Surface defect detection of steel strips based on classification priority YOLOv3-dense network

Jiaqiao Zhang[a,b], Xin Kang[a], Hongjun Ni[b] and Fuji Ren[a]

[a] Faculty of Engineering, Tokushima University, Tokushima, Japan; [b] College of Mechanical Engineering, Nantong University, Nantong, People's Republic of China.

Abstract: The steel strip is one of the essential raw materials in the machinery industry. Besides, the defects on the surface of the steel strip directly determine its performance. To achieve rapid and effective detection of surface defects on steel strips, a CP-YOLOv3-dense (classification priority YOLOv3 DenseNet) deep convolutional neural network was proposed in the present study. The model used YOLOv3 as the basic network, implemented priority classification on the target images, and then replaced the two residual network modules in the YOLOv3 network with two dense network modules. Therefore, the model can receive the multi-layer convolution features output by the dense connection block before making predictions, consequently enhancing the reuse and fusion of features. Finally, the six kinds of surface defects of steel strips were detected by the improved network model, and the results were compared with other deep learning networks. According to the results, the recognition precision of the CP-YOLOv3-dense network model is 85.7%, the recall rate is 82.3%, the mean average precision is 82.73%, and the detection time of each image is 9.68ms. The mean average precision is 6.65% higher than the original YOLO network and 10.6% higher than the DNN network. In addition, the detection speed is 1.77 times faster than the Faster RCNN network. The proposed CP-YOLOv3-dense network has stronger robustness and higher detection precision, which can be used for the identification of various steel strip surface defects.

Keywords: steel strip; defect detection; deep learning; neural network; surface technology

---

1. Introduction

Steel strip is an indispensable raw material in the machinery industry, and its quality is a key indicator determining its price. Due to the limitations of equipment and process conditions, the surface of the steel strip will inevitably have different forms and types of defects, and the size, number and distribution of the defects are various [1, 2]. For the diversity and complexity of steel strip surface defects, steel production companies in various countries attach great importance to surface quality inspection, and spend huge sums of money improving the level of detection technology.

Traditional steel strip surface defect detection methods are mostly manual inspections, and the defects are used to classify and locate through the eyes and experience of workers [3]. The proposed method has poor real-time performance with high false detection rate. Even for the most highly trained and experienced workers, under their best working conditions, the detection rate of metal surface defects is only approximately 80%. Recently, with the development of machine learning, numerous scholars have applied this technology to different fields, including industrial inspection [4-6].

A lot of scholars have proposed "deep learning + defect detection" methods, which have been applied to the classification and detection of surface defects on steel strips, having achieved satisfying results. Qiwu Luo et al. employed the selectively dominant local binary patterns (SDLBPs) algorithm to classify the surface defects of hot-rolled strips so as to obtain higher classification accuracy and time efficiency, yet they failed to achieve target defect detection [7]. X.L. Zhang and other scholars proposed a sinusoidal phase grating projection method to detect the depth and surface profile of cracks in continuous slab casting, which is suitable for the detection of defects on the surface of the slab [8]. Additionally, the YOLO network has also been used to detect the surface defects of steel strips. However, the image datasets used in most studies are relatively simple. The network model is directly applied instead of improving the network based on the actual defects of the steel strips, causing that the applicability of the YOLO network is low [9]. In addition, some other deep learning network models have also been used in the field of steel strip surface defect detection, such as CNN [10-13], Pyramid Feature Fusion and Global Context Attention Network (PGA-Net) [14], and semi-supervised convolutional neural network [15, 16], generating a certain effect.

In the present study, we take the images in the NEU-DET Dataset as the research object, and propose a CP-YOLOv3-dense deep convolutional neural network based on the characteristics of defects in the images. The innovation of our work lies in the following aspects. First, an improved YOLO network model is proposed, which uses dense network modules instead of residual network modules to enhance the multiplexing and fusion of features. Secondly, according to the characteristics of defect images in the database, the principle of classification priority is proposed, which not only solves the problem of a small number of training images, but also avoids the prediction of defect categories during the detection process, thereby improving the detection accuracy. Finally, a dense labeling method suitable for small surface defects of steel strips is proposed through comparative experiments. The experiment results demonstrate that the network based on CP-YOLOv3-dense is superior to the original YOLOv3 and other network in terms of precision and speed in detecting surface defects on steel strips.

The rest of this paper is organized as follows. Section II summarizes the related works. Section III introduces the methods and principles involved in the experiment. The experimental results and some related discussions are presented in Section IV. Finally, Section V concludes our paper.

2. Related Works

2.1. Image Target Detection based on Deep Learning

Computer-based image processing consists of three levels, respectively, classification, detection and segmentation. The task of target detection is to find all the targets of interest in the image and obtain the category information and location information of this target. Because various types of objects have different appearances, shapes, and postures together with the interference of lighting, occlusion and other factors during imaging, target detection has always been the most challenging problem in the field of machine vision.

The traditional target detection method uses a sliding window frame to decompose a picture into millions of sub-windows at different positions and different scales. For each window, a classifier is used to determine whether the target object is included, and a specific method needs to be designed according to the characteristics of the target to be detected. For example, Harr feature and Adaboosting classifier are used for face detection [17, 18]. HOG (histogram of gradients) and Support Vector Machine are used for pedestrian detection [19-21]. These methods have poor versatility with low detection accuracy and speed.

The deep learning model has gradually become a hot research direction for image target detection due to its powerful representation ability, coupled with the accumulation of data volume and the progress of computing power [22]. Initially, image target detection based on deep learning has an accuracy that cannot be achieved by traditional methods, greatly improving the accuracy of the detection results and enabling target detection to be put into practical applications. Secondly, the deep learning algorithm is extremely versatile. The same algorithm model can be used to detect multiple targets, and the features obtained by deep learning have a very strong migration ability, which greatly broadens the detection range of the model [23]. From the classic CNN convolutional neural network to the more state-of-the-art object detectors, such as RFBNet [24], CenterNet 25] and CornerNet [26], the detection accuracy and speed of deep learning models continue to increase.

Currently, image target detection based on deep learning has been applied to all walks of life. Face detection is more common in people [27], which can be used not only for smartphones and mobile payments, but also for tracking fugitives [28]. Additionally, target detection can monitor the growth of crops in the agricultural field and detect diseases and insect pests in time [29, 30]. In the industrial field, it can be used to detect surface defects and equipment abnormalities [31]. In the medical field, it can be used for medical diagnosis [32, 33]. In the commercial field, it can be used for coin identification, invoice testing [34], etc.

2.2. Application of Deep Learning in Surface Defect Detection of Steel Strips

In recent years, the deep learning algorithms represented by convolutional neural networks have been extensively used in the field of industrial defect detection, such as the detection of surface defects on glass [35, 36] and cloth [37, 38]. Similarly, the deep learning has gradually replaced traditional machine vision methods, which has become the mainstream algorithm for surface defect detection of steel strips. Based on the different structures of deep learning models, they can be divided into two categories, respectively, two-stage detection algorithm and one-stage detection algorithm. Figure 1 presents a comparison of the structure of the two detection algorithms.

The two-stage detection algorithm divides the detection problem into two stages, which first generates region proposals, and then classifies the region proposals (generally, location refinement is also required). The typical representative of this type of algorithm is the R-CNN algorithm, such as R-CNN, Fast R-CNN and Faster R-CNN. At present, scholars have conducted a lot of research on the detection of surface defects of steel strips based on the two-stage detection algorithm, having achieved considerable results. Qirui Ren et al. [10] used the Faster R-CNN model to detect the surface defects of the steel strips, and made certain improvements to the Faster R-CNN model. First, the convolutional layer used for feature extraction in Faster R-CNN is replaced by deep separable convolution. Then, the center loss is added to the original loss function, thereby increasing the network operating speed and improving the ability of distinguishing different defects. When Weiyang Lin [39] et al. conducted the surface defect detection of hot-rolled steel, the feature maps were generated by RCNN model based on ResNet 50. Their experimental results demonstrated that the detection method based on deep learning is more effective than the traditional method and can detect the surface defects of the steel strips more accurately. Kangyu Li [40] and Rubo Wei [41] employed improved Faster R-CNN to detect surface defects on steel strips, and improved detection accuracy by adopting multi-scale feature fusion or introducing weighted regions of interest.

The one-stage detection algorithm does not require the region proposal stage, directly generating the category probability and position coordinate value of the object. Therefore, the detection speed of one-stage detection algorithm is faster, and the typical algorithms include YOLO and SSD. Due to the relatively short development time of the one-stage detection algorithm, few related studies have used it to detect surface defects on steel strips, so it is still in the preliminary exploration stage. Jiangyun Li [9] earlier attempted to use the YOLO network model for surface defect detection of steel strips, and the YOLO network used was composed of 27 convolutional layers to achieve end-to-end surface defect detection of steel strips. Renjie Tang [42] et al. employed two detection algorithms, respectively, Faster R-CNN and YOLO, to merge type-related variables into the Generator, and then proposed a GANs model for steel strips defect detection. However, they did not conduct in-depth research on the role of the YOLO network. Reference [39] gave up this type of algorithm directly, because YOLO and SSD networks are challenging in detecting small defects. Therefore, on the premise of ensuring the detection accuracy, it is urgently necessary to carry out related research on the one-stage detection algorithm to further improve the detection speed of the surface defects of the steel strip.

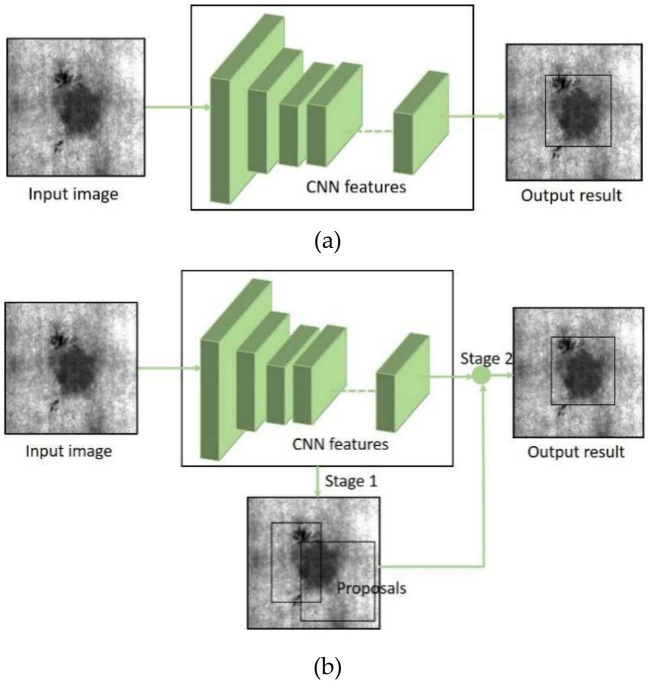


Figure 1. Comparison of the structure of the two detection algorithms; (a) One-stage detection algorithm; (b) Two-stage detection algorithm.

3. Methodology

3.1. The Classification Priority YOLOv3 Network

The YOLO network model was first proposed in 2016 [43, 44]. Its later version, YOLOv3 [45], is not only faster in detection, but also is more suitable for small target detection. The YOLO network covers twenty-four convolutional layers, four maximum pooling layers, and two fully connected layers. The convolutional layer is used to extract image features, the maximum pooling layer is adopted to reduce image pixels, and the fully connected layer is employed to predict image categories and locations. YOLO uses the features of the entire image to predict the bounding box and classify the targets within the box, indicating that the YOLO network can use the full information existing in the image to achieve target defect classification and position detection.

Figure 2 shows a YOLO network model for surface defect detection of steel strip. Obviously, the input image of this model is divided into S × S grids. If a target object falls into one of the grids in the image, the grid is responsible for predicting this object. Each grid predicts B bounding boxes, and each predicted bounding box contains 5 parameters (x, y, w, h, confidence). (x, y) is the coordinate of the bounding box relative to the center of the grid cell boundary, (w, h) represents the length and width of the bounding box relative to the entire image, and the confidence denotes the confidence score of each bounding box. The confidence score reflects the probability that the bounding box contains the target defect and the case where the bounding box coincides with the ground truth box (intersection-over-union, IoU). That is, the confidence includes two parts: one is the probability Pr(object) of whether the grid contains the target object, and the other is the accuracy of the bounding box (IoU). If there is a target defect in the bounding box, Pr(object) = 1, and the confidence score is equal to the IoU value. If there is no target object in the bounding box, Pr(object) = 0, and the confidence score is 0. IoU is the ratio of the intersection and union of the bounding box and the ground truth box. The calculation formula is:

$$\mathrm{IoU} = \frac{b_{\mathrm{gt}} \cap b_{\mathrm{bd}}}{b_{\mathrm{gt}} \cup b_{\mathrm{bd}}} \tag{1}$$

where, b_{gt} represents the ground truth box and b_{db} represents the bounding box.

When there are C defects in the image, the conditional probability of the C classes is $Pr(Class_i/object)$, which indicates the probability that the grid contains the target object and the object belongs to the *i*-th class object. The calculation formula is presented as follows:

$$\mathrm{Pr}\,(\mathrm{Class}_i/\mathrm{Objet}) * \mathrm{Pr}\,(\mathrm{Object}) * \mathrm{IoU}_{\mathrm{pred}}^{\mathrm{truth}} = \mathrm{Pr}\,(\mathrm{Class}_i) * \mathrm{IoU}_{\mathrm{pred}}^{\mathrm{truth}} \tag{2}$$

Due to the particularity that there is only one type of defect in a single image, we propose a classification priority YOLOv3 network model. Based on the principle of classification priority, we first classify the surface defects of the steel strips. The image dataset used in our experiment contains a total of 1,800 images and thus it is not extremely large. If we train the classification model from scratch, it often gets poor results and takes considerable time. Therefore, we first pre-train the convolutional network (ConvNet) on the Imagenet dataset, and then replace and retrain the classifier on the newly constructed steel strip surface defect dataset, and fine-tune the weight of the pre-trained network by continuing backpropagation [46]. Since the dataset used in our experiment is small and different from the original ImageNet dataset, we chose to train a linear classifier. The best model was saved for the classification of steel strip surface defects. Therefore, through the principle of classification priority, the prediction of the defect category can be omitted. Thus, the defect probability calculation can be corrected as:

$$\mathrm{Pr}\,(\mathrm{Class}_i/\mathrm{Object}) * \mathrm{Pr}\,(\mathrm{Object}) * \mathrm{IoU}_{\mathrm{pred}}^{\mathrm{truth}} = \mathrm{IoU}_{\mathrm{pred}}^{\mathrm{truth}} \tag{3}$$

Compared with the traditional YOLO network, since the defect categories have been prioritized, the YOLO network does not need to predict the probability $Pr(Class_i)$ of the defect category. In addition, the value of $Pr(Class_i)$ belongs to [0, 1]. Thus, the defect detection accuracy of the improved YOLO model is higher.

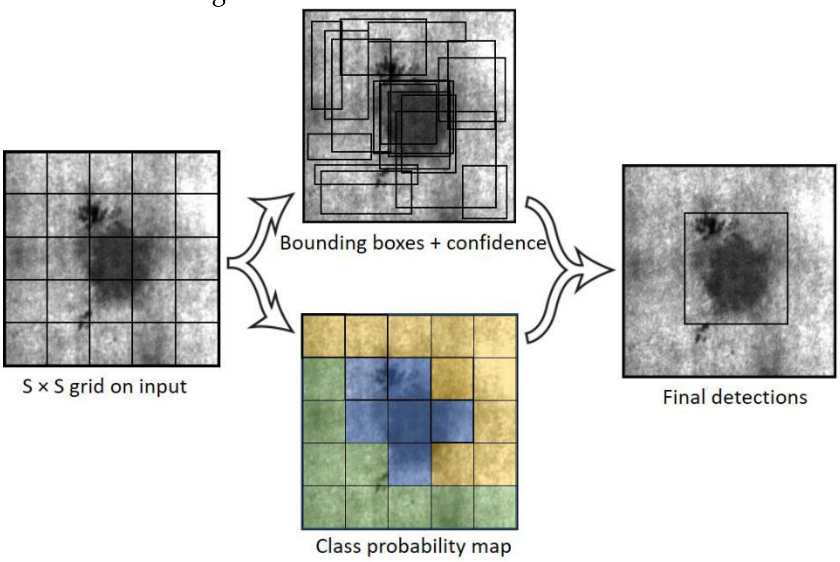


Figure 2. The YOLO network model for surface defect detection of steel strips.

After obtaining the confidence score of each bounding box, we set a threshold to remove the bounding boxes with low scores, and perform NMS (non-maximum suppression) processing on the remaining bounding boxes with high scores so as to achieve the final detection result.

In our experiment, each picture is divided into 7 × 7 grids, and each grid will predict 6 bounding boxes. After priority classification, there are only one type of defects in the image. Therefore, our final prediction is a S × S × (B × 5 + C) = 7 × 7 × (6 × 5 + 1) tensor.

3.2. *Design of CP-YOLOv3-Dense Network*

DenseNet was proposed by Gao Huang [47] et al. in 2017. From the perspective of features, through feature reuse and bypass settings, it not only greatly reduces the amount of network parameters, but also alleviates the gradient vanishing problem to a certain extent. Consequently, we combine the YOLO network with DenseNet to propose a new type of CP-YOLOv3-dense network structure. The DenseNet network structure contains 3 dense convolutional blocks, where each dense convolutional block contains 4 convolutional layers. In each dense convolutional block, each convolutional layer can obtain the output of all previous convolutional layers as input. Besides, adjacent convolutional layers are connected by a convolutional layer and a pooling layer. In the dense convolutional blocks:

$$x_n = \mathrm{H}_n([x_0, x_1, \cdots, x_{\mathrm{n}-1}]) \quad (n = 1, 2, 3, 4) \tag{4}$$

where, x_0 denotes the input feature map of the module, x_n represents the output of the n-th layer, $[\mathrm{x}_0, \mathrm{x}_1, \cdots, \mathrm{x}_{\mathrm{n}-1}]$ stands for the stitching of $\mathrm{x}_0, \mathrm{x}_1, \cdots, \mathrm{x}_{\mathrm{n}-1}$ and Hn is a function for processing stitched feature maps. H( ) indicates the connection between BN-ReLU-Conv (1, 1) and BN-ReLU-Conv (3, 3).

Figure 3 presents the CP-YOLOv3-dense network structure proposed in the present study for the detection of surface defects of steel strips, and the detailed network parameter settings are shown in Figure 4 [48]. The basic network is the YOLOv3 network, and DenseNet is used to replace the original transmission layer with lower resolution. Therefore, the model can receive the multi-layer convolutional features output by densely connected blocks before making predictions, thereby enhancing the reuse and fusion of features.

The first impression of the term "dense connection" is that it greatly increases the amount of network parameters and calculations, but in fact it is not the case. On the contrary, DenseNet is more efficient than other networks. DenseNet reuses image features through dense connections, which reduces the amount of computation on each layer of the network. In addition, DenseNet does not need to re-learn redundant feature maps, and the operation of dimensional stitching brings rich feature information, resulting that many feature maps can be obtained with less convolution. Therefore, DenseNet has much less parameters than ResNet convolutional network. The output of each layer of DenseNet will be superimposed on the input of the next layer. In order to avoid a sudden increase in the number of channels, the number of convolutional output channels of each layer of DenseNet is designed to be very small. Finally, the parameter amount of DenseNet in 40 layers is only 1M. After replacing the last three fully connected layers with global pooling layers, the parameters amount of convolutional network VGG-11 with only 10 layers could reache 9M. Therefore, the improved YOLO network model proposed by us has fewer parameters and lower space complexity. However, due to the channel superposition, the improved YOLO network needs to read memory frequently, which slows the training and prediction speed, resulting in a higher time complexity of the model.

In our experiment, The input images are adjusted to 512 × 512 pixels, and the 32 × 32 and 16 × 16 downsampling layers in the original YOLO network are replaced by DenseNet. For example, in second layer combination of the DenseNet, which replaces the 16 × 16 down-sampling layer, the 640 channel feature maps are spliced by the feature map x_0 and the output feature map x, that is, $[x_0, x_1]$ used as the input of H_2. H_2 performs BN operation and activation function ReLU nonlinear mapping on $[x_0, x_1]$, and uses 256 1 × 1 convolution kernels to generate 256 feature maps. After performing BN and ReLU operations, 128 3 × 3 convolution kernels are used for convolution. Finally, the x_2 with 128 feature maps is output. After that, x_2 and $[x_0, x_1]$ are spliced into 768 channel feature maps $[x_0, x, x_2]$, which are used as the input of H_3. Similarly, H_3 also outputs 128 channel feature maps x_3, and so on.

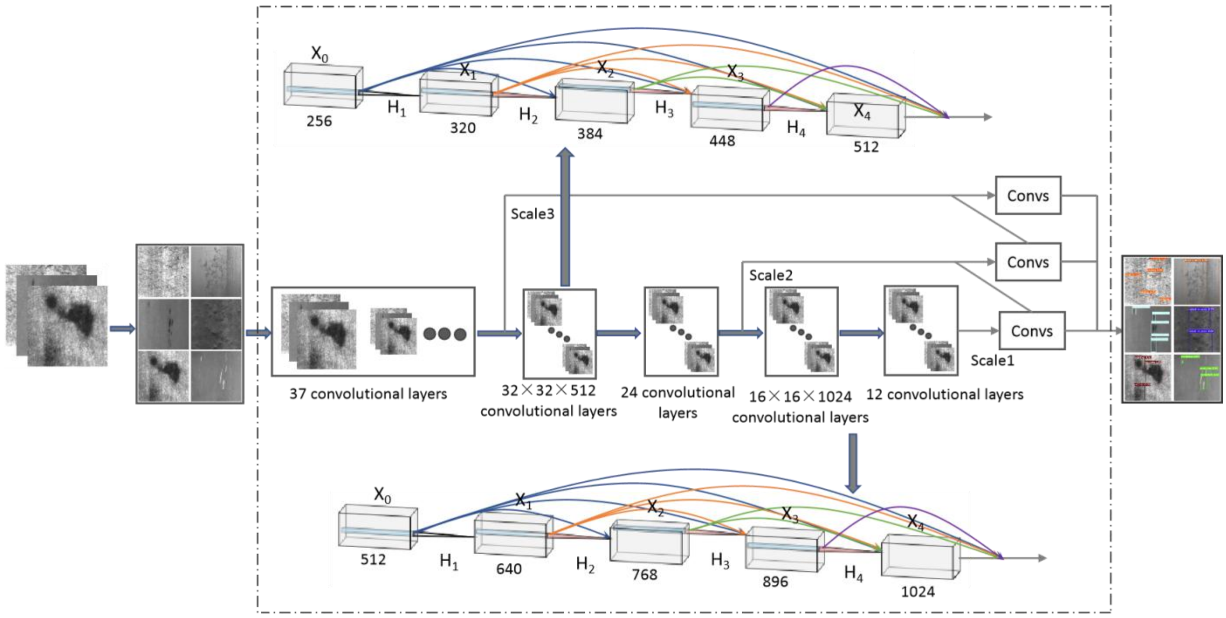


Figure 3. The improved YOLOv3 network structure proposed in this paper.

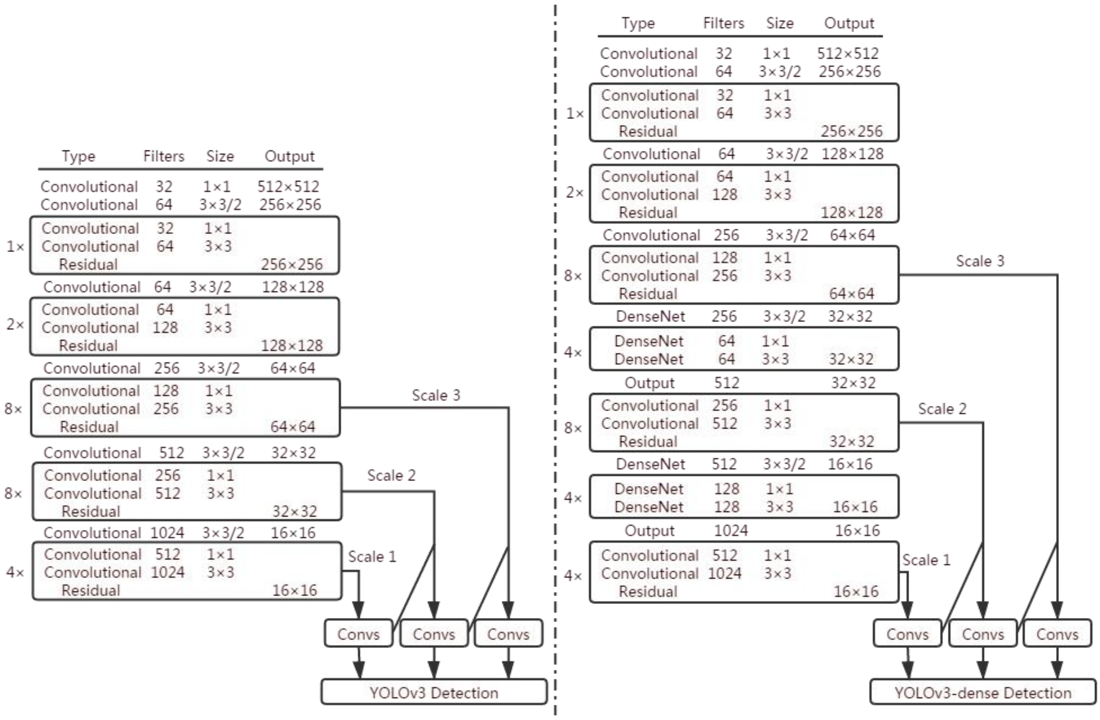


Figure 4. The parameters comparison between the improved and the original YOLOv3 network.

3.3. The Evaluation Indicators of Network Performance

The target defect detection results can be divided into 4 categories, respectively, true positive (TP), false positive (FP), true negative (TN), and false negative (FN) [49-51]. The confusion matrix of the detection results is shown in Table 1.

Table 1. Confusion matrix for evaluation.

Labeled	Predicted	Confusion matrix

Positive	Positive	TP
Positive	Negative	FN
Negative	Positive	FP
Negative	Negative	TN

The calculation formula for precision and recall is as follows:

$$\mathrm{Precision} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{5}$$

$$\mathrm{Recall} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{6}$$

Precision is generally used to evaluate the global accuracy of the model, reflecting the proportion of true positive samples among the predicted positive samples determined by the classifier. The recall rate reflects the proportion of true positive samples among the labeled positive samples.

It is further possible to obtain the parameter F_1 score so as to evaluate the performance of the network model:

$$\mathrm{F_1} = \frac{2 \times \mathrm{P} \times \mathrm{R}}{\mathrm{P} + \mathrm{R}} \tag{7}$$

With the recall as the horizontal axis and the precision as the vertical axis, a precision-recall (P-R) curve can be drawn. Average precision (AP) is the area under the P-R curve. Generally, the better a classifier, the higher the AP score. Mean average precision (mAP) is the mean score of multiple categories of APs, which can be obtained by the following calculation formula：

$$\mathrm{mAP} = \frac{1}{N}\sum_{n=1}^{N} \mathrm{P}_n \cdot \mathrm{R}_n \times 100\% \tag{8}$$

Due to the priority classification, the detection target just has one type of defect, so the AP and mAP are equal in our experiment.

When training the model, the activation function used is the Leaky ReLU function. Compared with the traditional ReLU function, the first half of the Leaky ReLU function is set to 0.01x instead of 0, which not only inherits the advantages of the ReLU function, but also does not cause Dead ReLU problems (Dead ReLU problem means that some neurons may never be activated, and the corresponding parameters can never be updated). The specific function is expressed as follows:

$$\phi(x) = \begin{cases} x, & \text{if } x > 0 \\ 0.1x, & \text{otherwise} \end{cases} \tag{9}$$

where, x represents the output of the convolution layer.

The detection model uses the sum of mean square error as a loss function to optimize model parameters, that is, the sum of mean square error of the S × S × (B × 5 + C) dimensional vector output by the network and the corresponding S × S × (B × 5 + C) dimensional vector of the real image. Since the priority classification has been performed, the classification error can be omitted. Finally, the error formula can be expressed as:

$$loss = \sum_{i=0}^{\mathrm{S}^2} \mathrm{coordError} + \mathrm{IoUError} \tag{10}$$

where, coordError indicates the coordinate error between the prediction data and the calibration data, and IoUError denotes the IoU error.

Because different types of errors contribute different values to the loss scores, λcoord = 5 is used to correct the coordError when calculating the loss score. When calculating the IOU error, the contribution of the IoU error to the network loss is different between the bounding box containing the target defect and the bounding box containing no target defect. If the same weight is used, when calculating the network parameter gradient, the confidence score of the bounding box that does not contain the object is approximately 0. Additionally, the influence of the confidence error of the bounding box that contains the object is enlarged in disguise. Therefore, λnooobj = 0.5 is used to correct the IoUError. The revised loss score calculation formula is presented as follows:

$$\begin{aligned} loss = {} & \lambda_{\mathrm{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathrm{I}_{i,j}^{obj} [(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2] \\ & + \lambda_{\mathrm{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathrm{I}_{i,j}^{obj} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right] \\ & + \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathrm{I}_{i,j}^{obj} \left(C_i - \hat{C}_i \right)^2 + \lambda_{noobj} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathrm{I}_{i,j}^{noobj} \left(C_i - \hat{C}_i \right)^2 \end{aligned} \tag{11}$$

Where, $\mathrm{x_i}, \mathrm{y_i}, \mathrm{w_i}, \mathrm{h_i}$ are the parameter values of the ground truth box; $\hat{\mathrm{x}}_\mathrm{i}, \hat{\mathrm{y}}_\mathrm{i}, \hat{\mathrm{w}}_\mathrm{i}, \hat{\mathrm{h}}_\mathrm{i}$ are the parameter values of the bounding box; S is the number of divided meshes; B is the number of bounding boxes predicted for each grid; $\mathrm{I}_{\mathrm{i,j}}^{\mathrm{obj}}$ determines whether the *j*-th bounding box of the *i*-th grid contains the target defect; Ci is the confidence score of the ground truth box of the target defect, $\hat{\mathrm{C}}_\mathrm{i}$ is the confidence score of the bounding box of the target defect; $\mathrm{I}_{\mathrm{i,j}}^{\mathrm{noobj}}$ indicates that the *j*-th bounding box of the *i*-th grid contains no target defects.

In the experiment, we use the average test time of an image to characterize the detection rate of the model. The calculation formula is as follows:

$$t = \frac{\sum_{i=1}^{N} T_i}{N} \tag{12}$$

where, *t* is the average detection time, *T* is the detection time of each image, and *N* is the total number of images to be detected.

The detection time depends on both of the complexity of the neural network and the number of bounding boxes generated in the detection process. The difference in the detection rate of different defects is caused by the unequal number of ground truth boxes.

4. Experiments and Discussions

The configuration of the hardware and software platforms used in our experiment is presented as follows: GPU is NVIDIA Corporation GP102 [TITAN X], operating system is Ubuntu, and the deep learning framework is DarkNet.

4.1. Selection of Image Dataset

The dataset is the foundation of image processing based on deep learning. In the current experiment, we chose an open source dataset whose name is NEU-DET Dataset [52]. This dataset contains 6 types of hot-rolled steel strip defects, including crazing (Cr), inclusion (In), patches (Pa), pitted surface (PS), rolled-in scales (RS), and scratches (Sc). Each defect has 300 images, and each image contains one type of defects mentioned above. Defects in the image are labeled by the LabelImg software, saving as annotation files in XML format. The steel strip defects in the image are marked with a rectangular frame called ground truth box. Additionally, the coordinate information of the ground truth box is recorded in the annotation files. The entire dataset contains over 5000 ground truth boxes. Figure 5 presents the examples of annotated defect images in the NEU-DET dataset.

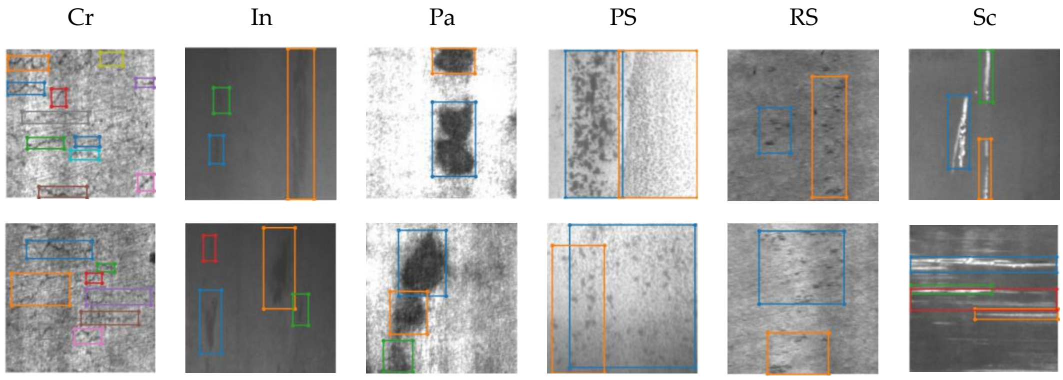


Figure 5. Examples of annotated defect images in the NEU-DET dataset.

4.2. *Test Results of Surface Defects Detection of Steel Strips*

The surface defects images of the steel strips were trained by the CP-YOLOv3-dense network proposed in the current work. During the training process, an asynchronous stochastic gradient descent with a momentum term of 0.9 is used, the initial learning rate of the weight is 0.001 and the attenuation coefficient is set to 0.0005. More training samples were generated by adjusting the saturation, exposure and overall tone. The final test results are shown in Table 2, and the visualization of part of the defect image detection results can be found in Figure 6.

Table 2. The detection results of different types of defects.

Parameters	Defect types						Average value
	Crazing	Inclusion	Patches	Pitted Surface	Rolled-in Scale	Scratches	
mAP/%	71.4	82.4	91.9	82.8	77.7	90.2	82.73
P	0.725	0.912	0.976	0.821	0.763	0.942	0.857
R	0.703	0.875	0.921	0.793	0.751	0.892	0.823
F_1	0.714	0.893	0.948	0.807	0.757	0.916	0.839
t/ms	14.35	7.57	12.98	5.89	9.81	7.48	9.68

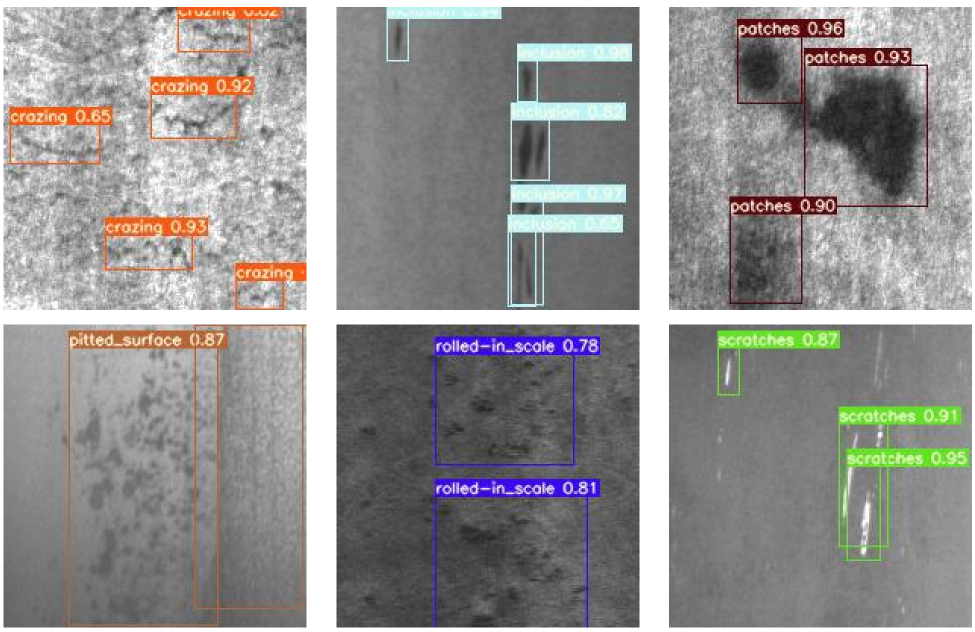


Figure 6. Visualization of part of the defect image detection results

Figue 7 is the change curve of the loss function during the training process. Obviously, the loss value drops rapidly during the first 30 epochs, indicating that the model is quickly fitting. Then, the loss value gradually decreases with the number of epochs, and tends to 0. When the number of epochs is 200, the loss value has been basically unchanged and divided into 3 gradients. The loss function of the defect Pa converges the best, and the corresponding loss value is less than 0.05.

Figure 8 shows a curve that the mean average precision of six kinds of defect detection changes with the number of epochs. Obviously, the mean average precision increases rapidly with the number of epochs, and then tends to remain stable. With the defect Pa as an example, when the number of epochs is 186, the mean average precision reaches the maximum value of 91.9%. Therefore, the weight parameters of 186 iterations are selected as the optimal model parameters.

Although our experiment results are reasonable and have been able to meet the detection requirements of steel strips, there is still an opportunity to further improve the performance of our model. Especially for defects Cr and PS, the detection accuracy is not high enough. It can be observed that the defect Cr is small and narrow, showing a linear shape. However, when the image features are extracted by convolutional networks, the filters used are squares with a size of S×S, which will result in loss of features and a decrease in detection accuracy. At present, the model proposed in current work realizes the reuse and fusion of features. Next, we will consider improving the feature extraction methods, such as simultaneous sample and feature selection [53], using linear discriminant analysis [54], etc., to further improve the performance of our model. The problem of lower detection accuracy of defect PS is mainly caused by labeling errors. The defects in the image cluster together, and the dividing lines between different defects are difficult to distinguish, resulting in a large gap between the bounding boxes and the ground truth boxes. In the future work, We will use auxiliary annotation tools and cross-checking algorithm to improve the accuracy of data annotation, avoid large errors caused by manual annotation, and thus improve the accuracy and availability of the annotation data.

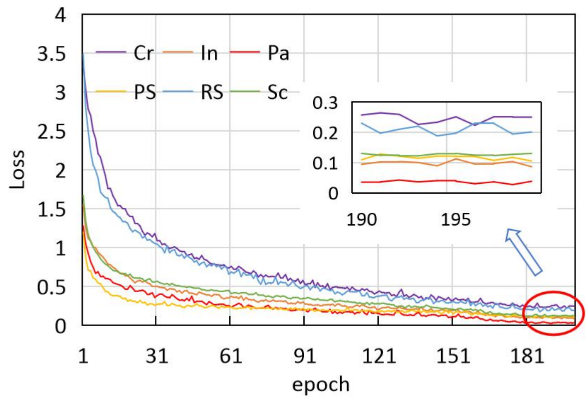


Figure 7. The change curves of loss value.

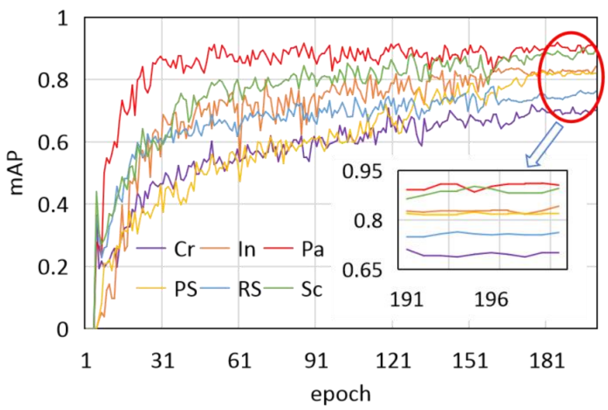


Figure 8. The change curves of mean average precision.

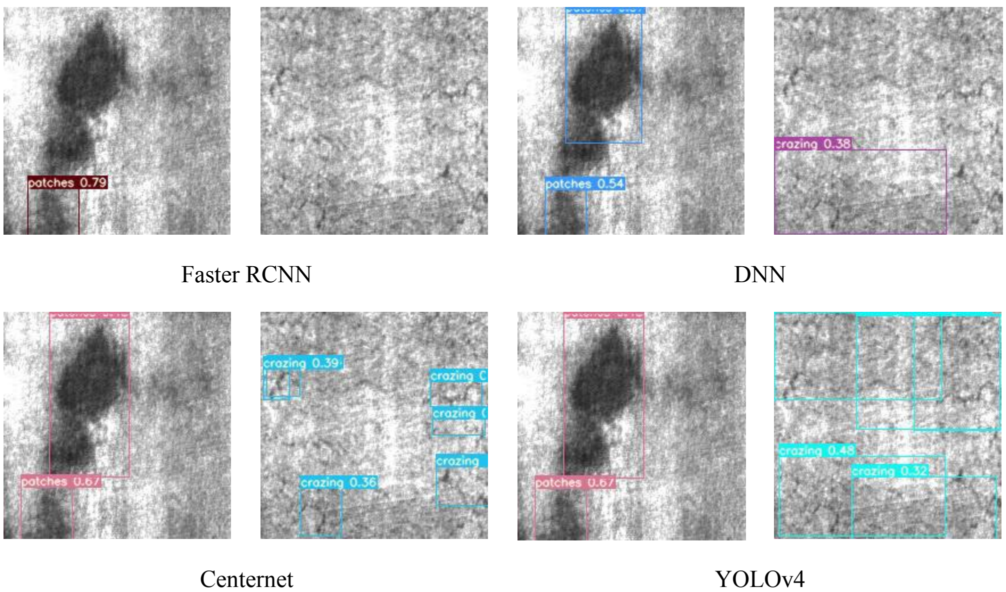
4.3. *Comparison Experiment of Different Models*

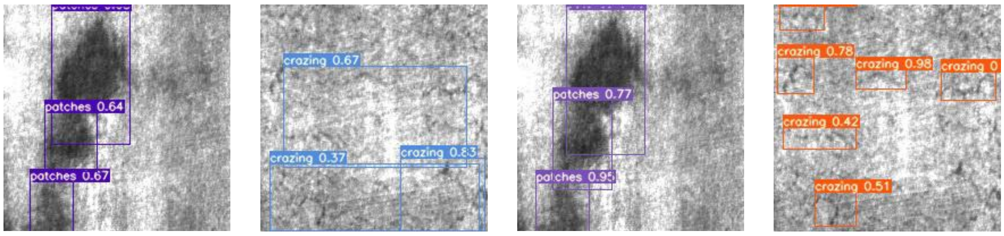
To verify the effectiveness of the proposed model, it is compared with other deep learning models. We employ classic object detectors (Faster RCNN and DNN), more state-of-the-art object detectors (Centernet and YOLOv4) and the original YOLOv3 network to detect the surface defects of the steel strips in the dataset, and compare the detection results with the methods proposed in the present study, as shown in Table 3. According to the mean average precision of the six defect detections, the CP-YOLOv3-dense network model we proposed is 6.65% higher than the original YOLOv3 network and 10.6% higher than the DNN network used in reference [55], and it is slightly higher than the state-of-the-art object detectors. The detection speed is slightly lower than the original YOLO network. This lies that the improved network needs to read memory frequently due to channel superposition, which slows down the model training and prediction speed.

According to the different shape of defects, the six types of defects in the dataset can be divided into two categories, namely planar defects and linear defects. An image of defect Patches (planar defects) and an image of defect Crazing (linear defects) were selected randomly from the detection results of various models in Table 3, which are shown in Figure 9. The classic object detector has a large number of missed detections, and the detection accuracy is low. The state-of-the-art object detectors are effective in detecting planar defects, but when detecting linear defects, the detection accuracy is much lower than our proposed method. This is because the linear defect is small and has a long and narrow shape, thus it is easy to cause feature loss during the convolution process. The CP-YOLOv3-dense network proposed by us uses the output of each layer as the input of the next layer to realize the multiplexing and fusion of features, which is more suitable for the detection of linear defects.

Table 3. Comparison of detection results of different deep learning networks.

Methods	Network	mAP/ %	F_1	t/ms
Faster RCNN	VGG16	73.12	0.698	17. 14
DNN [55]	ResNet34	74.80		
Centernet	DLA34	82.01	0.850	12.43
YOLOv4	DarkNet	80.86	0.817	7.85
YOLOv3	DarkNet	77.57	0.754	8.95
Our method	DarkNet+DenseNet	82.73	0.839	9.68





YOLOv3 Our method

Figure 9. The comparison of defect pa detection results using different deep learning networks.

4.4. Improvement of Cluster based on Images in Dataset

YOLO network improves the performance of target detection by introducing anchor box as a priori box [49]. Additionally, it is conducive to the learning of the neural network by selecting a suitable a priori frame, thereby improving the accuracy of steel strips defect detection. Therefore, we use K-means algorithm to cluster the target frame of the dataset in the process of detection. We define the following distance function through IoU.

$$d(\text{box}, \text{centroid}) = 1 - \text{IoU}(\text{box}, \text{centroid}) \tag{13}$$

where, centroid represents the center of the cluster, box represents the sample, IOU (box, centroid) represents the intersection ratio of the cluster center box and the cluster box.

Figure 10 shows the clustering experiment results of different steel strip surface defect datasets, revealing the relationship between K value and distance. Considering the influence of K value on the model parameters, K = 6 was selected in our experiment. At this time, the shape of the anchor box generated by clustering is more in consistence with the shape of defects in NEU-DET dataset. The ratio of the length and width of the anchor box is obtained through clustering, and then multiplied by the resized picture size. Finally, the anchor parameters in our experiment are obtained which can be found in Table 4.

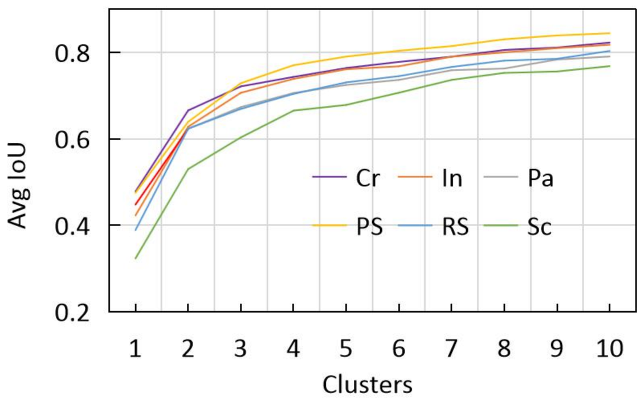


Figure 10. The relationship between K value and IoU.

Table 4. The setting of anchor parameters.

Defect names	Avg IoU	Anchor parameters
Crazing	0.7757	[56, 161], [47, 112], [94, 434], [38, 82], [69, 273], [31, 60]
Inclusion	0.7720	[29, 62], [47, 103], [67, 221], [87, 420], [38, 78], [51, 147]
Patches	0.7379	[136, 165], [89, 129], [125, 275], [132, 94], [181, 311], [69, 64]
Pitted Surface	0.8053	[71, 76], [286, 421], [434, 436], [106, 390], [185, 430], [123, 159]
Rolled-in Scale	0.7456	[109, 150], [246, 221], [154, 165], [224, 125], [147, 293], [103, 78]
Scratches	0.7128	[33, 395], [441, 62], [35, 154], [60, 441], [90, 441], [197, 35]

4.5. *The Effect of the Ground Truth Box on the Detection Results*

The image dataset used in our experiment is the NEU-DET Dataset. During the experiment, a very strange phenomenon occurred. During the training process using the CP-YOLOv3-dense network, the mAP of the defect Crazing will not increase with the increase of epochs, and it will always oscillate back and forth between the value of 0.1 and 0.3, which can be found in Figure 11. We observed the morphology of defect Crazing, finding that this defect's shape was small and densely distributed. The original images in NEU-DET Dataset are labeled with larger ground truth boxes. Each box contains several smaller Crazing defects, and this labeling method is unreasonable. We relabeled the images containing Crazing defects, and smaller ground truth boxes were selected. Each box contains only one defect, and then the CP-YOLOv3-dense network is used to train the relabeled dataset. The mAP value improved steadily. Under the CP-YOLOv3-dense network, the mAP of the original defect Crazing detection is 0.353 and the detection time is 7.67ms. The mAP of the relabeled defect Crazing detection is 0.714, and the detection time is 14.35ms. In the end, the detection accuracy was improved by 102.27%, and the detection speed was decreased by 87.1% due to the increase of bounding boxes. Figure 12 shows the comparison of the detection results of defect Crazing.

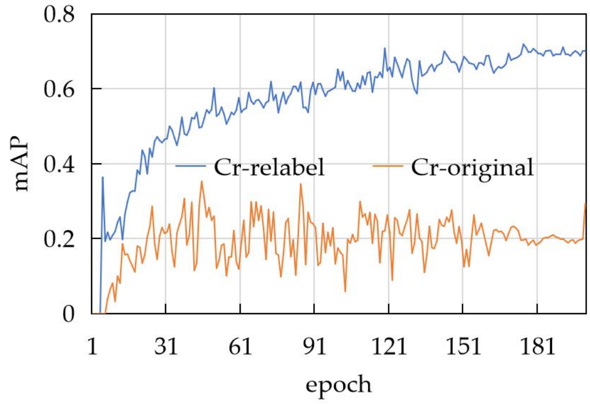


Figure 11. The mAP of defect Crazing changes with epochs.

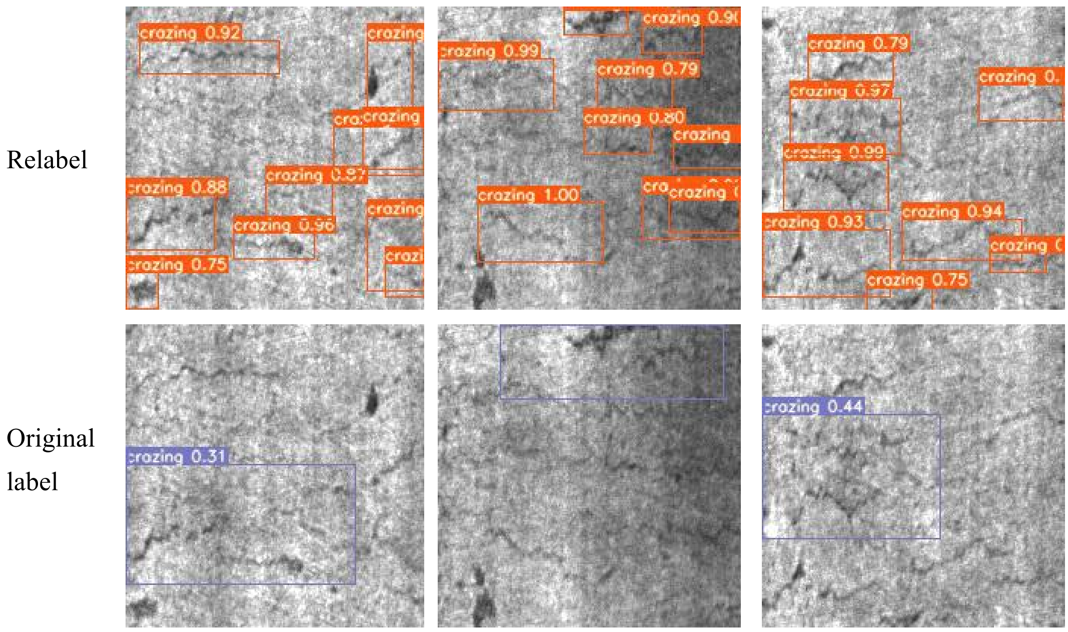


Figure 12. Comparison of detection results of defect Crazing with different labels.

5. Conclusions

(1) We propose a CP-YOLO-dense network for the detection of surface defects on steel strips. Through performing priority classification, the improved YOLO network does not need to predict the probability of the defect category, thereby improving the detection accuracy. Secondly, DenseNet is used to replace the original transmission layer with lower resolution. Therefore, the model can receive multi-layer convolutional features output by densely connected blocks before making predictions, consequently enhancing feature multiplexing and fusion. The results demonstrate that the recognition precision of the CP-YOLOv3-dense network model is 85.7%, the recall rate is 82.3%, the mean average precision is 82.73%, and the detection time of each image is 9.68ms, which are superior to other deep learning networks

(2) The K-means clustering algorithm is used to perform cluster analysis on the images in the NEU-DET dataset to find an appropriate size of anchor box. The results demonstrate that when K = 6, the shape of the anchor box generated by clustering is more in line with the appearance of defects in the NEU-DET dataset. Through testing the CP-YOLO-dense network model detection speed, we found that the detection speed of the improved network is 1.77 times faster than Faster RCNN network.

(3) This study refines annotation for images containing defect Crazing. The mAP curve of the relabeled dataset steadily rises during training, and the average detection precision is improved by 102.27% under the CP-YOLOv3-dense network.

(4) Currently, the model proposed by us realizes the reuse and fusion of features, but there is still an opportunity to further improve its performance. In future work, we will consider further improving the detection accuracy by changing the feature extraction methods.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

This work was supported by the Research Clusters Program of Tokushima University; JSPS KAKENHI [grant number 19K20345] and A Priority Academic Program Development of Jiangsu Higher Education Institutions (PAPD).

References

1. He, Y.; Song, K.; Dong, H.; Yan, Y. Semi-supervised defect classification of steel surface based on multi-training and generative adversarial network. *Opt. Lasers Eng.* **2019**, 122, 294-302. DOI.10.1016/j.optlaseng.2019.06.020.
2. Liu, Y.; Xu, K.; Wang, D. Online surface defect identification of cold rolled strips based on local binary pattern and extreme learning machine. *Metals*. **2018**, 8, 197. DOI. 10.3390/met8030197 .
3. Neogi, N.; Mohanta, D.K.; Dutta, P.K. Review of vision-based steel surface inspection systems. *J. Image Video Process.* **2014**. DOI. 10.1186/1687-5281-2014-50.
4. Weimer, D.; Scholz-Reiter, B.; Shpitalni, M. Design of deep convolutional neural network architectures for automated feature extraction in industrial inspection. *CIRP Ann. Manuf. Technols*. **2016**, 65,417-420. DOI. 10.1016/j.cirp.2016.04.072.
5. Park, JK.; Kwon, BK.; Park, JH.; Kang, DJ. Machine learning-based imaging system for surface defect inspection. *Int. J. of Precis. Eng. and Manuf.-Green Tech.* **2016**, 3, 303-310. DOI. 10.1007/s40684-016-0039-x.
6. Ravikumar, S.; Ramachandran, KI.; Sugumaran, V. Machine learning approach for automated visual inspection of machine components. *Expert. Syst. Appl.*. **2011**, 38, 3260-3266. DOI. 10.1016/j.eswa.2010.09.012.
7. Luo, Q.; Fang, X.; Sun, Y.; Liu, L.; Ai, L.; Yang, C.; Simpson, O. Surface Defect Classification for Hot-Rolled Steel Strips by Selectively Dominant Local Binary Patterns," *IEEE Access*. **2019**, 7, 23488-23499. DOI. 10.1109/ACCESS.2019.2898215.
8. Zhang XL, Ouyang Q, Peng S, et al. Continuous casting slab surface crack depth measurement using sinusoidal phase grating method.. Ironmak Steelmak, 2014, 41(5): 387-393.

9. Li, J.; Su, Z; Geng, J.; Yin, Y. Real-time detection of steel strip surface defects based on improved yolo detection network. *IFAC*. **2018**, 51, 76-81. DOI. 10.1016/j.ifacol.2018.09.412.
10. Ren, Q.; Geng, J.; Li, J. Slighter Faster R-CNN for real-time detection of steel strip surface defects. 2018 Chinese Autom. Congr. (CAC), Xi'an, China. **2018**, 2173-2178. DOI. 10.1109/CAC.2018.8623407.
11. Lin, C.Y.; Chen, C.H.; Yang,C.Y.; Akhyar, F.; Hsu, C.Y.; Ng, H.F. Cascading convolutional neural network for steel surface defect detection. *Int. Conf. Appl. Hum. Factors Ergon*. **2019**, 202-212. DOI. 10.1007/978-3-030-20454-9_20.
12. He, D.; Xu K.; Zhou, P. Defect detection of hot rolled steels with a new object detection framework called classification priority network. *Comput. Ind. Eng*. **2019**, 128, 290-297. DOI.10.1016/j.cie.2018.12.043.
13. Tao, X.; Zhang, D.; Ma, W.; Liu, X.; Xu, D. Automatic metallic surface defect detection and recognition with convolutional neural networks. *Appl. Sci.* **2018**, 1575. DOI. 10.3390/app8091575.
14. Dong, H.; Song, K.; He, Y.; Xu, J.; Yan, Y.; Meng, Q. PGA-Net: pyramid feature fusion and global context attention network for automated surface defect detection. *IEEE Trans. Industr. Inform.*. **2019**. DOI: 10.1109/TII.2019.2958826.
15. Gao, Y.; Gao, L.; Li, X.; Yan, X. A semi-supervised convolutional neural network-based method for steel surface defect recognition. *Robot. Comput. Integr. Manuf*. **2020**, 61, 101825. DOI. 10.1016/j.rcim.2019.101825.
16. Saludes-Rodil, S.; Baeyens, E.; Rodríguez-Juan, C.P. Unsupervised classification of surface defects in wire rod production obtained by eddy current sensors. *Sensors*. **2015**, 15, 10100-10117. DOI. 10.3390/s150510100.
17. D. Chen, and Z. Liu, "Generalized Haar-Like Features for Fast Face Detection," Int. Conf. Mach. Learn. Cyberne., Hong Kong, 2007, pp. 2131-2135, DOI. 10.1109/ICMLC.2007.4370496.
18. Han, P.; Liao, J. Face detection based on adaboost. Int. Conf. Apperc. Comput. Intell. Anal. Chengdu, **2009**, 337-340. DOI. 10.1109/ICACIA.2009.5361085.
19. Wang, Z.; Jia, Y.; Huang, H.; Tang, S. Pedestrian Detection Using Boosted HOG Features. Proc. IEEE Conf. Intell. Transport. Syst. Beijing, Oct. **2008**, 1155-1160. DOI. 10.1109/ITSC.2008.4732553.
20. Gan G.; Cheng, J. Pedestrian Detection Based on HOG-LBP Feature. Int. Conf. Comput. Intell Secur, Hainan, **2011**, 1184-1187. DOI. 10.1109/CIS.2011.262.
21. Bauer, S.; Köhler, S.; Doll, K.; Brunsmann, U. FPGA-GPU architecture for kernel SVM pedestrian detection. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW), Jun, **2010**, 61-68. DOI. 10.1109/CVPRW.2010.5543772.
22. Guo, Y.; Liu, Y.; Oerlemans, A.; Lao, S.; Wu, S.; S.Lewa, M. Deep learning for visual understanding: A review. *Neurocomputing*, **2016**, 187, 27-48.
23. Zhao, Z.; Zheng, P.; Xu, S.; Wu, X. Object detection with deep learning: a review. *IEEE Trans. Neural Netw. Learn Syst*. **2019**, 30, 3212-3232. DOI. 10.1109/TNNLS.2018.2876865.
24. Deng, L.; Yang, M.; Li, T.; He, Y.; Wang. C. RFBNet: deep multimodal networks with residual fusion blocks for RGB-D semantic segmentation. 2019, arXiv:1907.00135.
25. Duan, K.; Bai, S.; Xie, L.; Qi, H.; Huang, Q.; Tian, Q. CenterNet: Keypoint Triplets for Object Detection. 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea (South), **2019**, 6568-6577.
26. Law H.; Deng. J. Cornernet: Detecting objects as paired keypoints. The European Conference on Computer Vision (ECCV), **2018**, pp. 734-750.
27. Ren F.; and Huang, Z. Facial expression recognition based on AAM – SIFT and adaptive regional weighting. *IEEJ Trans. Electr. Electron. Eng*. **2015**, 10, 713-722, 2015.
28. Kasar, M.M.; Bhattacharyya, D.; Kim, T.H. Face recognition using neural network: a review. *Int. J. Secur. its Appl*. **2016**, 10, 81-100, 2016.
29. Ren, F.; Liu, W.; Wu, G. Feature reuse residual networks for insect pest recognition. *IEEE Access*. **2019**, 7, 122758-122768. DOI. 10.1109/ACCESS.2019.2938194.
30. Liu, W.; Wu, G.; Ren, F. Stochastic channel reuse residual networks for plant disease severity detection. IEEE Int. Conf. Cloud Comput. Intell. Syst. (CCIS), Singapore, **2019**, 57-61. DOI. 10.1109/CCIS48116.2019.9073714.
31. Zhang J.; Zhang S. Error analysis and feature detection of electrochemical machining micro-hole. *J. Mech. Electr. Eng*. **2019**, 36, 32-35. DOI. 10.3969/j.issn.1001-4551.2019. 01.007.
32. Ren F.; Zhou, Y. CGMVQA: a new classification and generative model for medical visual question answering. *IEEE Access*. 2020, 8, 50626-50636. DOI. 10.1109/ACCESS.2020.2980024.

33. Latif, S.; Usman, M.; Rana, R.; Qadir, J. Phonocardiographic sensing using deep learning for abnormal heartbeat detection. *IEEE Sens. J*. **2018**, 18, 9393-9400. DOI. 10.1109/JSEN.2018.2870759.
34. Zhang, J.; Ren, F.; Ni, H.; Zhang, Z.; Wang, K. Research on information recognition of VAT invoice based on computer vision. IEEE Int. Conf. Cloud Comput. Intell. Syst. (CCIS), Singapore, **2019**, 126-130. DOI. 10.1109/CCIS48116.2019.9073749.
35. Yuan, Z.; Zhang, Z.; Su, H.; Zhang, L.; Shen, F.; Zhang, F. Vision-based defect detection for mobile phone cover glass using deep neural networks. *Int. J. Precis. Eng. Manuf*. 2018, 19, 801-810. DOI. 10.1007/s12541-018-0096-x.
36. Lv, Y.; Ma, L.; Jiang, H. A mobile phone screen cover glass defect detection model based on small samples learning. IEEE Int. Conf. Signal Image Process. (ICSIP), Wuxi, China, **2019**, 1055-1059, DOI. 10.1109/SIPROCESS.2019.8868737.
37. Jeyaraj P.R.; Samuel Nadar, E.R. Computer vision for automatic detection and classification of fabric defect employing deep learning algorithm. *Int. J. Cloth. Sci. Technol*. **2019**, 31, 510-521, 2019, DOI. 10.1108/IJCST-11-2018-0135.
38. Li, Y.; Zhang, D.; Lee, D. Automatic fabric defect detection with a wide-and-compact network. *Neurocomputing*. 2018, 329, 329-338. DOI. 10.1016/j.neucom.2018.10.070
39. Lin, W.; Lin, C.; Chen, G.; Hsu, C. Steel surface defects detection based on deep learning. Int. Conf. Appl. Hum. Factors Ergon. (AHFE), Orlando, USA, Jul. **2018**, 141-149. DOI. 10.1007/978-3-319-94484-5_15
40. Li, K.; Wang, X.; Ji, L. Application of multi-scale feature fusion and deep learning in detection of steel strip surface defect. Int. Conf. Artif. Intell. Adv. Manuf. (AIAM), Dublin, Ireland, **2019**, 656-661. DOI. 10.1109/AIAM48774.2019.00136.
41. Wei, R.; Song, Y.; Zhang, Y. Enhanced faster region convolutional neural networks for steel surface defect detection. *ISIJ Int*. **2020**, 60, 539-545.
42. Tang R.; Mao, K. An improved GANs model for steel plate defect detection. Int. Conf. Commun. Netw. Artif. Intell, Guangzhou, China, Dec. **2019**, 27-29.
43. Redmon, J.; Divvala, S.; Girshick R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Las Vegas, NV, **2016**, 779-788. DOI. 10.1109/CVPR.2016.91.
44. Redmon J.; Farhadi, A. YOLO9000: Better, Faster, Stronger. Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Honolulu, HI, **2017**, 6517-6525. DOI. 10.1109/CVPR.2017.690.
45. Redmon J.; Farhadi, A. Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767, **2018**.
46. Shao, S.; McAleer, S.; Yan, R.; Baldi, P. Highly Accurate Machine Fault Diagnosis Using Deep Transfer Learning. *IEEE Trans. Industr. Inform*. **2019**, 15, 2446-2455. DOI. 10.1109/TII.2018.2864759.
47. Huang, G.; Liu, S.; Maaten, L.V.D.; Weinberger, K.Q. CondenseNet: an efficient densenet using learned group convolutions. Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Salt Lake City, UT, **2018**, 2752-2761. DOI. 10.1109/CVPR.2018.00291.
48. Tian, Y.; Yang, G.; Wang, Z.; Wang, H.; Li, E.; Liang, Z. Apple detection during different growth stages in orchards using the improved YOLO-V3 model. *Comput. Electron. Agric*. **2019**, 157, 417-426. DOI. 10.1016/j.compag.2019.01.012.
49. Ren F.; Zhang, Q. An Emotion Expression Extraction Method for Chinese Microblog Sentences. *IEEE Access*. **2020**, 8, 69244-69255. DOI. 10.1109/ACCESS.2020.2985726.
50. Wang, J.; Li, Q.; Gan, J.; Yu, H.; Yang, X. Surface Defect Detection via Entity Sparsity Pursuit With Intrinsic Priors. *IEEE Trans. Industr. Inform*. **2020**, 16, 141-150. DOI. 10.1109/TII.2019.2917522.
51. Ren, F.; Xue, S. Intention Detection Based on Siamese Neural Network With Triplet Loss. *IEEE Access*. **2020**, 8, 82242-82254.
52. Song, K.; Yan, Y. A noise robust method based on completed local binary patterns for hot-rolled steel strip surface defects. *Appl. Surface Sci*. **2013**, 285, 858-864.
53. Ali, L.; Zhu, C.; Zhou, M.; Liu, Y. Early diagnosis of Parkinson's disease from multiple voice recordings by simultaneous sample and feature selection. *Expert Syst. Appl*. **2019**, 137,. 22-28. DOI. 10.1016/j.eswa.2019.06.052.
54. Ali, L.; Zhu, C.; Zhang, Z.; Liu, Y. Automated detection of parkinson's disease based on multiple types of sustained phonations using linear discriminant analysis and genetically optimized neural network. *IEEE J. Transl. Eng. Health Med*. **2019**, 7, 1-10. DOI. 10.1109/JTEHM.2019.2940900.

55. He, Y.; Song, K.; Meng, Q.; Yan, Y. An end-to-end steel surface defect detection approach via fusing multiple hierarchical features. *IEEE Trans. Instrum. Meas*. **2020**, 69, 1493-1504. DOI. 10.1109/TIM.2019.2915404.