



Detection and recognition of concrete cracks on building surface based on machine vision

Xiaofei Zhu¹

Received: 18 February 2021 / Accepted: 1 October 2021 / Published online: 6 November 2021
© Springer-Verlag GmbH Germany, part of Springer Nature 2021

Abstract

Surface cracks need to be detected regularly to ensure the safety of concrete buildings. For the sake of efficiency and accuracy, concrete surface cracks are detected by machine vision technology. This paper briefly introduced the convolutional neural network (CNN) algorithm used for identifying concrete surface cracks. Then, the traditional CNN algorithm was improved by the particle swarm optimization (PSO) algorithm, and it was compared with the support vector machine (SVM) algorithm and the traditional CNN algorithm in the simulation experiment. The results showed that the improved CNN algorithm effectively identified the concrete surface cracks with different cracking degrees; moreover, the precision ratio, recall ratio, and F value of the improved CNN algorithm were superior to those of SVM and traditional CNN algorithms in recognizing cracks on the concrete surface, and the training and testing time was shorter than that of SVM and CNN algorithms.

Keywords Machine vision · Concrete · Crack detection · Convolutional neural network

1 Introduction

Most infrastructure belongs to public resources, which can be used by anyone. Once there is a problem in public infrastructure construction, people's personal and property safety will be damaged; therefore, its safety is very important [1]. However, every building material has a life cycle, so does concrete. With the increase in service time, the concrete will gradually age, and the structure will be damaged under the action of load, natural disasters, and environmental erosion. The accumulation of damages will reduce the bearing capacity of the structure, causing accidents such as collapse [2]. The main purpose of this paper is to realize rapid recognition of the cracks on the surface of concrete buildings to make timely maintenance and prolong the service life of buildings. In the actual inspection process, the most commonly used inspection method is manual inspection, but this method is time-consuming, laborious, inefficient, and extremely dependent on the subjective judgment of inspectors, which is severely restricted by the external environment

[3]. Therefore, there is a need for machine detection technology that does not rely on manpower and is not restricted by environmental conditions. Studies on machine vision detection of cracks in China and abroad were basically similar in principle and the flow of related algorithms. For example, when using machine vision to identify cracks on the surface of a concrete building, the flow usually includes image preprocessing, image feature extraction, and identification of extracted features [4]. In the face of different application environments, specific links and measures will change. Lins [5] developed a crack measurement system based on the machine vision concept and verified the application of the method in actual structure through experiments. In order to automatically extract the unobvious cracks on the premise of retaining the width information, Zhao et al. [6] designed an anisotropic clustering algorithm based on the surface geometric characteristics. The experimental results showed that the cracks extracted by the method were very similar to the manually traced real cracks on the ground. Islam et al. [7] proposed an automatic crack detection method based on machine vision, composed of a complete convolutional neural network (FCN) and a codec framework for semantic segmentation. The experimental results showed that the method was very effective for concrete crack classification and the recall rate and average F1 score were about 92%. In the above-mentioned related studies, Lins et al. mainly used

✉ Xiaofei Zhu
fx31590@163.com

¹ School of Architecture and Urban Planning, Henan University of Urban Construction, Longxiang Avenue, Xincheng District, Pingdingshan 467036, Henan, China

machine vision to measure the crack size and identify surface cracks based on the size; however, this method can only determine the size of the crack but cannot determine whether it is a crack, and the dark lines on the surface may also be identified as cracks for size calculation. Zhao et al. also applied machine vision to the calculation of crack size. In this paper, cracks on the surface of concrete buildings were identified by the convolutional neural network (CNN) algorithm. The CNN algorithm was optimized by PSO to replace the adjustment means of hyperparameters. In order to prevent the CNN algorithm from overfitting in the training process and fall into the locally optimal solution. Then, a simulation experiment was performed on the improved CNN algorithm in the MATLAB software. The final experimental results verified the effectiveness of the improved CNN algorithm in identifying cracks on the concrete surface. The novelty of this paper is that parameters in the algorithm were adjusted by PSO to avoid the CNN algorithm falling into locally optimal solution in the iterative process as much as possible. The improvement of the CNN algorithm and the research results of the improved CNN algorithm in the surface crack recognition of concrete buildings in this study provide an effective reference for the application of machine vision in identifying cracks.

2 Recognition of concrete surface cracks by machine vision

Under the influence of the load borne by the concrete building and the change of the external environment, cracks will occur in the concrete structure, which will further affect the stability of the structure, aggravate the influence of the load and external factors, cause more serious cracks, and cause a vicious circle [8]. Therefore, in the face of concrete buildings, it is necessary to detect the building structure regularly to find the surface cracks in time. The traditional surface crack detection is manual. The efficiency of the method is low as the staff directly inspect the concrete building surface, and the accuracy depends on the experience and concentration degree of the inspectors, leading to unstable detection results [9]. Also, it is difficult to dispatch staff to implement the manual detection method in the face of difficult detection occasions, such as overpass piers and concrete structures below the water surface. After the emergence and development of machine vision technology, machines replace humans to detect cracks on the surface of concrete, which improves the accuracy of detection and ensures the stability of high accuracy [10]. With the help of intelligent machine equipment equipped with machine vision technology, it can replace human detection on high-risk occasions to ensure the safety of staff [11].

2.1 Image preprocessing

The functions of preprocessing [12] include: (1) reducing the amount of data in color images captured by the device; (2) removing the “noise” contained in images to reduce the interference in feature extraction. The gray conversion formula is as follows:

$$\text{Gray}(x_{i,j}) = \omega_R R(x_{i,j}) + \omega_G G(x_{i,j}) + \omega_B B(x_{i,j}), \quad (1)$$

where $x_{i,j}$ is the pixel with a coordinate of (i, j) , $\text{Gray}(x_{i,j})$ represents the gray value of the corresponding pixel, $R(x_{i,j})$, $G(x_{i,j})$, and $B(x_{i,j})$ are the components of the red, green, and blue channels of the corresponding pixel [13], and ω_R , ω_G , and ω_B are the weights of corresponding channels. Due to the influence of unstable light sources, camera jitter, and settings, the camera will mix the unavoidable “noise” into the image. The “noise” needs to be removed in order to recognize the image later. In this study, the noise was removed by filtering [14]. The bilateral filtering method is adopted in this study to remove the “noise” as much as possible and preserve the image features in the filtering process.

$$\left\{ \begin{array}{l} I_D(i, j) = \frac{\sum_{k,l} I(k, l) \omega(i, j, k, l)}{\sum_{k,l} \omega(i, j, k, l)} \\ \omega(i, j, k, l) = d(i, j, k, l) \cdot r(i, j, k, l) \\ d(i, j, k, l) = \exp\left(-\frac{(i-k)^2 + (j-l)^2}{2\sigma_d^2}\right) \\ r(i, j, k, l) = \exp\left(-\frac{\|I(i, j) - I(k, l)\|^2}{2\sigma_r^2}\right) \end{array} \right. , \quad (2)$$

where $I_D(i, j)$ is the gray value of pixel (i, j) after filtering, $I(k, l)$ is the gray value of the neighborhood pixel of pixel (i, j) , $\omega(i, j, k, l)$ is the weight coefficient, $d(i, j, k, l)$ is the filtering kernel of the space domain, $r(i, j, k, l)$ is the filtering kernel of the gray domain [15], and σ_d and σ_r are standard deviations of the filtering kernels.

2.2 Convolutional neural network algorithm

The basic process of image recognition by the CNN algorithm [16] is as follows. ① The preprocessed image is input into the input layer. ② The image is convoluted in the convolution layer, and the convolution kernel slides on the image according to some step size. The convolution formula of the convolution kernel in the sliding process is as follows:

$$a_{i,j} = f\left(\sum_{m=1}^M \sum_{n=1}^N \omega_{m,n} x_{i+m-1, j+n-1} + w_b\right),$$

where $f(\bullet)$ is the activation function [17], $a_{i,j}$ is the element with a coordinate of (i, j) in the feature map after activation,

$x_{i+m-1, j+n-1}$ is the element whose coordinate is $(i+m-1, j+n-1)$ in the original map, which is selected for convolution by the convolution kernel, $\omega_{m,n}$ is the weight of the convolution kernel in the m -th row and n -th column, and w_b is the bias of the convolution kernel.

③ After convolution, the feature image is pooled in the pooling layer, i.e., image compression, to reduce the amount of data. The pooling operation [18] is divided into maximum pooling and average pooling. In this study, the maximum pooling operation was used. The target box slides on the feature image by a length and the largest pixel in the target box are taken as the compression result of the target box.

④ Steps (2) and (3) are repeated according to the requirements, and then, the output is output to the fully connected layer. The crack images are classified by the softmax [19] in the fully connected layer.

The CNN algorithm needs to be trained before it is officially used. In the training process, it is trained by the training samples according to steps ①~④. Then, the calculated results are compared with the expected results, and the weight in the convolution formula is adjusted reversely according to the difference. The formula of reversely adjusting the weight parameter is:

$$\Delta\omega(t) = -\eta\nabla E(t) + a\Delta\omega(t-1), \tag{4}$$

where $\Delta\omega(t)$, $\Delta\omega(t-1)$ are the weight adjustment amount of this time and last time, $E(t)$ is the output error of this time, a is the momentum factor, and η is the learning rate [20].

2.3 The improvement of the CNN algorithm

The training and use of the traditional CNN algorithm are shown above. In the training process, it uses the method of reversely adjusting the weight. The learning method is to adjust the weight gradually to let the rules in the neural network gradually fit the actual rules. The efficiency and effect of the adjustment have a great relationship with the learning rate. The learning rate of the traditional CNN algorithm is fixed and depends on experience. If the value of the learning rate is too large, it is difficult to converge; if it is too small, it will fall into the locally optimal solution because of little change. This study improved the weight adjustment method in the process of training to improve the training effect of the CNN algorithm (Fig. 1).

The improved CNN algorithm adjusts the weight using the PSO algorithm [21], and the flow is as follows.

① For the images as training samples, i.e., the concrete images with cracks and the concrete images without cracks, graying and bilateral filtering were carried out. The specific formulas are shown above.

- ② The parameters of CNN and PSO algorithms were initialized. The parameters of the hidden layer in the CNN algorithm were generated by the initial population of the PSO algorithm.
- ③ In the improved CNN algorithm, each particle in the population of the PSO algorithm represented a parameter setting scheme in the hidden layer of the CNN algorithm. The parameters represented by the PSO population were substituted into the CNN algorithm.
- ④. The preprocessed image was input into the CNN algorithm, and convolution and pooling operations were carried out. The convolution formula refers to Eq. (3). The maximum pooling operation was used. The convolution operation was performed on the preprocessed concrete surface image using the convolution kernel. Every convolution kernel could obtain the feature map. Then, the maximum pooling operation was performed on feature maps of the concrete surface obtained by convolution to compress the data volume.
- ⑤. The classification probability of the concrete surface image features was calculated after convolution and pooling operations in the fully connected layer. When the calculation results could take more than two classification labels, i.e., with and without cracks in this study, the one with higher probability was selected, or the final classification was determined according to the set threshold t . Then, the result was compared with the classification result of whether there were cracks in the concrete surface image in the training samples, and the error was calculated. As the final objective of this study was to determine whether there were cracks on the concrete surface, i.e., the result was not a number, but a classification label, the classification error was calculated by cross-entropy. Then, whether the improved CNN algorithm reached the termination condition of iteration was determined; if it did, then the training finished.
- ⑥. If the termination condition of iteration was not reached, the population particles were adjusted using the formula of the PSO algorithm. The formula of the PSO algorithm is:

$$\begin{cases} v_i(t+1) = \varpi v_i(t) + c_1r_1(P_i(t) - x_i(t)) + c_2r_2(G_g(t) - x_i(t)) \\ x_i(t+1) = x_i(t) + v_i(t+1) \end{cases}, \tag{5}$$

where $v_i(t+1)$ and $x_i(t+1)$ are the speed and position of particle i after one time of iteration, $v_i(t)$ and $x_i(t)$ are the speed and position of particle i before iteration, ϖ is the inertia weight of the particle, c_1 and c_2 are learning factors, r_1 and r_2 are random numbers between 0 and 1, $P_i(t)$ is the optimal position that particle i has experienced, and $G_g(t)$ is the optimal position that the

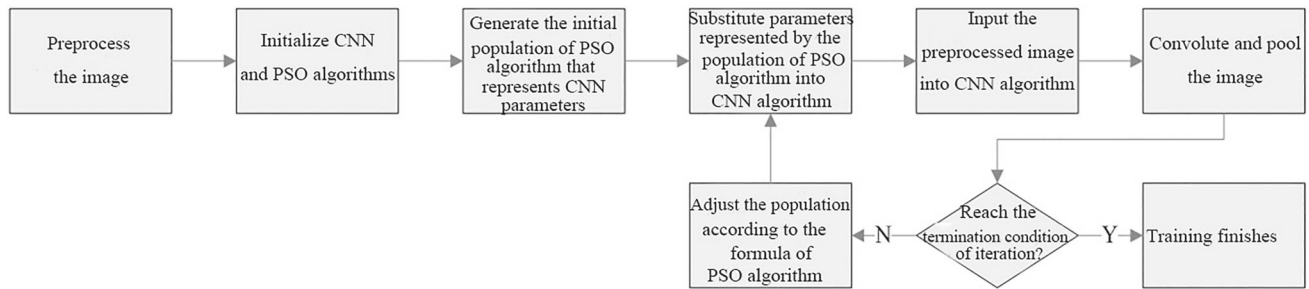


Fig. 1 The training process of the improved CNN algorithm

particle swarm has experienced. When the PSO algorithm adjusted the parameters of the CNN algorithm, the standard for judging whether the particle position was good or bad was the gap between the actual result and the expected result after the convolution, pooling, and classification of the CNN algorithm. The smaller the gap was, the better the particle position was, and the better the represented parameter was.

7. ⑦ The parameters represented by the particles adjusted by the PSO algorithm were substituted into the CNN algorithm, and steps ③~⑥ were repeated.

The process of improving the CNN algorithm with the PSO algorithm for training is as shown above. The improvement of the PSO algorithm to the traditional CNN algorithm was that the parameters in the iteration process of the CNN algorithm were no longer adjusted reversely, but the CNN parameters that needed to be adjusted are used as the coordinates of the particles in the search space in the PSO algorithm. Every iteration of the improved CNN algorithm in the training process was accompanied by the iteration of the PSO algorithm. In the iterative process of the PSO algorithm, the particle swarm gradually gathers at the optimal parameter point in the search space. Unlike the gradual adjustment of the traditional CNN algorithm, the PSO algorithm adjusted the parameters according to the step size that was determined by the flying speed and direction of particles; therefore, the algorithm was prevented from falling into a locally optimal solution in the parameter adjustment process.

3 Experimental analysis

3.1 Experimental environment

In this study, the improved CNN algorithm was simulated by the MATLAB software [22]. The experiment was carried out with a laboratory server configured with Windows 7 system, I7 processor, and 16 G memory.

3.2 Experimental data

In this study, a total of 3000 concrete building surface images were collected, of which 1800 images had no concrete surface cracks and 1200 images contained surface cracks with different cracking degrees. The images of some concrete surfaces with cracks are shown in Fig. 2. Figure 2 shows the concrete surface cracks with three cracking severity degrees, i.e., severe cracking, moderate cracking, and mild cracking, from left to right. There were 325 images of severe cracking, 512 images of moderate cracking, and 363 images of mild cracking. In the simulation experiment, 60% of the images were randomly selected from the images with or without cracks as the training set and the remaining 40% as the testing set.

3.3 Experimental setup

The improved concrete surface crack identification algorithm adopted in this study was developed by introducing the PSO algorithm based on the CNN algorithm. Some structural parameters of the CNN algorithm are shown in Table 1. The relevant parameters of the PSO algorithm are as follows. The number of particle swarms was 30, the two learning factors were set as 1.5, the maximum number of iterations was 1500, and the inertia weight was 0.8.

The main purpose of the simulation experiment was to verify the performance of the improved CNN algorithm in recognizing surface cracks on concrete buildings; therefore, traditional CNN and SVM algorithms were also used for comparison. The basic parameters of the traditional CNN algorithm are shown in Table 1. The difference between the traditional CNN algorithm and the improved CNN algorithm lays in parameter adjustment. The traditional CNN algorithm used the reverse adjustment method; therefore, the initial weight was randomly generated between 0 and 0.5, the learning rate was 0.1, and the maximum number of iterations was 1500. The kernel function of the SVM algorithm was the radial basis function, the penalty parameter was set as 1, and the local binary coding (LBP) was used for extracting features [23].



Fig. 2 Images of concrete surface cracks

Table 1 Parameters of the CNN algorithm

Structure name	Setting	Structure name	Setting
Convolution layer	Three layers	Convolution kernel	64, in a size of 5×5
Pooling layer	Three layers; maximum pooling	Pooling box	In a size of 2×2 ; moving step length: 2
The activation function of the convolution layer	Relu function	Image size in the input layer	150×100

3.4 Evaluation indicators

This study evaluated the performance of three concrete surface crack recognition algorithms by the precision, recall, and F value calculated by the confusion matrix [24]. Table 2 shows a confusion matrix. In the confusion matrix, TP indicates that there were cracks on the concrete surface actually, and the recognition algorithm also judged that there were cracks; FN indicates that there were cracks on the surface actually, but the algorithm judged that there were no cracks; FP indicates that there were no cracks on the surface actually, but the algorithm judged that there were cracks; TN indicates that there were no cracks on the surface actually, and the algorithm also judged that there were no cracks. The calculation formula of indicators for evaluating the performance of the algorithm is:

$$\begin{cases} P = \frac{TP}{TP + FP} \\ R = \frac{TP}{TP + FN} \\ F = \frac{2 \cdot P \cdot R}{P + R} \end{cases} \quad (6)$$

where P represents the precision ratio, which reflects the number of samples with actual cracks when being judged with cracks on the surface, R represents the recall ratio, which reflects the number of samples that are accurately predicted

Table 2 Confusion matrix

	The algorithm judges that there are cracks	The algorithm judges that there are no cracks
With cracks	TP	FN
Without cracks	FP	TN

among the samples with actual cracks, and F is a comprehensive evaluation of precision and recall, which can avoid the difficulty in evaluating when there are contradictions between precision and recall [25].

3.5 Experimental results

Limited by space, this paper only shows some concrete surface cracks and the recognition results of three recognition algorithms. In this study, three typical pictures with different cracking degrees were selected as examples, as shown in Fig. 3. It was seen from the original images in Fig. 3 that image (1) has the most serious cracks, image (2) takes the second place, and image (3) only has shallow cracks. The recognition results of the three recognition algorithms are also shown in Fig. 3. The red lines in the images were the recognition results of cracks. Figure 4 shows that the three recognition algorithms effectively identified the cracks in the face of image (1) with the most serious cracking; in the face of image (2) with the second most serious cracking, the SVM algorithm did not effectively identify the cracks, while the traditional and improved CNN algorithms identified the cracks; in the face of image (3) with the least serious cracking, SVM and traditional CNN algorithms did not effectively identify the cracks, but the improved CNN algorithm did.

The performance test results of three concrete surface crack identification algorithms are shown in Fig. 4. The precision of the SVM algorithm, the traditional CNN algorithm, and the improved CNN algorithm was 77.8%, 90.9%, and 97.8%, respectively. The recall of the SVM algorithm, the traditional CNN algorithm, and the improved CNN algorithm was 72.9%, 83.3%, and 95.5%, respectively. The F value of the SVM algorithm, the traditional CNN algorithm, and the improved CNN algorithm was 75.3%, 86.9%, and 96.6%, respectively. It can be seen from Fig. 4 that the SVM algo-

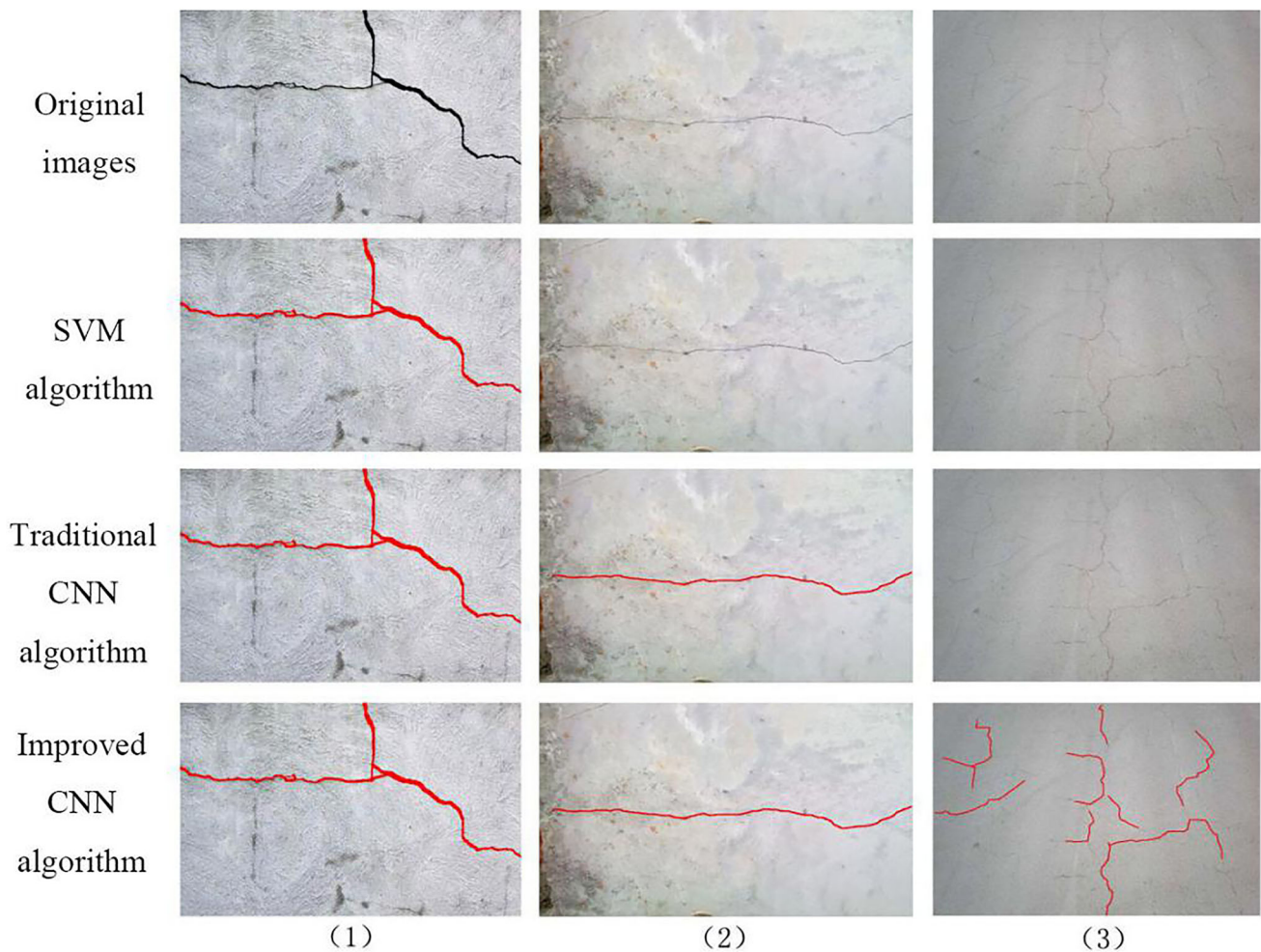


Fig. 3 Part of the original images and the recognition results of the three recognition algorithms

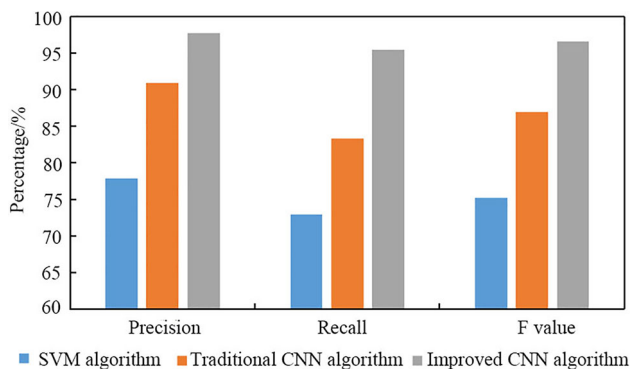


Fig. 4 Surface crack identification performance of the three algorithms

rithm had the lowest precision and recall, and the improved CNN algorithm had the highest precision and recall. To further verify the performance of the three algorithms, the precision and recall were comprehensively evaluated by the F value, and the final result was that the improved CNN algo-

rithm had the largest F value and the SVM algorithm had the smallest F value.

The reason for the above result was that the SVM algorithm needed to extract the features of the image first. For the SVM algorithm, the quality of feature extraction could directly affect the recognition performance. The basic principle of the SVM algorithm for image recognition was to find the “hyperplane” in the vector space composed of the extracted features to make it divide the classification area as much as possible. But in the face of the nonlinear characteristic law, the SVM algorithm was difficult to fit it completely. Compared with the SVM algorithm, the CNN algorithm did not need to rely on the feature extraction algorithm. The convolution kernel pooling operation of the CNN algorithm has the function of feature extraction, and its activation function could effectively fit the nonlinear law. The improved CNN algorithm also used the PSO algorithm to help adjust its parameters, avoiding the locally optimal solution in the learning process of the traditional CNN algorithm.

Table 3 Training and testing time of the three recognition algorithms

	The SVM algorithm	The traditional CNN algorithm	The improved CNN algorithm
Training time	20.2 min	18.4 min	15.3 min
Testing time	835 ms	621 ms	587 ms

The time taken by the three recognition algorithms in training and testing is shown in Table 3. It can be seen from Table 3 that the SVM algorithm took the longest time in the training phase, while the improved CNN algorithm took the least time. The reason for the above result was that the SVM algorithm needed to obtain image features through the feature extraction algorithm before training, but the convolutional and pooling operation in the CNN algorithm had the function of feature extraction. In the training process of the traditional CNN algorithm, the adjustment of internal parameters was carried out step by step; however, the improved CNN algorithm adopted the way of population evolution, which was a multi-scheme parallel contrast adjustment, not a single step-by-step adjustment. In testing, the SVM algorithm took the longest time, while the improved CNN algorithm took the least time. The main reason for the above result was that the SVM algorithm needed to extract additional image features, while the CNN algorithm did not need to. Although the improved CNN algorithm took less time than the traditional CNN algorithm, the difference was relatively small. The structure of the two algorithms was similar after training. The improved CNN algorithm used the PSO algorithm to avoid the locally optimal solution as much as possible.

4 Conclusion

This paper briefly introduced the CNN algorithm used for concrete surface crack recognition. Then, the traditional CNN algorithm was improved by the PSO algorithm. The simulation experiment was carried out on the improved CNN algorithm in MATLAB software. SVM and traditional CNN algorithms were used for comparison to verify the effectiveness of the improved CNN algorithm. The results are as follows. (1) Facing the concrete surface images with different cracking degrees, only the improved CNN algorithm could effectively recognize the concrete surface images with different cracking degrees, the traditional CNN algorithm was difficult to recognize the images with slight cracking, and the SVM algorithm could effectively recognize the images with relatively serious cracking only. (2) In recognizing the concrete surface images with or without images, the precision, recall, and F value of the SVM algorithm were 77.8%, 72.9%, and 75.3%, respectively, those of the traditional CNN

algorithm were 90.9%, 83.3%, and 86.9%, respectively, and those of the improved CNN algorithm were 97.8%, 95.5%, and 96.6%, respectively. (3) The training and testing time of the SVM algorithm was 20.2 min and 835 ms, respectively, that of the traditional CNN algorithm was 18.4 min and 621 ms, respectively, and that of the improved CNN algorithm was 15.3 min and 587 ms, respectively.

Funding Not applicable.

Availability of data and material The data that support the findings of this study are available from the corresponding author upon reasonable request.

Code availability Not applicable.

Declarations

Conflicts of interest The author declares no conflict of interests.

References

1. Yao, Y., Tung, S.T.E., Glisic, B.: Crack detection and characterization techniques—An overview. *Struct. Control Hlth.* **21**(12), 1387–1413 (2015)
2. Yuan, W., Xue, D.: Review of tunnel lining crack detection algorithm based on machine vision. *Chin. J. Sci. Instrum.* **38**(12), 3100–3111 (2017)
3. Li, D., Jiang, D., Bao, R., Chen, L., Kerns, M.K.: Crack detection and recognition model of parts based on machine vision. *J. Eng. Sci. Tech. Rev.* **12**(5), 148–156 (2019)
4. Mohan, A., Poobal, S.: Crack detection using image processing: A critical review and analysis. *Alex. Eng. J.* **57**, 787–798 (2018)
5. Lins, R.G., Givigi, S.N.: Automatic crack detection and measurement based on image analysis. *IEEE T. Instrum. Meas.* **2016**, 583–590 (2016)
6. Zhao, G., Wang, T., Ye, J.: Anisotropic clustering on surfaces for crack extraction. *Mach. Vision Appl.* **26**(5), 675–688 (2015)
7. Islam, M.M.M., Kim, J.M.: Vision-based autonomous crack detection of concrete structures using a fully convolutional encoder–decoder network. *Sensors* **19**(19), 4251 (2019)
8. Ni, F.T., Zhang, J., Chen, Z.Q.: Zernike-moment measurement of thin-crack width in images enabled by dual-scale deep learning. *Comput-Aided Civ. Inf.* **34**(5), 367–384 (2019)
9. Cubero-Fernandez, A., Lozano, F.J.R., Villatoro, R., Olivares, J., Palomares, J.M.: Efficient pavement crack detection and classification. *J. Image Video Proc.* **2017**, 39 (2017)
10. Aldea, E., Hégarat-Masclé, S.L.: Robust crack detection for unmanned aerial vehicles inspection in an a-contrario decision framework. *J. Electron. Imaging* **24**(6), 061119 (2018)
11. Luo, Q., Ge, B., Tian, Q.: A fast adaptive crack detection algorithm based on a double-edge extraction operator of FSM. *Constr. Build. Mater.* **204**(APR.20), 244–254 (2019)
12. Mahdavi-pour, Z., Abdullah, M.Z.: Micro-crack detection of polycrystalline silicon solar wafer. *IETE Tech. Rev.* **2015**, 428–434 (2015)
13. Chen, F.C., Jahanshahi, R.: NB-CNN: deep learning-based crack detection using convolutional neural network and Naïve Bayes data fusion. *IEEE T. Ind. Electron.* **65**(99), 4392–4400 (2018)

14. Ni, F.T., Zhang, J., Chen, Z.Q.: Pixel-level crack delineation in images with convolutional feature fusion. *Struct. Control Health* **26**(1), e2286.1–e2286.18 (2019)
15. Medina, R., Llamas, J., Gómez-García-Bermejo, J., Zalama, E., Segarra, M.J.: Crack detection in concrete tunnels using a gabor filter invariant to rotation. *Sensors* **17**(7), 1670 (2017)
16. Chen, F.C., Jahanshahi, M.R.: ARF-Crack: rotation invariant deep fully convolutional network for pixel-level crack detection. *Mach. Vision Appl.* **31**(6), 1–12 (2020)
17. Shi, P., Fan, X.N., Ni, J., Khan, Z., Li, N.: A novel underwater dam crack detection and classification approach based on sonar images. *PLoS ONE* **12**(6), e0179627 (2017)
18. Teo, T.W., Abdullah, M.Z.: In-line photoluminescence imaging of crystalline silicon solar cells for micro-crack detection, pp. 66–70 (2016).
19. Zhang, J., Gui, Y.T., Ao, B.Z.: Passive RFID sensor systems for crack detection & characterization. *NDT & E Int.* **86**(MAR.), 89–99 (2017)
20. Lin, C.S., Chen, S.H., Chang, C.M., Shen, T.W.: Crack detection on a retaining wall with an innovative, ensemble learning method in a dynamic imaging system. *Sensors (Basel, Switzerland)* **19**(21), 4784 (2019)
21. Yang, J., Wang, W., Lin, G., Li, Q., Sun, Y.Q., Sun, Y.X.: Infrared thermal imaging-based crack detection using deep learning. *IEEE Access* **7**, 182060–182077 (2019)
22. Padalkar, M.G., Joshi, M.V.: Auto-inpainting heritage scenes: a complete framework for detecting and infilling cracks in images and videos with quantitative assessment. *Mach. Vision Appl.* **26**(2–3), 317–337 (2015)
23. Yang, C., Liu, P., Yin, G., Wang, L.: Crack detection in magnetic tile images using nonsubsampling shearlet transform and envelope gray level gradient. *Opt. Laser Technol.* **90**, 7–17 (2017)
24. Chen, F.C., Jahanshahi, M.R., Wu, R.T., Joffe, C.: A texture-based video processing methodology using bayesian data fusion for autonomous crack detection on metallic surfaces. *Comput-Aided Civ. Inf.* **32**(4), 271–287 (2017)
25. Pahlberg, T., Thurley, M.J., Popovic, D., Hagman, O.: Crack detection in oak flooring lamellae using ultrasound-excited thermography. *Infrared Phys. Technol.* **88**, 57–69 (2018)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.