

## Article

# A Deep-Convolutional-Neural-Network-Based Semi-Supervised Learning Method for Anomaly Crack Detection

Xingjun Gao, Chuansheng Huang, Shuai Teng  and Gongfa Chen \* 

School of Civil and Transportation Engineering, Guangdong University of Technology, Guangzhou 510006, China  
\* Correspondence: gongfa.chen@gdut.edu.cn

**Abstract:** Crack detection plays a pivotal role in structural health monitoring. Deep convolutional neural networks (DCNN) provide a way to achieve image classification efficiently and accurately due to their powerful image processing ability. In this paper, we propose a semi-supervised learning method based on a DCNN to achieve anomaly crack detection. In the proposed method, the training set for the network only requires a small number of normal (non-crack) images but can achieve high detection accuracy. Moreover, the trained model has strong robustness in the condition of uneven illumination and evident crack difference. The proposed method is applied to the images of walls, bridges and pavements, and the results show that the detection accuracy comes up to 99.48%, 92.31% and 97.57%, respectively. In addition, the features of the neural network can be visualized to describe its working principle. This method has great potential in practical engineering applications.

**Keywords:** anomaly crack detection; deep convolutional neural network; semi-supervised learning; infrastructure cracks; neural network visualization



**Citation:** Gao, X.; Huang, C.; Teng, S.; Chen, G. A Deep-Convolutional-Neural-Network-Based Semi-Supervised Learning Method for Anomaly Crack Detection. *Appl. Sci.* **2022**, *12*, 9244. <https://doi.org/10.3390/app12189244>

Academic Editor: Diogo Ribeiro

Received: 26 July 2022

Accepted: 13 September 2022

Published: 15 September 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

During the service life of infrastructure such as buildings, bridges and roads, crack defects are always inevitable, which may lead to structural performance degradation and may even bring about catastrophic failures and enormous loss of human lives [1,2]. To reduce the adverse effect of crack defects, it is necessary to conduct defect detection and inspection for regular maintenance.

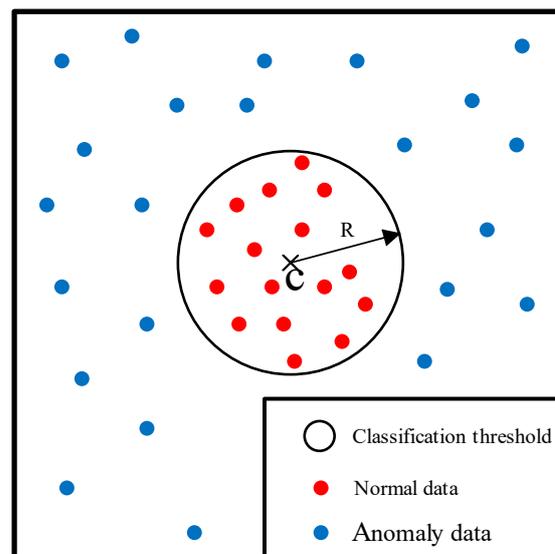
Traditional methods for crack detection are mainly based on manual visual inspection by certified inspectors, which may be labor intensive, time consuming and highly subjective [3]. In the last few years, crack detection techniques based on image processing have developed rapidly, and several approaches such as edge detection [4], region growth [5], threshold segmentation [6] and morphological operations [7] have emerged and been applied to the detection of infrastructure defects. With these methods, the cracks can be easily extracted from the background according to the edge, color, shape and other information of the images. However, it is worth noting that the key to the success of such image-processing-based approaches is to select an appropriate threshold value for accurate image classification. An improper parameter may cause relatively poor detection accuracy [8], especially for scenes with complex backgrounds.

With the great improvement of computer hardware, image recognition and detection based on machine learning (ML) have achieved great breakthroughs. The basic idea of ML is to extract common crack features from the training set and to apply them to the testing set for detection. Generally, ML can be divided into supervised learning, unsupervised learning and semi-supervised learning based on learning mechanisms. Unsupervised learning can learn the differences between cracks and backgrounds autonomously without manual labeling of image data, and it can remove the influence of subjective factors. Some

unsupervised learning approaches, such as the K-means algorithm [9] and principal component analysis (PCA) [10], have been used to discover the laws hidden in image data and to extract cracks from the background, and they perform poorly on images including noise and uneven illumination. To extract cracks effectively from complex backgrounds, some supervised learning-based methods, for instance, the support vector machine (SVM) [11], random forest (RF) [12] and decision tree (DT) [13], have been developed and applied to crack classification. With a large number of labeled high-quality image data, the classifier can be trained to obtain featured information of crack classification. Generally, these methods heavily depend on the quality and quantity of manually extracted features. For infrastructure with complex conditions, it is difficult to obtain universal features suitable for all cracks and to achieve a desirable detection effect. Recently, the success of deep learning in the field of computer vision provides an opportunity for the development of crack classification models. The image classification models developed based on convolutional neural networks (CNNs) have satisfactory prediction accuracy that even exceeds that of humans [14] and can effectively remove the influence of uneven illumination [15] and noise [16]. They work well in pavement crack detection [17] and concrete structure crack detection [18], and the model training is time consuming. To improve the training efficiency, transfer learning [19,20] is integrated into the crack classification model. Although high-performance models based on transfer learning can be obtained and perform well in crack detection, a huge number of samples is required to train the models. Generally, infrastructure in normal service may have very few crack defects of concern, resulting in a relatively high cost of obtaining the desired training samples. Moreover, crack labeling used in neural network models is usually performed manually and is labor intensive, which limit the wide application of this kind of method in crack detection.

Semi-supervised learning combines supervised learning with unsupervised learning together, and it can train classifiers with few labeled samples. Typical semi-supervised learning algorithms, for example, self-training [21], hybrid models [22], graph-based ones [23] and SVM-based [24] ones can be applied to wall crack detection [25], pavement crack detection [26] and steel structure surface defect detection [27]. During the training, classifiers are trained by a small amount of labeled data and then are employed to classify a great amount of unlabeled data. Subsequently, the mislabeled samples are picked out and corrected manually and are reused as the training data in the next round. In these methods, a large amount of anomaly (crack) data is the key to obtaining a high-performance classifier. In practical engineering, it is generally difficult to obtain abnormal crack data mainly due to the low frequency of abnormal events, high artificial manufacturing cost and the difficulty of obtaining a large number of open labeled abnormal data sets.

To address this issue, normal data instead of anomaly data are paid attention to and collected to train anomaly classifiers. Such a technique is called semi-supervised anomaly detection. With normal data as a training set, a classifier with a specific threshold is obtained and then employed to determine whether the testing images are abnormal. Currently, this technology has been already widely used in cancer detection [28], ultrasound detection [29], disease detection of industrial products [30–32] and infrastructure diseases [33,34]. Although anomaly detection technology can meet practical needs to some extent, there is very limited research on how to explain it. To further investigate the working principles of anomaly detection, an anomaly detection method named deep support vector data description (DSVDD) has been exploited to learn the neural network transformation from input space to output space [35]. In this way, most of the normal data (red dots) are mapped into a hypersphere characterized by the center  $c$  and the radius  $R$  of minimum volume, and anomalies (blue dots) fall outside, as shown in Figure 1. The boundary between normal and anomaly samples is defined as the classification threshold. To visualize the neural network, a fully convolutional data description (FCDD) has been developed [36], in which the abnormal images are displayed in the form of a thermal map. In such a way, it provides a more intuitive explanation of the working principles of anomaly detection.



**Figure 1.** Diagram of DSVDD data division.

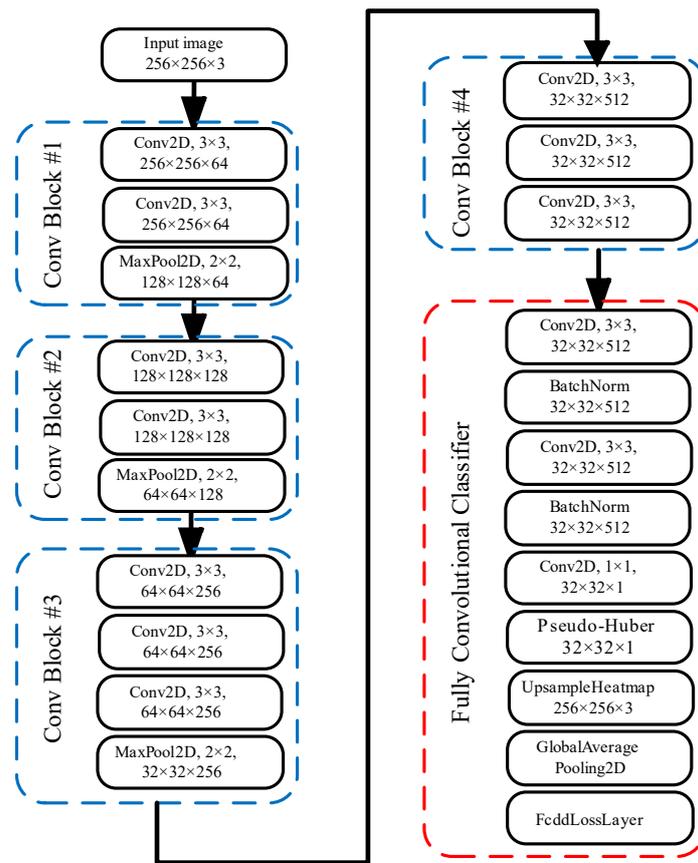
Inspired by the transfer learning and semi-supervised anomaly detection methods, the transfer learning model based on the VGG-16 is employed to conduct semi-supervised anomaly detection. In this study, cracks on the infrastructure image are seen as abnormal, whereas images without cracks are considered normal. With a part of normal images as the training set, the classifiers are trained and employed to classify other images. In this way, the classifiers can achieve high accuracy and testing speeds and describe the working principles of the neural network.

The rest of this paper is organized as follows. In Section 2, the methodology, the experimental flow and the performance evaluation indicators of the network model are briefly introduced. In Section 3, the effects of crack images in the training set on network model performance are compared. In Section 4, the effectiveness of the proposed method in different data sets is validated in terms of experiments. Finally, conclusions are made in Section 5.

## 2. Methodology

### 2.1. CNN Architecture

To conduct semi-supervised anomaly detection effectively, an improved VGG-16 network model with the network architecture, as shown in Figure 2, is used. Here, we use the VGG-16 network model [37] as the backbone network mainly based on its several advantages, such as excellent performance in the task of transfer learning [19,20] with simple architecture, strong generalization ability and flexibility but with relatively high training efficiency. However, the traditional VGG-16 still has some limitations. For example, in the classification task, only the label of the testing sample can be output, but not the visual feature map. To address this issue, some modifications are made. At first, the first four Conv blocks of the traditional VGG-16 model are frozen, and then the remaining part, which contains the fifth Conv block (including three convolution layers and one maximum pooling layer), three fully connected layers and one softmax layer, is replaced by a fully convolutional classifier, with its architecture shown in the red dashed box. In other words, both the traditional VGG-16 neural network model and the improved one have the same first four Conv blocks, and the latter one introduces a fully convolutional classifier to output the visual feature map and has random initialization and trainable convolutional layers.



**Figure 2.** The network architecture of improved VGG-16. The two-dimensional array denotes the convolution or pool kernel size, and the three-dimensional array denotes the size of the output image and the channels.

In algorithm implementation, the procedure of the improved VGG-16 is summarized as follows. In the training stages, some normal crack images of the same size are used as the input, and their characteristics are extracted after passing through four Conv blocks in sequence. In this process, the model training only takes a short time, due to the pre-trained network layers being frozen and the model not needing to relearn the parameters. Subsequently, the feature map obtained by the first four Conv blocks is input into the “fully convolutional classifier”, which includes nine layers, as shown in the red dashed box in Figure 2. In the first four layers, the characteristics are further extracted, and parameter learning and batch normalization on the feature map are performed. The fifth layer named Conv2D in the red dashed box is a  $1 \times 1$  convolutional layer. In this layer, it compresses the network to output into a single channel of data. The sixth layer is a pseudo-Huber loss function [38], which limits the output heatmaps to positive numbers. An upsampling layer is followed by the loss function layer to resize the output heatmap to the same size as the input image. Then, the global average pool layer is employed to calculate the average thermal scores. Finally, the FCDD objective loss function [36] is evaluated in the FCDDLossLayer.

The FCDD objective loss function [36] is calculated according to Equation (1), which is a variant of the Hypersphere Classifier objective [39], as given in Equation (2).

$$FCDD_{loss} = \min_w \frac{1}{n} \sum_{i=1}^n (1 - y_i) \frac{1}{u \cdot v} \|A(X_i)\|_1 - y_i \log \left( 1 - \exp \left( -\frac{1}{u \cdot v} \|A(X_i)\|_1 \right) \right) \quad (1)$$

$$HC_{loss} = \min_w \frac{1}{n} \sum_{i=1}^n (1 - y_i) h(\varphi(X_i; W) - c) - y_i \log(1 - \exp(-h(\varphi(X_i; W) - c))) \quad (2)$$

where  $n$  is the number of samples,  $i$  denotes the  $i$ -th sample,  $X_1, \dots, X_n$  represents a collection of samples and  $y_1, \dots, y_n$  are their labels.  $y_i = 1$  denotes an anomaly sample, and  $y_i = 0$  denotes a normal sample.  $W$  is the weight of the neural network  $\varphi$ , and  $c$  represents the feature center mapped by the training set.  $h$  is the pseudo-Huber loss, which is determined by  $h(a) = \sqrt{\|a\|_2^2 + 1} - 1$ .  $FCDD_{loss}$  represents the FCDD objective loss function, and  $HC_{loss}$  represents the Hypersphere Classifier objective loss function.

From Equation (2), it can be known that the value of  $X_i$  has a great effect on the loss function. For example, if  $y_i = 0$ , the loss whole function tends to be zero when  $X_i \rightarrow c$ ; otherwise, if  $y_i = 1$ , the whole loss function increases as the value of  $\|X_i - c\|$  increases. Here,  $c$  is the bias term in the last layer of the neural network and is omitted in Equation (1).  $\|A(X_i)\|_1$  represents the sum of all terms in  $A(X)$  ( $A(X) = \sqrt{\varphi(X; W)^2 + 1} - 1$ ). Since  $\|A(X_i)\|_1$  can satisfy the minimization of normal samples and maximize the anomalous samples,  $\|A(X_i)\|_1$  is taken as the thermal scores in heatmaps.  $u \cdot v$  represents the size of the output image of the network before upsampling.

## 2.2. Experimental Process

The overall flow chart is shown in Figure 3. Firstly, all images are proportionally randomly divided into the training set, calibration set and testing set. Image augmentation and random 50% confetti noise methods are applied to the training set to train the improved VGG-16 model. Secondly, binary processing is carried out on the image labels of the calibration set by using the conversion function, i.e., the non-crack images are assigned 0, and the crack images are assigned 1. The trained network is used to predict the average thermal scores and binary labels of all images in the calibration set. Then, a receiver operating characteristic (ROC) curve is created, where the  $x$ -axis and the  $y$ -axis are the false positive rate and the true positive rate, respectively. The ROC curve is a stepped line, and it increases sequentially from left to right. To express the performance of the classifier more intuitively, the area under the ROC curve (AUC) is proposed, and as the value of AUC becomes larger, the network performance becomes better. Another measure indicator of the ROC curve, the maximum Youden index, is used to determine the best boundary of the classifier, namely the thermal threshold. The thermal threshold is the score corresponding to the  $x$ -axis of the maximum difference between the true positive rate and the false positive rate. Finally, the image labels of the original testing set are binarized. Then, the trained classification network model is used to predict the preprocessed testing set images, and average thermal scores are obtained. The classification results are obtained with a comparison on the thermal threshold and the average thermal scores, i.e., if the average thermal score of a sample is greater than the thermal threshold, it is considered a crack image; otherwise, it is considered a non-crack image.

In the experiments, all the images in the testing set are classified and labeled with corresponding class labels, and then a comparison is made between these labels and the real ones to determine the confusion matrix of the model. Subsequently, the evaluation indicators given in Section 2.3 can be calculated in light of the confusion matrix. To obtain reliable results, a simple method is used to handle classification with the imbalanced datasets, including the training set, calibration set and test set. At first, the image sequences of the dataset are shuffled and randomly divided into the training set, calibration set and testing set according to proportion. Secondly, steps ①–③ in Figure 3 are recycled 10 times to obtain 10 neural network models with different evaluation indicators. Lastly, the 10 models are compared, and the best one is selected as the classifier for the corresponding dataset.

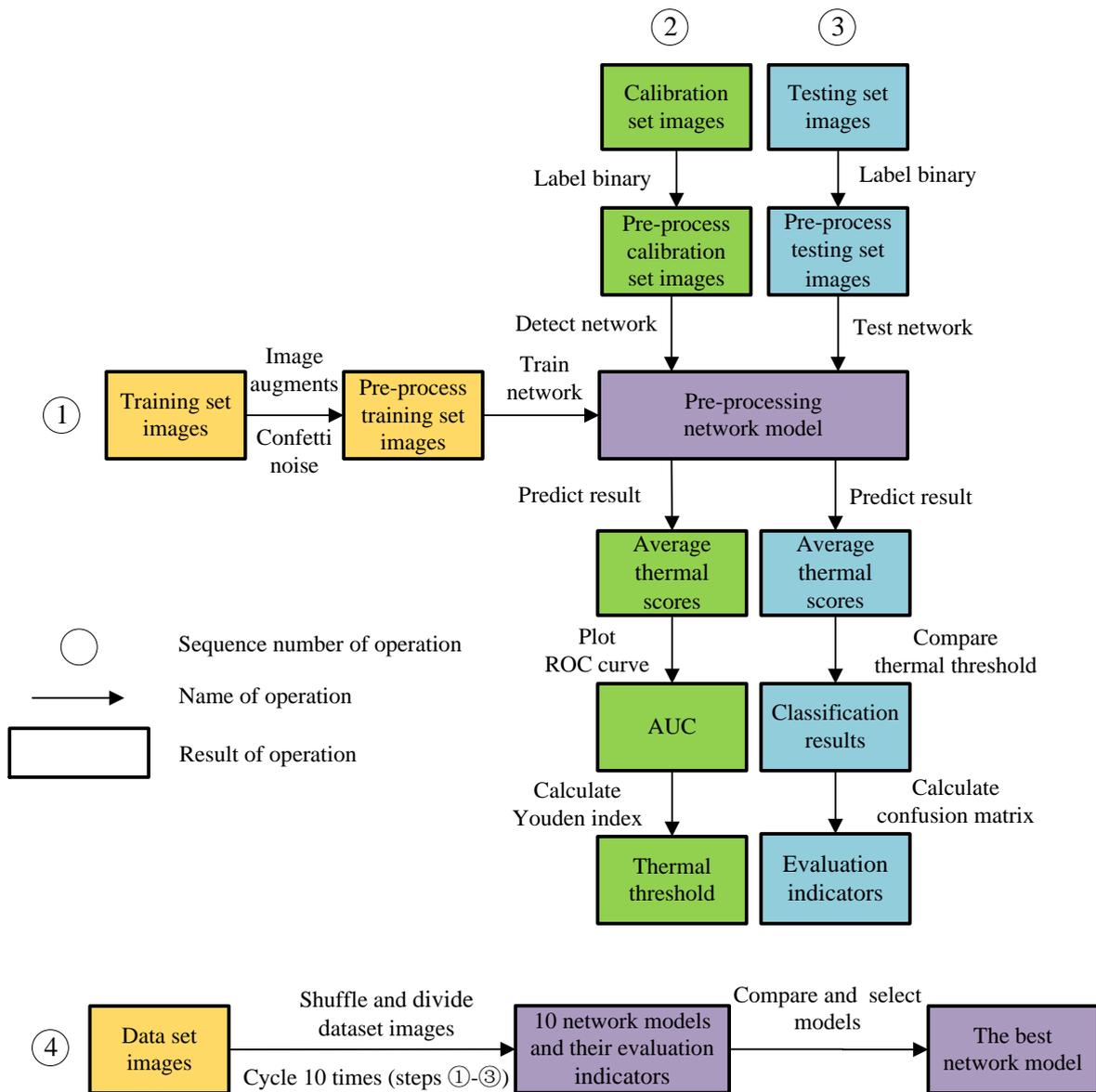


Figure 3. Flow chart of the proposed method.

### 2.3. Evaluation Indicators

Accuracy, Precision, Recall and F1-Score are commonly used indicators to evaluate a classification network model, as shown in Equations (3)–(6).

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN} \tag{3}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{4}$$

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

$$\text{F1-Score} = \frac{2TP}{2TP + FP + FN} = \frac{2 \times P \times R}{P + R} \tag{6}$$

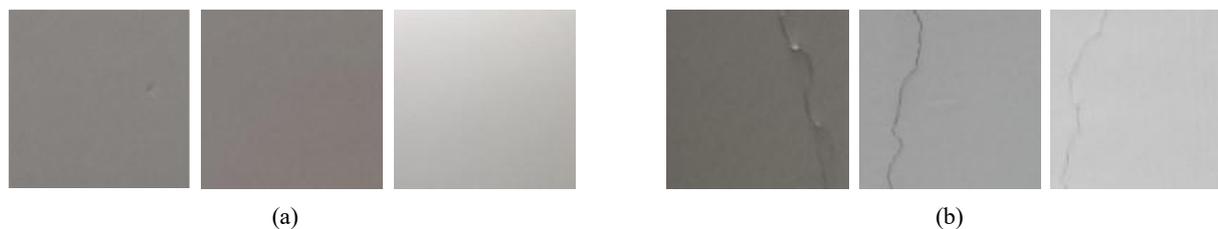
where TP is the number of crack samples that are correctly predicted; FP is the number of non-crack samples that are predicted as cracks; FN is the number of crack samples

that are predicted as non-cracks; and TN is the number of non-crack samples that are predicted correctly.

### 3. Results

#### 3.1. Wall Dataset and Training Process

The wall dataset consists of 421 indoor wall images of buildings, including 171 crack images and 250 non-crack images. These images are captured by a smartphone in uneven illumination conditions. The dimensions of the images are  $3000 \times 4000$  pixels. To speed up the training, the original image resolution is reduced to  $256 \times 256$ . The dataset contains images of uneven illumination, holes and various cracks, as shown in Figure 4. To compare the performance of the model, a testing group and a control group are set up in the experiment. The differences are that the training set of the testing group has 4 crack images (about 1~2% images of the wall dataset), whereas the training set of the control group does not have crack images. The detailed allocation of the testing group and the control group image samples are shown in Tables 1 and 2, respectively. The numbers in parentheses represent the number of additional images expanded with image augmentation techniques, and the numbers outside parentheses represent the total number of images in the same class.



**Figure 4.** Examples of images contained in the wall dataset: (a) non-crack images; (b) crack images.

**Table 1.** Image samples allocated from the wall testing group dataset.

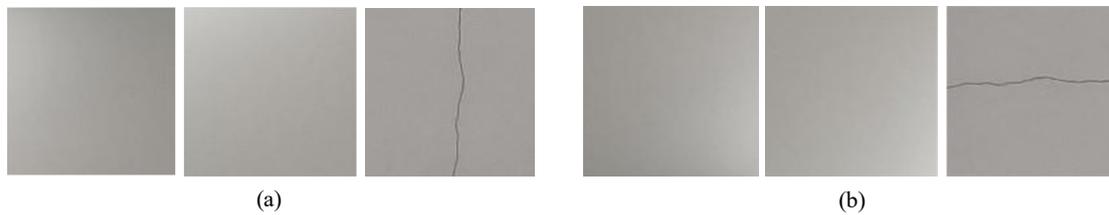
Number of Images	Training (Expansion)	Calibration	Testing	Total (Expansion)
Non-crack	250 (125)	50	75	375 (125)
Crack	8 (4)	50	117	175 (4)
Total	258 (129)	100	192	550 (129)

**Table 2.** Image samples allocated from the wall control group dataset.

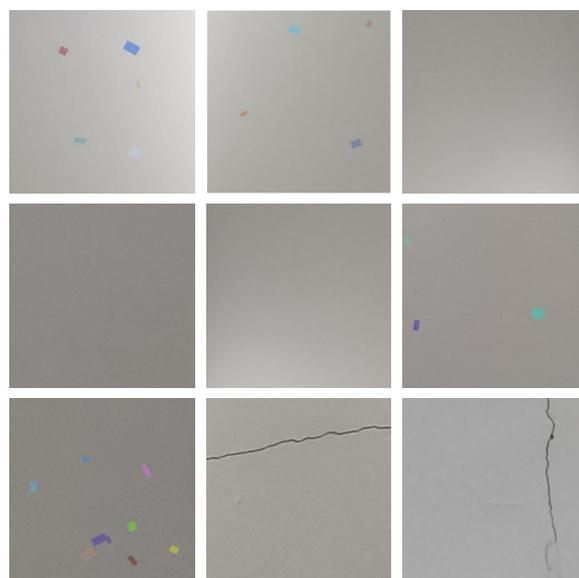
Number of Images	Training (Expansion)	Calibration	Testing	Total (Expansion)
Non-crack	250 (125)	50	75	375 (125)
Crack	0	50	121	171
Total	250 (125)	100	196	546 (125)

To alleviate the adverse effects of over-fitting, it is necessary to use image augmentation techniques to improve the generalization ability of the neural network model. Generally, image augmentation techniques contain traditional image augmentation (rotation, flipping, random cropping, color jittering, etc.), conditional generative adversarial networks (GAN) [40], deep convolutional GAN [41], single image GAN [42], other forms of GAN [43], etc. The application of these technologies significantly improves the recognition performance of intelligent algorithms. Taking into account the computation efficiency and the implementation difficulty of the algorithm, the traditional image augmentation method is applied to the training set to realize the expansion of the samples, including 90-degree rotations and 180-degree flips (horizontal and vertical). The partial results of image augmentation are shown in Figure 5. In addition, we use confetti noise to randomly simulate

anomalous regions in normal images for 50% of the training set images, which is beneficial to stabilize the loss function during training and to improve the CNN's robustness. The partial results of random confetti noise are shown in Figure 6.

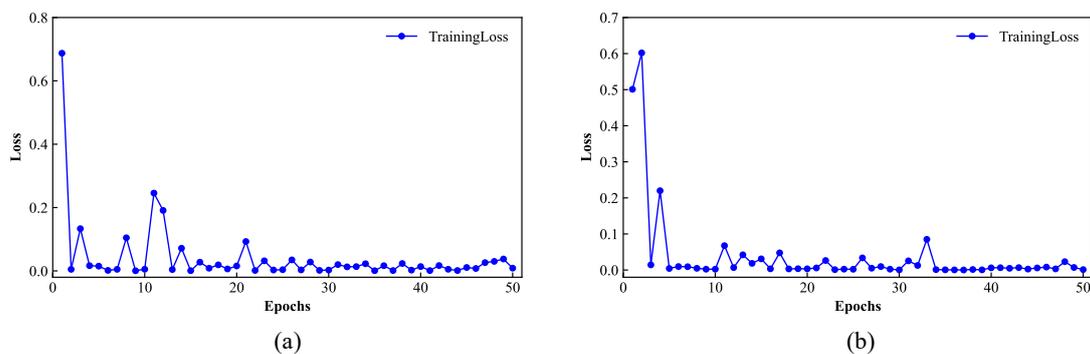


**Figure 5.** Examples of the training images contained in the wall testing group dataset: (a) original images; (b) expanded images.



**Figure 6.** Results of random confetti noise.

The image preprocessing, network training and testing evaluation in this paper are all run on a laptop equipped with a Core (TM) i7-10700 @ 2.90GHz CPU with 16 GB RAM and an NVIDIA Quadro P620 GPU. The crack detection models are trained with 50 epochs, and the learning rate and batch size of training in the Adam algorithm are set as 0.0001 and 2, respectively. Each image classification label is assumed to be correct, and the improved VGG-16 model is trained. According to the training process in Figure 7, the network tends to be stable with about 35 epochs and a training time of 630 s.

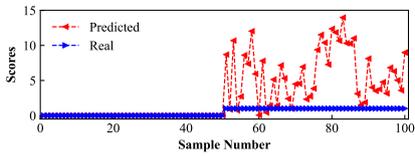
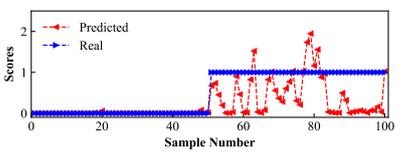
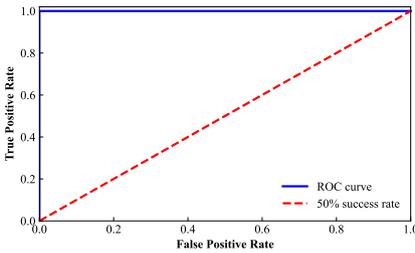
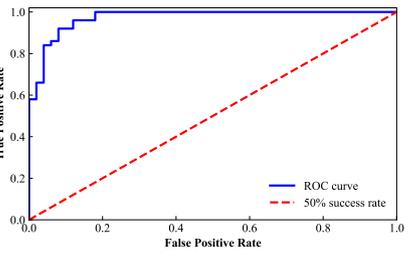
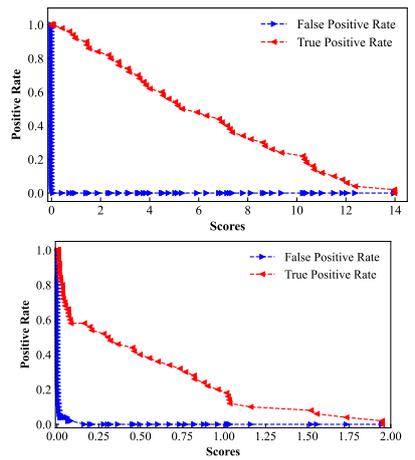
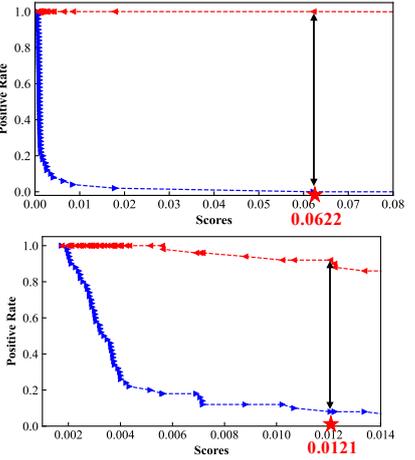


**Figure 7.** The training process of the wall image classification network: (a) testing group; (b) control group.

### 3.2. Calibration Process

The improved trained VGG-16 model is applied to the calibration set images after the label binary processing. The testing group and control group are evaluated for comparison, and the calibration process results are shown in Table 3. From the prediction results, all the data of the non-crack images in the testing group are close to 0, and some data in the control group are slightly higher than 0. Compared with the thermal threshold of 0.0121, it is shown that, in the control group, there are non-crack images predicted as cracks. Looking at the crack images again, the data in the testing group are almost all greater than 1, and some data in the control group are near 0. Such results reflect that the cracks are predicted as non-cracks in the control group. In terms of score range, the range of the testing group is significantly larger than that of the control group. These results show that existing crack images in the training set can enlarge the thermal score range and is more beneficial for the classifier in classifying the images.

**Table 3.** Calibration process results of the testing group and control group.

	Testing Group	Control Group
Calibration set prediction result		
ROC curve	 AUC: 1	 AUC: 0.9732
Scores-Positive rate (upper: overall, and lower: detail)		

Note: The red star represents the thermal threshold.

### 3.3. Testing Process

The improved trained VGG-16 model is also applied to the testing images after the label binary processing. The evaluation indicators of the testing set can be calculated, as shown in Tables 4 and 5. The mean of Accuracy, Precision, Recall and F1-Score of the testing group achieve better performance than those of the control group, which is consistent with the calibration conclusion. Since the center radius of the average thermal scores of the control group is small, for a slight change, it may identify the non-cracks as cracks. In other

words, the FP increases, resulting in low robustness and overall performance degradation. In engineering practice, besides the overall performance of the network model, the testing speeds are also a very important indicator of concern. The testing speeds of the two groups are similar, and each image only takes about 0.0553 s and 0.0571 s, respectively. To compare the model performance between the testing group and the control group, their confusion matrices are given in Figure 8. It can be seen from Figure 8 that all the values of TP, TN and accuracy of the testing group are larger than those of the control group.

**Table 4.** Experimental results of wall testing group image classification.

Situation	Training Set	Calibration Set		Testing Set				
	Training Time (s)	AUC	Thermal Threshold	Accuracy	Precision	Recall	F1-Score	Testing Speeds (s/img)
1	628	1.0000	0.0622	99.48%	100.00%	99.15%	99.57%	0.0523
2	630	1.0000	0.2739	98.96%	100.00%	98.29%	99.14%	0.0525
3	627	1.0000	0.2026	99.48%	100.00%	99.15%	99.57%	0.0550
4	638	1.0000	0.0539	98.44%	100.00%	97.43%	98.70%	0.0579
5	631	0.9980	0.4289	96.88%	100.00%	94.87%	97.37%	0.0573
6	621	1.0000	0.0813	99.48%	100.00%	99.15%	99.57%	0.0565
7	618	1.0000	0.0947	99.48%	100.00%	99.15%	99.57%	0.0590
8	651	1.0000	0.3309	98.44%	100.00%	97.43%	98.70%	0.0552
9	637	1.0000	0.5644	98.96%	100.00%	98.29%	99.14%	0.0552
10	620	1.0000	0.2005	98.96%	100.00%	98.29%	99.14%	0.0526
Mean	630	0.9998	0.2293	98.86%	100.00%	98.12%	99.05%	0.0553

**Table 5.** Experimental results of wall control group image classification.

Situation	Training Set	Calibration Set		Testing Set				
	Training Time (s)	AUC	Thermal Threshold	Accuracy	Precision	Recall	F1-Score	Testing Speeds (s/img)
1	636	0.9732	0.0121	95.41%	98.28%	94.21%	96.20%	0.0528
2	661	0.9992	0.0052	95.41%	99.12%	93.39%	96.17%	0.0558
3	647	0.9912	0.0018	94.90%	95.87%	95.87%	95.87%	0.0547
4	663	0.9940	0.0072	93.37%	95.76%	93.39%	94.56%	0.0584
5	664	0.9952	0.0026	94.90%	100.00%	91.74%	95.69%	0.0580
6	658	0.9296	0.0019	95.41%	100.00%	92.56%	96.14%	0.0579
7	642	0.9896	0.0019	94.90%	95.87%	95.87%	95.87%	0.0564
8	664	0.9880	0.0016	94.90%	95.12%	96.69%	95.90%	0.0606
9	665	0.9903	0.0233	92.86%	98.20%	90.08%	93.97%	0.0583
10	662	0.9900	0.0064	94.39%	100.00%	90.91%	95.24%	0.0582
Mean	656	0.9840	0.0064	94.65%	97.82%	93.47%	95.56%	0.0571

To further elaborate the principles of the classification network, the predicted heatmap by the network is exploited for analysis. Since the testing group only includes one false detection image, the heatmaps of FP and FN in Table 6 are from the control group, and the heatmaps of TP and TN are from the testing group. The average thermal scores of the corresponding images are shown in the figures below the heatmap, where T in brackets represents the testing group, and C represents the control group.

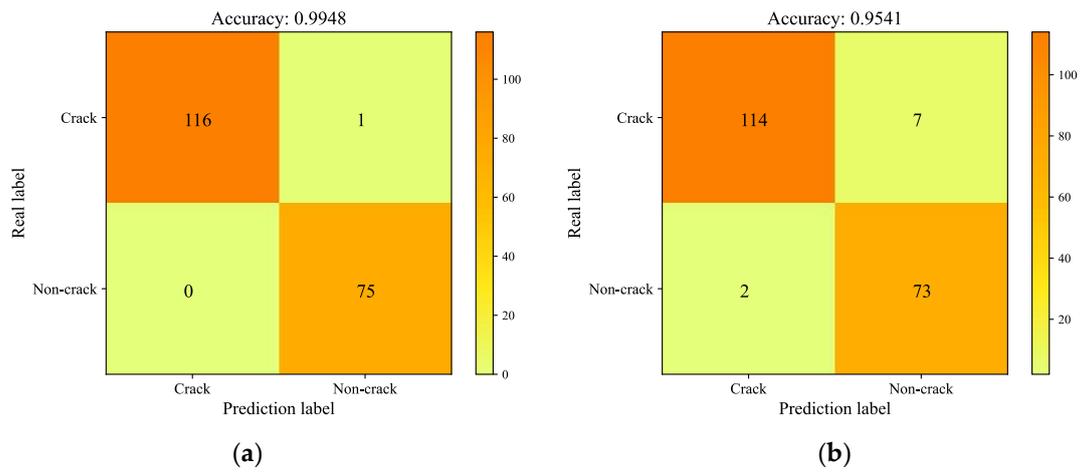


Figure 8. Confusion matrix of wall testing set images: (a) testing group; (b) control group.

Table 6. Original images and Heatmap results of the wall testing set.

	Original Image		Heatmap	
TP			 8.9854 (T)	 9.7063 (T)
TN			 0.0008 (T)	 0.0013 (T)
FP			 0.0481 (C)	 0.0125 (C)
FN			 0.0043 (C)	 0.0039 (C)

In heatmaps of the TP samples, some areas appear bright red. This means that the average thermal scores in the area are quite different from the center value of the training model. Comparing TP samples with the original images, it can be seen that the center of this area is exactly where the cracks are located. This means that both the classification and locations of cracks can be achieved in the proposed method.

From the heatmap results of the TN samples, the overall color is approximately the same, and no anomalous area appears. In the first image of the FP sample, a small part of the anomalous area appears in the center-right position, which corresponds to a defect spot in the original image. Although it may not be a crack, it can cause the inspectors' concern and make them further determine the disease type.

In the FN results, the network falsely identifies the cracks as non-cracks, and the cracks show up more clearly. To some extent, it is equivalent to performing a contrast enhancement process on the original images; therefore, the originally blurred cracks can be displayed clearly.

#### 4. Discussion

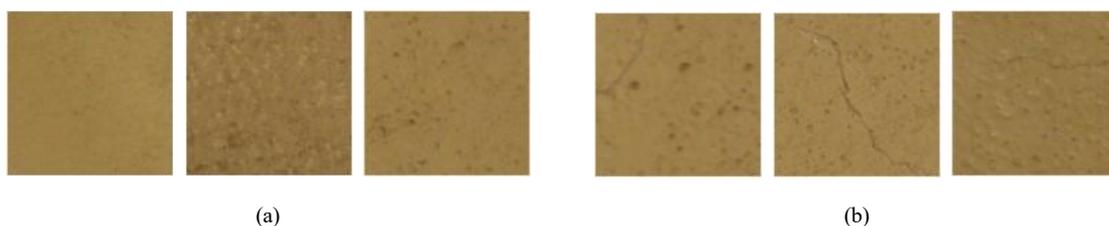
To study the influence of different input datasets on the method, each dataset is used to train and validate the model under the same parameters. Then, the trained model is used to test the bridge and pavement images, respectively. The evaluation indicators are used to validate the performance and effectiveness of the proposed method.

##### 4.1. Bridge Images

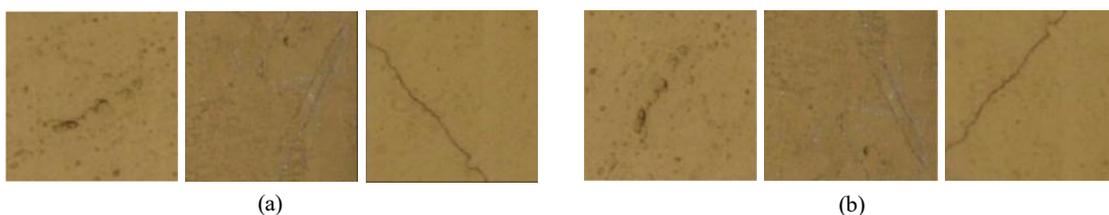
Firstly, 1124 RGB images with a resolution of  $256 \times 256$  pixels in the bridge dataset are used as the experimental samples, including 445 crack images and 679 non-crack images, with the same portion as the wall dataset in the last experiment. The number of samples in different sets is listed in Table 7. Moreover, the dataset contains images of black stains, surface roughness, holes and various cracks, as shown in Figure 9. The partial results of image augmentation are shown in Figure 10.

**Table 7.** Image samples allocated from the bridge dataset.

Number of Images	Training (Expansion)	Calibration	Testing	Total (Expansion)
Non-crack	680 (340)	134	205	1019 (340)
Crack	44 (22)	134	289	467 (22)
Total	724 (362)	268	494	1486 (362)

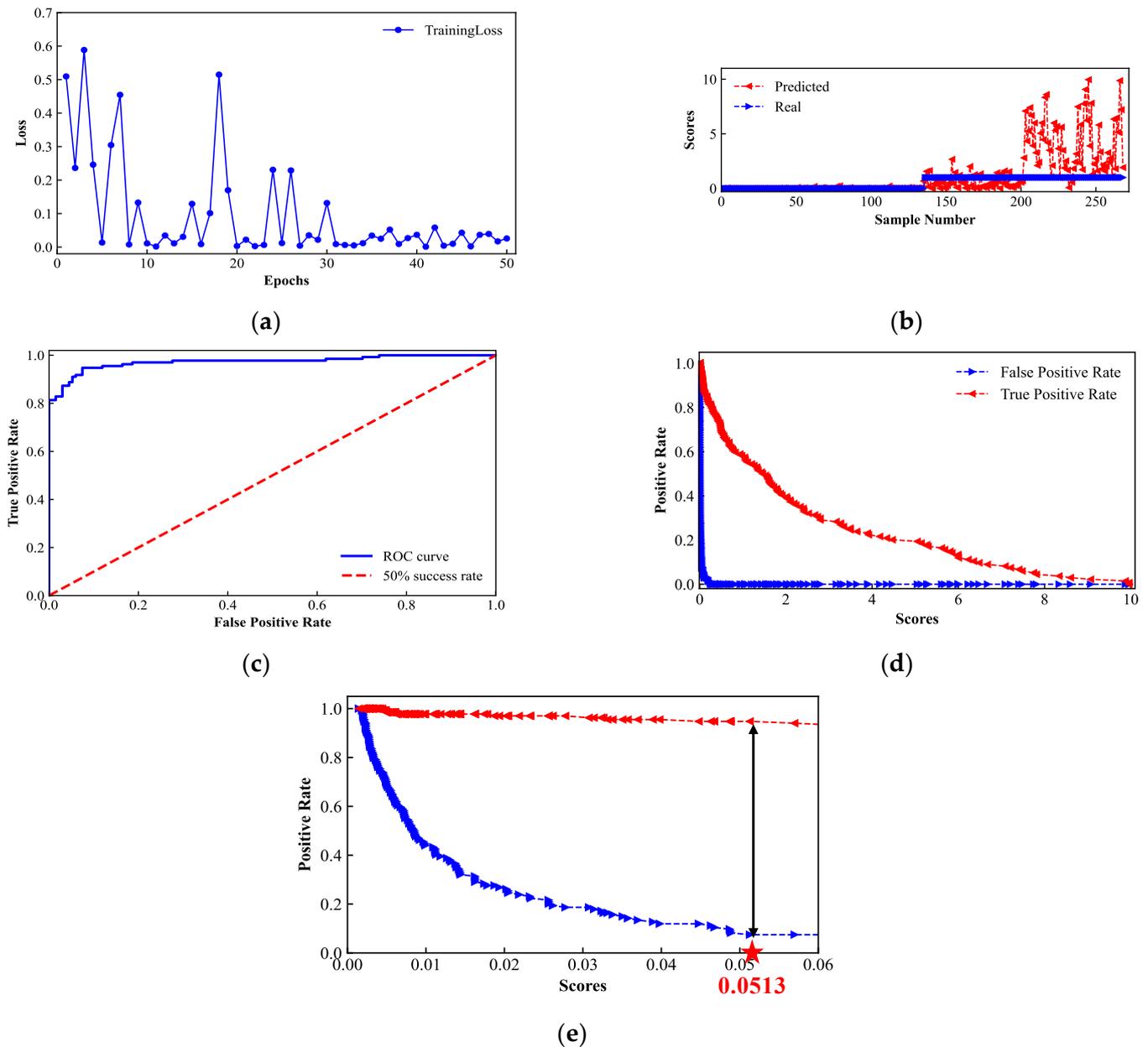


**Figure 9.** Example images contained in the bridge dataset: (a) non-crack images; (b) crack images.



**Figure 10.** Examples of the training images contained in the bridge dataset: (a) original images; (b) expanded images.

It can be seen from Figure 11a and Table 8 that the trained network model tends to be stable with about 50 epochs and 1767 s. The model is used to detect the calibration set. The prediction results are shown in Figure 11b. Generally, the crack detection prediction results are satisfactory. For non-crack images, the average thermal scores are roughly around 0; for crack images, some blurry or tiny-crack crack images tend to be 0~1, whereas clear or wide crack images are greater than 1. This further shows that the average thermal scores obtained by the proposed method can be approximately considered as the differences from the central threshold obtained with the network model, i.e., as the average thermal scores of the image become larger, the difference between cracks and non-crack images becomes greater, and the possibility of defects becomes higher. In the ROC curve, the AUC value can reach 0.9730, and finally, the thermal threshold reaches 0.0513.

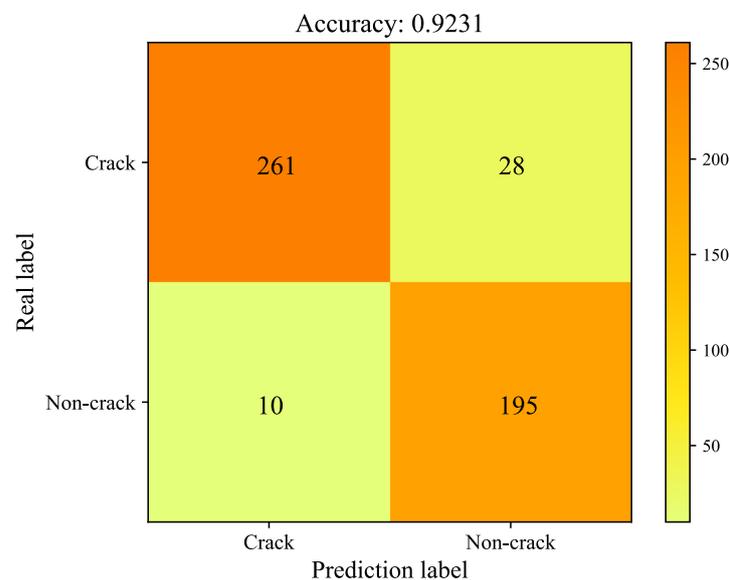


**Figure 11.** The results of the bridge image classification training and calibration process: (a) training process; (b) calibration set prediction results; (c) ROC curve; (d) overall diagram of scores-positive rate; (e) detailed diagram of a scores-positive rate. The red star represents the thermal threshold.

**Table 8.** Experimental results of bridge image classification.

Situation	Training Set	Calibration set		Testing Set				
	Training Time (s)	AUC	Thermal Threshold	Accuracy	Precision	Recall	F1-Score	Testing Speeds (s/img)
1	1767	0.9730	0.0513	92.31%	96.31%	90.31%	93.21%	0.0531
2	1759	0.9683	0.0556	90.28%	93.19%	89.97%	91.55%	0.0581
3	1763	0.9761	0.1799	90.89%	98.03%	86.16%	91.71%	0.0567
4	1880	0.9710	0.0516	92.11%	95.96%	90.31%	93.05%	0.0622
5	1815	0.9620	0.1316	90.08%	98.00%	84.78%	90.91%	0.0585
6	1922	0.9676	0.0966	92.51%	95.65%	91.35%	93.45%	0.0630
7	1823	0.9735	0.1229	90.69%	95.17%	88.58%	91.76%	0.0574
8	1911	0.9583	0.1046	92.31%	97.00%	89.62%	93.17%	0.0621
9	1814	0.9700	0.0485	91.90%	96.98%	88.93%	92.78%	0.0599
10	1772	0.9809	0.1012	92.11%	94.01%	92.39%	93.19%	0.0614
Mean	1823	0.9701	0.0944	91.52%	96.03%	89.24%	92.48%	0.0592

The trained model is then applied to the testing set. The prediction results are shown in Table 8. The bridge crack image classifier of the confusion matrix is shown in Figure 12. The main indicators, Accuracy and F1-Score, reach 92.31% and 93.21%, respectively, and the testing speed reaches 0.0531 s/img. Some non-crack concrete images containing continuous grooves have relatively high average thermal scores, resulting in them being misclassified as cracks. Nevertheless, some small hairy cracks are very close to the background and have low average thermal scores, resulting in being misclassified as non-crack images.

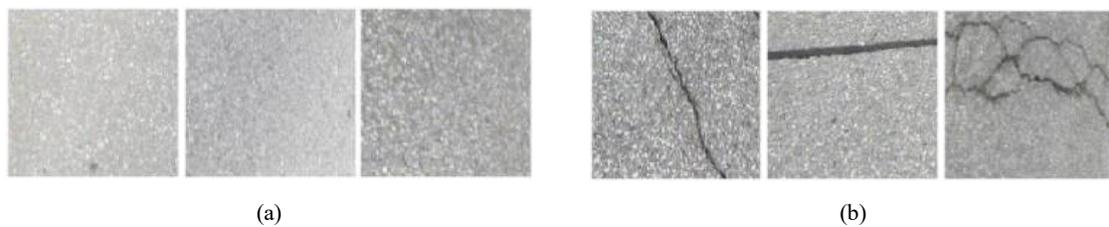
**Figure 12.** Confusion matrix of bridge testing set images.

#### 4.2. Pavement Images

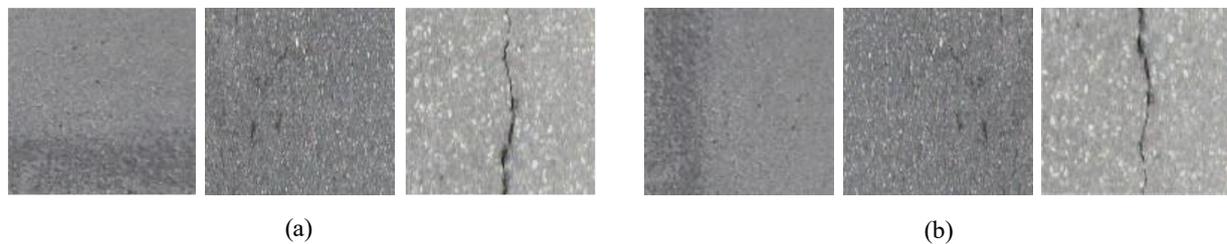
Secondly, 1157 RGB images with a resolution of  $256 \times 256$  pixels in the pavement dataset are used as the experimental samples, including 308 crack images and 849 non-crack images. The specific division results are listed in Table 9. The dataset contains images of uneven illumination, black stains and various cracks, as shown in Figure 13. The partial results of image augmentation are shown in Figure 14.

**Table 9.** Image samples allocated from the pavement dataset.

Number of Images	Training (Expansion)	Calibration	Testing	Total (Expansion)
Non-crack	850 (425)	170	254	1274 (425)
Crack	44 (22)	170	116	308 (22)
Total	894 (447)	340	370	1582 (447)



**Figure 13.** Examples of images contained in the pavement dataset: (a) non-crack images; (b) crack images.

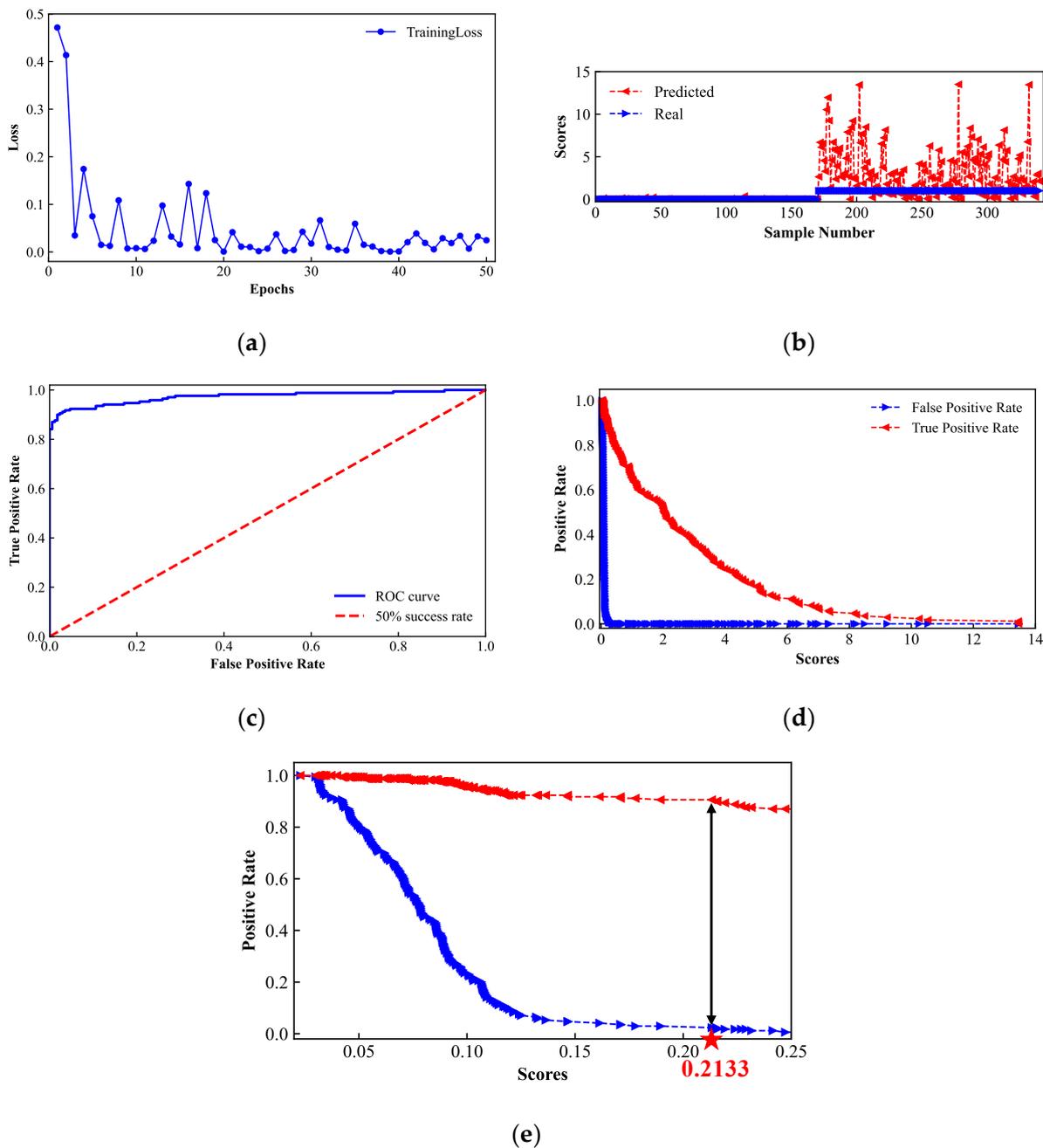


**Figure 14.** Examples of the training images contained in the pavement dataset: (a) original images; (b) expanded images.

As shown in Figure 15a and Table 10, the trained network model tends to be stable with about 50 epochs and 2178 s. The model is used to detect the calibration set. The prediction results are shown in Figure 15b. Generally, the crack detection prediction results are satisfactory. For the non-crack images, the average thermal scores are roughly around 0, and for the crack images, most of the data are far away from 0. For the ROC curve, the value of AUC is 0.9726, and the thermal threshold is 0.2133.

**Table 10.** Experimental results of pavement image classification.

Situation	Training Set	Calibration Set		Testing Set				
	Training Time(s)	AUC	Thermal Threshold	Accuracy	Precision	Recall	F1-Score	Testing Speeds (s/img)
1	2185	0.9916	0.0835	96.22%	94.74%	93.10%	93.91%	0.0505
2	2263	0.9910	0.0924	96.22%	97.22%	90.52%	93.75%	0.0536
3	2190	0.9957	0.0795	97.03%	94.12%	96.55%	95.32%	0.0512
4	2376	0.9784	0.0348	93.51%	87.70%	92.24%	89.92%	0.0562
5	2253	0.9898	0.1428	95.95%	98.10%	88.79%	93.21%	0.0516
6	2213	0.9912	0.0380	95.95%	89.15%	99.14%	93.88%	0.0522
7	2178	0.9726	0.2133	97.57%	99.08%	93.10%	96.00%	0.0510
8	2335	0.9934	0.0427	95.41%	90.24%	95.69%	92.89%	0.0527
9	2226	0.9903	0.1401	97.03%	94.87%	95.69%	95.28%	0.0565
10	2243	0.9911	0.1174	96.22%	95.54%	92.24%	93.86%	0.0519
Mean	2246	0.9885	0.0985	96.11%	94.08%	93.71%	93.80%	0.0527



**Figure 15.** The results of the pavement image classification training and calibration process: (a) training process; (b) calibration set prediction results; (c) ROC curve; (d) overall diagram of scores-positive rate; (e) detailed diagram of a scores-positive rate. The red star represents the thermal threshold.

The obtained trained network model is applied to the testing set. The prediction results are shown in Table 10. The pavement crack image classifier of the confusion matrix is shown in Figure 16. The main indicators, Accuracy and F1-Score, reach 97.57% and 96.00%, respectively. Moreover, the testing speeds reach 0.0510 s/img. Some non-crack images containing tree leaves have relatively high average thermal scores, resulting in them being misclassified as cracks. Some small hairy cracks have low average thermal scores, resulting in being misclassified as non-crack images. The effectiveness of the proposed method is validated again.

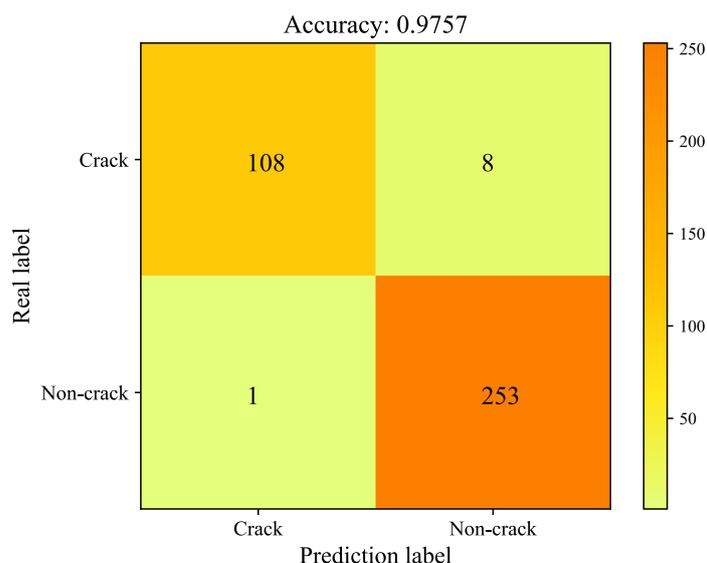


Figure 16. Confusion matrix of pavement testing set images.

### 4.3. Comparative Experiments

To validate the effectiveness of the improved VGG-16 model in anomaly detection, three classical convolutional neural networks, AlexNet [44], ResNet-18 and ResNet-34 [45], are also used as the backbone network model to conduct numerical experiments on the same dataset. The detection results of four network models are listed in Table 11. From Table 11, we can see that the improved VGG-16 model obtains the highest accuracy and F1-Score for all three cases. Moreover, the other evaluation indicators (Precision and Recall) of the improved VGG-16 model are also at the same level as the other three models.

Table 11. Experimental results of different backbone networks in each dataset.

Data Sets	Backbone Network	AUC	Thermal Threshold	Accuracy	Precision	Recall	F1-Score
Wall	VGG-16	0.9998	0.2293	98.86%	100.00%	98.12%	99.05%
	AlexNet	0.9756	0.1568	95.65%	98.65%	96.32%	97.48%
	ResNet-18	0.9903	0.2365	97.86%	100.00%	97.04%	98.50%
	ResNet-34	0.9964	0.1985	98.37%	100.00%	97.86%	98.89%
Bridge	VGG-16	0.9701	0.0944	91.52%	96.03%	89.24%	92.48%
	AlexNet	0.9056	0.0539	83.96%	91.26%	79.09%	84.75%
	ResNet-18	0.9568	0.1232	89.65%	87.25%	93.74%	90.36%
	ResNet-34	0.9365	0.0789	86.32%	93.72%	82.08%	87.51%
Pavement	VGG-16	0.9885	0.0985	96.11%	94.08%	93.71%	93.80%
	AlexNet	0.9563	0.0626	89.36%	90.17%	85.34%	87.69%
	ResNet-18	0.9875	0.1259	94.53%	92.47%	90.53%	91.49%
	ResNet-34	0.9899	0.0768	95.69%	96.95%	87.59%	92.03%

## 5. Conclusions

In this paper, we establish a DCNN based on a semi-supervised learning method for anomaly crack detection to address the challenges posed by the difficulty of obtaining crack data for in-service infrastructure. Some examples are used to demonstrate the high efficiency and excellent performance of the trained model in addressing crack detection problems, including:

- (1) The DCNN is used to realize automatic crack detection, and the influence of subjective factors can be removed.
- (2) Only a small number of labeled datasets are required to train the semi-supervised model, and the computation time and manpower for labor-intensive labeling tasks are greatly reduced.
- (3) The model is tailor-made for crack detection in infrastructure and has good robustness for large changes in structural roughness, uneven illumination and crack differences.
- (4) The neural network can be visualized to describe its working principle, in terms of the heatmaps of different testing results.
- (5) The effectiveness of the DCNN based on a semi-supervised learning method for anomaly crack detection is validated by several experiments.

The method can be used to address the issues of the lack of crack data and interfering with non-crack data effectively. It offers great potential for further applications in engineering practice. In the future, we will conduct a study on how to use the semi-supervised learning anomaly detection method based on a DCNN in more complex backgrounds and multi-defect classification detection. Image segmentation is the process of dividing an image into meaningful pixel-level regions. It is one of the important tasks for crack shape estimation. In future work, we will also explore image segmentation techniques; therefore, classification accuracy can be enhanced.

**Author Contributions:** X.G.: Conceptualization, Methodology Software, Data curation, Writing—original draft preparation, Visualization and Funding acquisition. C.H.: Writing—original draft preparation and Data curation. S.T.: Conceptualization, Methodology, Investigation, Visualization, Writing—original draft preparation and Writing—review and editing. G.C.: Writing—original draft preparation, Writing—review and editing and Project administration. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by [the National Natural Science Foundation of China] grant number [51808135] and [Youth Fund of Guangdong University of Technology] grant number [17QNZD005].

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Biondini, F.; Frangopol, D.M. Life-Cycle performance of deteriorating structural systems under uncertainty: Review. *J. Struct. Eng.* **2016**, *142*, F4016001. [[CrossRef](#)]
2. Rafiei, M.H.; Adeli, H. A novel machine learning-based algorithm to detect damage in high-rise building structures. *Struct. Des. Tall Spec. Build.* **2017**, *26*, 1400. [[CrossRef](#)]
3. Pan, Y.; Zhang, G.; Zhang, L. A spatial-channel hierarchical deep learning network for pixel-level automated crack detection. *Autom. Constr.* **2020**, *119*, 103357. [[CrossRef](#)]
4. Weng, X.; Huang, Y.; Wang, W. Segment-based pavement crack quantification. *Autom. Constr.* **2019**, *105*, 14. [[CrossRef](#)]
5. Zhou, Y.; Wang, F.; Meghanathan, N.; Huang, Y. Seed-Based approach for automated crack detection from pavement images. *Transp. Res. Rec. J. Transp. Res. Board* **2016**, *2589*, 162–171. [[CrossRef](#)]
6. Tsai, Y.C.; Kaul, V.; Mersereau, R.M. Critical assessment of pavement distress segmentation methods. *J. Transp. Eng.* **2010**, *136*, 11–19. [[CrossRef](#)]
7. Su, T.C.; Yang, M.D. Application of morphological segmentation to leaking defect detection in sewer pipelines. *Sensors* **2014**, *14*, 8686–8704. [[CrossRef](#)]
8. Hu, W.; Wang, W.; Ai, C.; Wang, J.; Wang, W.; Meng, X.; Liu, J.; Tao, H.; Qiu, S. Machine vision-based surface crack analysis for transportation infrastructure. *Autom. Constr.* **2021**, *132*, 103973. [[CrossRef](#)]
9. Huyan, J.; Li, W.; Tighe, S.; Deng, R.; Yan, S. Illumination compensation model with k-means algorithm for detection of pavement surface cracks with shadow. *J. Comput. Civ. Eng.* **2020**, *34*, 869. [[CrossRef](#)]
10. Abdel-Qader, I.; Pashaie-Rad, S.; Abudayyeh, O.; Yehia, S. PCA-Based algorithm for unsupervised bridge crack detection. *Adv. Eng. Softw.* **2006**, *37*, 771–778. [[CrossRef](#)]

11. Wang, S.; Qiu, S.; Wang, W.; Xiao, D.; Wang, K.C.P. Cracking classification using minimum rectangular cover-based support vector machine. *J. Comput. Civ. Eng.* **2017**, *31*, 672. [[CrossRef](#)]
12. Shi, Y.; Cui, L.; Qi, Z.; Meng, F.; Chen, Z. Automatic road crack detection using random structured forests. *IEEE Trans. Intell. Transp. Syst.* **2016**, *17*, 3434–3445. [[CrossRef](#)]
13. Oliveira, H.; Correia, P.L. CrackIT—An image processing toolbox for crack detection and characterization. In Proceedings of the 2014 IEEE International Conference on Image Processing (ICIP), Paris, France, 27–30 October 2014.
14. Wang, W.; Yang, Y. Development of convolutional neural network and its application in image classification: A survey. *Opt. Eng.* **2019**, *58*, 040901. [[CrossRef](#)]
15. Cha, Y.-J.; Choi, W.; Büyüköztürk, O. Deep learning-based crack damage detection using convolutional neural networks. *Comput.-Aided Civ. Infrastruct. Eng.* **2017**, *32*, 361–378. [[CrossRef](#)]
16. Protopapadakis, E.; Voulodimos, A.; Doulamis, A.; Doulamis, N.; Stathaki, T. Automatic crack detection for tunnel inspection using deep learning and heuristic image post-processing. *Appl. Intell.* **2019**, *49*, 2793–2806. [[CrossRef](#)]
17. Bang, S.; Park, S.; Kim, H.; Kim, H. Encoder-decoder network for pixel-level road crack detection in black-box images. *Comput.-Aided Civ. Infrastruct. Eng.* **2019**, *34*, 713–727. [[CrossRef](#)]
18. Da Silva, W.R.L.; da Lucena, D.S. Concrete cracks detection based on deep learning image classification. *Proceedings* **2018**, *2*, 489. [[CrossRef](#)]
19. Yang, Q.; Shi, W.; Chen, J.; Lin, W. Deep convolution neural network-based transfer learning method for civil infrastructure crack detection. *Autom. Constr.* **2020**, *116*, 103199. [[CrossRef](#)]
20. Gopalakrishnan, K.; Khaitan, S.K.; Choudhary, A.; Agrawal, A. Deep convolutional neural networks with transfer learning for computer vision-based data-driven pavement distress detection. *Constr. Build. Mater.* **2017**, *157*, 322–330. [[CrossRef](#)]
21. Alipour, M.; Harris, D.K. A big data analytics strategy for scalable urban infrastructure condition assessment using semi-supervised multi-transform self-training. *J. Civ. Struct. Health Monit.* **2020**, *10*, 313–332. [[CrossRef](#)]
22. Liang, H.; Zou, J. Rock image segmentation of improved semi-supervised SVM–FCM algorithm based on chaos. *Circuits Syst. Signal Process.* **2019**, *39*, 571–585. [[CrossRef](#)]
23. Zhao, Y.; Ball, R.; Mosesian, J.; de Palma, J.-F.; Lehman, B. Graph-Based semi-supervised learning for fault detection and classification in solar photovoltaic arrays. *IEEE Trans. Power Electron.* **2015**, *30*, 2848–2858. [[CrossRef](#)]
24. Wu, D.; Liu, C.; Fan, H.; Song, B. Research on abnormal detection of one-class support vector machine based on ensemble cooperative semi-supervised learning. *J. Phys. Conf. Ser.* **2019**, *1237*, 052007. [[CrossRef](#)]
25. Liu, Y.; Yeoh, J. Vision-Based semi-supervised learning method for concrete crack detection. In Proceedings of the Construction Research Congress (CRC) on Construction Research and Innovation to Transform Society, Tempe, AZ, USA, 8–10 March 2020.
26. Wang, W.; Su, C. Semi-supervised semantic segmentation network for surface crack detection. *Autom. Constr.* **2021**, *128*, 103786. [[CrossRef](#)]
27. Zhang, G.; Pan, Y.; Zhang, L. Semi-supervised learning with GAN for automatic defect detection from images. *Autom. Constr.* **2021**, *128*, 103764. [[CrossRef](#)]
28. Quellec, G.; Lamard, M.; Cozic, M.; Coatrieux, G.; Cazuguel, G. Multiple-Instance learning for anomaly detection in digital mammography. *IEEE Trans. Med. Imaging* **2016**, *35*, 1604–1614. [[CrossRef](#)] [[PubMed](#)]
29. Posilovic, L.; Medak, D.; Milkovic, F.; Subasic, M.; Budimir, M.; Loncaric, S. Deep learning-based anomaly detection from ultrasonic images. *Ultrasonics* **2022**, *124*, 106737. [[CrossRef](#)]
30. Le, M.; Luong, V.S.; Nguyen, D.K.; Dao, V.-D.; Vu, N.H.; Vu, H.H.T. Remote anomaly detection and classification of solar photovoltaic modules based on deep neural network. *Sustain. Energy Technol. Assess.* **2021**, *48*, 101545. [[CrossRef](#)]
31. Kähler, F.; Schmedemann, O.; Schüppstuhl, T. Anomaly detection for industrial surface inspection: Application in maintenance of aircraft components. *Procedia CIRP* **2022**, *107*, 246–251. [[CrossRef](#)]
32. Liu, J.; Song, K.; Feng, M.; Yan, Y.; Tu, Z.; Zhu, L. Semi-supervised anomaly detection with dual prototypes autoencoder for industrial surface inspection. *Opt. Lasers Eng.* **2021**, *136*, 106324. [[CrossRef](#)]
33. Chow, J.K.; Su, Z.; Wu, J.; Tan, P.S.; Mao, X.; Wang, Y.H. Anomaly detection of defects on concrete structures with the convolutional autoencoder. *Adv. Eng. Inform.* **2020**, *45*, 101105. [[CrossRef](#)]
34. Sattar, S.; Li, S.; Chapman, M. Developing a near real-time road surface anomaly detection approach for road surface monitoring. *Measurement* **2021**, *185*, 109990. [[CrossRef](#)]
35. Ruff, L.; Vandermeulen, R.; Goernitz, N.; Deecke, L.; Siddiqui, S.A.; Binder, A.; Müller, E.; Kloft, M. Deep One-Class Classification. In Proceedings of the 35th International Conference on Machine Learning, Stockholm, Sweden, 10–15 July 2018.
36. Liznerski, P.; Ruff, L.; Vandermeulen, R.A.; Franks, B.J.; Kloft, M.; Müller, K.R. Explainable Deep One-Class Classification. In Proceedings of the International Conference on Learning Representations, Virtual Event, 3–7 May 2021. [[CrossRef](#)]
37. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv* **2014**, arXiv:1409.1556. [[CrossRef](#)]
38. Huber, J.P. Robust estimation of a location parameter. *Ann. Math. Stat.* **1964**, *35*, 73–101. [[CrossRef](#)]
39. Ruff, L.; Vandermeulen, R.A.; Franks, B.J.; Müller, K.; Kloft, M. Rethinking assumptions in deep anomaly detection. *arXiv* **2020**, arXiv:2006.00339. [[CrossRef](#)]
40. Mirza, M.; Osindero, S. Conditional generative adversarial nets. *arXiv* **2014**, arXiv:1411.1784. [[CrossRef](#)]

41. Radford, A.; Metz, L.; Chintala, S. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv* **2015**, arXiv:1511.06434. [[CrossRef](#)]
42. Akhenia, P.; Bhavsar, K.; Panchal, J.; Vakharia, V. Fault severity classification of ball bearing using SinGAN and deep convolutional neural network. *Proc. Inst. Mech. Eng. Part C J. Mech. Eng. Sci.* **2021**, *236*, 3864–3877. [[CrossRef](#)]
43. Figueira, A.; Vaz, B. Survey on synthetic data generation, evaluation methods and GANs. *Mathematics* **2022**, *10*, 2733. [[CrossRef](#)]
44. Krizhevsky, A.; Sutskever, I.E.G. Hinton. Imagenet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.* **2012**, *25*, 1097–1105. [[CrossRef](#)]
45. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016. [[CrossRef](#)]