Research Article

Semantic Recognition and Location of Cracks by Fusing Cracks Segmentation and Deep Learning

Qing An ¹, **Xijiang Chen** ^{1,2}, **Xiaoyan Du** ², **Jiewen Yang** ², **Shusen Wu** ³,
and **Ya Ban** ⁴

¹School of Artificial Intelligence, Wuchang University of Technology, Wuhan, Hubei 430223, China

²School of Safety and Emergency Management, Wuhan University of Technology, Wuhan, Hubei 430079, China

³State Key Laboratory of Materials Processing and Die and Mould Technology, School of Materials Science and Engineering, Huazhong University of Science and Technology, Wuhan, Hubei 430074, China

⁴Chongqing Measurement Quality Examination Research Institute, Chongqing 404100, China

Correspondence should be addressed to Xijiang Chen; cxj_0421@163.com and Shusen Wu; ssw636@hust.edu.cn

Received 24 June 2021; Revised 19 July 2021; Accepted 2 August 2021; Published 9 August 2021

Academic Editor: Ning Cai

Copyright © 2021 Qing An et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

For a long time, cracks can appear on the surface of concrete, resulting in a number of safety problems. Traditional manual detection methods not only cost money and time but also cannot guarantee high accuracy. Therefore, a recognition method based on the combination of convolutional neural network and cluster segmentation is proposed. The proposed method realizes the accurate identification of concrete surface crack image under complex background and improves the efficiency of concrete surface crack identification. The research results show that the proposed method not only classifies crack and noncrack efficiently but also identifies cracks in complex backgrounds. The proposed method has high accuracy in crack recognition, which is at least 97.3% and even up to 98.6%.

1. Introduction

At present, the use of Chinese concrete structure is fairly wide, involving many engineering industries. But with various cracks generated in the process of construction and using, potential safety hazard has been brought to engineering construction and maintenance. Now, the most common detection method is still manual detection, which is not only inefficient, but also brings safety problems. And the more advanced nondestructive detection methods are costly and cannot realize absolute noncontact detection. In recent years, with the continuous development of computer vision technology, the realization of high-precision crack detection based on computer technology has become a hot spot. The effects of detection methods are mainly focused on the sample-data cluster formation protocol to tackle the leaderless cluster formation problems [1]. They are mainly divided into traditional image processing methods (including edge detection [2], threshold segmentation [3], and region segmentation [4])

and machine learning methods (including deep learning [5] and clustering segmentation [6]).

For the traditional image processing methods, Xu et al. [7] used phase angle and gray distribution to improve the Canny operator, which can identify fine surface cracks, but generate more noise. Rivera et al. [8] tried to detect the cracks on the concrete surface by morphological methods and crack splicing, which can remove noise well, but the splicing effect of the crack's overall skeleton is not very good. Su and Yang [9] proposed an algorithm of image segmentation enhancement, named morphological segmentation based on edge detection-II (MSSED-II), to concrete crack segmentation. Lu et al. [10] used a new double-threshold algorithm combined with morphological denoising to realize the identification of strain hardening cementitious composites crack images, but this method cannot identify cracks with complex backgrounds. Based on the edge detection algorithm, Nikraves and Nezamivand Chegini [11] proposed a detection method using wavelet transform,

which can identify crack images with insignificant grayscale differences better and can identify cracks under a certain complex background, but the effect is not significant enough.

Compared with traditional image processing methods, Rao et al. [12] proposed a faster and simpler single-level convolutional neural network based on real-time target detection technology, which can not only reduce the impact of background on crack identification to a certain extent, but also identify various surface cracks of concrete bridges, but the accuracy is only 66%. Based on deep fully convolutional neural network, Ren et al. [13] proposed a multiscale crack feature extraction network structure for concrete tunnels named CrackSegNet, which has a strong ability to split the entire crack, but it still has trouble in crack extraction with complex background. Dung and Anh [14] introduced “Full Convolutional Neural Network” to realize effective detection of crack targets in a complex background and reduce false marks, but the processing efficiency is still not high enough. Based on the improved and optimized convolutional neural network “GoogLeNet,” Ye et al. [15] have realized the high-precision identification of multiple cracks with complex background, but the identification accuracy of the crack data set is not very high.

For the accurate identification of concrete surface cracks, we proposed a method of concrete surface crack identification in this paper. It is based on the combination of convolutional neural network and clustering segmentation algorithm and can achieve accurate identification and size calculation of concrete surface crack images.

2. Preliminary Identification of Cracks

2.1. Crack Identification Network (CIN). According to the diversity of crack shapes and the uncertainty of crack magnitude, this paper proposes a convolution neural network structure called crack identification network (CIN) based on the convolutional neural network, as shown in Figure 1. Set serial number as i , and layer i as L_i ; convolution operation i as C_i ; and pooling operation i as P_i . L_1, L_3, L_4, L_6, L_8 , and L_9 are convolutional layers; L_2, L_5, L_7 , and L_{10} are pooling layers; L_{11} is fully connected layer; L_{12} is ReLu layer; L_{13} is dropout layer; and L_{14} is softmax layer.

The basic structure of the model consists of six convolutional layers, four pooling layers, a fully connected layer, an input layer, and an output layer. All convolutional layers and fully connected layers are batch normalized to improve the generalization ability of the model. In order to ensure the images which may be too large or too small not to affect the recognition effect, set the input images size as $256 \times 256 \times 3$. The three dimensions are length, width, and RGB components (red, green, and blue components). In order to perform feature compression better and simplify network complexity, set pooled window size of L_2, L_5, L_7 , and L_{10} as $2 \times 2, 4 \times 4, 4 \times 4$, and 6×6 and step size as 2 and apply the maximum pooling rule. In order to obtain deeper image features and ensure that the complexity of the network would not increase, set convolution kernel size of convolutional layers L_1, L_3, L_6 , and L_8 as 3×3 and convolution kernel size of convolutional layers L_4 and L_9 as 5×5 ; step size

of all convolutional layers is 1. After convolution and pooling, through the Flatten operation, a vector containing 512 elements is output to the ReLu layer. Then, the partial neuron connection is randomly cut off by the dropout layer, and the dependence between network model parameters is reduced, the robustness of the model is improved, and overfitting is effectively prevented. Finally, the softmax layer is used to judge whether the data after the above processing is a crack image.

According to the constructed crack identification network, the cracks can be preliminarily identified through the following steps. First, we established a large number of crack data sets, and the data set was divided into a training set and a verification set. Next, we used the training set and validation set to train and validate the crack identification network model, and last, we used test set to test the model to predict the effect of the model prediction. The overall identification process is shown in Figure 2.

2.2. Data Set Making. The crack acquisition mechanism shown in Figure 3 was constructed to obtain the crack image to make a data set. We used the CCD industrial camera (Basler acA1300–30 gm) mounted on the drone to collect 1000 original images, most of which were crack pictures with complex backgrounds (such as crack pictures with rough surfaces, moss, bumps, depressions, and stains), and each image was different due to different photographing conditions. The size of the collected original images was 4896×3672 pixels. They are too large to input directly, so they are divided into 512×512 subimages in steps of 248, and 1000 original images are divided into 46000 subimages. In order to detect cracks and eliminate noise more effectively, some subimages were rotated. The final data set contained 58,000 subimages, 48,000 of which were used for training and 10,000 for testing and verifying the effectiveness of the network.

In order to train out a crack detection model with high robustness and high accuracy, the generated data set contains a wide range of crack image types. For example, the crack images contain cross cracks, thick and thin cracks, and cracks with debris. The noncrack images contain concrete surfaces with shadows, bumps, and textures, as shown in Figure 4.

It can be seen from Figure 4 that the blue dotted box in the first image of Figure 4(a) marks the intersection of the cross crack, the box in the second image marks the fallen leaves in the crack image, and the last two images are examples of thick and fine cracks. In Figure 4(b), the blue dotted elliptical frame in the first noncracked image marks the shaded part of the concrete surface. The elliptical frame in the second image marks the depression on the concrete surface. The elliptical frame in the third image marks the bumps on the concrete surface, and the oval frame in the last image marks the texture of the concrete surface.

In order to verify the recognition accuracy of the recognition model effectively and ensure the simplicity of the detection method, we used the 5-fold cross-validation [16] to verify and train 5 sets of models to obtain the training results of the 5 sets of models. Then, we selected a training result

FIGURE 1: Schematic of crack identification network (CIN).

FIGURE 2: Flowchart of identification.

FIGURE 3: Crack collection mechanism.

randomly to show the accuracy and loss rate curves of the training set and the verification set, which is shown in Figure 5.

It can be seen from Figure 5(a) that the model convergence speed is very fast. In the case with a small amount

of sample data, the loss curve drops fast. The curve tends to be stable at 20 epochs, and it indicates that the sample training is almost completed. It can be seen from Figure 5(b) that the accuracy of the verification set continues to rise, and the loss value continues to decrease. The accuracy of the

FIGURE 4: Example of crack and noncrack images. (a) Crack images. (b) Noncrack images.

FIGURE 5: Training and validation results. (a) Training results. (b) Validation results.

verification set tends to 100% at 20 epochs; the loss value tends to 0%, and it indicates that training has finished at 20 epochs.

Figure 6 shows the results of the 5-fold cross-validation. The accuracy rates of the five sets of model training sets are all above 99%, and the average accuracy rate of the validation set is 99.1%. The accuracy of the validation set of the five groups of models has a little change, but the range of change is small, which shows that the trained crack network model has good robustness and stability.

Some images were selected from the data set randomly to test the effect of crack identification of the crack identification network, as shown in Figure 7.

From Figure 7(b), it is clearly visible that the crack identification network based on deep learning can classify cracks and noncracks accurately. On the one hand, as shown in Figure 7(b), in which the first and second layers belong to the classified cracks Images, while the third and fourth layers

belong to the classified noncrack images. On the other hand, the cracks can be identified and located cracks accurately, as shown in the blue box in Figure 7(b). It can be seen from Figure 7(a) that the resolution of the original image is different. For example, the resolution of the first image is significantly greater than that of the second. However, the first and second images in Figure 7(b) can identify the cracks accurately, indicating that the method is not affected by the resolution. At the same time, there are other miscellanies around the cracks, such as the leaves, as shown in Figure 7(a). The blue boxes in the second and seventh pictures of Figure 7(b) show the cracks around the leaves clearly. It indicates that the method is not affected by the surrounding miscellanies. As shown in the first and seventh pictures of Figure 7(b), the cracks in the first picture of Figure 7(a) are significantly larger than those in the seventh picture of Figure 7(a), but the identification effect is basically the same. The fourth picture in Figure 7(a) shows cross

FIGURE 6: Accuracy rate of training and validation for 5 different groups.

FIGURE 7: Example of identification results. (a) Original image. (b) Crack classification and identification results.

cracks, but the fourth picture in Figure 7(b) gives three cross blue box identification results, indicating that the method can automatically decompose the cross crack identification result into straight crack identification result. The noncrack image in the blue dotted oval frame in Figure 7(a) contains large convex and concave concrete surfaces, and the solid blue line in the oval frame shows the concrete surface with large texture. The identification result shown in Figure 7(b) is not affected by those, indicating that the method is not affected by the surface characteristics of the crack. From the above analysis, it can be seen that the crack identification network we proposed is not affected by the image resolution, the surrounding miscellanies, nor the surface characteristics of the crack.

Although the crack identification network accurately positions the determined crack images, the positioning results lack the identification of complete cracks and continuity and cannot eliminate the influence of background miscellanies on the cracks. If we only clip and quantitative calculate according to above results, a large error will occur,

so it is necessary to divide the cracks more accurately to achieve quantitative identification of cracks.

3. Accurate Crack Segmentation

After the crack identification in the previous section, the initial crack image was obtained, but the crack in the crack image was not segmented accurately. For this, we achieve accurate segmentation of cracks based on improved *K*-means clustering algorithm.

3.1. Improved *K*-Means Clustering Segmentation Algorithm.

Although the *K*-means clustering algorithm [17] has fast and efficient segmentation characteristics, it has high requirement for high clustering center selection and is very easy to converge to the local optimal solution, thereby missing the global optimal solution. In view of this, we used dynamic particle swarm optimization (DPSO) to improve the

K-means clustering algorithm. The specific steps of improved *K*-means are as follows:

Step 1. Convert the color image to grayscale image to obtain the initial grayscale image of the crack, and then set the initial cluster number k .

Step 2. Determine the clustering center m according to the *K*-means algorithm.

Step 3. After the clustering center is determined, calculate the fitness value f_i of the particles i [18], which is shown in the following equation:

$$f_i = \sum_{i=1}^n \sum_{j=1}^k \|x_i - m_j\|^2. \quad (1)$$

Here, x_i is the i th data point in the data point set X and m_j is the j th cluster center.

Step 4. Determine the particle inertia coefficient w and learning factor s according to equations (2) and (3) [18]:

$$w = \begin{cases} w_{\min} - \frac{(w_{\max} - w_{\min}) \times (f_i - f_{\min})}{f_{\text{avg}} - f_{\min}}, & f_i \leq f_{\text{ave}}, \\ w_{\max}, & f_i > f_{\text{ave}}. \end{cases} \quad (2)$$

Here, $w_{\max}$ is the maximum inertia coefficient; $w_{\min}$ is the minimum inertia coefficient; f_i is the current adaptation value of particles; f_{ave} is the current average adaptation value of all particles; and $f_{\min}$ is the minimum adaptation value of all particles.

$$\begin{cases} s_1 = s_{1,\text{int}} + \frac{s_{1,\text{fin}} - s_{1,\text{int}}}{t_{\max}} \times t, \\ s_2 = s_{2,\text{int}} + \frac{s_{2,\text{fin}} - s_{2,\text{int}}}{t_{\max}} \times t. \end{cases} \quad (3)$$

Here, s_1 and s_2 represent the particle's self-learning ability and the learning ability to the excellent collective; $s_{1,\text{int}}$ and $s_{2,\text{int}}$ represent the particle's initial learning ability; $s_{1,\text{fin}}$ and $s_{2,\text{fin}}$ represent the particle's final learning ability; $s_{1,\text{int}} > s_{2,\text{int}}$; $s_{1,\text{fin}} > s_{2,\text{fin}}$; and $t_{\max}$ represents the maximum time for algorithm operation.

Step 5. Use the inertia coefficient w and learning factor s obtained in step 4 to update the particle velocity V_i [18] and position X_i [18], which are shown in the following equation:

$$\begin{cases} x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t+1), & j = 1, 2, 3, \dots, d, \\ v_{ij}(t+1) = wv_{ij}(t) + s_1 r_1 [p_{ij} - x_{ij}(t)] + s_2 r_2 [p_{gj} - x_{ij}(t)]. \end{cases} \quad (4)$$

Here, r_1 and r_2 are two random numbers which are evenly distributed on the interval $[0, 1]$; t is time; p_{ij} is the position of particle i during the update process; and p_{gj} is the best position that particle i experienced during the update process.

Step 6. When the maximum number of iterations is reached or the particle swarm fitness variance converges to δ^2 fixed value, the global optimal solution is output to obtain the particle swarm fitness variance δ^2 [18], which is shown in the following equation; otherwise, repeat step two to step four.

$$\delta^2 = \frac{1}{n} \sum_{i=1}^n (f_i - f_{\text{ave}})^2. \quad (5)$$

Step 7. Take the optimal solution of step 5 as the optimal clustering center, and then obtain k clusters.

Step 8. Perform criterion function judgment. The index function of the Davidson Burger Index (DBI) [19] is adopted as the criterion function of the final clustering result, and its calculation formula is shown as follows:

$$\text{DBI} = \frac{1}{k} \sum_{i=1}^{k-1} \sum_{j=i+1}^k \max \left\{ \frac{S_i + S_j}{\text{Dis}(i, j)} \right\}. \quad (6)$$

Here, $\text{Dis}(i, j)$ represents the distance between the i th cluster and the j th cluster; S_i represents the sum of the standard error of the Euclidean distance between each data point in the i th cluster and the center m_i of the cluster; and k is the total number of clusters in the data set.

Step 9. If DBI converges, the clustering result at this time is output as the final result; if it does not converge, repeat steps 2 to 8 until it converges, and then output the result. The overall flowchart of the improved *K*-means algorithm is shown in Figure 8.

3.2. Algorithm Performance Analysis. We used the evaluation index of precision, recall, and *F*-measure to evaluate the effectiveness of the algorithm. The specific evaluation index [20] is defined as shown in Figure 9.

In Figure 9, P stands for precision rate; R stands for recall rate; F stands for *F*-measure value; A stands for the number of images that are crack images and correctly identified as crack; B stands for the number of images that are background images but identified as crack; C represents the number of images that are crack images but identified as background; and D represents the number of images that are background images and are correctly identified as background.

3.3. Comparison with Traditional Algorithms. Based on the crack data set in Section 2, taking a few simple crack images as an example, compare the segmentation effects of the improved *K*-means algorithm with the traditional improved Otsu [21], improved Canny [22] and improved median filtering [23]. Taking the presence and absence of rough surfaces and protrusions as examples, the comparison of segmentation effects is performed as shown in Figures 10 and 11, respectively.

It can be seen from Figure 10 that, for the first crack image with rough surface, although the improved Otsu

FIGURE 8: Overall flowchart of the improved K-means algorithm.

algorithm, Canny algorithm, and median filter algorithm can all segment the crack, but there is still some noise, but the improved K-means algorithm has almost no noise, and the contour of the crack is also clearer. For the second crack image, because the surface is smooth, all algorithms have few noise, but the crack skeleton integrity and continuity of the improved K-means algorithm are higher than the traditional algorithms. It can be seen from the above comparison that

the precise segmentation of the improved K-means algorithm has good noise resistance and accuracy.

As can be seen from Figure 11, for the first image with raised cracks, although the improved Otsu algorithm, Canny algorithm, and median filter algorithm can all segment the cracks, the cracked skeletons are all thick, and the bulge is also segmented out. In addition to almost no noise, the improved K-means algorithm can also restore the outline of

FIGURE 9: Evaluation index.

FIGURE 10: Comparison of rough surface and smooth surface. (a) Original image; (b) segmentation of improved Otsu algorithm; (c) segmentation of improved Canny algorithm; (d) segmentation of improved median filter; and (e) segmentation of this paper's algorithm.

the thin original crack skeleton more accurately. For the second crack image without bumps, because the surface has no bumps, the traditional algorithm only has a small amount of noise, but the crack continuity is poor. But the improved *K*-means algorithm does not have noise, while ensuring integrity and continuity of the crack skeleton well. It can be seen from the above comparison that the precise segmentation of the improved *K*-means algorithm has good noise resistance and accuracy.

Then, combine the algorithms with the crack identification network, and compare and analyze the overall identification effect. The result of evaluation indicators is shown in Figure 12.

As can be seen from Figure 12, based on the data set of this article, the indicators of the improved Otsu algorithm and the improved Canny algorithm are less than 80%, while the improved median filter algorithm only has a precision rate slightly higher than 80%, and the other two indicators are also no more than 80%; and the improved *K*-means algorithm has a recall rate and *F*-measure value of more than 90%, and the accuracy rate has reached 97%. It can be seen from the above that the processing effect of the improved *K*-means algorithm is much higher than the above traditional algorithms.

3.4. Comparison with Clustering Algorithms. Based on the crack data set, the improved *K*-means segmentation algorithm is used to identify crack pictures on the surface of the concrete with moss, fallen leaves, or water stains, and the identification results are combined with the *K*-means algorithm [24], means shift algorithm [25], and fuzzy *C*-means algorithm for comparative analysis, as shown in Figure 13.

It can be seen from Figure 13 that when identifying the first and second cracks with moss in Figure 13(a), the improved *K*-means algorithm removes almost all noise and guarantees the integrity of the crack skeleton well. For the third cracked image with fallen leaves in Figure 13(a) (where the fallen leaves are marked with a blue dotted box), the other three algorithms have failed to eliminate the noise from the fallen leaves. The two cracks connected to the second small segment were clearly identified. For the identification of the fourth crack with fallen leaves and shadows in Figure 13(a) (where the fallen leaves are marked with a blue dotted box and the shadow is marked with a blue solid line), the *K*-means algorithm, mean shift algorithm, and fuzzy *C*-means algorithm can hardly identify cracks because the cracks are too fine, and neither can remove the noise caused by the fallen leaves and shadows. However, the improved *K*-means algorithm identified the entire fine

FIGURE 11: Comparison of salient surface and flat surface. (a) Original image; (b) segmentation of improved Otsu algorithm; (c) segmentation of improved Canny algorithm; (d) segmentation of improved median filter; and (e) segmentation of this paper's algorithm.

FIGURE 12: Comparison of evaluation indicators of each algorithm.

cracks well when the fallen leaves and shadow noises are completely removed. For the last picture with water stains in Figure 13(a) (where water stains are marked with solid blue boxes), the other three types of algorithms all have a lot of noise and the upper left part of the crack is missing, and the lower rightmost part of the crack is unable to be identified due to the influence of water stains and the grayscale difference of the concrete surface, but the improved *K*-means algorithm completes the clear identification of the overall crack without noise, and the small arc in the middle of the crack is also clearly identified. The above analysis shows that the crack identification method is not only not affected by moss, shadow, water stains, and miscellanies, but also not affected by the size of the crack and the surface characteristics. Even for such cracks under complex backgrounds, this paper's identification method can still identify and segment the crack skeleton with high accuracy, and it also guarantees the great advantage of no noise and the integrity and continuity of the crack skeleton. And the comparison of the evaluation indexes of the algorithms is shown in Figure 14.

It can be seen from Figure 14 that based on the data set, the recall and *F*-measure values of the improved *K*-means

algorithm are both above 90%, and the accuracy rate is 97%, which is still much higher than other three clustering algorithms. It can be inferred that the performance of the improved *K*-means algorithm is better than the other clustering algorithms.

4. Quantitative Identification of Cracks

Use deep learning and improved *K*-means algorithm to accurately extract the crack but cannot determine the size of the crack. For this, this section calculates the physical size of the crack accurately according to the premise determination of the crack pixel size.

4.1. Crack Physical Size. The physical size of the cracks, namely, the area, length, average width, and occupation ratio, needs to be calculated. The calculation method is as follows:

- (1) *Crack Area.* Calculate the number of pixels included in each connected domain, according to the actual physical size corresponding to each pixel, as shown

FIGURE 13: Comparison with traditional algorithms. (a) Original image; (b) segmentation of improved Otsu algorithm; (c) segmentation of improved canny algorithm; (d) segmentation of improved median filter; and (e) segmentation of this paper's algorithm.

FIGURE 14: Comparison of evaluation indicators of each algorithm.

FIGURE 15: Crack marking. (a) Original image; (b) identification of CIN; (c) segmentation; (d) marking.

in the following equation. Thus, the area of each connected domain is obtained, and the areas of all connected domains are summed.

$$P_a = \alpha \cdot P_i. \quad (7)$$

Here, P_a is the actual physical size of the crack, the unit is mm^2 ; P_i is the pixel size of the crack, the unit is pixel^2 ; and α is the zoom ratio of the crack, the unit is $\text{mm}^2/\text{pixel}^2$.

- (2) *Crack Length*. Thin the image of the crack skeleton. The crack width after thinning is a single pixel; then calculate half of the perimeter to get the number of pixels of the crack at this time, and get the crack physical length according to formula (7).
- (3) *Average Crack Width*. The average crack width is the ratio of the crack area to the length. The calculation formula of the average crack width can be obtained by combining the calculation method of the crack length, which is shown as follows:

$$d_i = \frac{2S_i}{C_i}, \quad 1 \leq i \leq n. \quad (8)$$

Here, S_i is the area of the i th crack; C_i is the perimeter of the i th crack; d_i is the average width of the i th crack; and n is the total number of cracks.

- (4) *Crack Occupancy Ratio*. It is the ratio of the area of the crack to the total area of the image.

4.2. Crack Marking. To facilitate the calculation of the size of the crack, the acquired crack image needs to be marked. Based on the data set, combined with the output of the CIN network model, the crack pictures were tested for marking through morphological method. The software used is MATLAB 2016b, the processor is Intel® Core™ i5-8300H CPU @2.30 GHz, the memory is 8.00 GB, the graphics card is NVIDIA GeForce GTX1050Ti, and the operating system is Windows 10.

In order to verify the identification performance better, the identification test was carried out by taking the crack shown in Figure 15 as an example. And the crack was marked in sections to facilitate the size calculation; at the same time, the grayscale image was reversed to improve the effect of marking. And, Figure 15(a) is the original image of the crack; the identification result of CIN is shown in

FIGURE 16: Example of size calculation. (a) Crack 1. (b) Crack 2.

TABLE 1: Pixel size of cracks 1 and 2.

Crack number	Segmentation number	Width (pixel)	Length (pixel)	Average width (pixel)	Overall length (pixel)	Area (pixel ²)	Occupation ration (%)
Crack 1	1	4	107	4.63	1193	5994	0.76
	2	4	63				
	3	4	97				
	4	5	152				
	5	6	163				
	6	6	286				
	7	5	258				
	8	3	67				
Crack 2	1	5	913	4.75	1384	6213	1.47
	2	6	106				
	3	4	240				
	4	4	125				

TABLE 2: Actual size comparison results of cracks 1 and 2.

Crack number	Average width of quantitatively calculation (mm)	Overall length of quantitatively calculation (mm)	Area of quantitatively calculation (mm ²)	Average width of crack gauge measurement (mm)	Overall length of crack gauge measurement (mm)	Area of crack gauge measurement (mm ²)
Crack 1	0.97	250.53	264.33	1.00	251.20	261.52
Crack 2	1.00	290.64	273.99	0.98	288.42	270.26

Figure 15(b); the segmentation of the identified crack is shown in Figure 15(c); and the marking of the split cracks is shown in Figure 15(d).

It can be seen from Figure 15 that the marking of the crack skeleton is very clear and unambiguous due to the precise identification and segmentation, and almost every large turning point is segmented to facilitate the subsequent size calculation.

4.3. Crack Size Calculation. The method of this paper was used to identify and segment cracks and mark them. Then, we chose two images as an example to perform the size calculation, which is shown in Figure 16. The crack size calculation method can achieve quantitative identification of cracks. The shooting equipment used in this article was CCD industrial camera (Basler acA 1300–30 gm), and the zoom ratio of the camera in image shooting is α 0.21 mm/pixel.

FIGURE 17: Original crack image for quantitative calculation.

TABLE 3: Comparison of statistical results.

Group number	Average width of quantitatively calculation (mm)	Average width of crack gauge measurement (mm)	Error (mm)	Overall length of quantitatively calculation (mm)	Overall length of crack gauge measurement (mm)	Error (mm)	Area of quantitatively calculation (mm ²)	Area of crack gauge measurement (mm ²)	Error (mm ²)
1	0.81	0.80	0.01	202.66	205.86	3.20	206.35	203.46	2.89
2	0.76	0.74	0.02	185.13	189.30	4.17	169.14	172.55	3.41
3	0.87	0.86	0.01	221.37	224.75	3.38	249.90	254.73	4.83
4	0.80	0.82	0.02	216.17	212.35	3.82	218.45	222.19	3.74
5	0.78	0.80	0.02	200.65	205.02	4.37	206.40	202.62	3.78
6	0.81	0.80	0.01	210.60	207.68	2.92	212.11	208.19	3.92
7	0.76	0.78	0.02	204.19	200.34	3.85	201.96	197.93	4.03
8	0.80	0.78	0.02	197.57	201.22	3.65	195.80	199.02	3.22
9	0.80	0.82	0.02	207.03	210.75	3.72	217.81	213.78	4.03
10	0.91	0.93	0.02	247.73	243.68	4.05	279.35	284.48	5.13

FIGURE 18: Comparison of accuracy of statistical results.

It can be seen from Figure 16 that segmented marking is adopted due to the different widths of each segment. Crack 1 in Figure 16(a) contains 8 segments, and crack 2 in Figure 16(b) contains 4 segments. Table 1 shows the pixel size and occupancy ratio of each segment of 1 and crack 2 as

well as the overall pixel size. Combining Table 1 and the zoom ratio, the actual sizes of the cracks 1 and 2 were obtained and compared with the measurement results of the crack measuring instrument. The results are shown in Table 2.

It can be seen from Tables 1 and 2 that the error between the size obtained by the identification method and the size obtained by the crack measuring instrument is not very large. In order to verify the accuracy more effectively, a part of the image was randomly extracted from the data set. This part of the image was divided into 10 categories, and each category selected a representative image, as shown in Figure 17. The results of these 10 crack images were identified and quantified and compared with the average width, overall length, and area measured by the crack measuring instrument. The statistics and comparison results are shown in Table 3, and the accuracy chart is shown in Figure 18.

It can be seen from Table 3 and Figure 16 that for 10 representative crack images, the accuracy of the average width calculated by the quantitative identification method is not less than 97.3% compared with the average width measured by the crack measuring instrument, and the highest accuracy rate is up to 98.84%; the accuracy of the overall length is no less than 97.8%, and the highest accuracy is even more than 98.6%; the accuracy of the area is no less than 97.9%, and the highest accuracy is up to 98.5%. From the above comparative analysis, we can see that the quantitative identification method has good stability, and the accuracy is extremely high. Compared with the instrument measurement, the error obtained is basically within the controllable range.

5. Conclusion

This paper proposes a concrete surface crack identification and size calculation method combining deep learning convolutional neural network, clustering segmentation, and morphological methods, which can effectively classify concrete cracks and noncrack images and accurately segment and quantitatively identify cracks. The main conclusions are as follows:

- (1) Using deep learning convolutional neural network for training, a preliminary crack recognition model is obtained. After verification and testing, the identification accuracy rate reached more than 99%. At the same time, the crack identification model of deep convolutional neural network is combined with clustering segmentation and morphological methods to propose a hybrid multialgorithm crack identification extraction method called CIN.
- (2) Compared with traditional segmentation algorithms and similar clustering segmentation algorithms, the improved *K*-means algorithm can identify a variety of concrete surface cracks in complex backgrounds while ensuring extremely high segmentation accuracy.
- (3) Based on the improved *K*-means algorithm, combined with morphological methods, the identification of cracks, quantitative identification, and calculation of physical size are realized. Experiments show that the accuracy of quantitative identification is at least 97.3%, and the highest can reach 98.6%.
- (4) It provides a certain theoretical basis for ensuring the safety of crack detection personnel on the concrete surface, reducing workload, and maintaining and safety testing of concrete structures. It also provides some research bases for the identification of similar cracks under higher precision and more complex conditions.

Data Availability

All the data generated or analyzed during this study are included within this article.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the Chongqing Technological Innovation and Application Development (Program SN: cstc2019jscx-msxmX0051), CRSRI Open Research Program (Program SN: CKWV2019758/KY), the Quality and Technical Supervision Bureau Research Project of Chongqing Province (CQZJKY2018004), and Chongqing Administration for Market Regulation Project (CQSJKY2019004).

References

- [1] X. Yang, L. Liao, Q. Yang, B. Sun, and J. Xi, "Limited-energy output formation for multiagent systems with intermittent interactions," *Journal of the Franklin Institute*, vol. 16, 2021.
- [2] L. Kong, "SIFT feature-based video camera boundary detection algorithm," *Complexity*, vol. 2021, Article ID 5587873, 11 pages, 2021.
- [3] Y. Dong, M. Li, and G. Cheng, "Evaluation and comparison of thresholding segmentation techniques," *In Proceedings of the Mechanics & Materials*, vol. 519-520, pp. 689-692, 2014.
- [4] X. Zhang, P. Xiao, X. Feng, and G. He, "Another look on region merging procedure from seed region shift for high-resolution remote sensing image segmentation," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 2019, no. 148, pp. 197-207, 2019.
- [5] F.-P. An, "Medical image classification algorithm based on weight initialization-sliding window fusion convolutional neural network," *Complexity*, vol. 2019, Article ID 9151670, 15 pages, 2019.
- [6] X. Chen, Q. AN, K. Yu, and Y. Ban, "A novel fire identification algorithm based on improved color segmentation and enhanced feature data," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1-15, 2021.
- [7] H. Xu, Y. Tian, S. Lin et al., "Research of image segmentation algorithm applied to concrete bridge cracks," in *Proceedings of the International Conference on Information Science and Technology*, Yangzhou, China, February 2013.
- [8] J. Rivera, G. Josipovic, E. Lejeune et al., "Automated detection and measurement of cracks in reinforced concrete components," *In Proceedings of the Computer Science*, vol. 112, no. 3, pp. 397-405, 2015.
- [9] T. Su and M. Yang, "Morphological segmentation based on edge detection-II for automatic concrete crack measurement,"

- In Proceedings of the Computer and Concrete*, vol. 21, no. 6, pp. 727–739, 2018.
- [10] C. Lu, J. Yu, and C. K. Y. Leung, “An improved image processing method for assessing multiple cracking development in strain hardening cementitious composites (SHCC),” *Cement and Concrete Composites*, vol. 74, pp. 191–200, 2016.
 - [11] S. Nikraves and S. Nezamivand Chegini, “Crack identification in double-cracked plates using wavelet analysis,” *Meccanica*, vol. 48, no. 9, pp. 2075–2098, 2013.
 - [12] T. Rao, X. Li, H. Zhang, and M. Xu, “Multi-level region-based convolution neural network for image emotion classification,” *Neurocomputing*, vol. 333, pp. 429–439, 2019.
 - [13] Y. Ren, J. Huang, Z. Hong et al., “Image-based concrete crack detection in tunnels using deep fully convolutional networks,” *Construction and Building Materials*, vol. 234, 2020.
 - [14] C. V. Dung and L. D. Anh, “Autonomous concrete crack detection using deep fully convolutional neural network,” *Automation in Construction*, vol. 99, pp. 52–58, 2019.
 - [15] X.-W. Ye, T. Jin, and P.-Y. Chen, “Structural crack detection using deep learning-based fully convolutional networks,” *Advances in Structural Engineering*, vol. 22, no. 16, pp. 3412–3419, 2019.
 - [16] C. Qiu, L. Kong, and S. Yang, “Automatic recognition of flavor types of flue-cured tobacco based on GA-SVM algorithm,” *Tobacco Science and Technology*, vol. 52, no. 2, pp. 101–108, 2019.
 - [17] Y. Zhang, Y. Zhou, X. Guo et al., “Self-adaptive k-means based on a covering algorithm,” *Complexity*, vol. 2018, Article ID 7698274, 16 pages, 2018.
 - [18] D. Nameirakpam and J. Yambem, “An image segmentation approach based on fuzzy C-means and dynamic particle swarm optimization algorithm,” *Multimedia Tools and Applications*, vol. 79, no. 25–26, pp. 18839–18858, 2020, prepublsh.
 - [19] L. Tang, Z. Yu, C. Ren et al., “A method for obtaining spatial structure information of stereo intersections based on vehicle GPS tracks,” *Journal of Transportation Engineering*, vol. 19, no. 5, pp. 170–179, 2019.
 - [20] A. K. Bhandari, I. V. Kumar, and K. Srinivas, “Cuttlefish algorithm-based multilevel 3-D otsu function for color image segmentation,” *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 5, pp. 1871–1880, 2020.
 - [21] R. Chai, “Otsu’s image segmentation algorithm with memory-based fruit fly optimization algorithm,” *Complexity*, vol. 2021, Article ID 5564690, 11 pages, 2021.
 - [22] L. Liu and F. Liang, “Ship infrared image edge detection based on an improved adaptive canny algorithm,” *International Journal of Distributed Sensor Networks*, vol. 14, no. 3, pp. 1–7, 2018.
 - [23] C. Sheela and G. Suganthi, “An efficient denoising of impulse noise from MRI using adaptive switching modified decision based unsymmetric trimmed median filter,” *Biomedical Signal Processing and Control*, vol. 2020, Article ID 101657, 55 pages, 2020.
 - [24] M. Ramalingam and R. Thangarajan, “Mutated k-means algorithm for dynamic clustering to perform effective and intelligent broadcasting in medical surveillance using selective reliable broadcast protocol in VANET,” *Computer Communications*, vol. 150, pp. 563–568, 2020.
 - [25] I. A. Iswanto, T. W. Choa, and B. Li, “Object tracking based on Meanshift and particle-kalman filter algorithm with multi features,” *Procedia Computer Science*, vol. 157, pp. 521–529, 2019.