Journal of Computers Vol. 34 No. 4, August 2023, pp. 203-213
doi: 10.53106/199115992023083404018

Intelligent Crack Detection and Analysis of Building Walls Based on DeepCrack Network

YingGang Xie[1,2*], XueWei Peng[1], YangPeng Xiao[1], YaRu Zhang[1]

[1] Key Laboratory of Information and Communication Systems, Ministry of Information Industry,
Beijing Information Science and Technology University, Beijing, 100101, China
{xieyinggang, xuewei.peng, yangpeng.xiao}@bistu.edu.cn

[2] Key Laboratory of the Ministry of Education for Optoelectronic Measurement Technology and Instrument,
Beijing Information Science and Technology University, Beijing, 100101, China

Received 1 July 2023; Revised 20 July 2023; Accepted 25 July 2023

Abstract. Crack detection is an important aspect to measure the structural stability of buildings. At present, the detection of building cracks still mainly adopts manual detection methods, which rely too much on personal experience, low detection accuracy, and consume a lot of manpower and material resources. In response to this issue, we use an end-to-end method to predict the pixel by pixel crack segmentation DeepCrack network model, and use CRF and GF methods to fuse the final prediction results. Firstly, the ResNet34 model was pre trained on the PASCAL VOC2007 dataset. The DeepCrack + CRF + GF model was used for training, and the Adaptive Threshold method was used to partition and binarize the training results. Finally, the constructed wall crack detection model achieved an AP value of 89.12%, accuracy and recall rates of 83.96%, 88.47%, and IoU value of 85.80%. On the premise of ensuring detection accuracy, the model is only 47 MB, making it possible to deploy it on embedded devices. It can be used in practical engineering applications to build an intelligent building crack detection system, saving a lot of manpower and resources.

Keywords: crack detection, DeepCrack, ResNet34, adaptive threshold

1 Introduction

With the rapid development of economy, the number of infrastructure construction in China continues to grow, and its structural stability detection has become a key issue in the industrial field. Crack detection is an important field to measure the structural stability of buildings [1]. As shown in Fig. 1, once cracks occur in a building, it can lead to corrosion of the steel bars inside the building, reducing the load-bearing capacity of the internal components of the building, and posing serious safety hazards to life and property [2]. Timely detection of building defects, filling and repairing defects, can prevent the continuous deterioration of various hazards, and inspect construction materials and quality, which can play a role in monitoring project quality [3, 4].

Fig. 1. Wall cracks in buildings

[*] Corresponding Author

Early detection and maintenance of cracks in building walls mainly relied on manual detection. The manual detection method is time-consuming and labor-intensive, has low detection accuracy, and is greatly influenced by human factors [5]. In addition, in most cases, cracks cannot be detected visually by the human eye due to the inaccessible nature of the area or the microscopic size of the cracks. This hidden crack can reduce the strength of the structure, leading to ductile or brittle failure, thereby posing a serious safety hazard. Therefore, deep learning technology is used to detect and classify cracks [6], and geometric parameters such as length, width and number of cracks are analyzed, providing a very valuable reference basis for judging the causes of structural stability detection.

Therefore, we optimized the DeepCrack network model and achieved good results for wall crack detection scenarios. The following are the main contributions of our work:

(1) Manually annotate the image dataset, enhance the obtained wall crack images, and use the Labelme open-source tool to polygonally annotate the wall cracks, establishing the wall crack image dataset.

(2) The ResNet34 network model was pre trained on the PASCAL VOC2007 dataset, and the DeepCrack + CRF + GF network model was used for training, resulting in improved detection performance.

(3) We have implemented partition based binarization on the training results and segmented the image into blocks. Instead of segmenting the entire image into pixel by pixel cracks, we have set dynamic N values for different images, dividing them into N regions and segmenting each region pixel by pixel.

The primary organization of this paper is as follows: Part II describes the related work, Part III describes the DeepCrack model structure and improved method used in this paper, Part IV demonstrates the effectiveness of the method in detail through experiments, and Part V concludes the paper.

2 Related Work

In recent years, deep learning technology has been widely applied in various fields, achieving great success in object detection, object classification, and semantic segmentation [7, 8]. Semantic segmentation is an image segmentation based on pixel level, which is more complex than object detection and classification. It not only determines the category of the target, but also extracts the contour of the target to achieve precise positioning. Compared to traditional image processing methods, it has a huge improvement in performance and effectiveness [9]. At present, many scholars have applied semantic segmentation based on deep learning to crack detection. Long et al. [10] proposed a fully convolutional neural network (FCN) based on convolutional neural network (CNN), which replaces fully connected layers with convolutional layers to achieve pixel level classification, marking the beginning of the use of FCN for semantic segmentation. Ronneberger et al. [11] subsequently proposed a U-Net network with a U-shaped symmetric structure, which encodes and decodes image features, and achieves information fusion between high-level and low-level networks. Liu et al. [12] used the U-Net method for crack detection, which greatly improved the segmentation effect compared to FCN, but there were still details loss, missed detection, and false detection. Zhu Suya et al. used U-Net network for crack detection, using threshold method and improved Dijkstra connection algorithm for precise extraction. This method improved detection accuracy, but was limited by the U-Net detection effect [13]. At present, crack segmentation is mostly improved based on U-Net networks. Although U-Net is suitable for small datasets and can perform crack segmentation well, it cannot effectively recover the information lost in the pooling layer, which can easily lead to missed and false detections of small cracks. Due to the lack of clear contrast between wall cracks and the background, uneven pixel count, and a large number of small cracks, crack segmentation is more difficult. Li Li [14] constructed an FCN model based on Alexnet, which can detect cracks in complex road backgrounds. Wang Sen et al. [15] replaced the encoding structure of FCN with Visual Geometry Group (VGG), eliminated the Dropout technique in the fully connected layer, modified the filter size, and expanded the network depth to improve crack detection capability. However, using FCN alone still has the obvious drawback of not being precise enough in segmentation results.

In response to the above issues, we adopt a deep layered convolutional neural network called DeepCrack, which uses end-to-end methods to predict pixel by pixel crack segmentation. DeepCrack is composed of extended FCN and Deeply-Supervised Nets (DSN) [16]. Unlike the standard method of using only the last layer of convolution, this model learns and aggregates multi-scale and multi-level features from low convolution layers to high convolution layers. DSN provides integrated direct supervision for the features of each convolution stage. In addition, guided filtering (GF) and conditional random field (CRF) [17] methods are also used to improve the final prediction results [18, 19]. The experimental results show that before starting the DeepCrack + CRF + GF model training, the ResNet34 model is pre trained on the PASCAL VOC2007 dataset to obtain weights, and the Adaptive Threshold [20] method is used to binarize the training results. At the same time, the dynamic learning rate balancing training time and loss convergence issues are introduced, achieving good results in detecting wall cracks.

3 Method

3.1 Structure of DeepCrack Network

We can describe crack segmentation as a labeling problem of binary image, where “0” and “1” mean “no crack” and “with crack” respectively. As shown in Fig. 2, the DeepCrack network architecture aggregates layered features obtained from multiple layers, using 13 convolutional layers corresponding to the top 13 convolutional layers used for target classification in the VGG16 network [21, 22]. The VGG16 structural model is shown in Fig. 3. The reason for discarding the fifth pooling layer in the VGG16 network is to obtain meaningful side outputs of different scales. However, the output plane generated after the fifth pooling is too small, and the correlated predicted feature maps are too fuzzy to generate precise results; The fully connected layer requires a large amount of computation and occupies a large amount of memory, therefore, the fully connected layer is also discarded. The DeepCrack network uses feature maps from each convolutional stage to predict crack segmentation results, known as side output; Connect all side outputs in series to produce the final fusion result; Adopting a deep supervision network to supervise the output and fusion results on both sides, forming an integrated direct supervision; Apply GF to refine the final fusion result. In this structure, there is no fully connected layer, and a side output layer is inserted after the conventional layer. As the input plane size of the side output layer decreases and the receiving field size increases, the final output layer obtains multi-scale and multi-level features [23].

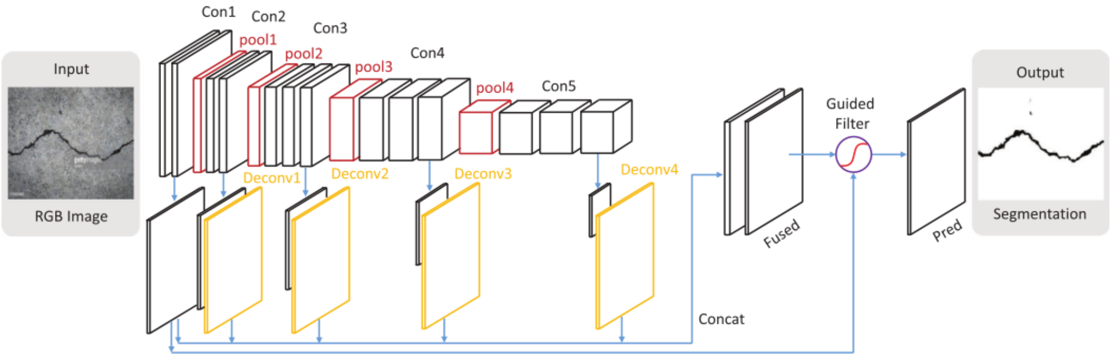


Fig. 2. DeepCrack network architecture

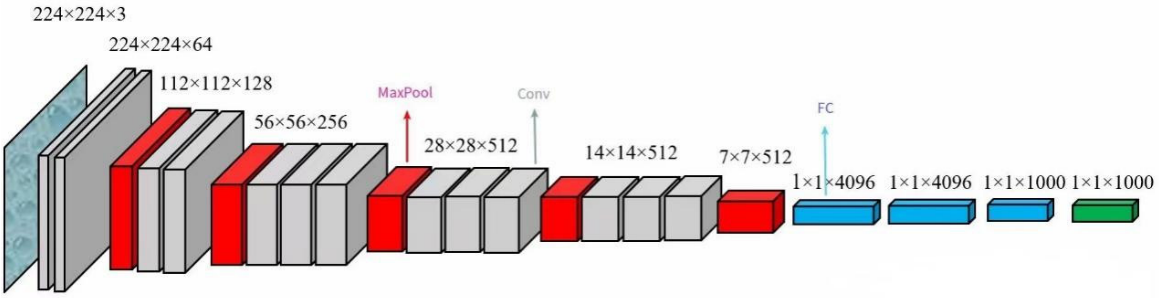
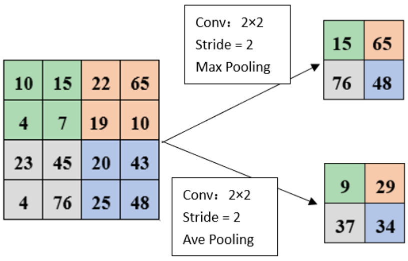


Fig. 3. VGG16 network structure

In DeepCrack network structure, each convolution layer is composed of convolution, batch normalization and rectifying linear unit (ReLU). Convolution is the process of using a filter bank to produce a set of feature mappings; Batch normalization is used to reduce internal covariance shift. The ReLU layer computes the activation function max (0, x) to enable the network to learn nonlinear tasks. The spatial pooling consists of four maximum pooling layers followed by conv1_2, conv2_2, conv3_3 and conv4_3 convolution layers, which can reduce the plane size, realize the translation invariance of image small migration, and greatly reduce the parameter size of the network. The plane size reduction is achieved by Max-pooling with a 2×2 pixel filter. Fig. 4 shows a 4×4 size image. After a step length of 2, the convolution kernel acts as a 2×2 maximum pooling layer. The side output feature is obtained from the convolution layer with kernel size of 1 and output number of M.



10	15	22	65
4	7	19	10
23	45	20	43
4	76	25	48

Conv：2×2
Stride = 2
Max Pooling

15	65
76	48

Conv：2×2
Stride = 2
Ave Pooling

9	29
37	34

Fig. 4. 4x4 image, after the action of the maximum pooling layer with step size of 2 and convolution kernel of 2x2

Except for the first side output layer, the other four side output layers are deconvolution layers used to upsample the plane size of the feature map to the same size as the input image. Then, the upsampled feature maps are connected to form the final features, followed by convolutional layers and softmax layers. The output of the Softmax layer is a probability N-channel mapping, where N is the number of classes. Based on the prediction of the softmax layer, we can obtain the prediction labels for each pixel through a fixed threshold.

Considering the difference in the size of the Receptive field, the deeper prediction is less affected by noise, while the lower prediction presents a more detailed boundary [24, 25]. Finally, CRF and GF were used to improve the trade-off between fusion prediction and output prediction, as shown in Fig. 5.

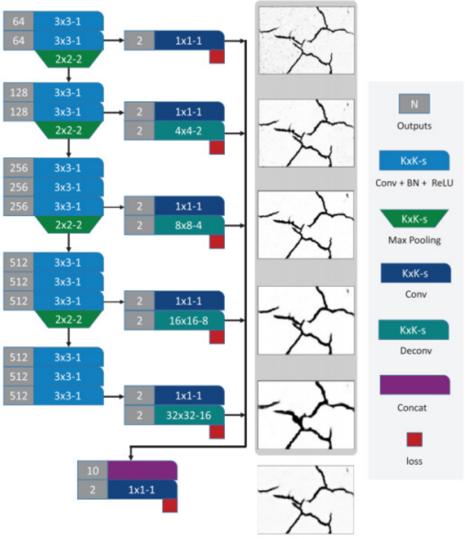


Fig. 5. Detailed of DeepCrack network model

3.2 Structure of ResNet Network

In order to better solve the gradient vanishing problem [26, 27], we comprehensively compared various deep learning algorithms and selected the Residual Network (ResNet) [28-30] model with better performance. ResNet solves the problem of gradient vanishing by introducing residual structures for structural reparameterization, allowing the network to be built deeply and greatly improving detection performance. Before starting the DeepCrack model training, the ResNet34 model was pre trained on the PASCAL VOC2007 dataset to obtain weight values. At the same time, the dynamic learning rate balance training time and loss convergence were introduced to achieve residual structure. The ResNet core structure model is shown in Fig. 6, and the learning process will change from directly learning features to adding some features to the previously learned features to obtain better features [31]. In the past, features were independently learned layer by layer, but now it has become a model where $H(x)=F(x) + x$. Among them, x is the feature at the beginning of the shortcut link, and $F(x)$ is the filling and increasing of x, which becomes the residual. Therefore, the goal of learning is to shift from learning complete information to learning residuals, greatly reducing the difficulty of learning quality features.

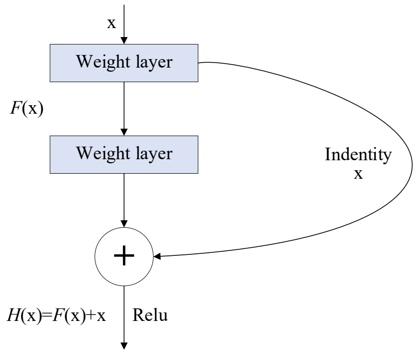


Fig. 6. Structure of Residual module

3.3 Loss Function

For crack segmentation, we define the training set as $S = \{(\boldsymbol{I}_n, \boldsymbol{G}_n), n = 1, ..., N\}$, among them, image samples $\boldsymbol{I}_n = w\{I_j^{(n)}, j = 1, ..., |\boldsymbol{I}_n|\}$, represents the original input image, $\boldsymbol{G}n = \{G_j^{(n)}, j = 1, ..., |\boldsymbol{G}n|\}$, $Y_j \in \{0,1\}$. Set C_0 and C_1 represents the total number of non cracked (negative) pixels and cracked (positive) pixels in the total training set. The prediction includes “side output m” (m = 1, ..., 5). The losses generated by each side’s output are called $a_m\boldsymbol{\Delta}(\boldsymbol{P}^{(m)}, \boldsymbol{G}, \boldsymbol{W}, \boldsymbol{w}^{(m)})$ [31]. The final fuse prediction, also known as L_{fuse} The loss of fuse. The goal of training is to learn a model that minimizes the difference between the final prediction of the network and the actual ground data. In order to learn meaningful crack segmentation features, DSN was applied to supervise each side output layer.

Each side output layer can serve as a pixel level classifier, with corresponding weights $w = \{w(1), ..., w(M)\}$, Where M is the number of side output layers. We express all parameters of the network as, and then simulate the loss function as shown in Equation (1).

$$\begin{aligned} L_{side}(I,G,W,w) &= \sum_{m=1}^{M} \alpha_m \ \ell_{side}(I,G,W,w^{(m)}) \\ &= \sum_{m=1}^{M} \alpha_m \ \Delta(P^{(m)},G,W,w^{(m)}). \end{aligned} \quad \textbf{(1)}$$

Where l_{side} is the image level loss function of side output, $\boldsymbol{P} = \{P_{\text{j}}, j\text{=}1, ..., |I|\}$, The m-th side output layer, if necessary, samples up to the original image size, a_m is a hyperparameter, expressed as the loss weight of each side output layer. In our image to image training, the modified cross entropy function Δ is shown in Equation (2).

$$\begin{aligned}\Delta = &-\sum_{j\in G_+} w_0 \log \Pr(P_j = 1 \big| I, W, w) \\ &-\sum_{j\in G_-} w_1 \log \Pr(P_j = 0 \big| I, W, w).\end{aligned} \tag{2}$$

Where $|\boldsymbol{G}|$, $|\boldsymbol{G}_+|$, $|\boldsymbol{G}_-|$ represents the total number of all pixels, all positive pixels, and all negative pixels of the input image I, respectively. w_0 and w_0=c_0c_1 represents the class loss weights of corresponding non cracked pixels and cracked pixels, respectively. Pr(•) is the probability of a pixel being positive or negative in the predicted graph.

4 Experimental Results and Analysis

All experiments were conducted based on the Pytorch deep learning framework with the programming language of Python. The hardware is configured with an Intel(R) Core (TM) i5-11260H CPU @ 2.40 GHz, 16 GB of RAM, an NVIDIA GeForce GTX 3050Ti (4GB) GPU, and a Windows 10 (64-bit) operating system. We selected the ResNet34 model for pre training on the PASCAL VOC2007 dataset. The main training parameters are listed in Table 1.

Table 1. Training parameters of DeepCrack model

Parameters	Value
Weight	ResNet34
Batch size	16
Learning rate	0.001
Epochs	150
Momentum	0.937

4.1 Data Preparation

The dataset of building wall cracks used in the experiment was from the Beijing Data Center. We also performed data augmentation on the images to expand the dataset. By randomly adding noise, flipping, adjusting brightness, and cutout to the images, we obtained 11075 usable wall crack detection images, and manually marked each wall crack image. The image resolution input to the DeepCrack model was set to 227 × 227 pixels, and the dataset was segmented according to the ratio of training set: validation set: test set=8:1:1. 8860 images were used for training, 1108 images were used for validation, and 1107 images were used for testing. Part of the training set is shown in Fig. 7, with a total of 7962 images with cracks and 898 images without cracks. Due to the fact that most walls are in a cracked state, there are relatively few images of intact and crack free surfaces in the database; The rest are wall images with cracks in different materials, patterns, structures, and crack shapes and sizes, so there are fewer complete wall images that can be used for training, verification, and testing compared to images with cracks. In order to train and validate the algorithm used in this article, we identified crack images from 11075 high-resolution images and obtained 10045 fuzzy crack images as positive examples.

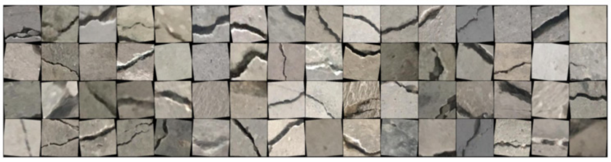


Fig. 7. Training set images

The datasets we use are all RGB images of the same size, so before inputting them into the network, we first convert the RGB images into grayscale images and then perform edge detection on them. In this article, we did not use the Opencv Threshold binarization method because it is suitable for images with distinct areas of different shades. However, due to different lighting and patterns in the dataset, it is easy to cause some images to be overall dark, which can lead to binarization and become completely black, losing all details. The Threshold function uses a threshold to binarize an image, causing all pixels below this threshold to become zero. We cannot manually specify a threshold for each image to be trained or detected, so we propose the idea of dividing the image into two regions, with each region being binarized separately. This means that binarization requires two different thresholds for the image above. If there are more than two areas with different brightness, then a threshold needs to be used for binarization of each area, so the value of N is adaptive and targeted. Therefore, we instead used the cv2. adaptive Threshold function for binarization processing [32, 33]. The use of partition domain binarization method can easily separate the crack and wall parts [34]. Fig. 8 shows the edge detection results after setting the threshold to a minimum value of 90 and a maximum value of 150. It can be observed that the edges can be clearly detected. Fig. 9 compares the original image with the real crack image, but there is still some error. We need to apply the DeepCrack model for training.

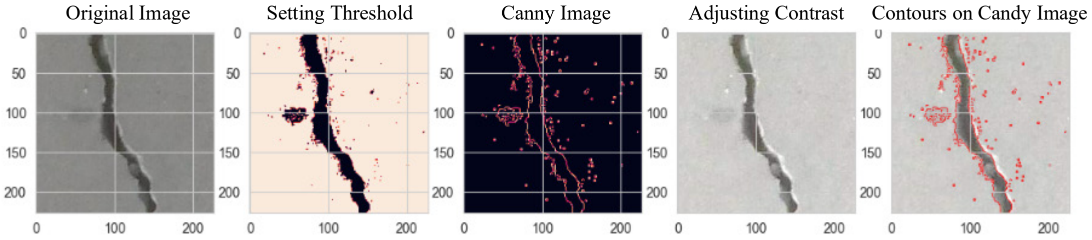
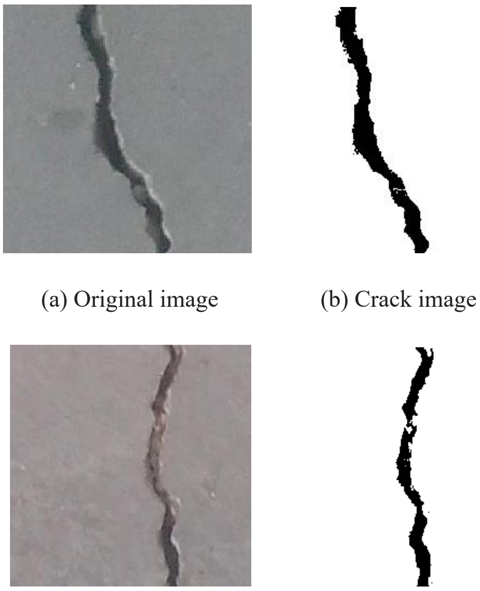


Fig. 8. Partition domain binarization processing



(a) Original image (b) Crack image

(c) Original image (d) Crack image

Fig. 9. Original image and real crack image

4.2 Evaluation Criteria

To evaluate the performance of DeepCrack on the validation set, we introduced three commonly used semantic segmentation evaluation metrics: global accuracy (G), class average accuracy (C), and Intersection over Union (IoU) [35]. Among them, G can measure the percentage of correctly predicted pixels, and the calculation Equation is shown in (3); C represents the prediction accuracy of all classes, as shown in Equation (4); IoU is equal to the ratio of the intersection and union between the "predicted border" and the "real border", calculated as shown in Equation (5).

$$G = \frac{\sum_i n_{ij}}{\sum_i t_i}. \tag{3}$$

$$C = \sum_i \frac{n_{ii}}{\sum_i t_i}. \tag{4}$$

$$IoU = \frac{\left(\frac{1}{n_{cls}}\right)\sum_i n_{ii}}{\left(t_i + \sum_j n_{ji} - n_{ii}\right)}. \tag{5}$$

Where n_{ij} is the number of pixels predicted as class j for class i, n_{cls} different classes, $t_i = \Sigma_j n_{ij}$ is the sum of pixels of class I (both true and false positives are included).

The definitions of Precision, and Recall are given in Equations (6) and (7), respectively.

$$Precision = \frac{TP}{TP + FP}. \tag{6}$$

$$Recall = \frac{TP}{TP + FN}. \tag{7}$$

where TP refers to the number of true-positive samples; FP refers to the number of false-positive samples; FN refers to the number of false-negative samples.

AP is comprehensive metrics for evaluating the detection accuracy of individual categories. The definitions of AP is given in Equation (8). AP is calculated by the enclosed area of the Precision-Recall (P-R) curve and coordinate axis.

$$AP = \int_0^1 p(r)dr. \tag{8}$$

where $p(r)$ refers to the P-R curve plotted by Precision and Recall values

4.3 Analysis of Experimental Results

In order to achieve better detection results of wall cracks, the results of various detection data are shown in Table 2. We compared the DeepCrack model, the DeepCrack + CRF + GF combination model, and the ResNet34 model pre trained on the PASCAL VOC2007 dataset. The DeepCrack + CRF + GF (Pre) model trained using the DeepCrack + CRF + GF model has the advantage of introducing residual structures for structural reparameterization to solve the gradient vanishing problem, allowing the network to be built deep [36], The detection performance has been greatly improved. From the detection results, it can be seen that the AP value based on the model has increased by 1.46%. At the same time, training fewer rounds during the training process can achieve better results. In addition, we also conducted Adaptive Threshold binarization on the results, using the DeepCrack + CRF + GF (Pre + AT) model in Table 2, to further optimize the detection results.

Table 2. Comparison of the recognition performance of different methods

Methods	Metrics					
	G (%)	C (%)	IoU (%)	Pr (%)	Re (%)	AP (%)
DeepCrack	96.82	91.45	80.16	79.48	78.61	82.92
DeepCrack+CRF+GF	97.57	95.36	84.67	83.24	82.15	87.13
DeepCrack+CRF+GF(Pre)	97.32	96.43	85.89	82.89	85.64	87.59
DeepCrack+CRF+GF(Pre+AT)	98.04	97.18	85.80	83.96	88.47	89.12

In addition, we also drew the change curve of the loss function of each iteration, as shown in Fig. 10, the change curve of the loss function of DeepCrack + CRF + GF (Pre), from which we can see that with the increase of the number of iterations, the loss continues to decrease, and better results are achieved after correcting the model parameters.

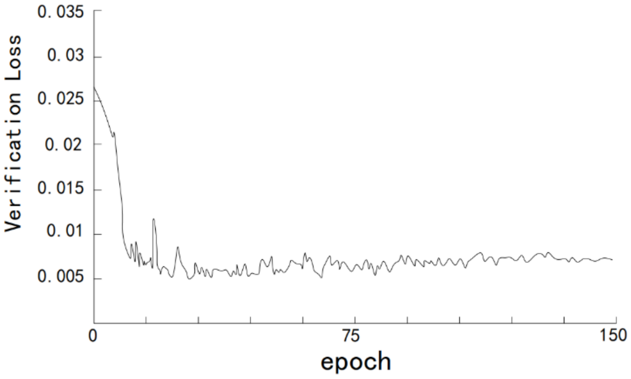
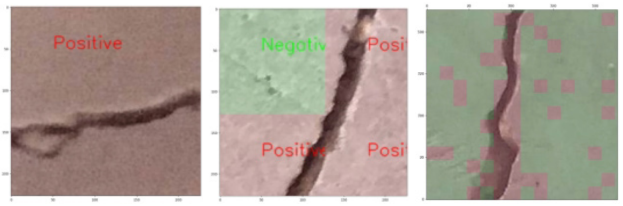


Fig. 10. Loss function change curve

We selected a test set and real images to test and validate the effectiveness of the model. Specifically, the model can accurately determine whether all images in the test set contain cracks, and can also perform block wise judgment to output the predicted severity of cracks. In addition, the model also performs well in the real crack images we have captured and collected, indicating that the model has strong generalization ability and practical application value. The specific results of the test set are shown in Fig. 11.

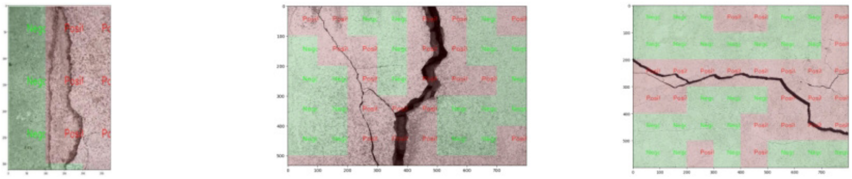


(a) (b) (c)

(a) Predict whether there are cracks in the image (b) Block the image to determine whether there are cracks (c) Analyze the proportion of cracks in the overall image and classify the cracks

Fig. 11. Prediction results on the test set

Fig. 12 shows the prediction results of the images taken from real life, and it can be seen from the figures that the model can accurately outline the shape of cracks and determine the severity of cracks; The ability to accurately predict multi crack images without training on multi crack datasets indicates that the model has good generalization ability; The model also has a certain ability to reflect abnormal situations such as road depressions. Overall, the model performs well in both the test set and the real environment.



(a) Prediction effect on real image (b) Prediction effect on real image (c) Prediction effect on real image

Fig. 12. Prediction effect on real images

5 Conclusion

Reliable crack detection is of great significance for engineering construction. We used the dataset provided by the Beijing Data Center and collected and captured images of wall cracks in real-life scenarios, and manually marked them to construct a wall crack image dataset. We use an end-to-end method to predict the pixel by pixel crack segmentation DeepCrack network model, and fuse the final prediction results using CRF and GF methods. Firstly, the ResNet34 model was pre trained on the PASCAL VOC2007 dataset. The DeepCrack + CRF + GF model was used for training, and the adaptive threshold method was used to partition and binarize the training results. The final constructed model achieved an AP value of 89.12%, accuracy and recall rates of 83.96%, 88.47%, and IoU value of 85.80%, respectively. From the actual scene detection results, it can be seen that the model has achieved good results and has certain application value. In the future, the research team will introduce more different building crack images to improve the generalization ability of the algorithm model and further improve the accuracy of the algorithm model.

Acknowledgement

This work is supported by Beijing Natural Science Foundation (Grant No.4192023 and 4202024); the Gold-Bridge Funds for Beijing.

References

[1] H.-L. Chen, Research on CRTSII track slab crack detection algorithm based on deep learning, [dissertation] Shanghai: Shanghai Institute of Technology, 2021.

[2] Y.-F. Jiang, Identification of apparent Defects in RC bridges with image processing and deep learning, [dissertation] Chongqing: Chongqing University, 2019.

[3] Q. Wen, Research and implementation of building surface crack detection technology based on deep learning, [dissertation] Beijing: Beijing University of Posts and Telecommunications, 2019.

[4] A. Hauashdh, J. Jailani, I. Abdul Rahman, N. Al-Fadhali, Factors affecting the number of building defects and the approaches to reduce their negative impacts in Malaysian public universities' buildings, Journal of Facilities Management 20(2)(2022) 145-171.

[5] J.-W. Wang, Research and system implementation of building surface crack detection based on deep learning, [dissertation] Xi'an: Xi'an University of Architecture and Technology, 2022.

[6] S. Lee, M. Jeong, C.-S. Cho, J. Park, S. Kwon, Deep learning-based PC member crack detection and quality inspection support technology for the precise construction of OSC projects, Applied Sciences 12(19)(2022) 9810.

[7] G.-R. Padalkar, M.-B. Khambete, Fusion-based semantic segmentation using deep learning architecture in case of very

small training dataset, International Journal of Image and Graphics 22(5)(2022) 2250043.

[8] L.-B. Wang, R. Li, C. Zhang, S.H. Fang, C.-X. Duan, X.-L. Meng, P.-M. Atkinson, UNetFormer: A UNet-like transformer for efficient semantic segmentation of remote sensing urban scene imagery, ISPRS Journal of Photogrammetry and Remote Sensing 190(2022) 196-214.

[9] R. Kamath, M. Balachandra, A. Vardhan, U. Maheshwari, Classification of paddy crop and weeds using semantic segmentation, Cogent Engineering 9(1)(2022) 2018791.

[10] J. Long, E. Shelhamer, T. Darrell, Fully convolutional networks for semantic segmentation, in: Proc. 2015 IEEE Conference on Computer Vision and Pattern Recognition, 2015.

[11] O. Ronneberger, P. Fischer, T. Brox, U-net: convolutional networks for biomedical image segmentation, in: Proc. 2015 Medical Image Computing and Computer-Assisted Intervention- MICCAI, 2015.

[12] Z.-Q. Liu, Y.-W. Cao, Y.-Z. Wang, W. Wei, Computer vision-based concrete crack detection using U-net fully convolutional networks, Automation in Construction 104(2019) 129-139.

[13] S.-Y. Zhu, J.-C. Du, Y.-S. Li, X.P. Wang, Method for bridge crack detection based on the U-Net convolutional networks, Journal of Xidian University 46(4)(2019) 35-42.

[14] L. Li, Research on road crack detection algorithms in complex background, [dissertation] Xi'an: Shaanxi Normal University, 2018.

[15] S. Wang, X. Wu, Y.-H. Zhang, Image crack detection in fully convolutional network images based on deep learning, Journal of Computer Aided Design and Graphics 30(5)(2018) 859-867.

[16] X. Wang, Y.-J. Li, H.-Y. Wang, L.-Z. Huang, S.-X. Ding, A video summarization model based on deep reinforcement learning with long-term dependency, Sensors 22(19)(2022) 7689.

[17] N.-G. Rezk, A.-F. Attia, M.-A. El-Rashidy, A. El-Sayed, E.-E. Hemdan, An efficient plant disease recognition system using hybrid convolutional neural networks (CNNs) and conditional random fields (CRFs) for smart IoT applications in agriculture, International Journal of Computational Intelligence Systems 15(1)(2022) 65.

[18] J. Ji, R. Shi, S.-T Li, P. Chen, Q.-G Miao, Encoder-decoder with cascaded CRFs for semantic segmentation, IEEE Transactions on Circuits and Systems for Video Technology 31(5)(2020) 1926-1938.

[19] N.-L. Zamarashkin D.-A. Zheltkov, Gpu acceleration of dense matrix and block operations for lanczos method for systems over GF(2), Lobachevskii Journal of Mathematics 40(11)(2019) 1881-1891.

[20] U. Ali, M.-T. Mahmood, Defocus blur segmentation using local binary patterns with adaptive threshold, Computers, Materials & Continua 71(1)(2022) 1597-1611.

[21] Q. Zou, Z. Zhang, Q.-Q. Li, X.-B. Qi, Q. Wang, S. Wang, Deepcrack: learning hierarchical convolutional features for crack detection, IEEE Transactions on Image Processing 28(3)(2018) 1498-1512.

[22] Z.-C. Hu, Z.-J. Yin, L. Qin, F.-X. Xu, A novel method of fault diagnosis for injection molding systems based on improved VGG16 and machine vision, Sustainability 14(21)(2022) 14280.

[23] J.-H. Kang, End-to-end dense stereo matching based on full convolutional neural network, [dissertation] Wuhan: Wuhan University, 2020.

[24] X.-Q. Sheng, Research on retinal vessel segmentation method based on supervised learning, [dissertation] Jiangxi: Jiangxi University of Science and Technology, 2020.

[25] Q. Quan, Methods for improving generalization of convolutional nurral networks on image classification and detection, [dissertation] Wuhan: Wuhan University, 2020.

[26] H.-D. Yu, J. Yin, Aging residual factorization machines: a multi-layer residual network based on aging mechanisms, Applied Sciences 12(11)(2022) 5318-5318.

[27] J.-Y. Chen, Z. Liu, Mask dynamic routing to combined model of deep capsule network and U-Net, IEEE transactions on neural networks and learning systems 31(7)(2020) 2653-2664.

[28] W.-J. Zhou, X.-F. Han, Y. Xu, R.-F. Chen, Z.-B. Zhang, Embryo evaluation based on resNet with adaptive GA-optimized hyperparameters, Journal of Internet Technology 23(3)(2022) 527-538.

[29] V.-K. Kaliappan, R. Thangaraj, P. Pandivan, K. Mohanasundaram, S. Anandamurugan, D. Min, Real-time face mask position recognition system using YOLO models for preventing COVID-19 disease spread in public places, International Journal of Ad Hoc and Ubiquitous Computing 42(2)(2023)73-82.

[30] Q.-X. Yao, H. Li, P. Gao, H.-J. Guo, C. Zhong, Mapping irregular local climate zones from sentinel-2 images using deep learning with sequential virtual scenes, Remote Sensing 14(21)(2022) 5564-5564.

[31] P. Zan, H. Zhong, Y.-T. Zhao, H.-Y, Xu, R. Hong, Q. Ding, J.-W. Yue, Research on improved intestinal image classification for LARS based on ResNet, Review of scientific instruments 93(12)(2022) 124101-124101.

[32] C. Liu, J. Wang, T. Zhang, D. Ding. Adaptive threshold gesture segmentation algorithm based on skin color, in: Proc. of the 2016 2nd International Conference on Advances in Mechanical Engineering and Industrial Informatics (AMEII 2016), 2016.

[33] X. Zhong, Research on target tracking algorithm for complex scene based on multi-feature fusion, [dissertation] Zibo: Shandong University of Technology, 2021.

[34] X.-J. Hu, X.-P. Lu, S.-F. Peng, Y.-Y. Su, Research on a segmentation quality evaluation method of high score image considering area difference, World Scientific Research Journal 8(11)(2022) 102-109.

[35] Y.-L. He, L. Sun, Z.-D. Sun, X.-Q. Hu, MODIS aerosol optical thickness product algorithm verification and analysis, Aerosol and Air Quality Research 21(11)(2021) 210019.

[36] Q.-Y. Ling, X.-F. Liu, Image recognition of esophageal cancer based on ResNet and transfer learning, International Core Journal of Engineering 8(5)(2022) 863-869.