

# A Lightweight Reconstruction Network for Surface Defect Inspection

Chao Hu, Jian Yao, Weijie Wu, Weibin Qiu and Liqiang Zhu

AI Lab, Unicom (Shanghai) Industry Internet Co., Ltd.,  
Shanghai 200000, China.

## Abstract

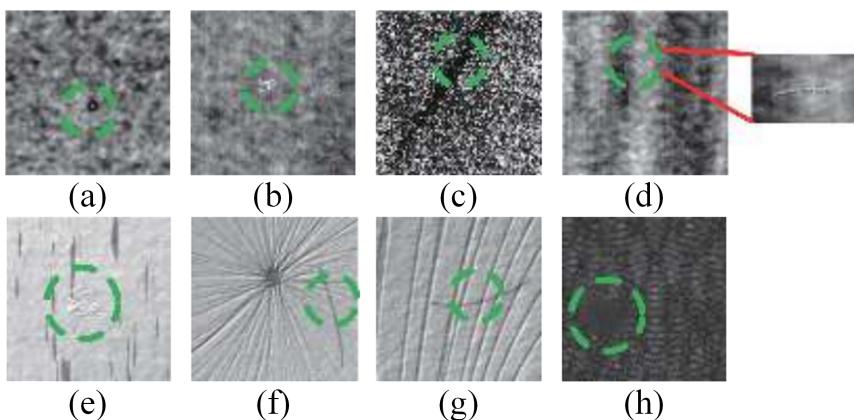
Currently, most deep learning methods cannot solve the problem of scarcity of industrial product defect samples and significant differences in characteristics. This paper proposes an unsupervised defect detection algorithm based on a reconstruction network, which is realized using only a large number of easily obtained defect-free sample data. The network includes two parts: image reconstruction and surface defect area detection. The reconstruction network is designed through a fully convolutional autoencoder with a lightweight structure. Only a small number of normal samples are used for training so that the reconstruction network can be A defect-free reconstructed image is generated. A function combining structural loss and  $L1$  loss is proposed as the loss function of the reconstruction network to solve the problem of poor detection of irregular texture surface defects. Further, the residual of the reconstructed image and the image to be tested is used as the possible region of the defect, and conventional image operations can realize the location of the fault. The unsupervised defect detection algorithm of the proposed reconstruction network is used on multiple defect image sample sets. Compared with other similar algorithms, the results show that the unsupervised defect detection algorithm of the reconstructed network has strong robustness and accuracy.

**Keywords:** Auto encoder; surface defects; abnormal defects; visual inspection; unsupervised defect



# 1 Introduction

Traditional machine learning methods can effectively solve various industrial product quality detection problems, such as bearings<sup>[21]</sup>, mobile phone screens<sup>[23]</sup>, coils<sup>[4]</sup>, rails<sup>[30]</sup>, steel beams<sup>[32]</sup>, etc. Such methods use artificial feature extractors to adapt to a specific product image sample data set, the characteristics of the classifier and support vector machine<sup>[10]</sup>, and the neural network<sup>[35]</sup> to determine whether the product is defective. However, when the surface defects of the detected product appear, such as complex background texture (including regular and irregular), large changes in the scale of defect features, and similar defect area features and background features (as shown in Figure 1), the traditional machine learning method relies on the ability of artificial features to represent product image samples. It does not adapt to such complex detection needs. Figure 1(a) is a dark defect, and Figure 1(b) is a light defect. Figure 1(c) is a large-scale defect covering an image. Figure 1(d) is a minor defect. Figure 1(f) (g) shows a defect similar to a texture. Figure 1(d) shows a defect with small chromatic aberration. Figure 1(h) is a fuzzy defect.



**Fig. 1** Various surface defects

Since AlexNet<sup>[33]</sup> was proposed, deep learning methods based on convolutional neural networks (CNNs) have become the mainstream methods<sup>[?]</sup> in surface defect detection. convolutional neural networks can automatically learn image features and extract more abstract image features through the superposition of multiple convolutional layers, which has better feature representation capabilities than manually designed feature extraction algorithms. According to the results of the network output, the algorithm of defect detection by deep learning method can be divided into defect classification method, defect identification method, and defect segmentation method.

Defect classification algorithms are usually trained using classic classification network algorithms to treat the detected samples, and the learned model can classify defects and non-defect categories. Such methods usually do not

involve the localization of defect areas. Wang et al.<sup>[24]</sup> proposed using 2 CNN networks for defect detection of 6 types of images. Xu et al.<sup>[1]</sup> proposed a CNN classification network that integrates visual geometry (VGG) and residual networks to detect and classify roll surface defects. Paolo et al.<sup>[26]</sup> and Weimer et al.<sup>[22]</sup> also use CNN's image feature representation capabilities to identify defects.

## 2 Related Work

In order to achieve accurate localization of defect areas, some researchers have improved and applied the excellent network in vision recognition tasks to surface defect detection. Such algorithms are primarily based on regional convolutional neural networks<sup>[34]</sup>, single-excitation multi-box detectors<sup>[8]</sup>, and You only look once (YOLO)<sup>[13]</sup> and other networks. Chen et al.<sup>[25]</sup> applied deep convolutional neural networks to fastener defect detection. Cha et al.<sup>[14]</sup> used regional convolutional neural networks (Region-CNN, R-CNN) for structural visual inspection in construction.

In order to achieve pixel-level detection accuracy, some researchers have used segmentation networks, such as Huang et al., to convert defect detection tasks into semantic segmentation tasks, which improves the accuracy of tile surface detection. Qiu et al. use a full convolutional network (FCN) to detect defect areas.

In many industrial situations, product defect types are unpredictable and occur only during the production process, making it difficult to collect a large number of defect samples. In response to these problems, researchers began focusing on small or unsupervised samples. Such methods all rely on a certain amount of training data. Various methods improved based on autoencoders are used for surface defect detection, such as convolutional autoencoder (CAE)<sup>[20]</sup>, stacked noise-canceling autoencoders<sup>[19]</sup> based on Fisher guidelines, robust autoencoders<sup>[3]</sup>, sparse denoising self-coding networks<sup>[31]</sup> that fuse gradient difference information, etc. Xi methods, such as Yu et al.<sup>[?]</sup>, use the YOLOV3 network to achieve high-accuracy detection results under the training conditions of a few defective samples. Mei et al.<sup>[5]</sup> proposed that a multi-scale convolutional denoising autoencoder network (MSCDAE<sup>textsuperscript[5]</sup>) reconstructs images and generates detection results using reconstruction residuals, compared with traditional unsupervised algorithms such as Phase only transform (PHOT<sup>[18]</sup>) and discrete cosine transform<sup>[5]</sup>. There is an excellent improvement in the model evaluation index. Yang et al.<sup>[9]</sup> improved the reconstruction accuracy of texture backgrounds by using feature clustering based on MSCDAE. The above-reconstructed networks all use the Mean square error (MSE) loss function with normal terms, and the data samples are mostly regular surface textures.

In addition to autoencoders, generative adversarial networks<sup>[6]</sup> are also used for unsupervised defect detection. By learning a large number of normal image samples, the generator in the network can learn the data distribution of

normal sample images. Zhao et al. [12]. combine generative adversarial networks and autoencoders, make defects on defect-free samples, and train generative adversarial networks to have the ability to recover images. He et al. [16]. used semi-supervision to generate anti-network and self-encoder training unlabeled steel surface defect data, extracted fine-grained features of the image, and classified them. Schlegl et al. [7]. propose anomaly detection to generate anomaly detection of lesion images under unsupervised conditions by generating adversarial networks. In practice, generative adversarial networks also have problems such as unstable performance and difficulty [27]. in training.

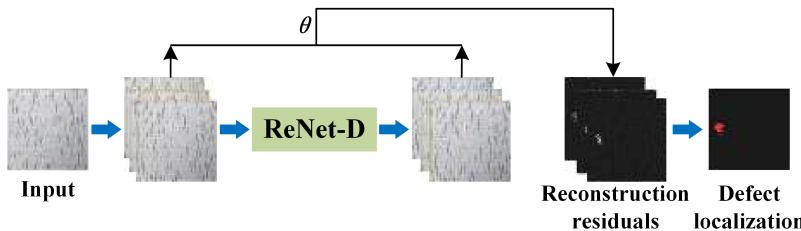
Considering the complexity and scarcity of defect samples in industrial application scenarios, this paper proposes a reconstituted network detection method (ReNet-D) based on a reconstructed network, which uses a small number of defect-free samples as objects for network model learning. The reconfiguration training of the sample image enables the network to have the ability to reconstruct the positive sample. When the abnormal sample is input, the trained network model can detect the abnormal area of the sample image. In this paper, the network structure, training block size, loss function coefficient, and other influencing factors of the ReNet-D method are analyzed and evaluated in detail to adapt to the detection requirements of regular textures and irregular textures time and compare experiments with other classical algorithms.

### 3 Methodology

In practical industrial applications, factors such as the scarcity of defective samples, significant differences in characteristics, and the accidental appearance of unknown defects make it challenging to apply supervised algorithms driven by large data samples. The unsupervised algorithm proposed in this paper solves the problem of missing insufficient sample data for model learning. The algorithm is divided into two training stages: the image reconstruction network training stage and the surface defect area detection stage. The reconstruction network is designed by the full convolutional autoencoder, and only a small number of normal samples are used for training so that the reconstruction network can generate defect-free reconstruction images. The defect detection stage takes the residuals of the reconstructed image and the image to be tested as the possible areas of defects, and the final detection results are obtained through conventional image operation. The model of the ReNet-D algorithm is shown in Figure 2.

#### 3.1 Reconstruction Network

The surface defects of industrial products have multi-scale characteristics, similar to the background texture and complex shape, which require high accuracy and operation time of the detection algorithm. Therefore, there are three requirements for refactoring the network design: 1) the network can adapt to defect areas of different scales; 2) The network needs to identify whether there



**Fig. 2** ReNet-D algorithm architecture

are defective features in the sample area; 3) Reconstruct the network model with as few parameters as possible.

The reconstruction process is usually decomposed into encoding transforms  $\varphi$  and decoding transforms  $\gamma$ , defined as follows:

$$\begin{aligned} \varphi : I &\rightarrow F \\ \gamma : F &\rightarrow I \\ (\varphi, \gamma) &= \arg \min \|I - \gamma(\varphi(F))\|^2 \end{aligned} \quad (1)$$

In Eq. (1),  $I \in \mathbf{R}^{W \times H}$  represents the spatial domain of the image sample, mapped to the hidden space by function  $\varphi$ ,  $F$  represents the corresponding image sample feature in the hidden space, and  $\varphi$  is implemented by the coding module.  $\gamma$  remaps the image sample feature  $F$  corresponding to the hidden layer space back to the spatial domain of the original image sample, which is implemented by the decoding module.

Where  $z = \varphi(I) \in F$ , the encoding and decoding process is described as:

$$\begin{aligned} z &= \sigma(W \circ I + b) \\ I' &= \sigma(W' \circ z + b') \end{aligned} \quad (2)$$

In Eq. (2),  $I'$  represents a refactored image,  $\circ$  represents convolution,  $\sigma$  signifies an activation function.  $W$  and  $W'$  represent the encoding convolutional kernel and the decoding convolutional kernel, respectively.  $b$  and  $b'$  indicate the encoding bias and the decoding bias.

The network structure of ReNet-D is shown in Figure 3. In order to adapt to a larger image, the original image is divided into several image blocks, usually  $16 \times 16$ ,  $32 \times 32$  and  $64 \times 64$ , as inputs to the network. ReNet-D uses the three convolutional kernels of  $1 \times 1$ ,  $3 \times 3$  and  $5 \times 5$  to obtain multi-scale features, and inputs the multi-scale features into the coding module. The results of the decoding module output are fed into three deconvolution layers of different scales to obtain the final reconstruction image, which can also obtain multi-scale features compared with the Gaussian pyramid sampling model of MSCDAE, but the computational cost is reduced. ReNet-D's CAE module

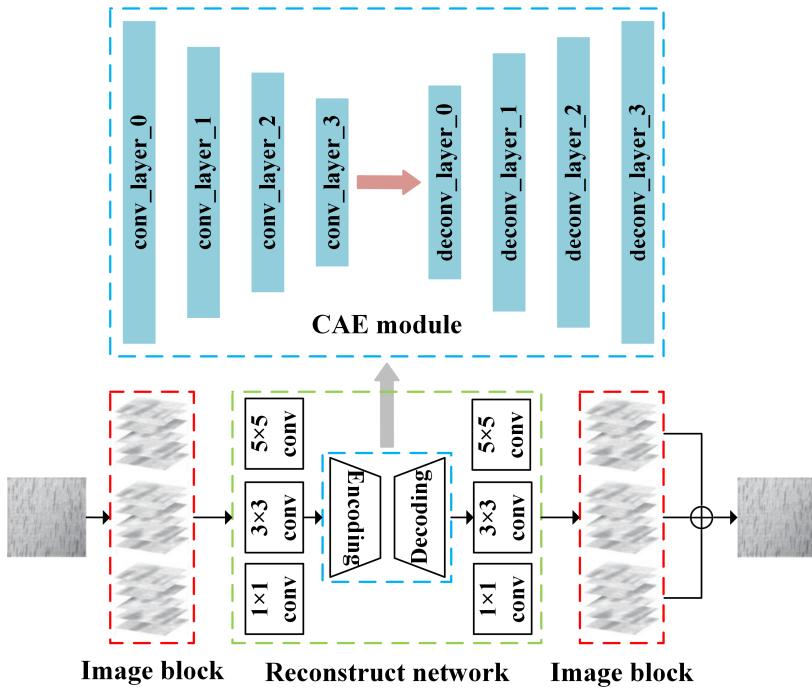


Fig. 3 ReNet-D network structure

consists of 4 convolutional modules and 4 deconvolution modules. Each convolutional module contains a convolutional layer, a batch normalization (BN<sup>[11]</sup>) layer and a nonlinear activation layer, and the first three convolutional modules also include a pooling layer that can change the image scale. The activation function uses Relu. The first 3 convolutional layers use  $5 \times 5$  convolutional kernels, and the last layer uses  $3 \times 3$  convolutional nuclei.

The mechanism of the self-encoder for defect detection is to complete the reconstruction of the defect-free background based on the high sensitivity to the defect-free environment, and to complete the imperfect reconstruction of the low sensitivity of the defective area, to realize the detection of defects. The autoencoder is a neural network with the same input and learning objectives. The depth of the network level also determines the reproduction ability of the autoencoder to the input image. If the model with a complex network structure is adopted, the ability to reproduce the sample features can be improved. Still, it also causes the reproduction ability of the defect area to be improved. ReNet-D adopts a lightweight structural design and limited capacity for model reconstruction. Still, through the design of multi-scale features and loss functions, the network can fully learn the characteristics of normal textures and obtain information on defect parts for the imperfect reconstruction of defect areas.

### 3.2 Loss Function

In the training phase of ReNet-D, the reconstruction error between the original and reconstructed images is used as a loss function to promote network convergence. The following analyzes and improves the existing loss function for evaluating reconstruction errors.

**Mean Squared error loss.** The textured background of most industrial products is irregular, the abnormal features are easy to integrate into the textured background, and the difference between the abnormal features and the normal texture background features is small. Define  $L2$  as the mean absolute error loss:

$$L2 = \|I_{src} - I_{rec}\|_2^2 + \lambda\|\omega\|_F \quad (3)$$

In the Eq. (3),  $I_{src}$  represents the original image of the input,  $I_{rec}$  represents the image of the model reconstruction,  $\omega$  represents the set of weights in the reconstruction network, and  $\lambda$  represents the penalty factor of the regularization term,  $0 < \lambda < 1$ . The reconstruction algorithm model with MSE as the loss function is suitable for image samples with regular texture backgrounds, such as textiles<sup>[5, 9]</sup>.

**Average absolute error loss.** The textured background of most industrial products is irregular, the abnormal features are easy to integrate into the textured background, and the difference between the abnormal features and the normal texture background features is slight. Define  $L1$  as the mean absolute error loss:

$$L1 = \|I_{src} - I_{rec}\|_1 + \lambda\|\omega\|_F \quad (4)$$

Compared with  $L1$  loss,  $L2$  loss is more sensitive to outliers and overpasses significant loss errors, such as the MSCDAE<sup>[5]</sup> method, so ReNet-D introduces  $L1$  loss to optimize network training.

**Structural loss.** When evaluating the effect of reconstructing the network model,  $L1$  loss and  $L2$  loss are compared on a pixel-by-pixel basis, and the regional structure of the image is not considered. For the detection of some irregular texture image samples, ReNet-D introduces the Structural similarity index (SSIM<sup>[28, 36]</sup>) to construct the loss function so that the network can apply complex and changeable texture background samples to reconstruct better results. The SSIM loss function optimizes the model from three indicators(brightness, contrast, and structure)<sup>[17]</sup>, and the results reflect the image details better than the  $L1$  or  $L2$  loss functions. For image pairs  $(x, y)$  for model input and output, SSIM is defined as:

$$\begin{aligned} SSIM(x, y) &= (l(x, y))^\alpha (c(x, y))^\beta (s(x, y))^\gamma \\ l(x, y) &= \frac{2u_xu_y + C_1}{u_x^2 + u_y^2 + C_1} \\ c(x, y) &= \frac{2\sigma_{xy} + C_2}{\sigma_x^2 + \sigma_y^2 + C_2} \\ s(x, y) &= \frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3} \end{aligned} \quad (5)$$

where  $\alpha > 0, \beta > 0, \gamma > 0$ ,  $l(x, y)$  represents brightness ratio,  $c(x, y)$  represents contrast comparison, and  $s(x, y)$  represents structural comparison.  $u_x$  and  $u_y$  are the mean of  $x$  and  $y$ , respectively, and  $\sigma_x$  and  $\sigma_y$  are the standard deviations of  $x$  and  $y$ .  $\sigma_{xy}$  is the covariance of  $x$  and  $y$ .  $C_1, C_2, C_3$  are non-zero constants, usually  $\alpha = \beta = \gamma = 1$ ,  $C_3 = C_2/2$ .

The loss function of SSIM is defined as:

$$L_{SSIM}(x, y) = 1 - SSIM(x, y) \quad (6)$$

The SSIM loss function is used to evaluate the difference between the output result of the last layer and the original image in the reconstruction network, and multiple deconvolutional layer results of different scales can be extracted and the corresponding convolutional layer results can be used at the same time to use the SSIM loss function to construct multi-scale SSIM<sup>[15]</sup> (Multi-scale SSIM, MS\_SSIM). For M scales, the MS\_SSIM loss function is defined as:

$$L_{MS\_SSIM}(x, y) = 1 - \prod SSIM(x, y) \quad (7)$$

**Loss Function of ReNet-D.** Compared with MSE loss, the  $L1$  loss has a weaker penalty for pixel-level error, which is suitable for irregular texture samples. At the same time,  $L_{SSIM}$  can train the reconstruction network to pay attention to the brightness change and color deviation of the sample image to retain the high-frequency information of the image, that is, the image edge and detail. To solve the problem of defect detection of regular and random texture image samples at the same time, this paper designs a loss function combining  $L1$  loss and  $L_{SSIM}$  as the loss function of the ReNet-D network model, as follows:

$$L_{ReNet-D} = \alpha L1 + (1 - \alpha) L_{SSIM} \quad (8)$$

In the Eq. (8),  $\alpha$  is the weight factor, and the value range is  $(0, 1)$ , which is used to balance the proportion of  $L1$  loss and  $L_{SSIM}$ , and this paper will experimentally compare the influence of different weight factors and loss functions on the ReNet-D detection results.

### 3.3 Defect Area Locate

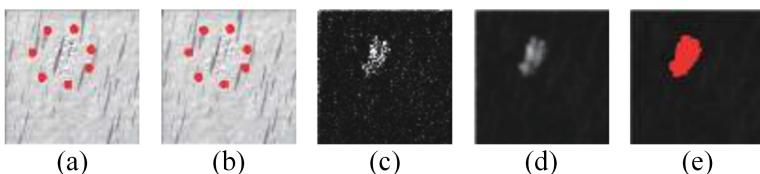
In the detection stage, the network will output an approximate defect-free image after the defect image input is trained to reconstruct the network. That is, the reconstructed network will "repair" the defective area into a normal area while maintaining the defect-free area. According to this characteristic, the pixel-level difference between the output image and the input image, after conventional image processing technology, can accurately locate the defect area, the processing process is as follows:

- Residual plot acquisition: The input image (shown in Figure 4(a)) and the ReNet-D reconstruction image (shown in Figure 4(b)) are used to make a difference shadow to obtain the reconstruction error of the network for the

defective area, and the obtained residual map is shown in Figure 4(c), which contains the position information of the abnormal area. Among them, Fig. 4(a) is the original image of the input model, Fig. 4(b) is the ReNet-D reconstruction diagram, Fig. 4(c) is the residual graph  $v$ , and  $(i, j)$  in Eq. (9). Fig. 4(d) is the residual map filtering, and Fig. 4(e) is the defect location.

$$v(i, j) = (I_{src}(i, j) - I_{rec}(i, j))^2 \quad (9)$$

- Denoising processing: Fig. 4(c) of the residual plot shows a lot of noise, forming pseudo-defects, affecting the judgment of the real defect area, and using mean filtering for denoising to obtain Fig. 4(d).
- Thresholding segmentation and defect localization.



**Fig. 4** The residual graph processing flow location

- The final result is obtained using the adaptive threshold method Figure 4(e).

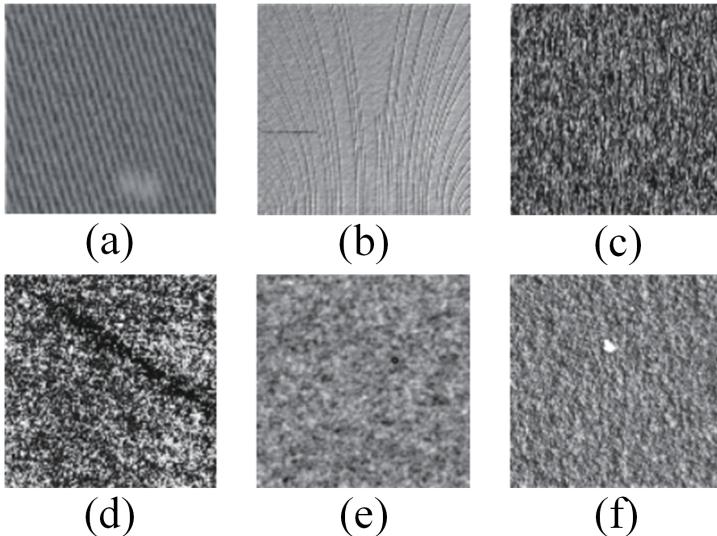
## 4 Experiments

In this paper, the proposed detection algorithm ReNet-D is extensively evaluated on the surface data of industrial products. Firstly, the data set used in the experiment is introduced, and then the critical indicators of model evaluation are introduced, and then the influencing factors of the detection effect of the ReNet-D algorithm include loss function, network structure, image block, and the detection effect of different types of defects of similar materials are analyzed in detail. Finally, the proposed detection algorithm is compared with other parallel unsupervised algorithms.

### 4.1 Dataset

To objectively evaluate the proposed detection algorithm, this experiment establishes a verification data set composed of texture samples of various materials, as shown in Figure 5, where Figure 5(a) is derived from the AITEX<sup>[2]</sup> dataset, which is derived from the textile industry, the sample is a regular texture, the number of positive and negative examples is 149/5, and Figure 5(b) – (e) samples are derived from the DAGM2007<sup>[29]</sup> data set, the data set has two characteristics: texture irregularity and defect area and picture scale, defects are hidden in the texture and structure, and texture the theory is very similar, where the number of positive and negative samples in Figure

5(b) is 100/29, the number of positive and negative samples in Figure 5(c) is 100/6, and the number of positive and negative samples in Figure 5(d) is 101/6. Figure 5(f) samples from the Kylberg Sintorn dataset, plus or minus sample size 50/5. In addition to the dataset shown in Figure 5, samples of the MVtech<sup>[27]</sup> unsupervised dataset were added for comparative experiments.



**Fig. 5** Surface defect data set used in the experiment

## 4.2 Model Evaluation Metrics

This paper evaluates the performance of the algorithm through pixel-level metrics, using three evaluation indicators: recall rate (Recall), precision rate (Precision) and the weighted harmonic average of the two (*F1-Measure*), which are defined as follows:

$$\begin{aligned}
 \text{Recall} &= \frac{TP_p}{TP_p + FN_p} \times 100\% \\
 \text{Precision} &= \frac{TP_p}{TP_p + FP_p} \times 100\% \\
 \text{F1-Measure} &= \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}
 \end{aligned} \tag{10}$$

Eq. (10),  $TP_p$  is the proportion of correctly segmented defect areas in the foreground,  $FP_p$  is the proportion of incorrectly segmented defect areas in the background, and  $FN_p$  is the proportion of defect areas that are not detected in the defect area. *F1-Measure* evaluates recall and precision. All tests were

performed on a computer equipped with a graphics processor, as shown in Table 1.

**Table 1** Computer system configuration

System	Ubuntu 16.04
Memory	128 GB
Graphics Processor	NVIDIA GTX-1080 Ti
CPU	Intel E5-2650 v4@2.2 GHz
Deep Learning Framework	Pytorch, CUDA 9.0, CUDNN 5.1

### 4.3 Loss Function Comparison Experiment

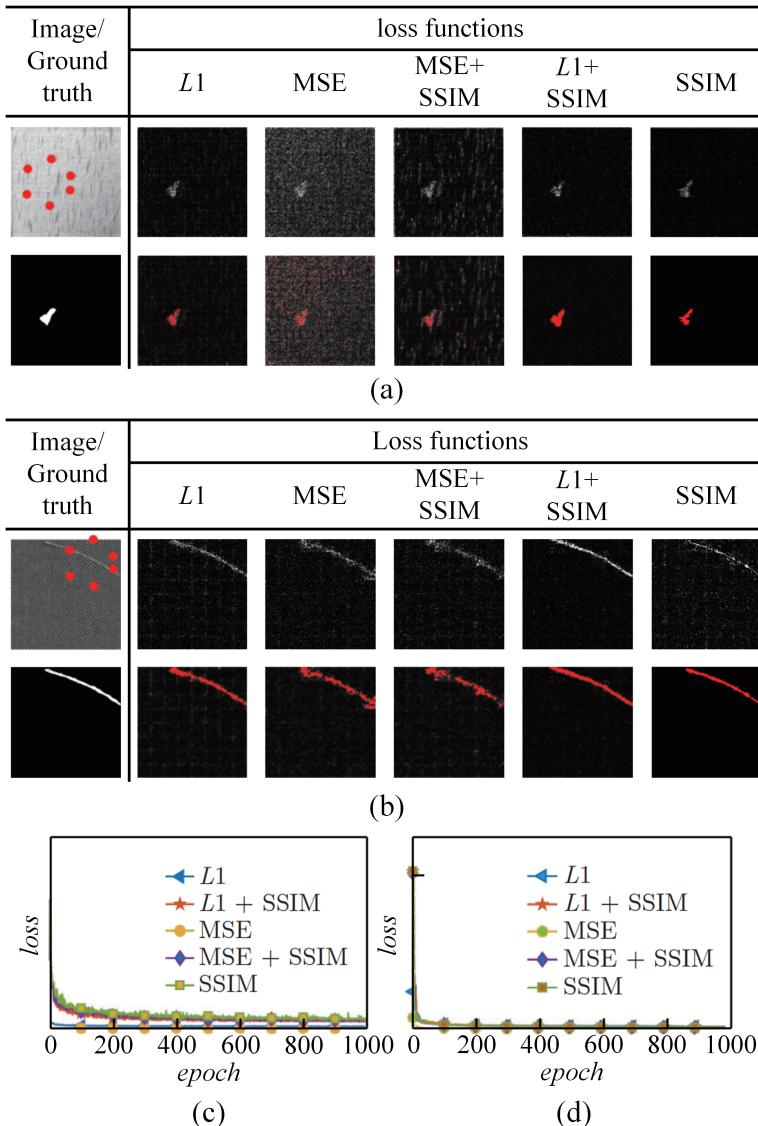
ReNet-D selects the loss functions MSE,  $L1$ , SSIM and the combination of the three for comparative experiments to evaluate the performance of the loss function proposed by Eq. (8) in the defect detection task. In this experiment, the ReNet-D network parameters are set As shown in table 2.

Figure 6(a) and Figure 6(b) are the experimental results of two different product surface defect samples under various loss functions, of which Figure 6(a) is an irregular texture sample and Figure 6(b) is a regular texture sample. Figure 6(c) is the convergence test of sample (a) under different loss functions, Figure 6(d) is the convergence test of sample (b) under different loss functions, and Figure 6(a) uses an irregular surface texture Defect image sample. From the comparison of residual results, it can be seen that the residual results obtained by MSE as the loss function of the algorithm have more noise points in other areas except the actual defect area, forming false defects; while using SSIM alone as the loss function, the detection The defect area is slightly smaller than the actual defect area; compared with other loss functions, the combination of the structural loss function SSIM and the  $L1$  loss function achieves better results.

Figure 6(b) Defect image sample with regular surface texture. From the comparison of residual results, it can be seen that the integrity of the defect area obtained by using the MSE loss function is poor, similar to the detection result of the MSE + SSIM. Structural loss function The combination of SSIM and  $L1$  loss function achieves better results, and the detection results are similar to using only the  $L1$  loss function. Figure 6(c) and Figure 6(d) compare the convergence trends of ReNet-D under different loss functions.

**Table 2** Default network parameters

Block Size	32×32
Number of samples	256
Iteration steps	1000
Loss weights $\alpha$	0.15



**Fig. 6** ReNet-D detection results under different loss functions

The comparison results in Table 3 show that for the defect samples with irregular surface texture in Figure 6(a), the combination of the structural loss function SSIM and the  $L1$  loss function achieves better recall and weighted harmonic mean results. It is slightly inferior to the loss function SSIM in precision. In the defect samples with regular surface texture shown in Fig. 6(b), only the use of the  $L1$  loss function achieves the highest recall rate, followed by the combination of the SSIM and  $L1$  loss functions. The loss function SSIM

**Table 3** Comparison of test results under different loss functions

Metrics	Samples	<i>L1</i>	MSE	MSE + SSIM	<i>L1 + SSIM</i>	SSIM
Recall	irregular samples	0.51	0.38	0.5	<b>0.75</b>	0.59
	regular samples	<b>0.76</b>	0.70	0.67	0.71	0.59
Precision	irregular samples	<b>0.93</b>	0.35	0.52	0.89	<b>0.93</b>
	regular samples	0.84	0.65	0.70	0.87	<b>0.96</b>
Weights	irregular samples	0.66	0.36	0.51	<b>0.82</b>	0.72
	regular samples	<b>0.80</b>	0.67	0.69	0.78	0.73

achieves the highest accuracy, followed by the variety of SSIM and *L1* loss functions. For weighted harmonic averaging, the *L1* loss function performs the best.

Further, by comparing the detection effects of the sample library used in this study, the loss function has the following rules:

- 1) For regular surface texture samples, the above four loss functions can be used to detect defects, among which the results of using only the MSE and MSE + SSIM loss functions are relatively poor, and the results of the other two loss functions are slightly different.
- 2) For irregular surface texture samples, the detection results obtained using the combined loss function of *L1 + SSIM* are better.

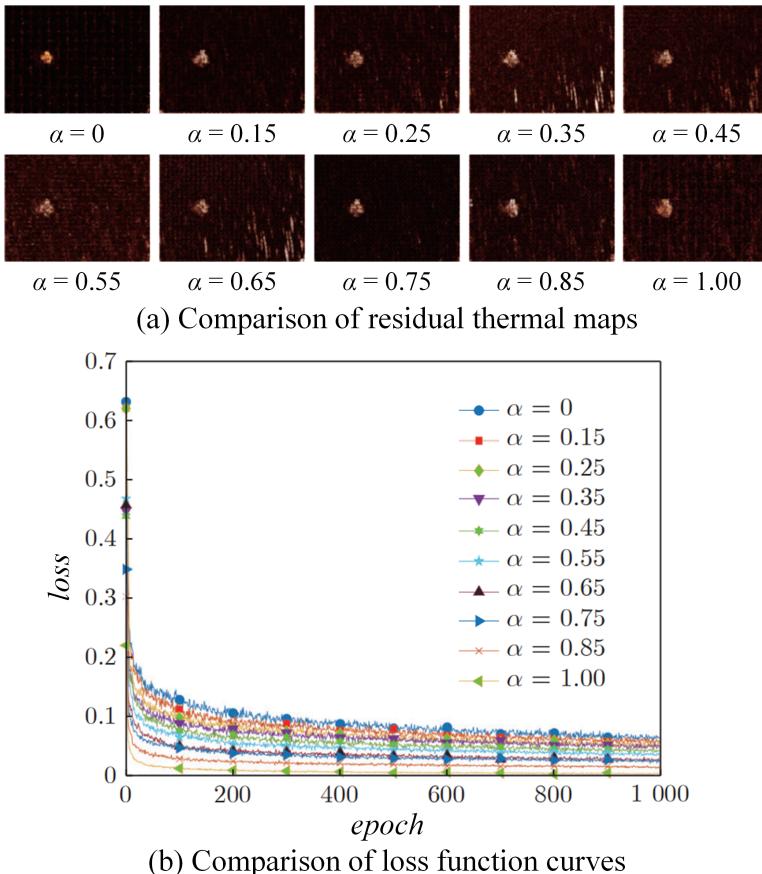
**Table 4** Comparison of test results under different loss functions

Weights	0	0.15	0.25	0.35	0.45	0.55	0.65	0.75	0.85	1
Recall	0.72	<b>0.79</b>	0.62	0.73	0.65	0.67	0.52	0.55	0.72	0.45
Precision	0.71	0.69	0.58	0.28	0.46	0.53	0.23	<b>0.89</b>	0.54	0.62
Weights	0.71	<b>0.73</b>	0.60	0.41	0.54	0.60	0.32	0.68	0.62	0.52

## 4.4 Ablation Experiments

### 4.4.1 Different weight coefficients

The above experiments show that the ReNet-D model uses the combined loss function of *L1 + SSIM*, which can be applied to defect detection of both regular and irregular surface textures. For normal texture samples, only using the *L1* loss, that is, the weight coefficient  $\alpha = 1$ , can obtain a relatively For irregular texture samples, different weight coefficients will produce different detection effects. This experiment uses the irregular texture samples in Figure 6(a). The weight coefficient  $\alpha$  ranges from 0 to 1, and the step size is set to 0.1 and



**Fig. 7** Comparison of ReNet-D performances under different weight coefficients

is used to adjust the proportion of SSIM loss and  $L1$  loss. The comparison experiment is shown in Figure 7, in which Figure 7(a) is the residual heat map comparison. Figure 7(b) is the training loss curve comparison.

According to Eq. (8), when  $\alpha$  increases, the influence of structural loss SSIM gradually decreases. It can be seen from Figure 7 and Table 4 that the residual map changes significantly. When  $\alpha = 0.15$ , the defect detection effect is better, the signal-to-noise ratio is the lowest, and the recall rate and weighted harmonic average are the best. Through experiments with multiple samples, the practical advice given in this paper is for regular texture samples to set the weight  $\alpha = 1$ , which only uses the  $L1$  loss as the training model. The loss function of irregular texture samples, set  $\alpha = 0.15$ , so that the weight of the structural loss effect is too large to obtain the best results.

#### 4.4.2 Different kinds of defects

The surface texture defect classes in the unsupervised dataset include leather, wood, carpet, grid and tiles, among which the tile class has the most messy textures. The training set includes 230 non-defective irregular texture normal type images; the test set includes 5 There are 17 damage defects, 18 tape defects, 16 gray smear defects, 18 oil stain defects, and 15 wear scar defects. In this experiment, the default network parameters are used to perform the Tile data set. After training, the detection results of 5 types of defects are obtained as shown in Figure 8.

	Greasy dirt	Breakage	Abrasion	Smear	Adhesive tape
Images					
Ground truth					
Results					

**Fig. 8** Test results of unsupervised samples

As shown in Table 5, the ReNet-D method can adapt to the detection of different types of defects in the Tile data set, among which oil stains, damage and wear scars have better performance, and its detection precision, recall and weighted harmonic average are better , but in the detection of smear and tape defects, the defect area with different color and texture from the background can be detected, but the overall shape of the generated defect is different from the ideal detection result, resulting in a low pixel-level evaluation index.

**Table 5** Test results of unsupervised samples

Category	Recall	Precision	weights
dirt	0.71	0.94	0.80
damaged	0.66	0.48	0.55
crack	0.63	0.89	0.70
smear	0.27	0.47	0.32
tape	0.16	0.35	0.20

The characteristic of ReNet-D is that the network has a better reconstruction effect for the components similar to the background, and the reconstruction effect is poor for the parts that are not similar to the environment, so this feature can be used to detect defects that are different from the background. The analysis of the tape defect characteristics found that the local area's color is very close to the texture and environment or even coincides with it. The detection effect of the region is not affected. Although it is not ideal regarding recall rate and other indicators, in industrial detection, local detection can be regarded as defect detection.

This paper compares the efficiency of the algorithms. The experiment uses  $1024 \times 1024$  pixel sample images. Under the same computing performance, the processing time of the four methods is compared, as shown in Table 6. After the ReNet-D algorithm model is trained, the size is less than 1 MB bytes, and the average detection time is 2.82 ms, which can meet the requirements of industrial real-time detection. Other methods are time-consuming, which limits their practical application.

**Table 6** Comparison of processing time (ms)

Methods	PHOT	LCA	MSCDAE	ReNet-D
Time	450	430	9746.59	2.82

## 5 Conclusions

This paper proposes a novel ReNet-D for the visual detection of surface defects using a reconstruction network. This method uses a fully convolutional autoencoder with a lightweight structure to design a reconstruction network. It can solve the problem of difficulty in obtaining defect samples in the industrial environment; in the detection stage, the trained model is used to reconstruct the input defect samples, and conventional image processing algorithms can accurately detect the defect area. In the training phase, only defect-free samples are used for training. This paper discusses the unsupervised influence of factors such as network structure and loss function in the algorithm on the surface defect detection task, and a combined loss function combining  $L1$  loss and structural loss is proposed for surface defect detection to adapt to the detection problem of regular texture and irregular texture samples at the same time. This paper The proposed ReNet-D method is compared with other unsupervised algorithms on multi-class sample data. The results show that the detection algorithm proposed in this paper has achieved good results and is suitable for transplanting to industrial detection environments. Due to the lightness of the quantitative network characteristics, ReNet-D has better reconstruction performance for some defects that are similar to the background texture and close to the color, resulting in inconspicuous shadow results. The

improvement makes the defect contrast more noticeable and achieves a better detection effect.

## References

- [1] Prahar M Bhatt, Rishi K Malhan, Pradeep Rajendran, Brual C Shah, Shantanu Thakar, Yeo Jung Yoon, and Satyandra K Gupta. Image-based surface defect detection using deep learning: A review. *Journal of Computing and Information Science in Engineering*, 21(4), 2021.
- [2] Prahar M Bhatt, Rishi K Malhan, Pradeep Rajendran, Brual C Shah, Shantanu Thakar, Yeo Jung Yoon, and Satyandra K Gupta. Image-based surface defect detection using deep learning: A review. *Journal of Computing and Information Science in Engineering*, 21(4), 2021.
- [3] Raghavendra Chalapathy and Sanjay Chawla. Deep learning for anomaly detection: A survey. *arXiv preprint arXiv:1901.03407*, 2019.
- [4] Haiyong Chen, Yue Pang, Qidi Hu, and Kun Liu. Solar cell surface defect inspection based on multispectral convolutional neural network. *Journal of Intelligent Manufacturing*, 31(2):453–468, 2020.
- [5] Tamás Czimmermann, Gastone Ciuti, Mario Milazzo, Marcello Chiurazzi, Stefano Roccella, Calogero Maria Oddo, and Paolo Dario. Visual-based defect detection and classification approaches for industrial applications—a survey. *Sensors*, 20(5):1459, 2020.
- [6] Vincent Dumoulin, Ishmael Belghazi, Ben Poole, Olivier Mastropietro, Alex Lamb, Martin Arjovsky, and Aaron Courville. Adversarially learned inference. *arXiv preprint arXiv:1606.00704*, 2016.
- [7] Maayan Frid-Adar, Idit Diamant, Eyal Klang, Michal Amitai, Jacob Goldberger, and Hayit Greenspan. Gan-based synthetic medical image augmentation for increased cnn performance in liver lesion classification. *Neurocomputing*, 321:321–331, 2018.
- [8] PM Harikrishnan, Anju Thomas, Varun P Gopi, P Palanisamy, and Khan A Wahid. Inception single shot multi-box detector with affinity propagation clustering and their application in multi-class vehicle counting. *Applied Intelligence*, 51(7):4714–4729, 2021.
- [9] Chuanfei Hu and Yongxiong Wang. An efficient convolutional neural network model based on object-level attention mechanism for casting defect detection on radiography images. *IEEE Transactions on Industrial Electronics*, 67(12):10922–10930, 2020.

- [10] Kazunori Imoto, Tomohiro Nakai, Tsukasa Ike, Kosuke Haruki, and Yoshiyuki Sato. A cnn-based transfer learning method for defect classification in semiconductor manufacturing. In *2018 international symposium on semiconductor manufacturing (ISSM)*, pages 1–3. IEEE, 2018.
- [11] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015.
- [12] Yu Jiang, Wei Wang, and Chunhui Zhao. A machine vision-based realtime anomaly detection method for industrial products using deep learning. In *2019 Chinese Automation Congress (CAC)*, pages 4842–4847. IEEE, 2019.
- [13] M Karthikeyan, TS Subashini, and R Jebakumar. Ssd based waste separation in smart garbage using augmented clustering nms. *Automated Software Engineering*, 28(2):1–17, 2021.
- [14] Shengyuan Li, Xuefeng Zhao, and Guangyi Zhou. Automatic pixel-level multiple damage detection of concrete structure using fully convolutional network. *Computer-Aided Civil and Infrastructure Engineering*, 34(7):616–634, 2019.
- [15] Anmin Liu, Weisi Lin, and Manish Narwaria. Image quality assessment based on gradient similarity. *IEEE Transactions on Image Processing*, 21(4):1500–1512, 2011.
- [16] Qiwu Luo, Xiaoxin Fang, Jiaojiao Su, Jian Zhou, Bingxing Zhou, Chunhua Yang, Li Liu, Weihua Gui, and Lu Tian. Automated visual defect classification for flat steel surface: a survey. *IEEE Transactions on Instrumentation and Measurement*, 69(12):9329–9349, 2020.
- [17] Chengkan Lv, Zhengtao Zhang, Fei Shen, Feng Zhang, and Hu Su. A fast surface defect detection method based on background reconstruction. *International Journal of Precision Engineering and Manufacturing*, 21(3):363–375, 2020.
- [18] Shuang Mei, Yudan Wang, and Guojun Wen. Automatic fabric defect detection with a multi-scale convolutional denoising autoencoder network model. *Sensors*, 18(4):1064, 2018.
- [19] Shuang Mei, Hua Yang, and Zhouping Yin. An unsupervised-learning-based approach for automated defect inspection on textured surfaces. *IEEE Transactions on Instrumentation and Measurement*, 67(6):1266–1277, 2018.
- [20] Waseem Rawat and Zenghui Wang. Deep convolutional neural networks for image classification: A comprehensive review. *Neural computation*,

29(9):2352–2449, 2017.

- [21] Hao Shen, Shuxiao Li, Duoyu Gu, and Hongxing Chang. Bearing defect inspection based on machine vision. *Measurement*, 45(4):719–733, 2012.
- [22] Domen Tabernik, Samo Šela, Jure Skvarč, and Danijel Skočaj. Segmentation-based deep-learning approach for surface-defect detection. *Journal of Intelligent Manufacturing*, 31(3):759–776, 2020.
- [23] Xian Tao, Dapeng Zhang, Wenzhi Ma, Xilong Liu, and De Xu. Automatic metallic surface defect detection and recognition with convolutional neural networks. *Applied Sciences*, 8(9):1575, 2018.
- [24] Xian Tao, Dapeng Zhang, Wenzhi Ma, Xilong Liu, and De Xu. Automatic metallic surface defect detection and recognition with convolutional neural networks. *Applied Sciences*, 8(9):1575, 2018.
- [25] Xian Tao, Dapeng Zhang, Wenzhi Ma, Xilong Liu, and De Xu. Automatic metallic surface defect detection and recognition with convolutional neural networks. *Applied Sciences*, 8(9):1575, 2018.
- [26] Shashanka Venkataramanan, Kuan-Chuan Peng, Rajat Vikram Singh, and Abhijit Mahalanobis. Attention guided anomaly localization in images. In *European Conference on Computer Vision*, pages 485–503. Springer, 2020.
- [27] Shashanka Venkataramanan, Kuan-Chuan Peng, Rajat Vikram Singh, and Abhijit Mahalanobis. Attention guided anomaly localization in images. In *European Conference on Computer Vision*, pages 485–503. Springer, 2020.
- [28] Shashanka Venkataramanan, Kuan-Chuan Peng, Rajat Vikram Singh, and Abhijit Mahalanobis. Attention guided anomaly localization in images. In *European Conference on Computer Vision*, pages 485–503. Springer, 2020.
- [29] Tian Wang, Yang Chen, Meina Qiao, and Hichem Snoussi. A fast and robust convolutional neural network-based defect detection model in product quality control. *The International Journal of Advanced Manufacturing Technology*, 94(9):3465–3471, 2018.
- [30] Zhenchao Wu, Ruizhe Yang, Fangfang Gao, Wenqi Wang, Longsheng Fu, and Rui Li. Segmentation of abnormal leaves of hydroponic lettuce based on deeplabv3+ for robotic sorting. *Computers and Electronics in Agriculture*, 190:106443, 2021.

- [31] Jin Xia-Ting, Wang Yao-Nan, Zhang Hui, Liu Li, Zhong Hang, and He Zhen-Dong. Deeprail: automatic visual detection system for railway surface defect using bayesian cnn and attention network. *Acta Automatica Sinica*, 45(12):2312–2327, 2019.
- [32] Qiaoning Yang, Weimin Shi, Juan Chen, and Weiguo Lin. Deep convolution neural network-based transfer learning method for civil infrastructure crack detection. *Automation in Construction*, 116:103199, 2020.
- [33] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014.
- [34] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014.
- [35] Mei Zhang, Jinglan Wu, Huifeng Lin, Peng Yuan, and Yanan Song. The application of one-class classifier based on cnn in image defect detection. *Procedia computer science*, 114:341–348, 2017.
- [36] Hang Zhao, Orazio Gallo, Iuri Frosio, and Jan Kautz. Loss functions for image restoration with neural networks. *IEEE Transactions on computational imaging*, 3(1):47–57, 2016.