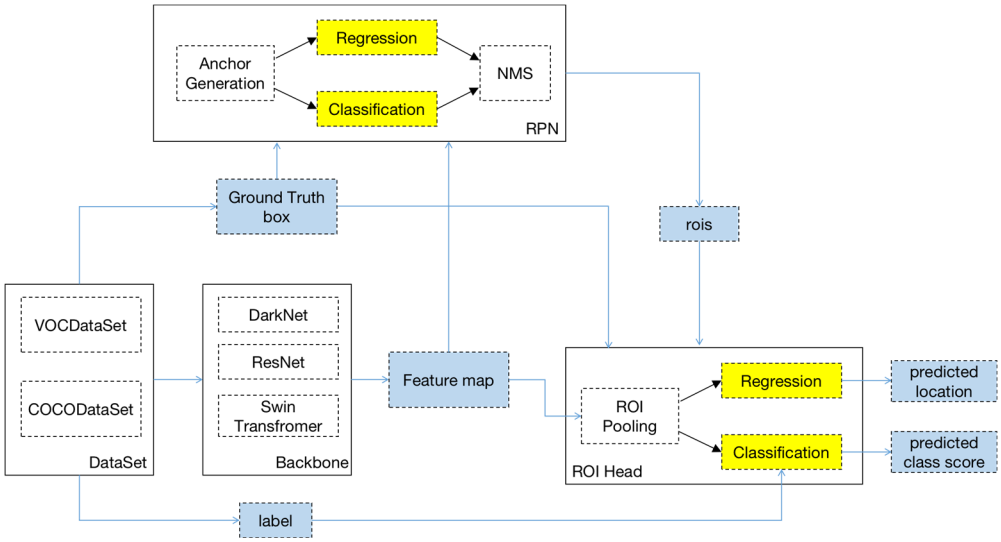

Figure 1. Architecture of Faster RCNN

The overall process of Faster RCNN is as follow:

1) Extraction of features: read in the original image and extract the image features through ResNet or other backbone networks.

2) Generating candidate regions: Using the extracted image features, the RPN network is used to obtain a certain number of ROIs (regions of interests). the RPN network essentially uses a convolutional neural network to generate candidate regions instead of selective search, and the candidate frame matrix of the target to be detected and its score can be output by inputting image information of any size. The mapping of candidate frames on the original image is called anchor.

3) Target classification and position regression: The region of interest and image features are input to the ROI header, and these regions of interest are classified to determine what category they belong to, while the position is fine-tuned. Edge regression is used to regress the position of anchor to a position closer to the GT (groundtruth) box. Given the original anchor as $A = (A_x, A_y, A_w, A_h)$, the $GT = (G_x, G_y, G_w, G_h)$, when the difference between the two is small, it is approximated as a linear transformation and a linear regression model is used to fine-tune the regression detection box to achieve close to the true value. The corresponding coordinate parameter regression is given by

$$\begin{aligned} t_x &= (x - x_a)/w_a, t_y = (y - y_a)/h_a \\ t_w &= \log(w - w_a), t_h = \log(h/h_a) \\ t_x^* &= (x^* - x_a)/w_a, t_y^* = (y^* - y_a)/h_a \\ t_w^* &= \log(w^* - w_a), t_h^* = \log(h^*/h_a) \end{aligned} \tag{1}$$

ISSN: 2688-8645

In Eq. (1). x, y, w, h denotes the horizontal and vertical coordinates as well as the width and height values of the predicted target bounding box, respectively. x_a, y_a, w_a, h_a denotes the horizontal and vertical coordinates as well as the width and height values of the candidate target proposal box, respectively. x^*, y^*, w^*, h^* denotes the horizontal and vertical coordinates and the width and height of the real target bounding box, respectively. $(t_x, t_y), (t_x^*, t_y^*)$ is the corresponding translation factor; and $(t_w, t_h), (t_w^*, t_h^*)$ is the corresponding scaling factor. By equation (1), the anchor box is regressed to the closest GT box as the prediction box.

2.2. Backbone Network

The classical network ResNet(Residual Networks) was proposed by Kaiming He et al. in 2015 in a paper titled "Deep Residual Learning for Image Recognition" [9]. ResNet is to solve the problem of "degradation" in deep neural networks, i.e., the use of shallow layers directly stacked into deep networks, not only makes it difficult to exploit the powerful feature extraction ability of deep networks, but also decreases the accuracy rate, which is not caused by overfitting.

ResNet is built from Residual Building Block, whose structure is shown in Figure 2: two mappings are proposed: identity mapping, which refers to the curve labeled x on the right side, and residual mapping, where residual refers to the F(x) part. The final output is F(x)+x. F(x)+x can be implemented by a feedforward neural network with shortcut connections. shortcut connections are connections that skip one or more layers. The weight layer in the figure refers to the convolution operation. If the network has reached the optimum and continues to deepen the network, the residual mapping will become 0, leaving only the identity mapping, so that the network will theoretically remain in the optimum state and the performance of the network will not decrease with increasing depth.

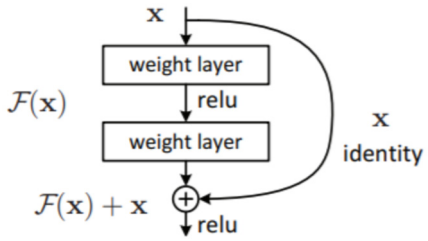

Figure 2. The Structure of Residual Building Block

The residual block consists of multiple cascaded convolutional layers and a shortcut connections. The output of the residual block is obtained by accumulating the output values of the two and passing them through the ReLU activation layer. Multiple residual blocks can be concatenated together to achieve a deeper network.

The residual blocks are designed in two ways, as shown in Figure 3: the left figure is for shallow networks, such as ResNet-18/34; the right figure is for deeper networks, also known as "bottleneck" building blocks, such as ResNet-50/101/152, and the purpose of using this approach is to reduce the number of parameters. The purpose of using this approach is to reduce the number of parameters.

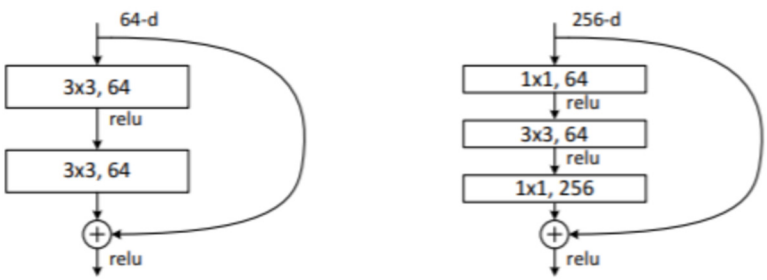

Figure 3. Two Types of Residual Building Block

ISSN: 2688-8645

The left residual structure is called BasicBlock and the right residual structure is called Bottleneck. the following demonstrates the analysis of Bottleneck:

1) The role of the 1× 1 convolution kernel in the first layer is to perform a dimensionality reduction operation on the feature matrix, which reduces the depth of the feature matrix from 256 to 64; The 1× 1 convolution kernel in the third layer is a boosting operation of the feature matrix, which boosts the depth of the feature matrix from 64 to 256.

The main reason for reducing the depth of the feature matrix is to reduce the number of parameters. If the BasicBlock is used, the number of parameters should be: 256× 256× 3×3× 2= 1179648. Using Bottleneck, the number of parameters is: 1×1 ×256 ×64 +3×3 ×64× 64+1 ×1× 256 ×64=69632.

2) First descend then ascend in order to have the same shape of the eigenmatrix output on the main branch and the eigenmatrix output on the shortcut branch for the addition operation.

2.3. Multi-scale Feature Fusion

In the Faster RCNN algorithm, because the shallow network has few convolution operations, the image has high resolution and more location detail information, but contains weak semantic information and strong noise. With deeper network layers and more convolutional operations, the deeper network features have stronger semantic information, but the image resolution is reduced and the perception of details is poorer. Especially for small defects, not only the location information is lost in the deep neural network, but also the semantic information is blurred. Therefore, how to fuse the high and low layer features is an important means to improve the model performance.

FPN is an applicable multi-scale target detection algorithm proposed by Kaiming He and other authors[10]. Most of the original target detection algorithms (e.g., Faster RCNN) only use top-level features for prediction, but we know that the low-level features have less semantic information, but the target location is accurate; the high-level features have more semantic information, but the target location is coarse. FPN is based on the pyramid hierarchy inherent in CNN, and the top-down path is constructed by skip The FPN is based on the pyramid hierarchy inherent in CNN, and the top-down path is constructed by skip connection, and only a small cost is required to generate the feature pyramid, and each scale of the feature pyramid has high-level semantic feature, and finally the target is detected at each level of the feature pyramid. The structure of FPN is shown in Figure 4.

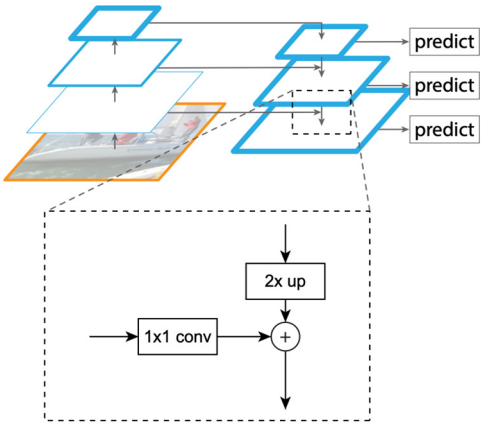

Figure 4. The Structure of FPN Network

2.4. Loss Function

The total loss function of Faster RCNN consists of classification loss and regression loss. The specific loss function of this method for an image is defined as shown in equation (2).

ISSN: 2688-8645

$$L(\{p_i\},\{t_i\}) = \frac{1}{N_{\text{cls}}}\sum_i L_{\text{cls}}(p_i, p_i^*) + a\frac{1}{N_{\text{reg}}}\sum_i P_i^* L_{\text{reg}}(t_i, t_i^*) \quad (2)$$

where p_i is the predicted classification probability of the ith anchor, when it is a positive sample $p_i^* = 1$; when it is a negative sample, the $p_i^* = 0$. t_i is the 4 parametric coordinates of the predicted bounding box of the ith anchor, and t_i^* is the 4 parametric coordinates of its corresponding Ground Truth. N_{cls} is the minimum batch size in the training process. N_{reg} is the number of anchor positions and L_{reg} is the position regression loss function based on the smooth function, and L_{cls} is the classification loss function implemented by the aforementioned focal loss.

3. Algorithm Implementation Framework

3.1. Pytorch

PyTorch [11] is an open source deep learning framework that provides a variety of tensor operations and can automatically perform gradient calculations through automatic derivation, facilitating the construction of various dynamic neural networks. In January 2017, PyTorch was launched by the Facebook Artificial Intelligence Research Institute(FAIR) based on Torch. Torch is a scientific computing framework supported by a large number of machine learning algorithms, and is a tensor manipulation library similar to Numpy, which is characterized by being particularly flexible, but not very popular because it uses a niche programming language is Lua, which led to the emergence of PyTorch. So actually Torch is the predecessor of PyTorch, they have the same underlying language, but just use a different top-level wrapper language.

Pytorch is a Python-based renewable computing package that provides two advanced features: (1) tensor computation with powerful GPU acceleration(such as NumPy).(PyTorch can be seen as both a numpy with GPU support and a powerful deep neural network with automatic derivation. In addition to Facebook, it has been adopted by organizations such as Twitter, CMU, and Salesforce.

3.2. Mmdetection

Mmdetection[12] is a PyTorch-based object detection library open-sourced by the Chinese University of Hong Kong-Shangtang Technology Joint Lab, which provides a variety of publicly available core modules for visual detection, through the combination of which a variety of well-known detection frameworks can be quickly built.q Its main features include:(1) modular design: by combining different components, it is easy to build custom object detection frameworks.(2) Support for multiple frameworks out of the box: the toolkit directly supports popular detection frameworks, such as Faster RCNN, Mask RCNN, RetinaNet, etc.(3) Efficient: All basic bbox and mask operations now run on GPU. The training speed is about 5%~20% faster than Detectron on different models.(4) State-of-the-art: refactored from the code base of MMDet team, which won the COCO Detection 2018 challenge.

Currently, mmdetection has been an important component project of OpenMMLab, which is the most complete open source algorithm system for computer vision in the area of deep learning. Since its open source in 2018, more than 15 algorithm libraries have been released, covering many algorithm fields such as classification, detection, segmentation, and video understanding, with more than 250 algorithm implementations and 2000 pre-trained models. All tasks and all algorithms are developed based on a unified underlying architecture with abstraction of training interfaces, so that users can quickly implement new ideas based on OpenMMLab: algorithm research or business algorithm development in corresponding fields based on existing algorithm libraries, or add their own new algorithm tasks based on OpenMMLab's unified architectural design.

ISSN: 2688-8645

4. Experiments and Results Analysis

4.1. Experimental Environment

The experimental running environment of this study is listed as follow:

1) Intel Xeon Silver 4210R CPU;
2) 64G memory;
3) NVIDIA RTX3090 24G GPU;
4) Ubuntu 20.04 64-bit OS.

The model is built based on Pytorch deep learning framework and mmdetection framework.

4.2. Dataset

The dataset used in this study is NEU-DET [13], which contains 1800 grayscale images of steel surface defects, 300 images of each defect type, each with a resolution of 200×200. The dataset contains six major types of defects commonly found on steel surfaces, namely: crazing, inclusion, patches, pitted surface, rolled-in scale and scratches, and the sample images are shown in Figure X. The training validation set and the test set were randomly divided according to the ratio of 8:2. In the training validation set, 75% of the images are used for training and 25% for validation. Finally, the original dataset is made into PASCAL VOC dataset format for training, and the location and class of defects in each image are marked using a file in XML format.

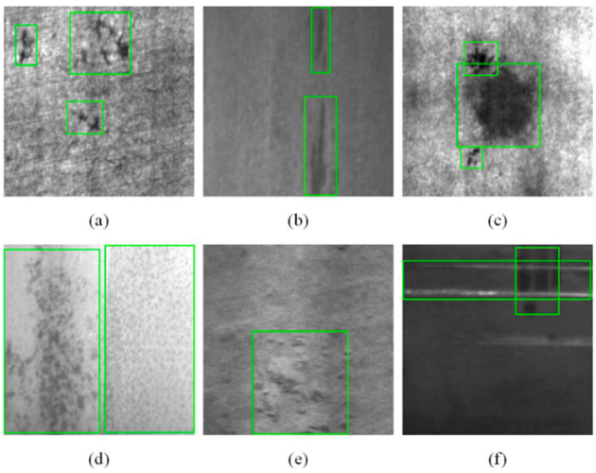

Figure 5. Examples of defect images in NEU:(a) crazing, (b) inclusion, (c) patches, (d) pitted surface, (e) rolled-in scale, and(f) scratches

4.3. Training Process

With the built-in data enhancement method of mmdetection framework, the original image is rotated, cropped, blurred and color adjusted. The model is trained with a batch size of 2, an AdamW optimizer, a weight decay of 0.05, and an initial learning rate of 0.0001, and the learning rate is updated using the StepLR mechanism.

4.4. Evaluation Indicators

In order to quantitatively analyze the experimental results and verify the effectiveness of the proposed method, precision, recall, AP (average precision), mAP (mean average precision), and FPS (frame per second) are selected as evaluation indexes in this paper. Among them, recall indicates the proportion of correct prediction in the prediction target to the total prediction samples, and precision indicates the proportion of correct prediction in the prediction target of a certain category to the total correct samples, which are defined by the following formulas.

ISSN: 2688-8645

$$\text{recall} = \frac{\text{TP}}{\text{TP+FN}} \tag{3}$$

$$\text{precision} = \frac{\text{TP}}{\text{TP+FP}} \tag{4}$$

TP(True Positive) means the actual sample is positive and the prediction is also positive; FN(False Negative) means the actual sample is positive and the prediction is negative; FP(False Positive) means the actual sample is negative and the prediction is positive. And the positive and negative samples are divided according to the threshold value of IoU, and the intersection ratio of actual and predicted edges is used to determine whether the successful prediction is a positive sample, when IoU>0.7 is a positive sample, otherwise it is a negative sample. Therefore, the recall rate represents the proportion of the number of correctly predicted positive samples to all the predicted positive samples, and the accuracy rate represents the proportion of the number of correctly predicted positive samples to all the positive samples.

Since precision and recall are a pair of contradictory indicators, it is common to use the index enclosed by recall and recall. The area of the formed curve, i.e., AP, indicates the combined model performance under different thresholds. And mAP is the mean value of AP for all categories.

4.5. Analysis of Experimental Results

The AP as well as mAP results for the six surface defects in the NEU-DET dataset are shown in Table 1, and the performance of the final model is shown in Table 2.

From Table 1, it can be seen that the Faster RCNN algorithm based on feature fusion has different detection capabilities for different defect types. patches and scratches types have relatively high APs, indicating easy detection, while crazing types have low APs, indicating difficult detection.

Table 1. AP of six surface defects and mAP of our model

patches	crazing	rolled-in scale	scratches	inclusion	pitted surface	mAP
0.859	0.362	0.608	0.845	0.770	0.807	0.708

As can be seen from Table 2, the Faster RCNN algorithm based on feature fusion mAP reaches 0.708. the weight file size is 473 MB, which is a deep learning model of average size. In terms of model execution speed, the algorithm reaches 43.3 frame/s, which can meet the application of industrial scenarios.

Table 2. Performance of our model

recall/%	mAP/%	FPS/(frame/s)	weight file size/MB
0.871	0.708	43.3	473

5. Conclusion

Steel surface defects are characterized by large size variations and many morphological variations, etc. In this paper, a Faster RCNN algorithm based on ResNet50 as backbone is applied in the steel surface defect detection task. The algorithm utilizes the feature learning capability of ResNet50 combined with FPN to perform multi-scale feature fusion on its multiple output feature maps. The effectiveness of the algorithm is verified through experiments on the NEU-DET dataset. Although the detection capability varies for different defect types, where

ISSN: 2688-8645

individual classes are difficult to detect, the overall mAP reaches 0.708. In addition, the model execution speed reaches 43.3 frame/s, which satisfies the application in industrial scenarios. This study explores the application of Faster RCNN in the field of surface defect detection, and subsequent attempts can be made to use more advanced backbone or other techniques in the field of target detection to improve the overall detection accuracy of the algorithm.

Acknowledgments

This work is supported by the Fundamental Scientific Research Projects of Wenzhou (G2021 0041).

References

[1] Tang Bo, Kong Jianyi, Wu Shiqian. Review of surface defect detection based on machine vision [J]. Journal of Image and Graphics, 2017, 22(12): 1640–1663.

[2] Tao X, Zhang D, Ma W, et al. Automatic metallic surface defect detection and recognition with convolutional neural networks[J]. Applied Sciences, MDPI, 2018, 8(9): 1575.

[3] Chen Y, Ding Y, Zhao F, et al. Surface defect detection methods for industrial products: A review[J]. Applied Sciences, MDPI, 2021, 11(16): 7657.

[4] O'Mahony N, Campbell S, Carvalho A, et al. Deep learning vs. traditional computer vision[A]. Science and information conference[C]. Springer, 2019: 128–144.

[5] Zhou X, Wang Y, Zhu Q, et al. A Surface Defect Detection Framework for Glass Bottle Bottom Using Visual Attention Model and Wavelet Transform[J]. IEEE Transactions on Industrial Informatics, 2020, 16(4): 2189–2201.

[6] Li Xudong, Ye Mao, Li Tao. Review of object detection based on convolutional neural networks[J]. Application Research of Computers, 2017, 34(10): 2881-2886+2891.

[7] Ren S, He K, Girshick R, et al. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017, 39(6): 1137–1149.

[8] Redmon J, Farhadi A. YOLOv3: An Incremental Improvement[J]. arXiv, 2018.

[9] He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[A]. Proceedings of the IEEE conference on computer vision and pattern recognition[C]. 2016: 770–778.

[10] Lin T-Y, Dollar P, Girshick R, et al. Feature Pyramid Networks for Object Detection[A]. 2017 IEEE Conference on Computer Vision and Pattern Recognition(CVPR)[C]. Honolulu, HI: IEEE, 2017: 936–944.

[11] PyTorch[EB/OL]. Baidu Baike. /2022-10-11. https:// baike.baidu. com/item/ PyTorch/ 24269838? fr= aladdin.

[12] MMDetection Contributors. OpenMMLab Detection Toolbox and Benchmark[J]. 2018.

[13] He Y, Song K, Meng Q, et al. An End-to-End Steel Surface Defect Detection Approach via Fusing Multiple Hierarchical Features[J]. IEEE Transactions on Instrumentation and Measurement, 2020, 69 (4): 1493–1504.