Computing and Informatics, Vol. 43, 2024, 1432–1454, doi: 10.31577/cai_2024_6_1432

BONDING DEFECT DETECTION BASED ON IMPROVED SINGLE SHOT MULTIBOX DETECTOR

Huifang Gao, Yong Jin*

School of Information and Communication Engineering
North University of China, 030051 Taiyuan, China
e-mail: {g15167013798, xiandaijiance601}@163.com

Maozhen Li

Department of Electronic and Computer Engineering
Brunel University London, Uxbridge UB8 3PH, U.K.
e-mail: Maozhenli@brunel.ac.uk

Youxing Chen

School of Information and Communication Engineering
North University of China, 030051 Taiyuan, China
e-mail: chenyouxing@nuc.edu.cn

Junbin Zang

Key Laboratory of Instrumentation Science and Dynamic Measurement
North University of China, 030051 Taiyuan, China
e-mail: zangjunbin@nuc.edu.cn

Xiaoliang Fan

School of Earth Sciences and Engineering
Nanjing University, 210023 Nanjing, China
e-mail: 13394897195@163.com

Abstract. To solve the problem of time-consuming and low efficiency in manual defect detection, this paper proposes a bonding defect detection algorithm based on improved Single Shot MultiBox Detector (SSD). DenseNet is used to replace VGG of the SSD algorithm to improve the detection effect of bonding defect. A novel feature fusion network is designed, in which dilated convolution is used to reduce the size of the low-level feature map, and it is fused with the high-level feature map, and then the Convolutional Block Attention Module (CBAM) attention mechanism is used to increase the ability to extract the features. Focal loss is used to control the ratio of positive and negative samples for training and suppress easily separable samples, so that the samples involved in training have better distribution and the model has better detection performance. Then, the defect data set is constructed and a comparison experiment is carried out. The results show that the mAP, Precision, and Recall of the improved SSD network are increased to 75.9 %, 77.3 %, and 75.6 %, respectively, which can better identify bonding defect.

Keywords: SSD, defect detection, DenseNet, dilated convolution, CBAM, focal loss

1 INTRODUCTION

The bonding structure is widely used in various industrial products because of its high specific strength and modulus. However, defects such as debonding, cracking, and delamination easily occur during in the bonding process, which destroys the integrity of the bonding structure. The ability to properly identify these defects is critical for optimizing production techniques and improving quality.

At present, X-ray is the most commonly used nondestructive testing method, which can accurately reflect the location, shape, type, and size of defects by gray images [1, 2]. When detecting bonding defects, it is necessary to determine whether there are defects on X-ray images, and then determine the location and type of defects. The defect detection algorithms mainly include traditional algorithms and detection algorithms based on deep learning. There are many shortcomings in the traditional detection process due to the influence of human factors, and it is increasingly unable to meet the needs of industrial automation and intelligent development [3]. Deep learning can learn and recognize the features of the input image independently, which solves the inconvenience of manually extracting features, and better results are achieved. Using the target detection algorithm based on deep learning, the localization and classification of defects can be realized at the same time, which is mainly divided into two categories. One is the two-stage target detection algorithm represented by R-CNN [4], Fast R-CNN [5], and Faster R-CNN [6]. The other is a one-stage target detection algorithm represented by YOLO [7] and SSD. Specifically, the two-stage algorithm first generates candidate regions on the

* Corresponding author

image, and then performs classification and regression on each candidate region in turn. The one-stage algorithm directly completes the localization and classification of all targets on the whole image, omitting the operation of generating candidate regions. Among them, SSD (Single Shot Multibox Detector) is one of the current mainstream object detection frameworks. SSD was originally proposed by Wei Liu at the 14th European Conference on Computer Vision (ECCV) in 2016 [8], and has become another one-stage object detection algorithm after YOLO. SSD not only draws on the anchor mechanism and feature pyramid structure of Faster R-CNN, but also inherits the regression idea of YOLO [9].

SSD algorithm achieves a better detection effect while taking into account the detection speed, but there are some problems:

1. The backbone network is shallow and the feature extraction is insufficient.
2. The low-level feature layer has poor semantic information and the high-level has poor location information; both resulting in insufficient accuracy.
3. The cross-entropy loss does not pay attention to challenging samples, which affects the detection effect.

In order to solve the above problems, this paper takes the defects of debonding, cracking, and delamination as the research objects, and proposes an improved SSD defect detection algorithm. DenseNet is used as the backbone network, and dilated convolution and CBAM are used to fuse shallow and deep features, which greatly enhances the feature extraction and expression ability of the network. In addition, focal loss is introduced to enhance the learning ability of the network and further improve the model detection effect.

The rest of the paper is organized as follows. Section 2 briefly describes improved SSD network. Section 3 presents the loss function of improved SSD network. Section 4 conducts experiments and evaluates the performance of the improved SSD. Section 5 concludes the paper.

2 IMPROVED SSD NETWORK

The SSD network consists of three parts: backbone network, feature extraction network, and detection network. Figure 1 shows the structure of the SSD network. The backbone network is modified on the basis of VGG16 [10] by replacing the last two fully-connected layers FC6 and FC7 with convolutional layers Conv6 and Conv7, and then adding four groups of convolutional layers: Conv8, Conv9, Conv10, and Conv11. Then, the feature maps of Conv4_3 and Conv7 are combined with those of Conv8_2, Conv9_2, Conv10_2, and Conv11_2 to form a multi-scale feature extraction network. Finally, the detection network is used to output category confidences and location information, then all the calculation results are combined to calculate the loss.

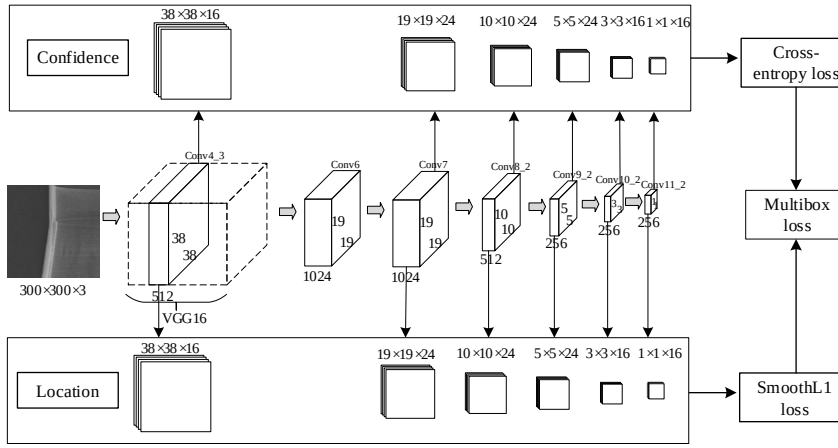


Figure 1. Model structure of SSD

2.1 Replacing the Backbone Network

The SSD algorithm is limited by the network depth of VGG-16. Related studies have shown that features can be enriched by increasing the number of network layers and the recognition accuracy improves with the deepening of network depth. Compared with VGG-16, the ResNet-50 and DenseNet-121 are good choices. To verify the performance of these two networks, the test is carried out on the ImageNet2012 dataset [11] and the results are shown in Table 1.

Model	Accuracy (%)	Size (M)	FLOPS (10^9)
VGG-16	71.7	537	15.3
ResNet-50	73.1	87	3.6
DenseNet-121	74.3	32	1.9

Table 1. Comparison results of different network performance

As we can see from Table 1, among the above three models, DenseNet-121 has the highest accuracy and the lowest size and FLOPS. Meanwhile, VGG-16 is composed of several convolutional layers and pooling layers, which has a simple structure and insufficient feature extraction. The idea of DenseNet-121 is basically the same as that of ResNet-50, which is to intensively connect all the previous layers with the later layers to realize feature reuse. However, ResNet-50 adds feature maps, while DenseNet-121 concatenates feature maps of different channels. The former highlights feature reuse, and the latter explores new features. Therefore, DenseNet-121 is used to replace VGG-16 network.

DenseNet [12] was proposed in 2017 and borrowed from ResNet [13]. Figure 2 shows the DenseNet, the later layer can also make use of the information extracted from the previous layer. At the same time, due to the reuse of features, each

layer only needs fewer convolution kernels, which reduces the network parameters to a certain extent and improves the network operation efficiency. In DenseNet, the Dense Block module consists of 1×1 convolution and 3×3 convolution, which 1×1 convolution is used to reduce the dimension of the input feature map, and the 3×3 convolution is used to extract features. The Transition layer structure is adopted between the Dense Block modules, which includes 1×1 convolution and 2×2 average pooling to gradually reduce the size of the feature map.

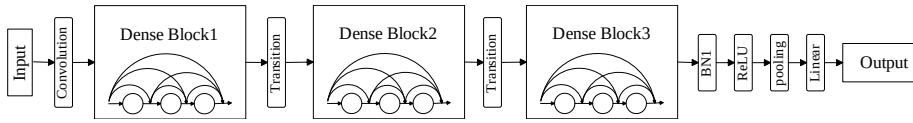


Figure 2. The structure of DenseNet

Table 2 is the structure of the proposed algorithm. It is modified on the basis of DenseNet, which strengthens the feature propagation and improves the repeated utilization of features at each layer. The feature maps output by the six layers of Dense block2, Dense block3, Conv5, Conv6-2, Conv7-2, and Conv8-2 are used for prediction, and concentric prior boxes are set on each reference point of each feature map.

	Layers	Output size	SSD_DenseNet
Backbone	Conv1	$150 \times 150 \times 64$	7×7 conv, stride 2, padding 3
	Pool1	$75 \times 75 \times 64$	3×3 max pool, stride 2
	Dense block1	$75 \times 75 \times 256$	$[1 \times 1$ conv, stride $1; 3 \times 3$ conv, stride $1] \times 6$
	Transition1	$75 \times 75 \times 128$	1×1 conv, stride 1
		$38 \times 38 \times 128$	2×2 avg pool, stride 2, ceil_mode = true
	Dense block2	$38 \times 38 \times 512$	$[1 \times 1$ conv, stride $1; 3 \times 3$ conv, stride $1] \times 12$
	Transition2	$38 \times 38 \times 256$	1×1 conv, stride 1
		$19 \times 19 \times 256$	2×2 avg pool, stride 2, ceil_mode = true
	Dense block3	$19 \times 19 \times 1\,024$	$[1 \times 1$ conv, stride $1; 3 \times 3$ conv, stride $1] \times 24$
	Transition3	$19 \times 19 \times 512$	1×1 conv, stride 1
		$10 \times 10 \times 512$	2×2 avg pool, stride 2, ceil_mode = true
	Dense block4	$10 \times 10 \times 1\,024$	$[1 \times 1$ conv, stride $1; 3 \times 3$ conv, stride $1] \times 16$
Others	Conv5	$10 \times 10 \times 512$	1×1 conv, stride 1
	Conv6-1	$10 \times 10 \times 128$	1×1 conv, stride 1
	Conv6-2	$5 \times 5 \times 256$	3×3 conv, stride 2
	Conv7-1	$5 \times 5 \times 128$	1×1 conv, stride 1
	Conv7-2	$3 \times 3 \times 256$	3×3 conv, stride 1
	Conv8-1	$3 \times 3 \times 128$	1×1 conv, stride 1
	Conv8-2	$1 \times 1 \times 256$	3×3 conv, stride 1

Table 2. Network structure of the improved algorithm

2.2 Feature Fusion Network

It is well known that in convolutional neural networks, detail information is beneficial for localization and semantic information is more suitable for classification objects. Low-level feature maps in the SSD network have higher resolution and contain more detail information, but they have less semantic information due to fewer convolution operations. High-level feature maps have stronger semantic information, but their resolution is low and their perception of details is poor. In the SSD (DenseNet) network, the features before Dense block2 belong to shallow features, and the features after Dense block2 belong to deep features. In Figure 3, the feature map output by Conv1 has obvious edge features, and the feature maps output by Dense block1 and Dense block2 are close to the texture features. The feature maps after Dense block3 are almost impossible to judge specific feature properties by naked eye.

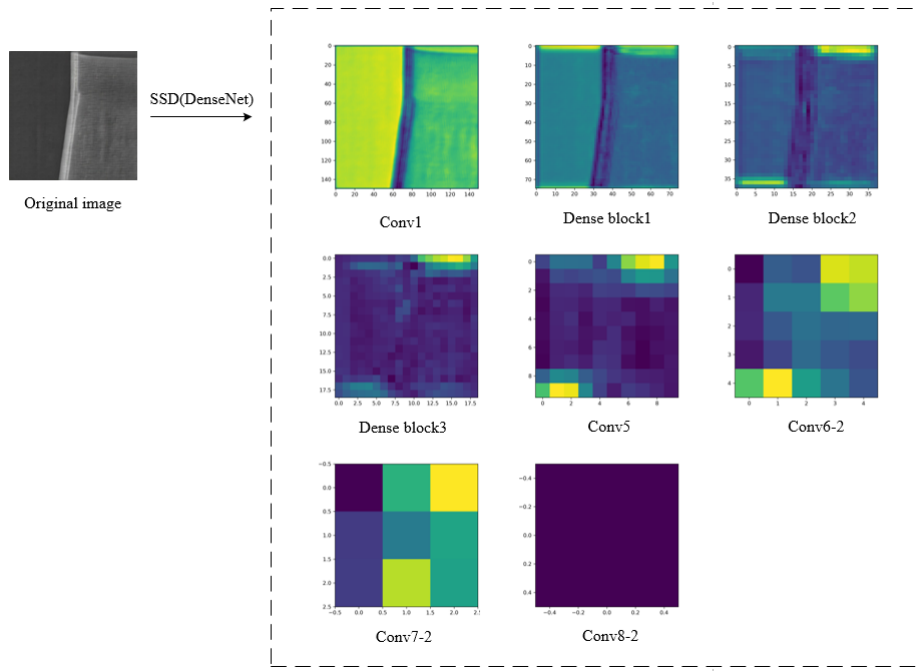


Figure 3. The feature maps of VGG (DenseNet)

After feature fusion, the shallow features with image detail information and the deep features with semantic information are fused together, which can improve the expression ability of the network. In this paper, a new feature fusion network with CBAM attention mechanism is designed. The network first reduces the size of the low-level feature map by downsampling operation, and then concatenates

it with the high-level feature map to form a multi-channel feature map, and finally passes through the CBAM attention mechanism. Common downsampling operations include pooling and dilated convolution [14]. The reason why dilated convolution is chosen instead of pooling is that information will be lost in the pooling process, while dilated convolution can enlarge the receptive field without losing information, so that the feature map can obtain more global information. Specifically, as shown in Figure 4, the output feature map $75 \times 75 \times 256$ of Dense block1 is downsampled using dilated convolution (kernel size is 3×3, step size is 2, padding is 2, dilation is 2), while the output feature map $38 \times 38 \times 512$ of Dense block2 uses ordinary convolution (kernel size is 3×3, step size is 1, padding is 1), and then the obtained two feature maps are concatenated after using ReLU [15] activation function. The fusion strategy of the other layers is also similar.

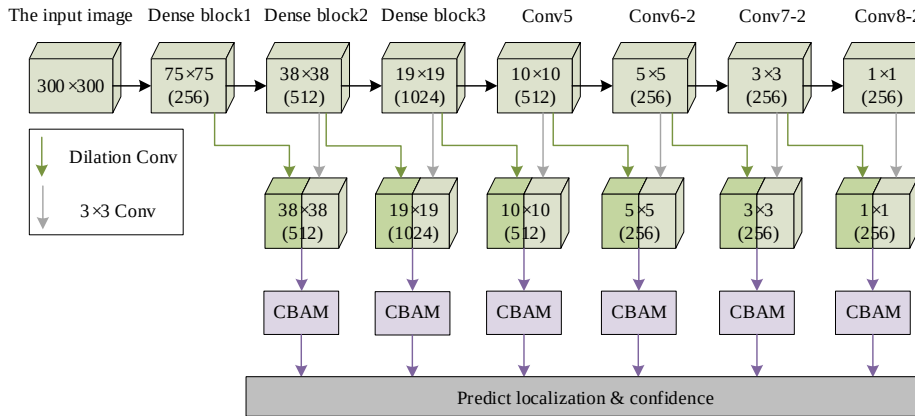


Figure 4. Detection flow chart of SSD model

Then, CBAM (Convolutional Block Attention Module) attention mechanism [16] is added to the output feature map after feature fusion, which strengthens the important detail features and weakens the useless interference features. The structure of CBAM is shown in Figure 5, it is composed of two parts: Channel Attention Module (CAM) and Spatial Attention Module (SAM). CAM is used to focus on what features are meaningful, while SAM is used to focus on where the meaningful features come from. In CAM, the input feature map $F(H \times W \times C)$ is first operated through global maximum pooling and global average pooling to generate two feature maps $(1 \times 1 \times C)$, which are then sent to the multi-layer perceptron (MPL). The one-dimensional channel attention map $Mc(F)$ is obtained by adding pixel by pixel and Sigmoid activation function, and $Mc(F)$ is multiplied by input feature map F to obtain the channel attention adjusted feature map $F1$. In SAM, $F1$ is performed through global maximum pooling and global average pooling to obtain a feature map with channel number 2, it passes a 7×7 convolutional layer to reduce the channel number to 1, and finally generates a spatial attention map $Ms(F1)$ through

a Sigmoid activation function, which is then multiplied with the feature graph $F1$. Its mathematical expression is as follows [17]:

$$M_c(F) = \sigma(MLP(AvgPool(F)) + MLP(MaxPool(F))), \tag{1}$$

$$M_s(F) = \sigma(f^{7\times7}[AvgPool(F); MaxPool(F)]), \tag{2}$$

where, σ is the Sigmoid activation function.

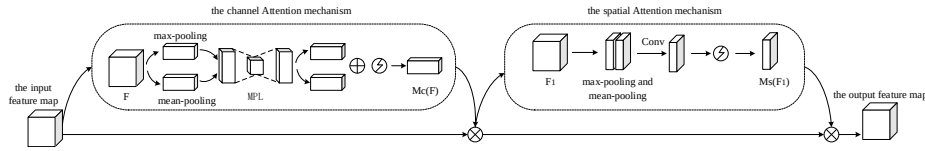


Figure 5. The structure of CBAM

3 LOSS FUNCTION OF IMPROVED SSD

It is a regression process to generate the recognition box by SSD model, and a classification process to judge the category of the recognition box. The overall loss function of SSD is represented by the weighted sum of localization loss (L_{loc}) and confidence loss (L_{conf}) in Equation (3) [18].

$$L(x, c, l, g) = \frac{1}{N}(L_{conf}(x, c) + \alpha L_{loc}(x, l, g)), \tag{3}$$

where, x represents the category matching information of the prediction box, and the matching value is 1, otherwise, the value is 0; c is the predictive value of category confidence; l is the position parameter of the prediction box; g is the position parameter of the real box; N is the number of the prior box that matches the real box; α is the weight coefficient and is set to 1.

3.1 Localization Loss Function

The localization loss is the smooth L1 loss between the prediction box and the real box, which can be expressed as:

$$L_{loc}(x, l, g) = \sum_{i\in pos}^{N} \sum_{m\in box} {x_{ij}}^k smooth_{L1}({l_i}^m - \hat{g}_j^m), \tag{4}$$

$$smooth_{L1}(x) = \begin{cases} 0.5x^2, & \text{if } |x| < 1, \\ |x| - 0.5, & \text{otherwise.} \end{cases} \tag{5}$$

Where, ${x_{ij}}^k = 1$ means that the i^{th} prediction box matches the j^{th} real box on category k, otherwise, ${x_{ij}}^k = 0$. ${x_{ij}}^k > 1$ means that there is more than one prediction box matching the j^{th} real box. $box = (cx, cy, w, h)$ indicates the center coordinates of the prediction box and its width and height. l is the transformation relationship between the prediction box and the prior box, while g is the transformation relationship between the real box and the prior box. Since l is the encoded value, it is necessary to encode the g to obtain $\hat{g}$. Where, ${d_i}^{cx}, {d_i}^{cy}, {d_i}^{w}, {d_i}^{h}$ represent the parameters of the prior box.

$$\begin{cases} \hat{g}_j^{cx} = \frac{{g_j}^{cx} - {d_i}^{cx}}{{d_i}^{w}} \\ \hat{g}_j^{cy} = \frac{{g_j}^{cy} - {d_i}^{cy}}{{d_i}^{h}} \\ \hat{g}_j^{w} = \log \frac{{g_j}^{w}}{{d_i}^{w}} \\ \hat{g}_j^{h} = \log \frac{{g_j}^{h}}{{d_i}^{h}} \end{cases} \tag{6}$$

3.2 Confidence Loss Function

Intersection-over-Union (IoU) between the prior box and ground truth is used to select the samples. The samples with the IoU greater than 0.5 and less than 0.5 are selected as positive and negative examples, respectively. If none IoU is greater than 0.5, the sample with the maximum IoU is selected as the positive example. The candidate regions generated with region growing are mostly negative examples, so the quantity ratio between positive and negative examples is normally set to 1:3 to prevent the quantity imbalance between them [19]. SSD uses the cross-entropy loss function to calculate the confidence loss, and all positive and negative samples participate in the calculation. Cross-entropy loss is a commonly used loss function in classification problems, which describes the distance between the real output and the prediction output. The loss function can be expressed in Equation (7):

$$L_{conf}(x, c) = -\sum_{i \in pos}^{N} {x_{ij}}^p \log(\hat{c}_i^p) - \sum_{i \in neg} \log(\hat{c}_i^0), \tag{7}$$

$$\hat{c}_i^p = \frac{\exp({c_i}^p)}{\sum_p \exp({c_i}^p)}, \tag{8}$$

where, the former represents the loss of a certain category (excluding the background), and the latter represents the loss of the background. $\hat{c}_i^p$ is the probability that the i^{th} prediction box predicts category p. ${x_{ij}}^p$ takes 1 to indicate that the i^{th} prior box matches the j^{th} real box of category p, otherwise it takes 0.

In the calculation of cross-entropy loss, the sample selection mechanism is used to effectively balance the number of positive and negative samples, but it does not pay attention to hard samples ($\hat{c}_i^p \leq 0.5$). In this paper, the focal loss function is used to solve this problem, which focuses more attention on hard samples and improves

the effectiveness of model training. The focal loss [20] function is as follows:

$$L_{Focalloss}(x,c) = -\sum_{i\in pos}^{N} x_{ij}{}^{p}(1-\hat{c}_i^p)^{\gamma}\log(\hat{c}_i^p) - \sum_{i\in neg}(\hat{c}_i^p)^{\gamma}\log(\hat{c}_i^0), \tag{9}$$

where, $\gamma \in [0,5]$, which is used to adjust the down-weighted rate of easy samples.

The focal loss introduces a modulating factor $(1-\hat{c}_i^p)^{\gamma}$ to reduce the contribution of easy samples ($\hat{c}_i^p$>0.5) to the loss function, so that the model focuses more on hard samples during training. For easy samples, the larger is $\hat{c}_i^p$, the smaller is $(1-\hat{c}_i^p)^{\gamma}$. For hard samples, $\hat{c}_i^p$ is small, $(1-\hat{c}_i^p)^{\gamma}$ will be large, so that the network tends to use such samples to update parameters. It can be seen from Figure 6, when $\gamma = 0$, focal loss is equivalent to cross-entropy loss. Even the easy samples have a high loss value, resulting in the high proportion of the loss value of easy samples in the algorithm. when γ increased, the weight of hard samples in the input samples increased, and $\gamma = 2$ is set in this paper. It showed that focal loss could balance the ratio of positive and negative samples and easy and hard samples by $\hat{c}_i^p$ and γ, so that the samples involved in training could be distributed more evenly and the reliability of detection algorithm could be further improved.

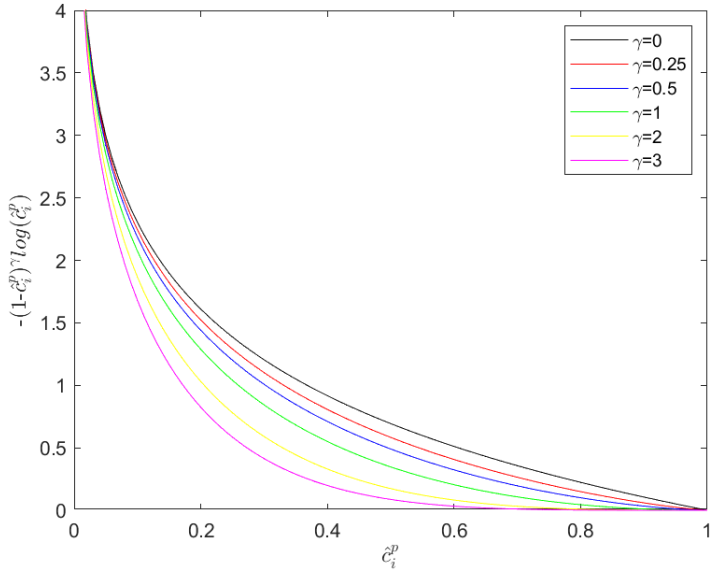


Figure 6. The graph of $-(1-\hat{c}_i^p)^{\gamma}log(\hat{c}_i^p)$

4 EXPERIMENT AND RESULT ANALYSIS

4.1 Generating Dataset

The dataset used in this experiment came from X-ray test samples of multi-layer metal and non-metal bonded tubular specimens collected over the past two years. The bonding structure defects of the sample mainly include: debonding, cracking, and delamination [21], as shown in Figure 7. Among them, debonding refers to poor bonding between layers, and it can be seen from the figure that there are obvious black images between inner and outer layers. Cracking is the outer or inner layer of cracking defects, the image is black dendritic. Delamination is the outer or inner layers that are poorly bonded inside, showing multiple vertical stripes. The annotation software called Make Sense [22] labels the images in the experiment according to the defect information. Label boxes are added, and the corresponding label files are generated for the areas with defects in images. Then, the dataset images are randomly divided into two groups: 80 % of the dataset is used for parameter learning and network training, whereas the other 20 % is used to test the generalization and recognition ability of the model, and the two datasets do not intersect each other.

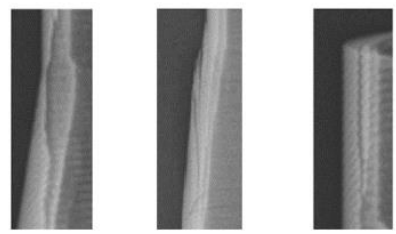


Figure 7. Defect images of adhesive structure

	Images	Labels	Debonding	Cracking	Delamination
The training set	349	626	352	109	165
The test set	93	157	90	28	39
Total	442	783	442	137	204

Table 3. Defect numbers in the dataset

4.2 Experimental Environment

The detailed software and hardware configuration of the experiment are shown in Table 4. The size of the input image was set to 300×300, and the stochastic gradient

descent algorithm was used to iteratively optimize the network in the training process. The momentum was set to 0.9, the batch size was 16, and the initial learning rate was 0.0001. When the number of iterations was 80 000 and 100 000, the learning rate was 0.00001 and 0.000001 respectively, and the total number of iterations was set to 120 000.

Name	Parameter
Software framework	PyTorch
Programming language	Python
System	Ubuntu 18.04.5 LTS
GPU	GeForce RTX 2080 Ti
CPU	Intel(R) Xeon(R) CPU E5-2690 V3 @ 2.60 GHz

Table 4. Experimental settings

4.3 Generation of Prior Boxes

SSD sets a series of prior boxes with different scales or aspect ratios for each unit of the feature map, and the detected target will adopt the prior boxes that are most suitable for their shape for training.

With an increase of the number of network layers, the size of the feature map decreases gradually, and the scale of the prior box increases linearly. The corresponding calculation formula is as follows:

$$S_k = S_{min} + \frac{S_{max} - S_{min}}{m-1}(k-1), \quad k \in [1, m], \tag{10}$$

where, m is the number of feature maps and is set to 5 in this work (the first feature map needs to be set separately, the scale factor is artificially set to 0.07). S_k is the ratio of the prior box size to the image size; S_{min} and S_{max} represent the minimum and maximum values of the ratio, respectively, and we find $S_{min} = 0.15$ and $S_{max} = 0.87$ to work better in our experiments.

For the first feature map, the scale is $300 \times 0.07 = 21$. For the other five feature maps, in order to facilitate the calculation, the scale ratio is first expanded by 100 times, then:

$$\delta = \frac{[S_{max} \times 100] - [S_{min} \times 100]}{m-1} = 18. \tag{11}$$

According to formula (10), there are:

$$S_k = S_{min} \times 100 + \delta(k-1). \tag{12}$$

In this way, $S_k \in [15, 33, 51, 69, 87]$, the data in S_k is divided by 100 and then multiplied by the size of the original map 300, and then the prior box size of the

first feature map is synthesized, and the prior box size of the six feature maps can be obtained as [21, 45, 99, 153, 207, 261]. For the last feature map, $S_{max\_size} = 300 \times \frac{87+18}{100} = 315$. The aspect ratio of the prior boxes is generally set as $\alpha_\gamma \in [1, 2, 3, 1/2, 1/3]$, but the prior boxes of 3 and 1/3 are not used when there are only four prior boxes.

The main parameters are shown in Table 5.

Layer	Out_size	Step	Prior_box_num	Min_size	Max_size
Dense block2	38×38	8	4	21	45
Dense block3	19×19	16	6	45	99
Conv5	10×10	32	6	99	153
Conv6-2	5×5	64	6	153	207
Conv7-2	3×3	100	4	207	261
Conv8-2	1×1	300	4	261	315

Table 5. The main parameters of the SSD network model

The coordinate of the center point of the prior box is $(\frac{i+0.5}{f_k}, \frac{j+0.5}{f_k})$, where f_k is the size of the k^{th} feature map, $i, j \in [0, f_k)$, and the width and height of the prior box are shown in Table 6.

Layer	Prior_box_ratio	Prior_box_size(width, height)
Dense block2	[1/2, 1, 2]	(21, 21), $(21 \times \sqrt{2}, 21/\sqrt{2})$, $(21/\sqrt{2}, 21 \times \sqrt{2})$, $(\sqrt{21 \times 45}, \sqrt{21 \times 45})$
Dense block3	[1/3, 1/2, 1, 2, 3]	(45, 45), $(45 \times \sqrt{2}, 45/\sqrt{2})$, $(45/\sqrt{2}, 45 \times \sqrt{2})$, $(45/\sqrt{3}, 45 \times \sqrt{3})$, $(45 \times \sqrt{3}, 45/\sqrt{3})$, $(\sqrt{45 \times 99}, \sqrt{45 \times 99})$
Conv5	[1/3, 1/2, 1, 2, 3]	(99, 99), $(99 \times \sqrt{2}, 99/\sqrt{2})$, $(99/\sqrt{2}, 99 \times \sqrt{2})$, $(99/\sqrt{3}, 99 \times \sqrt{3})$, $(99 \times \sqrt{3}, 99/\sqrt{3})$, $(\sqrt{99 \times 153}, \sqrt{99 \times 153})$
Conv6-2	[1/3, 1/2, 1, 2, 3]	(153, 153), $(153 \times \sqrt{2}, 153/\sqrt{2})$, $(153/\sqrt{2}, 153 \times \sqrt{2})$, $(153/\sqrt{3}, 153 \times \sqrt{3})$, $(153 \times \sqrt{3}, 153/\sqrt{3})$, $(\sqrt{153 \times 207}, \sqrt{153 \times 207})$
Conv7-2	[1/2, 1, 2]	(207, 207), $(207 \times \sqrt{2}, 207/\sqrt{2})$, $(207/\sqrt{2}, 207 \times \sqrt{2})$, $(\sqrt{207 \times 261}, \sqrt{207 \times 261})$
Conv8-2	[1/2, 1, 2]	(261, 261), $(261 \times \sqrt{2}, 261/\sqrt{2})$, $(261/\sqrt{2}, 261 \times \sqrt{2})$, $(\sqrt{261 \times 315}, \sqrt{261 \times 315})$

Table 6. The size of prior_box

4.4 Evaluation Indicators

In order to measure the robustness and accuracy of defect recognition, Precision, Recall, Average Precision (AP), and Mean Average Precision (mAP) are employed as the main evaluation indicators in this experiment.

Precision and Recall are defined as follows, respectively:

$$Precision = \frac{TP}{TP + FP}, \tag{13}$$

$$Recall = \frac{TP}{TP + FN}, \tag{14}$$

where TP is the number of IoU¿0.5 between the predicted and truth boxes, FP is the number of IoU¡0.5 between the predicted and truth boxes, and FN is the number of missed real boxes.

AP is the area enclosed by the Precision-Recall curve and coordinate axes. The mAP is the average of AP for different categories. Generally speaking, the higher AP value indicates the better target detection. The AP can be calculated by Equation (15), where $P(r)$ denotes the Precision-Recall curve.

$$AP = \int_0^1 P(r)dr, \tag{15}$$

$$mAP = \frac{1}{|C|}. \tag{16}$$

4.5 Result Analysis

To test the performance of the improved SSD, corresponding experiments were set for each improvement point of the algorithm, and the Precision, Recall, and mAP of the algorithm before and after the improvement are quantitatively analyzed.

4.5.1 A Comparison of the Underlying Backbone Networks

In this experiment, VGG-16 and DenseNet were respectively used as a backbone network. The experimental results are shown in Figure 8. For the model using DenseNet, the network structure is relatively more complex and the extracted features are more representational, so the detection effect is better. The mAP, Recall, and Precision are increased by 9.6 %, 8.7 %, and 9.3 %, respectively.

4.5.2 Evaluating Feature Fusion

Based on the backbone network of DenseNet, the feature fusion module is added, and Figure 9 shows its experimental results. The feature fusion module can effectively improve the Precision of the model and keep the mAP and Recall similar to the original model. It shows that the information gaps between different feature maps can be effectively filled through feature fusion and attention mechanism, and the primary and secondary information between different positions and channels of feature maps can be selectively activated or suppressed, which effectively improves the detection precision of defects.

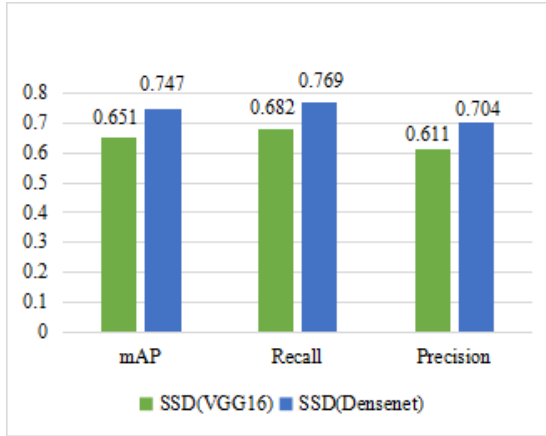


Figure 8. Performance comparison of different backbone network

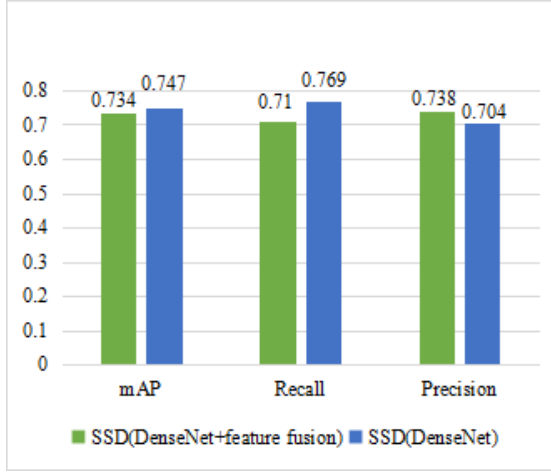


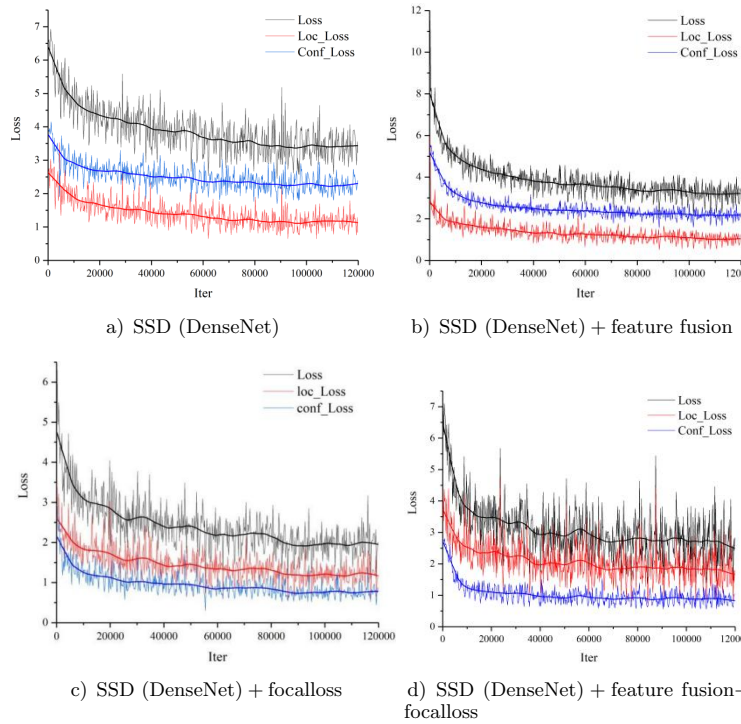
Figure 9. Performance comparison of feature fusion module

4.5.3 Evaluating Loss Function

Before and after adopting the improved strategy, the variation curve of the loss function is shown in Figure 10, where the horizontal coordinate represents the number of iterations and the vertical coordinate represents the loss value. It can be seen from the figure that with the increase of the number of iterations, the improved loss curve gradually decreases until it becomes stable. Compared with SSD (DenseNet), the localization loss in SSD (DenseNet) + focalloss remains almost the same, and the classification loss decreases significantly. Meanwhile, the convergence speed of SSD (DenseNet) + feature fusion is faster than that of SSD (DenseNet).

The training loss of SSD (DenseNet) converges around 4, and the convergence of SSD (DenseNet)+feature fusion+focalloss is around 3. After improvement, the convergence performance of the algorithm is obviously better than that of the original algorithm.

The trained network is used for target detection of defect images, and the results are shown in Table 7. It can be seen that the improved algorithm has significantly improved the target detection performance.



a) SSD (DenseNet)

b) SSD (DenseNet) + feature fusion

c) SSD (DenseNet) + focalloss

d) SSD (DenseNet) + feature fusion+ focalloss

Figure 10. Results of loss function

4.5.4 Comparative Experiment for Improvement Points

In this section, three improvement points of SSD algorithm are compared. Experiment 1: Only the feature extraction network of SSD algorithm is improved, and the VGG-16 network is replaced with the DenseNet network structure. Experiment 2: On the basis of the original SSD algorithm, only the feature fusion network is added. Experiment 3: Only the loss function of SSD algorithm is improved, and the focalloss function replaces the original confidence loss function. The bonding structure defect data set was used to test the above experiments, and the comparison results are shown in Table 8.

Models	AP (%)			mAP (%)	Recall (%)	Precision (%)
	Debonding	Cracking	Delamination			
SSD + DenseNet	63.6	71.4	89.1	74.7	76.9	70.4
SSD + DenseNet + feature fusion	69.7	62.5	87.9	73.4	71	73.8
SSD + DenseNet + focalloss	69.2	62	86.1	72.4	68.1	76.3
SSD + DenseNet + feature fusion + focalloss	69.1	68.2	90.5	75.9	75.6	77.3

Table 7. The results of the ablation experiment

	mAP (%)	Recall (%)	Precision (%)
Original SSD	65.1	68.2	61.1
Experiment 1	74.7	76.9	70.4
Experiment 2	64.5	72.1	62.8
Experiment 3	69.9	70.7	68.1

Table 8. Comparative experiment for improvement points

The mAP, Recall, and Precision of Experiment 1 have been significantly improved, which shows that using DenseNet as the feature extraction network can significantly improve the feature extraction ability, and then improve the detection accuracy of the model. In Experiment 2, after adding the feature fusion network, the Recall has been improved, and the mAP and Precision remain basically unchanged. In Experiment 3, the detection effect of the model with the introduction of focalloss function is better than that without the focalloss function, which shows that the focalloss function can balance the sample distribution, thereby improving the detection effect of the model. The improvement effect of Experiment 1 is significantly better than that of Experiment 2 and Experiment 3, and it can be inferred that the improvement of the feature extraction network plays a major role in improving the performance of target detection.

4.5.5 Horizontal Contrast Experiment

In order to further verify the detection performance of the improved algorithm in this paper, the SSD, YOLOv5s, and the improved SSD algorithm proposed in this paper were compared under the same experimental environment and defect data set.

It can be seen from Table 9 that the detection effect of the proposed algorithm is the best, followed by YOLOv5, and the original SSD is the worst. Compared with the original SSD algorithm, the mAP, Precision, and Recall of the proposed algorithm are increased by 10.8 %, 7.4 %, and 16.2 %, respectively. On the whole, the proposed algorithm is better than that of YOLOv5s, and its mAP, Precision,

Models	mAP (%)	Recall (%)	Precision (%)	FLOPs/G
YOLOv5s	70.1	74.5	69.1	16.4
Original SSD	65.1	68.2	61.1	30.54
Algorithm of this paper	75.9	75.6	77.3	11.23

Table 9. Detection results of different algorithms

and Recall are increased by 5.8 %, 1.1 %, and 8.2 %, respectively, which verifies the effectiveness of the proposed algorithm. However, in terms of the FLOPs, the algorithm of this paper is only about 1/3 of SSD and lower than YOLOv5s. The comparison of results adequately illustrated the superiority of the algorithm of this paper.

4.5.6 Visual Analysis of Detection Effect

The partial detection results of SSD algorithm for bonding defects are shown in Table 10. On the left are manually marked defects, and in the middle are the detection results of the original SSD, and on the right are the detection results of the improved SSD algorithm. It is obvious that the detection effect of the original SSD is bad, and the missed detection and false detection are serious. The improved SSD algorithm can detect more targets without any false detection at all.

5 CONCLUSIONS

This paper has presented a defect detection method based on an improved SSD algorithm. DenseNet is used as the basic feature extraction network, and the dilated convolution and CBAM attention modules are used to fuse the feature maps of each layer, which improves the feature reuse rate and further enhances the detection accuracy. The focal classification loss is introduced into the loss function of the algorithm to realize the balanced distribution of samples in the algorithm and improve the reliability of the model. The above improvements improve the mAP, Recall, and Precision of the model to different degrees, but the effect of improving the feature extraction network is the most obvious. The experiment results show that the mAP, Precision, and Recall of the improved SSD network are increased to 75.9 %, 77.3 %, and 75.6 %, respectively. Compared with the original SSD algorithm, the mAP, Precision, and Recall of the proposed algorithm are increased by 10.8 %, 7.4 %, and 16.2 %, respectively. The comparison experiments show that the optimized SSD algorithm has a better recognition effect than the original SSD algorithm.

In practical industrial production, the use of deep learning often faces the problem of a small amount of raw data and the unbalanced distribution of defect samples in the data set. To expand the defect image by deep convolutional generative adversarial network (DCGAN), not only the amount of data can be increased, but also the diversity of images can be increased, and the generalization ability of the

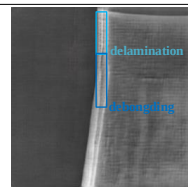
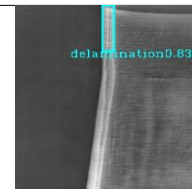
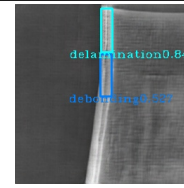
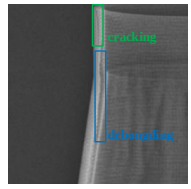
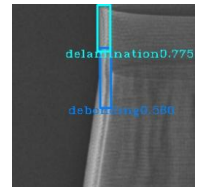
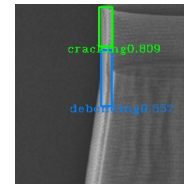
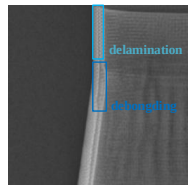
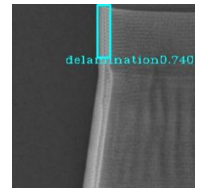
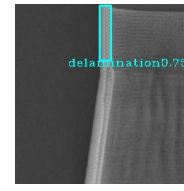
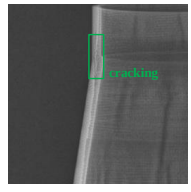
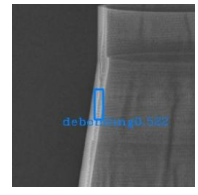
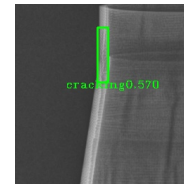
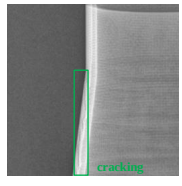
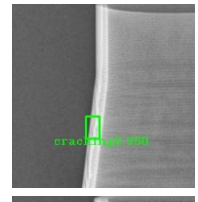
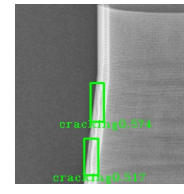
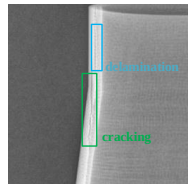
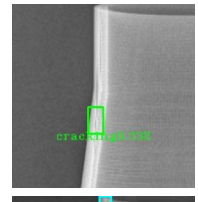
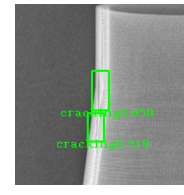
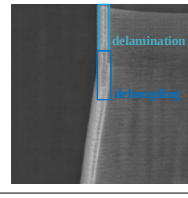
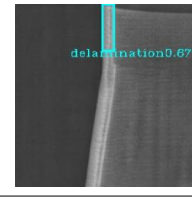
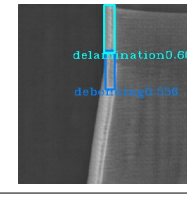
Image	Manual annotation	SSD	Algorithm of this paper
Image 1			
Image 2			
Image 3			
Image 4			
Image 5			
Image 6			
Image 7			

Table 10. A comparison of detection effects

trained model can be better improved. On this basis, it is worthwhile to improve the Precision and Recall of the model which is the direction of further in-depth research.

Acknowledgements

This work was supported by the Research Project supported by the Shanxi Scholarship Council of China under Grant No. 2022-145.

REFERENCES

[1] Schmid, M.—Bhogaraju, S. K.—Liu, E.—Elger, G.: Comparison of Nondestructive Testing Methods for Solder, Sinter, and Adhesive Interconnects in Power and Opto-Electronics. Applied Sciences, Vol. 10, 2020, No. 23, Art. No. 8516, doi: 10.3390/app10238516.

[2] Javadi, Y.—Sweeney, N. E.—Mohseni, E.—MacLeod, C. N.—Lines, D.—Vasilev, M.—Qiu, Z.—Vithanage, R. K. W.—Mineo, C.—Stratoudaki, T.—Pierce, S. G.—Gachagan, A.: In-Process Calibration of a Non-Destructive Testing System Used for In-Process Inspection of Multi-Pass Welding. Materials & Design, Vol. 195, 2020, Art. No. 108981, doi: 10.1016/j.matdes.2020.108981.

[3] Niu, J.—Chen, Y.—Yu, X.—Li, Z.—Gao, H.: Data Augmentation on Defect Detection of Sanitary Ceramics. IECON 2020 the 46th Annual Conference of the IEEE Industrial Electronics Society, 2020, pp. 5317–5322, doi: 10.1109/IECON43393.2020.9254518.

[4] Girshick, R.—Donahue, J.—Darrell, T.—Malik, J.: Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. 2014 IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 580–587, doi: 10.1109/CVPR.2014.81.

[5] Girshick, R.: Fast R-CNN. Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), 2015, pp. 1440–1448, doi: 10.1109/ICCV.2015.169.

[6] Ren, S.—He, K.—Girshick, R.—Sun, J.: Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. 2015, pp. 91–99, `https://proceedings.neurips.cc/paper_files/paper/2015/file/14bfa6bb14875e45bba028a21ed38046-Paper.pdf`.

[7] Redmon, J.—Divvala, S.—Girshick, R.—Farhadi, A.: You Only Look Once: Unified, Real-Time Object Detection. Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 779–788, doi: 10.1109/CVPR.2016.91.

[8] Liu, W.—Anguelov, D.—Erhan, D.—Szegedy, C.—Reed, S.—Fu, C. Y.—Berg, A. C.: SSD: Single Shot MultiBox Detector. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (Eds.): Computer Vision – ECCV 2016. Springer, Cham, Lecture Notes in Computer Science, Vol. 9905, 2016, pp. 21–37, doi: 10.1007/978-3-319-46448-0_2.

[9] Cao, J.—Song, C.—Song, S.—Peng, S.—Wang, D.—Shao, Y.—Xiao, F.: Front Vehicle Detection Algorithm for Smart Car Based on Improved SSD Model. Sensors, Vol. 20, 2020, No. 16, Art. No. 4646, doi: 10.3390/s20164646.

[10] Simonyan, K.—Zisserman, A.: Very Deep Convolutional Networks for Large-Scale Image Recognition. CoRR, 2014, doi: 10.48550/arXiv.1409.1556.

[11] Deng, J.—Dong, W.—Socher, R.—Li, L. J.—Li, K.—Fei-Fei, L.: ImageNet: A Large-Scale Hierarchical Image Database. 2009 IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 248–255, doi: 10.1109/CVPR.2009.5206848.

[12] Huang, G.—Liu, Z.—Van Der Maaten, L.—Weinberger, K. Q.: Densely Connected Convolutional Networks. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 2261–2269, doi: 10.1109/CVPR.2017.243.

[13] He, K.—Zhang, X.—Ren, S.—Sun, J.: Deep Residual Learning for Image Recognition. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770–778, doi: 10.1109/CVPR.2016.90.

[14] Yu, F.—Koltun, V.: Multi-Scale Context Aggregation by Dilated Convolutions. CoRR, 2015, doi: 10.48550/arXiv.1511.07122.

[15] Nair, V.—Hinton, G. E.: Rectified Linear Units Improve Restricted Boltzmann Machines. Proceedings of the 27$^{\text{th}}$ International Conference on International Conference on Machine Learning (ICML'10), 2010, pp. 807–814.

[16] Woo, S.—Park, J.—Lee, J. Y.—Kweon, I. S.: CBAM: Convolutional Block Attention Module. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (Eds.): Computer Vision – ECCV 2018. Springer, Cham, Lecture Notes in Computer Science, Vol. 11211, 2018, pp. 3–19, doi: 10.1007/978-3-030-01234-2_1.

[17] Fu, H.—Song, G.—Wang, Y.: Improved YOLOv4 Marine Target Detection Combined with CBAM. Symmetry, Vol. 13, 2021, No. 4, Art. No. 623, doi: 10.3390/sym13040623.

[18] He, K.—Gkioxari, G.—Dollár, P.—Girshick, R.: Mask R-CNN. 2017 IEEE International Conference on Computer Vision (ICCV), 2017, pp. 2980–2988, doi: 10.1109/ICCV.2017.322.

[19] Chu, J.—Guo, Z.—Leng, L.: Object Detection Based on Multi-Layer Convolution Feature Fusion and Online Hard Example Mining. IEEE Access, Vol. 6, 2018, pp. 19959–19967, doi: 10.1109/ACCESS.2018.2815149.

[20] Lin, T. Y.—Goyal, P.—Girshick, R.—He, K.—Dollár, P.: Focal Loss for Dense Object Detection. 2017 IEEE International Conference on Computer Vision (ICCV), 2017, pp. 2999–3007, doi: 10.1109/ICCV.2017.324.

[21] Jin, Y.—Gao, H.—Fan, X.—Khan, H.—Chen, Y.: Defect Identification of Adhesive Structure Based on DCGAN and YOLOv5. IEEE Access, Vol. 10, 2022, pp. 79913–79924, doi: 10.1109/ACCESS.2022.3193775.

[22] Make Sense Web Site. `https://www.makesense.ai/`.

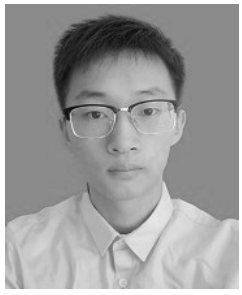
Huifang Gao is currently pursuing her Master's degree in the School of Information and Communication Engineering, North University of China. Her research interests are in deep learning and image processing.

Yong Jin received his Ph.D. degree from the North University of China, in 2013. He is currently Professor at the School of Information and Communication Engineering, North University of China. His research interests are in the areas of image processing, online inspections, and big data analytics.

Maozhen Li received his Ph.D. degree from the Institute of Software, Chinese Academy of Sciences, in 1997. He is currently Professor with the Department of Electronic and Computer Engineering, Brunel University London, U.K. His main research interests include high-performance computing, big data analytics, and intelligent systems with applications to smart grids, smart manufacturing, and smart cities. He has over 180 research publications in these areas, including four books. He has served over 30 IEEE conferences. He is also a Fellow of the British Computer Society and the IET. He is on the editorial board of several journals.

Youxing Chen received his Ph.D. degree from the North University of China, in 2010. He is currently Professor at the School of Information and Communication Engineering, North University of China. His research interests include the areas of image processing, signal processing, and non-destructive testing.

Junbin Zang received his B.Sc. degree in computer science and technology and his M.Sc. degree in precision instrument and machinery from the North University of China, Taiyuan, China, where he is currently pursuing his Ph.D. degree in instruments and electronic engineering at the Key Laboratory of Instrumentation Science and Dynamic Measurement, Ministry of Education. His current research interest includes the design of the MEMS stethoscope.



Xiaoliang Fan is currently pursuing his Ph.D. degree in the School of Earth Science and Engineering, Nanjing University. His research interests are soil desiccation cracking as well as machine identification and self-healing of cracks.