Advances in
Bridge Engineering

ORIGINAL INNOVATION

Open Access

Weakly-supervised structural surface crack detection algorithm based on class activation map and superpixel segmentation

Chao Liu[1] and Boqiang Xu[1*]

*Correspondence:
2110024@tongji.edu.cn

[1] Department of Bridge Engineering, College of Civil Engineering, Tongji University, Shanghai 200092, People's Republic of China

Abstract

This paper proposes a weakly-supervised structural surface crack detection algorithm that can detect the crack area in an image with low data labeling cost. The algorithm consists of a convolutional neural networks Vgg16-Crack for classification, an improved and optimized class activation map (CAM) algorithm for accurately reflecting the position and distribution of cracks in the image, and a method that combines superpixel segmentation algorithm simple linear iterative clustering (SLIC) with CAM for more accurate semantic segmentation of cracks. In addition, this paper uses Bayesian optimization algorithm to obtain the optimal parameter combination that maximizes the performance of the model. The test results show that the algorithm only requires image-level labeling, which can effectively reduce the labor and material consumption brought by pixel-level labeling while ensuring accuracy.

Keywords: Crack detection, Transfer learning, Convolutional neural networks, Class activation map, Superpixel

1 Introduction

The identification and detection of structural surface damages, especially cracks, can provide reliable data support for the operation and maintenance of the structure (Zhao et al. 2022, 2021). The traditional crack detection adopts manual detection, and the detection results are often subjective. Furthermore, the traditional detection method is often lack of universal standard, which leads to low accuracy. With the development of computer vision technology (Zakaria et al. 2022), the crack detection algorithm based on computer vision has the advantages of automation, high efficiency and no contact (Xu & Liu 2022a) to better solve the problems existing in the manual detection method (Spencer et al. 2019). Especially with the rapid development of deep learning technology in recent years, Convolutional Neural Network (CNN) model greatly improves the accuracy and efficiency of detection (Liu & Xu 2022; Xu & Liu 2022b). At present, the detection algorithm based on CNN model has been used to detect the surface damages of buildings (Guo et al. 2020), bridges (Deng et al. 2020) and tunnels (Stent et al. 2016).

The functions of CNN models are mainly distributed into three types: image classification (Krizhevsky et al. 2017), object detection (Girshick et al. 2014) and semantic segmentation (Long et al. 2015). The image classification model can judge the category of the input image, the object detection model can roughly judge the position of objects in the image, and the semantic segmentation model can detect the objects in the image pixel by pixel. In terms of accuracy, the semantic segmentation model is the most accurate, but it needs pixel-level labeled data as the training set, which requires a lot of manual work and has become an important factor restricting the application of CNN models in structural crack detection.

To address the difficulty in obtaining high-quality labeled datasets, weakly-supervised semantic segmentation algorithm (M. Zhang et al. 2020) has been proposed. Weakly-supervised semantic segmentation is a popular research direction in the field of computer vision, aimed at automatically identifying and segmenting different semantic regions. Unlike traditional supervised learning methods, weakly-supervised semantic segmentation uses less labeled information, which may only be image-level labeling, or even just textual descriptions.

Class Activation Map (CAM) technique (Long et al. 2015) is an interpretability method commonly used in image classification tasks, which has been applied to weakly-supervised semantic segmentation in recent years. The CAM technique generates activation maps for each class by weighting the feature maps after global average pooling, visualizing the regions of interest that the network focuses on for each class. In weakly-supervised semantic segmentation, the CAM technique can be used to generate pixel-level class activation maps, guiding the network to learn the semantic segmentation task (Huang et al. 2018; Kolesnikov & Lampert 2016). The application of CAM technique in weakly-supervised semantic segmentation can improve the segmentation performance of the network, while providing interpretable results that help understand the decision-making process of the network in the semantic segmentation task. However, applying CAM to existing weakly-supervised semantic segmentation models requires generating small seed regions that do not exceed the boundaries of the objects to be segmented. Considering the nature of cracks, which are typically thin and elongated, this can be challenging. Therefore, conventional weakly-supervised semantic segmentation algorithms may not perform well in crack detection tasks.

Superpixel segmentation algorithm is a type of image segmentation method that divides an image into several regions, each of which is called a superpixel. Superpixel refers to a group of pixels that are semantically close to each other in the image and are merged into one category, thus dividing the image into small blocks with certain semantic information.

Compared with traditional pixel-level segmentation methods, superpixel segmentation algorithm has the following advantages. First, superpixel segmentation algorithm divides the image into several superpixels, which reduces the number of pixels, thereby reducing the computational complexity and improving segmentation speed. Second, superpixel segmentation algorithm merges similar pixels in the image into one category, thereby reducing noise and unnecessary details and improving segmentation accuracy. Finally, superpixel segmentation algorithm does not require any prior information or annotated data, only preprocessing of the image is needed, thereby reducing the difficulty and implementation cost of the algorithm.

Therefore, this paper proposes a weakly-supervised crack identification algorithm that combines CAM and superpixel methods. The algorithm consists of three main steps. Firstly, a binary classification model based on the structure of the Vgg16 model is constructed to determine whether there are cracks in the image. The model is trained using transfer learning. Secondly, an improved method of Grad-CAM + + is proposed, which accurately reflects the rough location and distribution of cracks in the image. Finally, the SLIC algorithm is used to obtain the superpixel segmentation results of the crack image, and CAM is used to determine the category of the superpixel to obtain the semantic segmentation results of the cracks in the image. Additionally, Bayesian optimization is employed to optimize the two most critical parameters of the algorithm to obtain the optimal model. The experimental results show that the proposed method achieves high-precision weakly-supervised semantic segmentation of pixel-level recognition results using only image-level labeled datasets. Moreover, the use of Bayesian optimization significantly reduces the optimization cost of the algorithm, thereby improving its efficiency.

The main original contributions of this paper are as follows:

1. A new approach is proposed that combines Convolutional Neural Networks (Vgg16-Crack), an optimized Class Activation Map (CAM) algorithm, and Superpixel Segmentation (SLIC). This blend of techniques allows for accurate semantic segmentation of cracks, overcoming some limitations of other methods.
2. High-precision semantic segmentation is achieved by our method using only image-level labels, which significantly reduces the labor and material costs associated with pixel-level labeling. This makes our approach more feasible for real-world applications.
3. The use of a Bayesian optimization algorithm is introduced to obtain optimal parameter combinations, enhancing the performance of the model and reducing optimization costs.

Therefore, this paper proposes a detection method based on CAM, which can detect the distribution and position of cracks based on the image classification model, taking into account both the data labeling cost and detection accuracy, and has great practical value. The overall idea of the paper is shown in Fig. 1.

1.1 Training and testing of CNN model

1.1.1 Establishment of data set

The data set used in this paper is a public dataset, which containing 56000 concrete images (Maguire et al. 2018). The data set can be distributed into two categories: crack and no crack. In this paper, 4000 of them are used as training sets and 1000 as test sets.

1.2 Structure of CNN model

Convolutional neural networks (CNNs) have become a popular method for image classification tasks due to their ability to learn hierarchical representations of visual

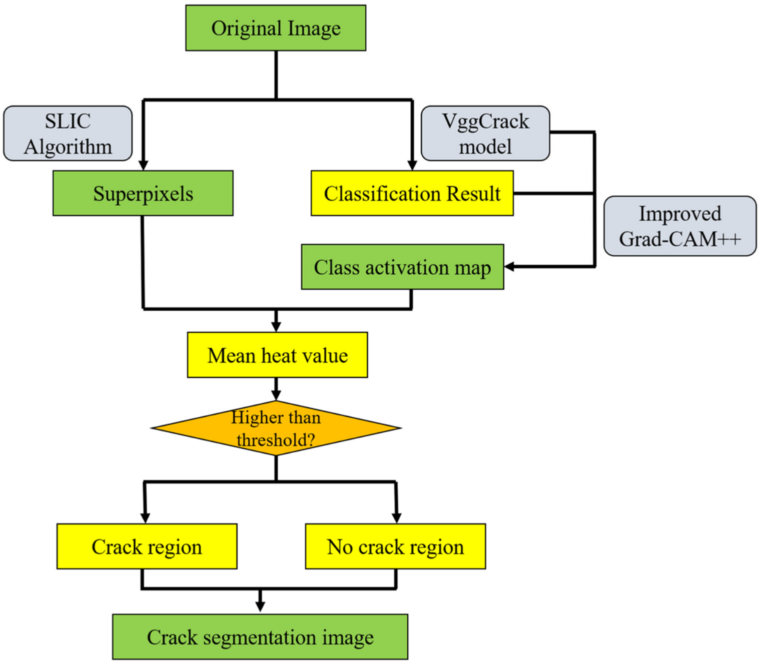


Fig. 1 Overall thinking of the paper

features. One prominent example of a CNN architecture is the Vgg model (Simonyan & Zisserman 2014).

The Vgg model is composed of a series of convolutional layers with small 3 × 3 filters, followed by max-pooling layers to reduce the spatial resolution of the feature maps. The use of small filters allows the model to learn more complex features, while the max-pooling layers help to reduce the number of parameters and control overfitting.

Vgg16 is a CNN model that belongs to the VggNet, it consists of 13 convolutional layers and 3 fully connected layers, with a filter size of 3 × 3 for the convolutional layers and a window size of 2 × 2 for the pooling layers. Vgg16 conducts a training process that utilizes extensive data augmentation and dropout techniques to avoid overfitting. The model takes input images of size 224 × 224 and outputs a probability distribution over 1000 possible categories.

The success of Vgg16 demonstrates that constructing CNN models with deeper and smaller convolutional filters can lead to better performance and has become a standard approach in image classification tasks. Furthermore, Vgg16 has also inspired deep learning research in other fields such as object detection, segmentation, and generative adversarial networks.

Therefore, the classification model in this paper is based on Vgg16 model. The number of output categories of the original Vgg16 model is 1000, while the crack detection model in this paper only needs to judge whether there is crack in the image. Therefore, the last two layers of the original Vgg16 model are changed to two outputs, as shown in Fig. 2. The above model is defined as Vgg16-Crack model.

The Vgg16-Crack model uses the categorical crosstropy loss as the loss function, as shown in Eq. (1):

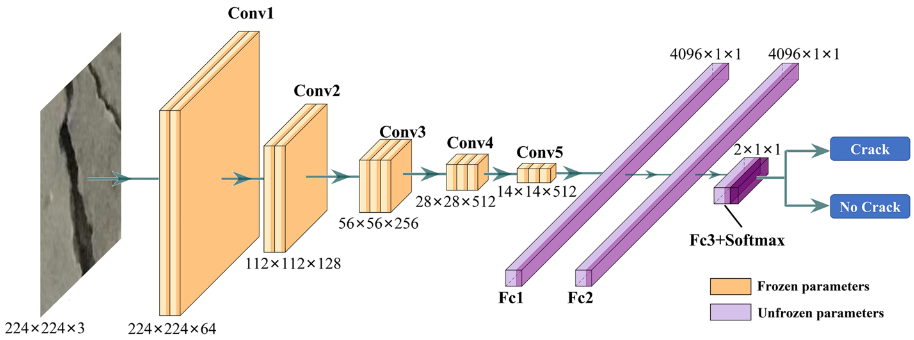


Fig. 2 Structure of Vgg16-Crack model

Table 1 Setting of hyperparameters

Hyperparameter	Value
Epoch	4
MBS	32
LR	0.0001

$$Loss = \frac{1}{N}\sum_{i} - \left[y_i \cdot \log(p_i) + \left(1 - y_i\right) \cdot \log(1 - p_i)\right] \tag{1}$$

>where y_i refers to each label of the sample, 1 represents the positive class (crack), 0 represents the negative class (no crack), and p_i is the probability that the input image is a positive class.

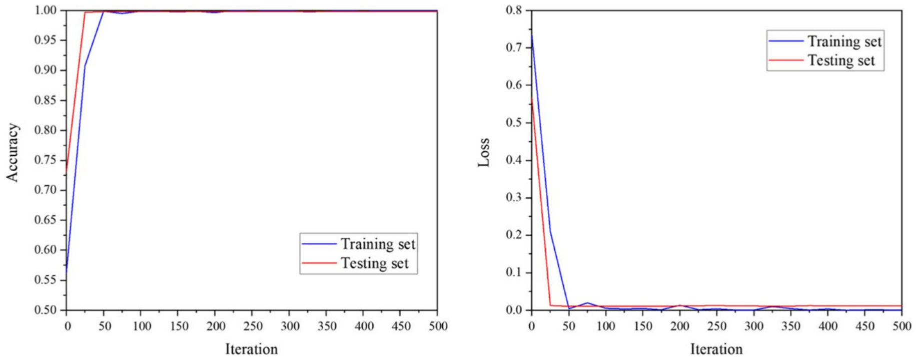
1.3 Model training

In order to improve the training efficiency, this paper adopts the methods of transfer learning (Weiss et al. 2016). Because the training set of the original Vgg16 is the ILSVRC-2012 data set containing more than 1 million images, the convolutional layers of original Vgg16 have strong feature extraction ability. Before training, take the parameters of the original Vgg16 model as the initialization value, and freeze the parameters of the convolution layer. Only the parameters of the full connection layer are trained during training.

Hyperparameters are the parameters set before training. The hyperparameters to be set in this paper include: epoch of training: Epoch, size of each batch: minibatchsize (MBS), and learning rate: LR. Because this paper adopts the transfer learning method for training, and the parameters of Vgg16-Crack only need to be fine-tuned, the smaller epoch and LR are adopted. Considering the limitation of video memory of the GPU, the value of MBS is relatively moderate. The setting of hyperparameters is shown in Table 1.

The training process of Vgg16-Crack is based on Tensorflow framework in Python 3.6., and the model is completed on a computer equipped with Intel i7-10700 k CPU and NVIDIA GEFORCE RTX3070ti.

There are 4000 images in the training set, and MBS = 32, epoch = 4, so the iteration number of training is 4000 / 32 × 4 = 500. In order to validate the generalization



(a) Accuracy curve

(b) Loss curve

Fig. 3 Model training

Table 2 Various situations of pavement crack identification

Identification Ground truth	Positive (Crack)	Negative (No Crack)
Positive (Crack)	TP	FN
Negative (No Crack)	FP	TN

ability of CNN model, during the training process, the images in the test set are input into the training Vgg16-Crack model and calculate the accuracy and loss, as shown in Fig. 3.

It can be seen from Fig. 3 that the accuracy of the model reaches to nearly 100% in both the training set and the test set. At the same time, the losses of the two data sets decrease rapidly with the progress of training and finally tend to be stable. It is worth noting that the loss of the test set does not rise with the training process, which indicates that there is no over fitting in the model.

1.4 Model testing

From the above analysis, it can be seen that the Vgg16-Crack model has good performance in the training set and test set and can accurately distinguish whether there is crack in the concrete image, and the model has good generalization ability. In order to further test the performance of the model in the test set, this paper will calculate the confusion matrix and its related test indicators.

Confusion matrix is a visualization tool in machine learning and deep learning, which can be used to compare classification results with real information of examples. Define an image with crack as positive, and one without crack as negative. TP represents the number of images with cracks accurately recognized, while FN represents those inaccurately recognized. FP represents the number of images without cracks accurately recognized, while TN represents those inaccurately recognized. Table 2 shows various situations of concrete crack identification.

The results will be tested by Precision, Recall, F1 and Accuracy. Definitions of these four indicators are as Eq. (2) ~ Eq. (5):

$$Precision = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{2}$$

$$Recall = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{3}$$

$$F1 = \frac{2}{1/Precision + 1/Recall} \tag{4}$$

$$Accuracy = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{FP} + \mathrm{TN} + \mathrm{FN}} \tag{5}$$

where the meanings of TP, FP, TN, and FN are the same as those in Table 2.

1000 images in the test set are input into the trained Vgg16-Crack model and the confusion matrix is calculated, as shown in Table 3.

According to Table 3, $Precision = 0.996$, $Recall = 1.000$, $F1 = 0.998$ and $Accuracy = 0.998$. The four evaluation indexes are all over 0.99, so Vgg16-Crack model has good classification performance.

2 Crack detection algorithm based on CAM

The image classification model can judge the category of the image (crack or no crack), but it can not give the distribution of cracks in the image. In fact, semantic segmentation model can provide pixel-level detection, but the semantic segmentation model has very high requirements for data labeling, and needs pixel-level labeled data as the training set. This paper presents a method based on CAM, which uses the classification model Vgg16-Crack to detect the crack distribution in the image.

2.1 Class activation map

Through multi-layer convolution operation, CNN model gradually extracts the information in the image, and finally generates feature maps. Then, the fully connected layers further extract the information in the feature map and output the probability that the image belongs to each category. Therefore, the feature maps output by the convolution layer reflect the features extracted by the model from the input image.

Based on this idea, Selvaraju et al. (Selvaraju et al. 2017) proposed Grad-CAM, and Chatopadhyay et al. (Chattopadhay et al. 2018) proposed Grad-CAM + +. This kind of algorithm is to calculate the gradient of the output of CNN model to the feature map first, then take it as the weight, finally calculate the weighted sum of all the feature maps, and convert it into a heat map.

Table 3 Confusion matrix

Identification **Ground truth**	**Positive**	**Negative**
Positive	500	0
Negative	2	498

The area with higher value in the heat map is the position that contributes more to the CNN model. If it has high degree of overlap with the expected detected target area (the area of cracks), it indicates that the CNN model is more accurate in image detection (Yang et al. 2020). The algorithm process is shown in Fig. 4.

2.2 Grad-CAM++

The core of Grad-CAM, Grad-CAM++and its derived CAM-based algorithm is the calculation of weight. The weight is determined according to the gradient of the output to the feature map, but different algorithms have different weight calculation methods. This paper is based on the advanced Grad-CAM++to detect cracks.

According to algorithm of Grad-CAM++, the weight corresponding to each feature map is shown in Eq. (6):

$$w_k^c = \sum_i \sum_j \alpha_{ij}^{kc} \cdot Relu\left(\frac{\partial Y^c}{\partial A_{ij}^k}\right) \tag{6}$$

Where

$$\alpha_{ij}^{kc} = \frac{\left(\frac{\partial S^c}{\partial A_{ij}^k}\right)^2}{2\left(\frac{\partial S^c}{\partial A_{ij}^k}\right)^2 + \sum_m \sum_n A_{mn}^k \left(\frac{\partial S^c}{\partial A_{ij}^k}\right)^3} \tag{7}$$

and A_{ij}^k is defined as the (i, j)th pixel in the k-th feature map of the output of the last convolutional layer, S^c is the score of the penultimate layer of category c, and the score obtained by the output layer of this category $Y^c = \exp(S^c)$.

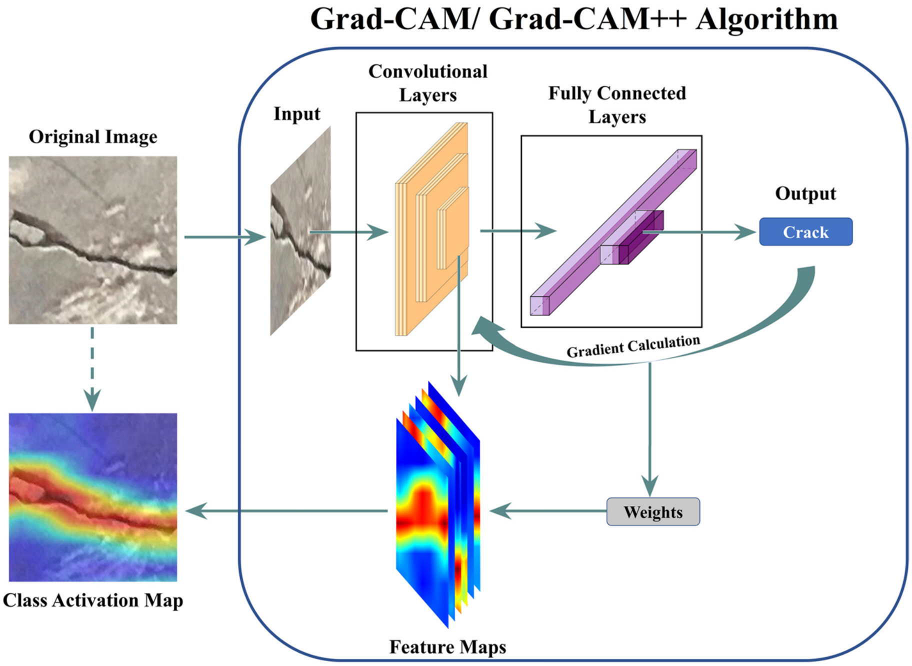


Fig. 4 Calculation process of CAM

The value of each pixel of the finally obtained CAM image is shown in Eq. (8):

$$I_{ij}^{c} = \sum\nolimits_{k} w_{k}^{c} \cdot A_{ij}^{c} \tag{8}$$

where w_k^c is defined by Eq. (6), and A_{ij}^c is defined in the same manner as described earlier.

After visualization, I_{ij}^c in Eq. (8) is CAM.

2.3 Improvement and optimization of Grad-CAM + +

In the actual test process, the CAM obtained by the Eq. (6) ~ Eq. (8) can not fully reflect the position and distribution of cracks in some cases. Therefore, two data augmentation methods (called DA1 and DA2) are used in this paper. The data augmentation method can improve the contrast of heat map and reflect the crack area more accurately.

DA1 linearly transforms each pixel in I_{ij}^c so that the maximum value is 1 and the minimum value is 0, that is the formula shown in Eq. (9):

$$I\prime_{ij}^{c} = \frac{I_{ij}^{c} - I_{min}}{I_{max} - I_{min}} \tag{9}$$

where I_{min} represents the minimum value in the image I, and I_{max} represents the maximum value in the image I.

DA2 squares the value of each pixel in the normalized I and normalizes it again, that is the formula shown in Eq. (10) and Eq. (11):

$$I\prime\prime_{ij}^{c} = \left(I\prime_{ij}^{c}\right)^{2} \tag{10}$$

$$I\prime\prime\prime_{ij}^{c} = \frac{I\prime\prime_{ij}^{c} - I\prime\prime_{min}}{I\prime\prime_{max} - I\prime\prime_{min}} \tag{11}$$

where I_{min} represents the minimum value in the image I, and I_{max} represents the maximum value in the image I.

The above data augmentation algorithm is also mathematically interpretable. In this paper, the basis for determining whether a pixel belongs to the crack area is whether the heat value is greater than a certain threshold. The use of the DA1 method helps ensure that each element's value falls within the range of 0 to 1. This normalization of the range allows the same threshold to be applicable to crack images under different conditions.

After the CAM has been processed by the DA1 method, the application of the DA2 method, which squares the heat values, alters the distribution of heat values within the CAM. This modification causes the heat values to concentrate around smaller values.

Under the same threshold, the area with heat values greater than the threshold is reduced after using data augmentation, as can be seen in Fig. 5.

In addition, the calculation method of Grad-CAM + + is improved. In the original Grad-CAM + + algorithm, the weight of the feature map is calculated by Eq. (1). In this paper, the weight calculation method in for Eq. (6) is optimized as Eq. (12):

$$w_{k}^{c} = \sum\nolimits_{i}\sum\nolimits_{j} \alpha_{ij}^{kc} \cdot Relu\left(A_{ij}^{k}\right) \tag{12}$$

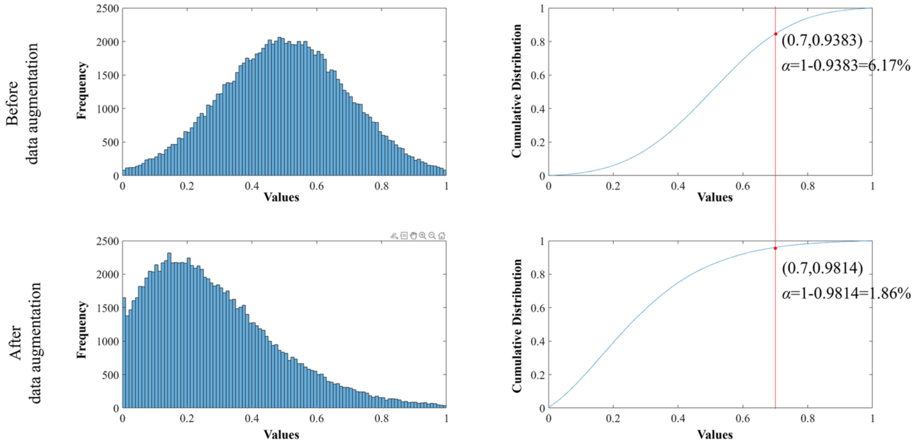


Fig. 5 Data distribution before and after data augmentation

where α_{ij}^{kc} is defined by Eq. (7), A_{ij}^{k} is defined as the (i, j)th pixel in the k-th feature map of the output of the last convolutional layer.

Compared with large data sets, crack data set in this paper have few category, and less information can be extracted from the image. Therefore, in Vgg16-Crack, a CNN model trained based on Vgg16, only a few feature maps are effective, and the values in most feature maps are all 0 or close to 0. It can be concluded that if the value in the feature map is large, this feature map has high weight. Therefore, this paper uses the method in Eq. (12) to calculate the weight.

A good performance was achieved by Eq. (12) in the dataset used in this paper. Furthermore, it is believed that the proposed method can be applied to most structural cracks, even those found in complex environments. For the original VGG network, millions of image datasets provided by computer vision researchers around the world were used, encompassing high complexity and 1000 categories. In contrast, for the structural surface crack detection algorithm concerned in this paper, the information contained in the images is relatively simple. The focus is only on the disease of the image, and the texture and color information of the disease are relatively singular.

Therefore, the feature extraction capability of the detection model based on the VGG16 deep learning model adopted in this paper is far greater than the images in the dataset. Even if the images may be in a complex environment, the information of the structural surface cracks can be effectively and accurately extracted by the convolutional layer of the VGG model.

In addition, after using Eq. (12) to obtain CAM, this paper also uses DA1 and DA2 to augment the data, so as to more accurately reflect the crack area.

2.4 Model validation

The algorithms mentioned above are tested with the images in the test set. Figure 6 shows partial results. As can be seen from Fig. 6, the original Grad-CAM+ +algorithm can roughly reflect the crack area in the image, but the accuracy is low. Furthermore, there are also some error detection, some areas with high heat value do not coincide with the crack area. Through data augmentation, the detection accuracy based on CAM

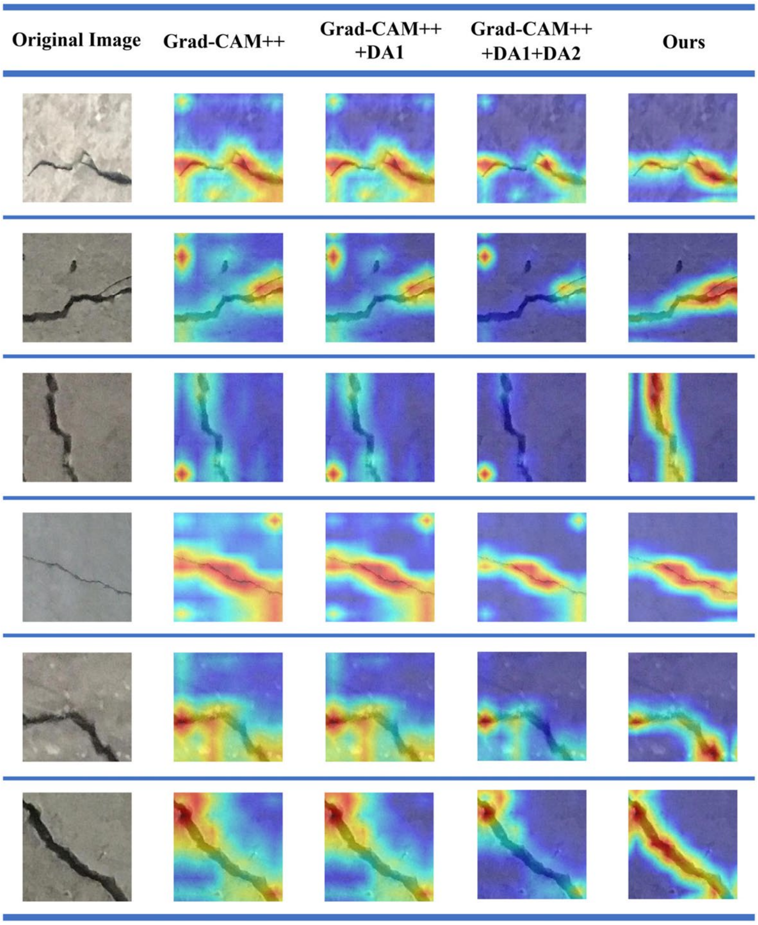


Fig. 6 Model testing

has been improved to a certain extent. But in some cases, the data augmentation algorithms locate the crack area incorrectly, and the area with high heat value is even far away from the crack.

The algorithm proposed in this paper can better solve the above problems. After using the optimized weight calculation formula and augmenting the data by DA1 and DA2, CAM can basically correctly reflect the position and distribution of the crack.

In order to further test the effect of the proposed algorithm in practical application, this paper selects a photograph of concrete surface. Firstly, this image is cropped into several images with a resolution of 224 × 224, and then Vgg16-Crack is used to classify each image, and generate the CAM of each "crack" category image. Finally, these CAMs are combined to obtain the detection results of the original picture, as shown in Fig. 7.

As can be seen from Fig. 7, the method proposed in this paper can ensure the effectiveness in practical application, and has great development potential and broad application prospects.

It is worth noting that even with the proposed improved method, the regions with high CAM heatmaps still exhibit relatively large widths. Therefore, for smaller cracks, preprocessing steps are required. For a crack detection task based on computer vision, the judgement of whether a crack is fine or not is not based on the actual width of the crack,

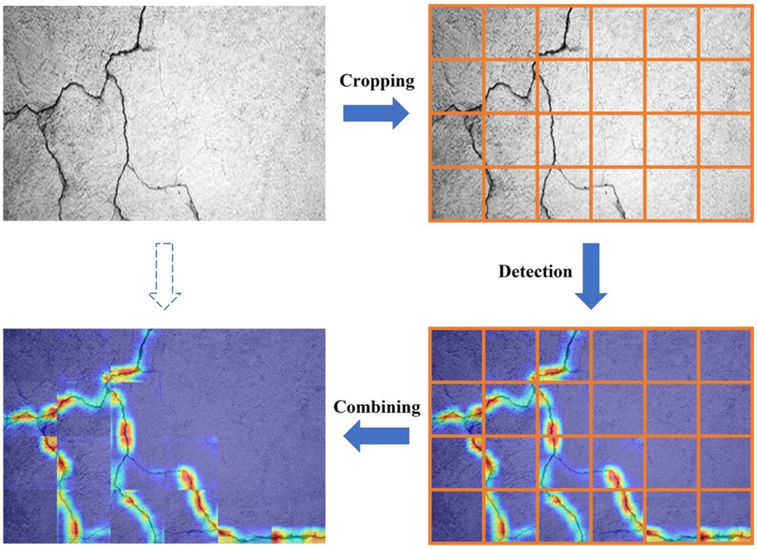


Fig. 7 Test results

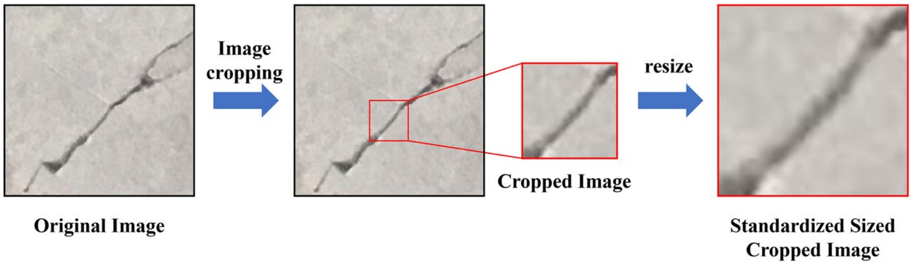


Fig. 8 Image cropping and resizing

but rather on the ratio of the pixel width of the crack in the image coordinate system to the image resolution. Therefore, it is possible to achieve this by cropping the image and adjusting its size, as shown in Fig. 8. If the issue of fine cracks is noted during imaging, the distance between the structural surface and the camera lens can be adjusted, or the camera's focal length can be changed during shooting to prevent the relative width of the crack in the image from being too small. In general, for fine cracks, simple adjustments to the image or camera can transform them into cracks of regular width, making them suitable for detection using our proposed method.

3 Superpixel-based crack segmentation

3.1 Superpixel segmentation

Superpixel segmentation algorithms (Ibrahim & El-kenawy 2020) have become a popular method for image processing tasks in computer vision due to their ability to group pixels that belong to the same object or region in an image. These algorithms can be used in a variety of applications, including object detection, image segmentation, and image recognition.

One prominent example of a superpixel segmentation algorithm is the SLIC (Simple Linear Iterative Clustering) algorithm (Achanta et al. 2012). The SLIC algorithm involves clustering pixels in a way that minimizes both color distance and spatial distance between pixels. This allows the algorithm to group pixels that are similar in color and spatial proximity into superpixels, which can then be used to segment the image into meaningful regions.

Other superpixel segmentation algorithms include the Quick Shift algorithm (Vedaldi & Soatto 2008), which uses a density-based approach to group pixels that have similar color and texture characteristics, and the Felzenszwalb and Huttenlocher algorithm (Felzenszwalb & Huttenlocher 2004), which uses a graph-based approach to group pixels that have similar color and intensity characteristics.

Superpixel segmentation algorithms have been shown to improve the efficiency and accuracy of image processing tasks in computer vision, and have become an important tool for researchers and practitioners in the field. Ongoing research aims to further improve the performance and applicability of these algorithms, as well as to explore new applications for superpixel segmentation in areas such as video processing (Giordano et al. 2015) and 3D reconstruction (Penza et al. 2016).

3.2 SLIC Algorithm

Simple Linear Iterative Clustering (SLIC) (Achanta et al. 2012) is a popular image segmentation algorithm, it is a fast and efficient algorithm that enables accurate and precise segmentation of images.

The SLIC algorithm is based on the k-means clustering method, which is a common unsupervised machine learning technique. The algorithm works by grouping pixels into clusters based on their color and spatial proximity. The SLIC algorithm is particularly useful for segmenting images with smooth and uniform regions, such as satellite images and medical images.

The SLIC algorithm can be summarized in the following steps:

1. The input image is first converted to a LAB color space, which separates the color and brightness components of the image.
2. The image is then divided into a grid of equally sized superpixels. The size of the superpixels is determined by a user-defined parameter, which controls the level of granularity in the segmentation.
3. The initial cluster centers are then placed at the center of each superpixel. These are typically initialized as the mean color and position of the pixels within each superpixel.
4. The algorithm then iteratively assigns each pixel to the nearest cluster center based on both color and spatial distance. This process is repeated until convergence is achieved.
5. Once convergence is achieved, the cluster centers are updated to the mean color and position of the pixels within each cluster.
6. Finally, the pixels in each cluster are assigned the label of the cluster center, resulting in a segmented image.

Overall, the SLIC algorithm is a powerful tool for image segmentation that is widely used in various applications, including computer vision, medical imaging, and remote sensing. It is a fast and efficient algorithm that produces accurate and precise segmentation results. Therefore, this paper uses SLIC algorithm to obtain the pre-processing segment superpixels of the original structural crack images.

The SLIC algorithm has two important parameters, namely region_size and ruler, which are used to control the size and smoothness of superpixels, respectively.

The region_size is a positive integer that specifies the number of pixels included in each superpixel. A smaller region_size generates smaller superpixels, while a larger region_size generates larger superpixels. The value of this parameter should be associated with the size of the input image and the expected size of the superpixels. Typically, it is recommended to set it to an integer multiple of the square root of the image size to obtain relatively uniform superpixels.

The ruler is a parameter that controls the similarity between pixels in the color space. It specifies the weighting factor used when calculating the distance between pixels. A smaller value of this parameter results in smaller differences between pixels, resulting in higher smoothness of the generated superpixels, but it may also lead to over-smoothing. Conversely, a larger value of ruler produces finer boundaries between superpixels, but also introduces more noise.

In general, these two parameters can be used to adjust the size and smoothness of superpixels to generate the desired results. However, their optimal values depend on the specific application and characteristics of the input image, and need to be adjusted according to the actual situation.

Figure 9 shows the superpixel segmentation results of crack images under different parameter combinations. It can be seen that when the superpixel size is small, the superpixels obtained by the SLIC algorithm can more accurately separate cracks from

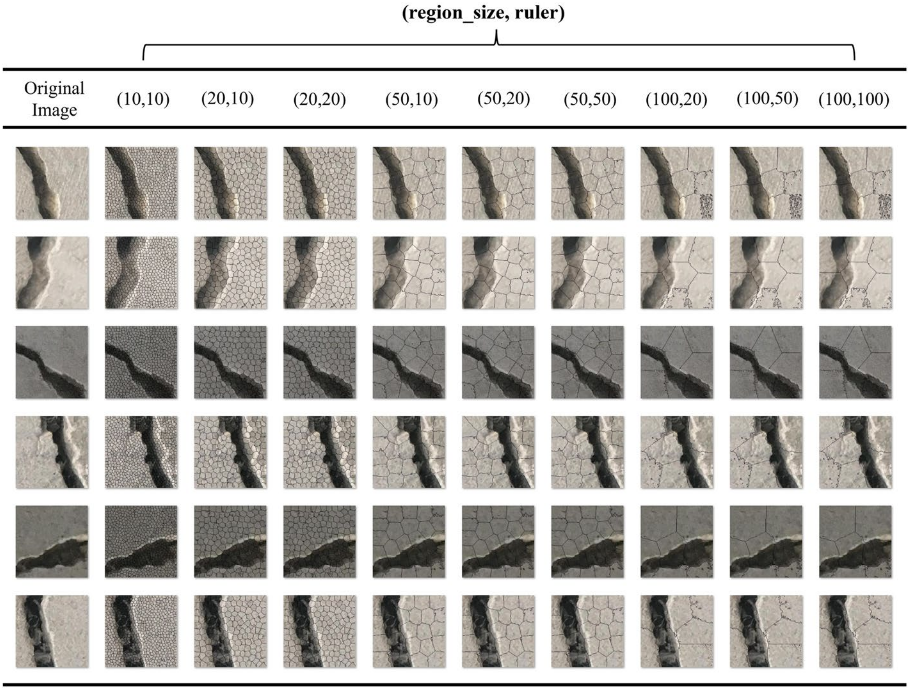


Fig. 9 Superpixel segmentation results under different parameter combinations

the background. However, considering that too small superpixels will greatly increase the complexity of the algorithm and reduce the efficiency of the whole method, it is necessary to choose parameters that generate moderately sized superpixels for subsequent processing.

Figure 10 shows the superpixel segmentation results of the same crack image when the region_size and ruler parameters are set to 5, 10, 20, 50, and 100. It can be seen from Fig. 10 that region_size has a significant impact on the final result, and the size of the superpixel increases with the increase of region_size. On the other hand, ruler has little effect on superpixel segmentation. When it changes and region_size remains the same, the obtained superpixels are almost unchanged. This is because for structural crack images, except for the crack edge area, other areas of the image are relatively smooth, so the smoothness parameter has little effect on the segmentation results.

Therefore, when considering subsequent parameter optimization, this paper only optimizes the parameter region_size, which has a more significant impact on the results.

4 Crack identification method combining CAM and SLIC Algorithm

According to Sect. 3, it is known that CAM provides the position and distribution of cracks in the original image based on the image classification model Vgg16-Crack, but this localization is imprecise and cannot obtain the accurate border of cracks. On the other hand, according to Sect. 4.2, the superpixel segmentation algorithm can effectively preprocess the original image by grouping pixels with similar semantic information into the same superpixel, which has clear boundaries, but this unsupervised method cannot determine the category of each superpixel.

Therefore, this paper proposes a method that combines CAM and superpixel segmentation. A single image is input into VggCrack to obtain its CAM, and at the same time,

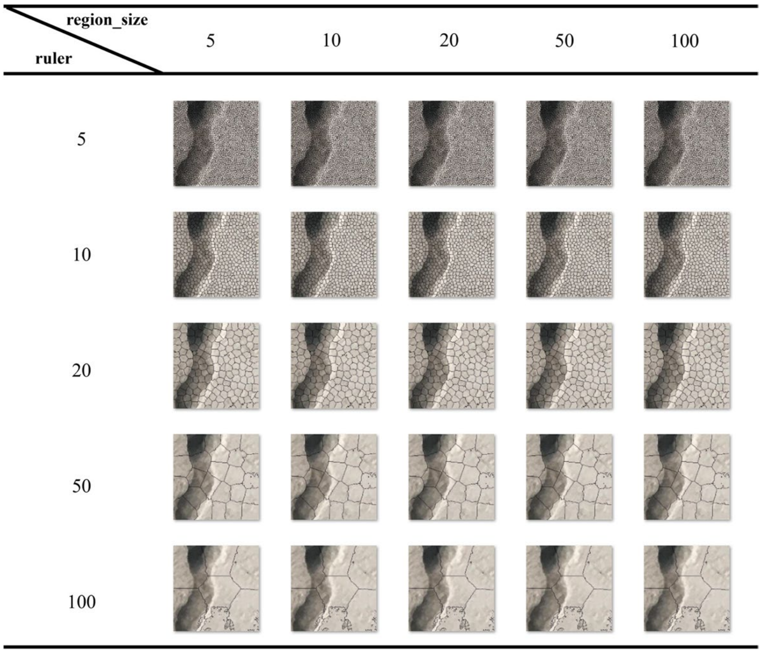


Fig. 10 Comparison of the effects of parameter changes on the egmentation results

the SLIC algorithm is used to process the image and obtain the superpixel points in the original image.

Assuming the original image is I, the pixel point at (i, j) is $I_{i,j}$, and the corresponding CAM of I is I^c. The SLIC algorithm segments k superpixels, and the set of pixels contained in the m-th superpixel is S_m.

In the actual recognition process, first, I^c and S_m are calculated. Then, for each m, the value of each pixel point in S_m of I^c is calculated, and the average value meanI_{sm} is obtained. If the average value is greater than the set threshold t, the superpixel can be considered to belong to the crack category. The algorithm flowchart is shown in Fig. 11.

5 Model parameter optimization based on Bayesian optimization

The algorithm obtained in Sect. 3.3 shows that for the task of crack recognition, the parameter "region_size" has a significant impact on the superpixel segmentation results, while the "ruler" parameter has almost no effect. On the other hand, when combining CAM to determine the category of superpixels, the choice of threshold has a significant impact on the final results. Therefore, for the model proposed in this paper, it is necessary to determine the optimal parameter combination $(\text{region\_size, threshold})_{\text{opt}}$ that can achieve the best model performance. Therefore, this paper manually annotated 100 images in the dataset pixel by pixel, and used them as the ground truth to evaluate the model performance under different parameter combinations.

It should be noted that under different parameter combinations, the superpixel segmentation in the dataset needs to be calculated multiple times, and as a machine learning model, the SLIC algorithm takes a certain amount of time to preprocess the image. Therefore, the method of determining the optimal model by traversing each point in the (region_size, threshold) parameter space will consume a lot of time, making the optimization difficulty significantly increased. In order to improve the efficiency of model optimization, this paper uses Bayesian optimization algorithm to obtain the optimal parameter combination.

Bayesian optimization (Frazier 2018) is a powerful algorithm that addresses the problem of optimizing complex, black-box functions. This algorithm is based on the principles of Bayesian inference, which allows for the efficient exploration of

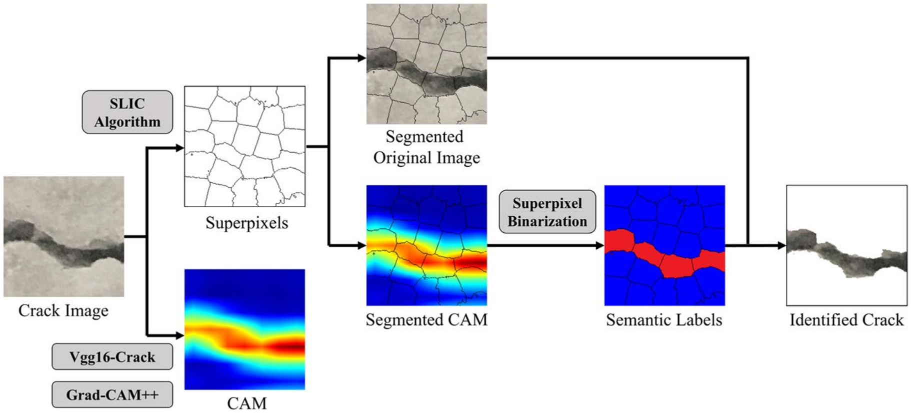


Fig. 11 Crack segmentation algorithm process

high-dimensional parameter spaces. One of the key advantages of Bayesian optimization is its ability to balance exploration and exploitation, which enables the algorithm to find optimal solutions quickly and efficiently.

In recent years, Bayesian optimization has gained wide-spread use across various fields, including machine learning (Snoek et al. 2012) and computer vision (Zhang et al. 2015). In particular, this algorithm has been applied to hyperparameter tuning of machine learning models and parameter optimization of computer vision algorithms. By using Bayesian optimization, researchers have been able to achieve state-of-the-art results in these fields.

The core steps of Bayesian optimization can be divided into four main stages: initialization, selection, evaluation, and updating. In the initialization stage, an initial set of candidate solutions is selected based on some prior knowledge or random sampling. In the selection stage, the algorithm uses the probabilistic model to select the most promising candidate solution to evaluate next. In the evaluation stage, the selected candidate solution is evaluated using the objective function. Finally, in the updating stage, the probabilistic model is updated to incorporate the new evaluation results, and the process repeats until the optimal solution is found.

One of the key advantages of Bayesian optimization is its ability to balance exploration and exploitation. By using the probabilistic model to guide the search, the algorithm can explore the solution space to find promising regions while also exploiting the most promising candidates. Additionally, the probabilistic model allows Bayesian optimization to handle noisy or incomplete evaluations, making it a robust algorithm for real-world optimization problems.

In this paper, we propose a novel algorithm that utilizes machine learning and deep learning algorithms from the computer vision field to identify the structural crack. Specifically, we aim to optimize parameters for both the SLIC algorithm from traditional machine learning and the CAM algorithm from deep learning. To achieve this, we employ Bayesian optimization to efficiently search the high-dimensional parameter space and find the optimal parameter combinations. First, the range of the two parameters are determined by experience: $10 \leq \text{region\_size} \leq 100$, $10 \leq \text{threshold} \leq 250$.

And as a semantic segmentation model, the objective function of Bayesian optimization is to select the most important evaluation metric in the semantic segmentation field, intersection over union (IoU).

During the optimization process, the Bayesian optimization algorithm is used for unsupervised semantic segmentation models with 40 different hyperparameter combinations. It should be noted that Bayesian optimization optimizes parameters in the real domain, while (region_size, threshold) are both positive integers, so their rounded-down values were treated as the actual values. The objective function values for each hyperparameter combination are shown in Table 4.

Figure 12 shows the relationship between parameters and the target in the parameter space calculated by Gaussian process regression. As can be seen from Table 4 and Fig. 12, when (region_size, threshold) = (52, 138), the target value reaches the maximum of 0.7003. Therefore, the model trained with this hyperparameter combination is used as the optimal model for crack semantic segmentation.

Table 4 Bayesian optimization process

Iteration	Target	Region_size	Region_size (rounded)	threshold	Threshold (rounded)
1	0.0835	38.48	38	220.4	220
2	0.001419	23.13	23	246.2	246
3	0.3245	49.04	49	189.4	189
4	0.297	19.14	19	36.2	36
5	0.4203	89.92	90	142.1	142
6	0.5637	34.15	34	125	125
7	0.559	35.46	35	122.1	122
8	0.183	100	100	10	10
9	0.5203	10	10	112.5	113
10	0.5593	10	10	152.8	153
11	0.6669	43.94	44	144.3	144
12	0.4376	83.52	84	86.88	87
13	0.6058	59.3	59	134	134
14	0	100	100	250	250
15	0.5919	36.05	36	157	157
16	0.5294	57.2	57	152.9	153
17	0.4178	10.28	10	78.96	79
18	0.4138	49.59	50	85.22	85
19	0.4725	99.96	100	114	114
20	0.666	46.21	46	136.3	136
21	0.6292	38.38	38	142.3	142
22	0.4499	10.31	10	177.9	178
23	0.3538	99.52	100	63.76	64
24	0.6778	49.4	49	141	141
25	0.6698	47.11	47	140.9	141
26	0.1317	100	100	184.7	185
27	0.1907	49.31	49	10	10
28	0.5842	62.1	62	110.1	110
29	0.5682	10	10	133.6	134
30	0.6527	51.22	51	123.4	123
31	0.2081	11.04	11	12.78	13
32	0.3062	62.81	63	52.58	53
33	0.6648	50.95	51	132.3	132
34	0.4945	78.68	79	116.1	116
35	0.6528	47.02	47	148.1	148
36	0.5732	49.48	49	110.3	110
37	**0.7003**	**51.84**	**52**	**138.4**	**138**
38	0.3674	31.2	31	63.33	63
39	0.4653	25.52	26	97.22	97
40	0.4361	99.73	100	92.86	93

Figure 13 illustrates the crack identification results obtained from several crack images processed using the method proposed in this paper and their comparison with the manually labeled ground truth. From Fig. 13, it can be seen that the method proposed in this article can accurately identify cracks in images, approaching the results of manual labeling.

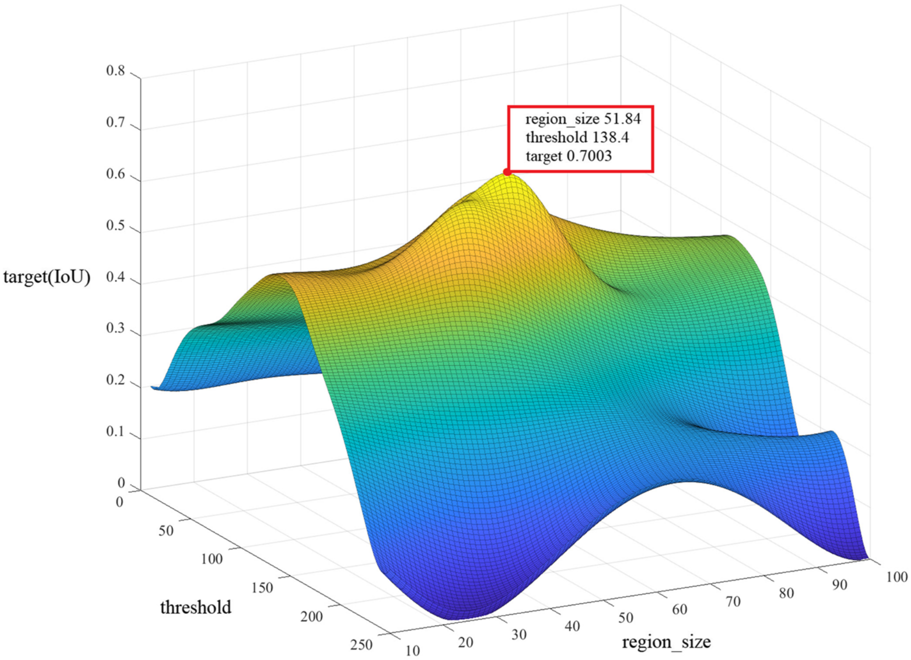


Fig. 12 Parameter-target surface based on Gaussian process regression

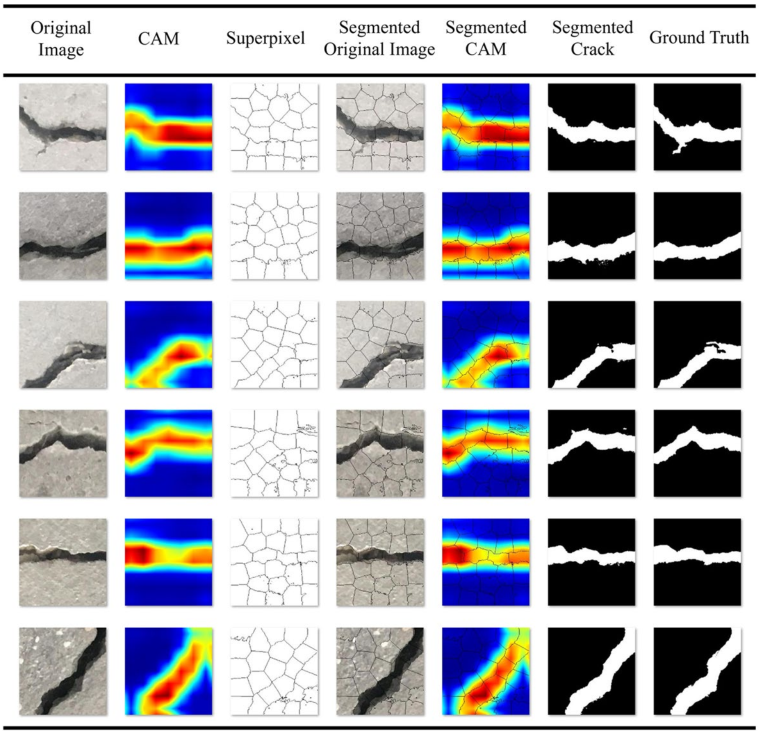


Fig. 13 Comparison between automated segmentation results and manual labeled results

6 Model test

To evaluate the performance of the proposed algorithm, this paper selects an additional 150 images. Of these 150 images, 100 came from the aforementioned public dataset, and the remaining 50 were sourced from on-site shots of bridge crack images. These images are then labeled at the pixel level, and the cracks in the images were identified using traditional threshold segmentation algorithm, strong supervised deep learning method, and the weakly-supervised algorithm proposed in this paper.

In the strong supervised deep learning method, this paper use the classic model in the semantic segmentation domain, U-Net. U-Net, widely applied in image segmentation tasks, adopts a U-shaped network structure which includes a contracting path known as the encoder and an expanding path referred to as the decoder. The encoder progressively reduces the size and channel number of the feature map through a series of convolution and pooling operations, extracting high-level semantic information from the image. The decoder, on the other hand, restores the feature map to its original size through upsampling and convolution operations and refines and reconstructs it in conjunction with the encoder's features. This encoder-decoder structure enables U-Net to simultaneously capture local details and global context information, thus achieving excellent performance in image segmentation tasks. Additionally, U-Net improves the accuracy of segmentation results by using skip connections to link the feature maps of the encoder and decoder, thereby aiding information transmission and gradient flow. Owing to its simple yet effective design, U-Net has become the preferred network architecture for many image segmentation tasks. Its structure is shown in Fig. 14.

This paper trains the U-Net model for crack detection using data from the training set. The model is trained for 10,000 iterations with a learning rate of 0.00001. The curve of the model's loss function during the training process is shown in Fig. 15. As can be observed, the model's loss function gradually decrease and remain constant, indicating that the model converges after 10,000 iterations of training.

Upon completion of the training, the aforementioned 150 images are crack-detected using three automated algorithms, with results shown in Fig. 16. Compared to the manually labeled results, the detection results obtained using traditional digital image processing methods are inferior, with discontinuous cracks and smaller cracks with slight

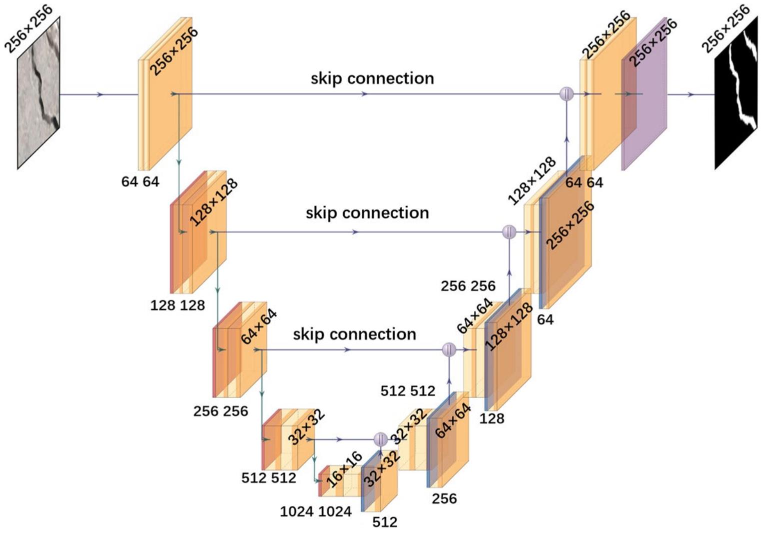


Fig. 14 Structure of U-Net

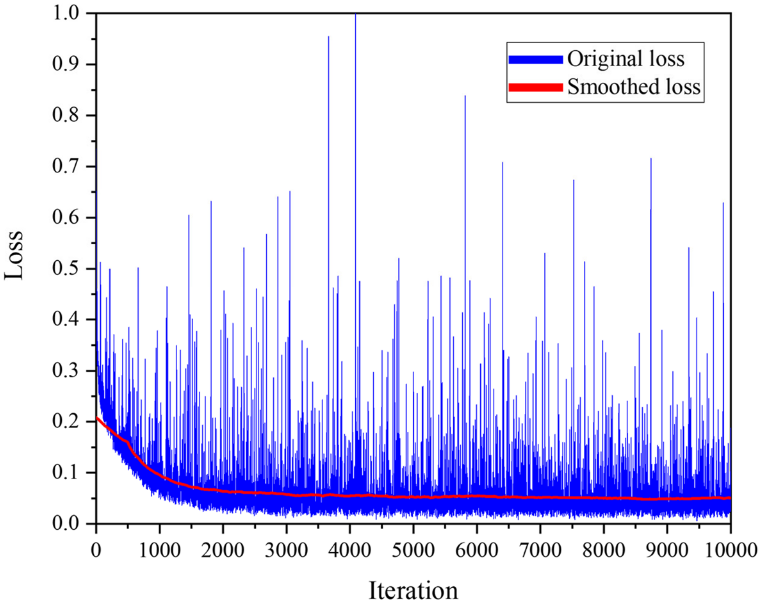


Fig. 15 Training process of U-Net

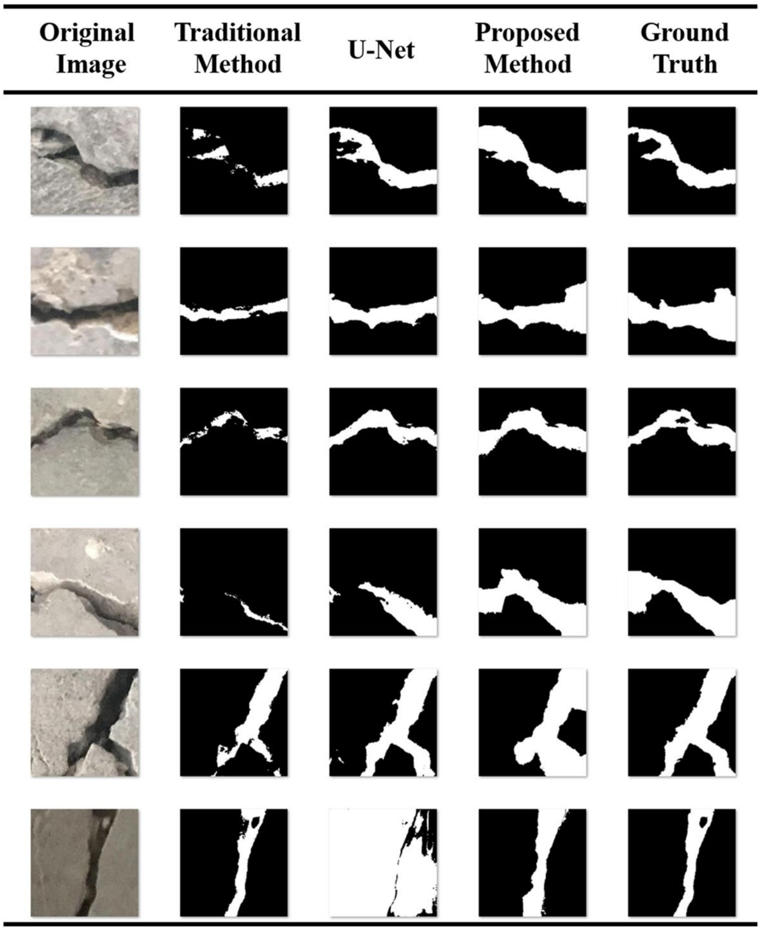


Fig. 16 Comparison of detection results between different methods and ground truth

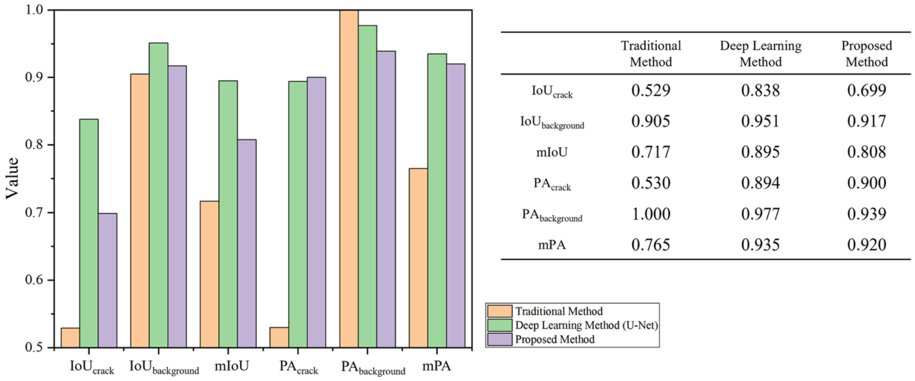
color differences from the background often undetectable. The detection model based on U-Net can effectively detect surface cracks in the structure, but when the background color deviates significantly from that of training set, severe errors also occur in deep learning methods. As shown in the last row of Fig. 16, where the background color of the crack image is darker, deep learning methods mistakenly recognize the background as a crack. The method proposed in this paper can effectively identify most cracks, but it also has a lack of precision in some aspects. For instance, if an image contains both coarse and fine cracks, the finer cracks are often overlooked by the algorithm.

To quantitatively analyze the performance of the three algorithms on the test set, this paper compares the results obtained by the three automated detection algorithms with the ground truth and calculates the sizes of the evaluation metrics for the three methods on the test set. In the field of computer vision semantic segmentation tasks, commonly used evaluation metrics include IoU (Intersection over Union), mIoU (mean Intersection over Union), PA (Pixel Accuracy), and mPA (mean Pixel Accuracy). These metrics can be used to evaluate the detection performance and accuracy of the semantic segmentation model on the dataset.

IoU is a metric used to measure the degree of overlap between prediction result and ground truth result, i.e., the area of intersection of the two results divided by the area of their union. mIoU is an indicator obtained by averaging the corresponding IoU for different categories and is used to evaluate the overall segmentation performance. Both IoU and mIoU have a value range between 0 and 1, with higher values indicating higher average accuracy of semantic segmentation. The segmentation in this paper has two classes: crack and background, so there are three IoU evaluation indicators: $\mathrm{IoU}_{\mathrm{crack}}$, $\mathrm{IoU}_{\mathrm{background}}$, and their mean value, mIoU.

PA is another commonly used evaluation metric in the field of semantic segmentation, used to measure the accuracy of pixel-level recognition. It compares the consistency between the category label of each pixel in the prediction results and the real label by calculating the ratio of the number of correctly classified pixels in each category to the total number of pixels in that category. mPA is an indicator obtained by averaging the corresponding PA for different categories and is used to evaluate the overall segmentation accuracy. The value range of PA is also between 0 and 1, with PA values closer to 1 indicating higher pixel classification accuracy by the semantic segmentation algorithm. Similar to IoU, there are also three PA evaluation indicators in this paper: $\mathrm{PA}_{\mathrm{crack}}$, $\mathrm{PA}_{\mathrm{background}}$, and mPA.

This paper compares three automated algorithms and evaluates their performance on the test set using the six evaluation metrics mentioned above. The results are shown in Fig. 17. From Fig. 17, it can be observed that among the three methods, the performance of the traditional digital image processing algorithm is poor, with IoU and PA metrics for cracks only around 0.5 and lower average metrics compared to the other two methods. For the supervised learning method, although the U-Net model slightly lags behind the proposed algorithm in some categories, it performs the best overall, indicating that supervised learning still achieves good accuracy in semantic segmentation. The performance of the proposed algorithm on the test set is slightly worse than the U-Net model but far superior to the traditional algorithm. This suggests that the proposed algorithm can achieve performance close to that of supervised learning models in the



	Traditional Method	Deep Learning Method	Proposed Method
IoU$_{crack}$	0.529	0.838	0.699
IoU$_{background}$	0.905	0.951	0.917
mIoU	0.717	0.895	0.808
PA$_{crack}$	0.530	0.894	0.900
PA$_{background}$	1.000	0.977	0.939
mPA	0.765	0.935	0.920

Fig. 17 Evaluation index comparison between different detection methods

task of semantic segmentation of cracks, while only requiring image-level annotations, which have much lower annotation costs compared to supervised learning. Therefore, the proposed method has great potential and broad application prospects in engineering applications.

7 Conclusion

In this paper, a weakly-supervised structural surface crack detection algorithm is proposed. Firstly, a CNN model Vgg16-Crack for classification is trained, and then the crack area in the image is detected based on Grad-CAM+ +algorithm. In this paper, the original Grad-CAM+ +algorithm is improved and optimized. Through data augmentation and weight calculation formula optimization, the generated CAM can accurately reflect the position and distribution of cracks in the image. Afterwards, the superpixel segmentation algorithm SLIC is used to preprocess the original crack image to generate multiple regions with similar semantic information. Then, this paper proposes a method that combines superpixel method with CAM to achieve more accurate semantic segmentation of cracks by binarizing the superpixels.

The algorithm proposed in this paper can use the classification model with low data labeling cost to detect the crack distribution in the image, which reduces the demand for human and material resources of the model. However, CAM can only give the approximate distribution of cracks, when the crack is relatively thin or short, the high heat value areas in CAM may not reflect its existence well, which may lead to incorrect discrimination of the superpixel where the thin or short crack is located, even if the superpixel can be preprocessed accurately. This can lead to a decrease in accuracy. In further work, image super-resolution and other techniques will be used to achieve accurate detection of multi-scale cracks under weak supervision, specifically targeting the above situation.

Acknowledgements
We would like to express our sincere gratitude to all those who have contributed to the completion of this research. We also thank the members of our research team for their hard work and dedication. Additionally, we would like to thank the participants for their participation in the study. This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Author's contributions
Chao Liu: Validation, Software, Writing—original draft, Data curation; Boqiang Xu: Supervision, Project administration, Conceptualization, Methodology.

Funding

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Availability of data and materials

The datasets used and/or analysed during the current study are available from the corresponding author on reasonable request.

Declarations

Competing interests

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Received: 30 June 2023 Accepted: 8 October 2023
Published online: 03 November 2023

References

Achanta R, Shaji A, Smith K, Lucchi A, Fua P, Süsstrunk S (2012) SLIC superpixels compared to state-of-the-art superpixel methods. IEEE Trans Pattern Anal Mach Intell 34(11):2274–2282

Chattopadhay A, Sarkar A, Howlader P, Balasubramanian VN (2018) Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. 2018 IEEE winter conference on applications of computer vision (WACV). p 839-847

Deng J, Lu Y, Lee VCS (2020) Concrete crack detection with handwriting script interferences using faster region-based convolutional neural network. Comp-Aided Civil Infrastr Eng 35(4):373–388

Felzenszwalb PF, Huttenlocher DP (2004) Efficient graph-based image segmentation. Int J Comput Vision 59:167–181

Frazier PI (2018) Bayesian optimization recent advances in optimization and modeling of contemporary problems. Informs, p 255-278

Giordano D, Murabito F, Palazzo S, Spampinato C (2015) Superpixel-based video object segmentation using perceptual organization and location prior. Proceedings of the IEEE conference on computer vision and pattern recognition, p 4814-4822

Girshick R, Donahue J, Darrell T, Malik J (2014) Rich feature hierarchies for accurate object detection and semantic segmentation. Proceedings of the IEEE conference on computer vision and pattern recognition, p 580-587

Guo J, Wang Q, Li Y, Liu P (2020) Façade defects classification from imbalanced dataset using meta learning-based convolutional neural network. Comp-Aided Civil Infrastr Eng 35(12):1403–1418

Huang Z, Wang X, Wang J, Liu W, Wang J (2018) Weakly-supervised semantic segmentation network with deep seeded region growing. Proceedings of the IEEE conference on computer vision and pattern recognition, p 7014-7023

Ibrahim A, El-kenawy E-SM (2020) Applications and datasets for superpixel techniques: a survey. J Comp Sci Inform Syst 15(3):1–6

Kolesnikov A, Lampert CH (2016) Seed, expand and constrain: Three principles for weakly-supervised image segmentation. Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14. p 695-711

Krizhevsky A, Sutskever I, Hinton GE (2017) Imagenet classification with deep convolutional neural networks. Commun ACM 60(6):84–90

Liu C, Xu B (2022) A night pavement crack detection method based on image-to-image translation. Comp-Aided Civil Infrastr Eng 37(13):1737–1753

Long J, Shelhamer E, Darrell T (2015). Fully convolutional networks for semantic segmentation. Proceedings of the IEEE conference on computer vision and pattern recognition. p 3431-3440

Maguire M, Dorafshan S, Thomas RJ, (2018). SDNET2018: A concrete crack image dataset for machine learning applications.

Penza V, Ortiz J, Mattos LS, Forgione A, De Momi E (2016) Dense soft tissue 3D reconstruction refined with super-pixel segmentation for robotic abdominal surgery. Int J Comput Assist Radiol Surg 11:197–206

Selvaraju RR, Cogswell M, Das A, Vedantam R, Parikh D, Batra D (2017) Grad-cam: Visual explanations from deep networks via gradient-based localization. Proceedings of the IEEE international conference on computer vision. p 618-626

Simonyan K, Zisserman A, (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

Snoek J, Larochelle H, Adams RP (2012) Practical bayesian optimization of machine learning algorithms. Adv Neural Inform Process Syst 25.

Spencer BF Jr, Hoskere V, Narazaki Y (2019) Advances in computer vision-based civil infrastructure inspection and monitoring. Engineering 5(2):199–222

Stent S, Gherardi R, Stenger B, Soga K, Cipolla R (2016) Visual change detection on tunnel linings. Mach vis Appl 27:319–330

Vedaldi A, Soatto S (2008) Quick shift and kernel methods for mode seeking. Computer Vision–ECCV 2008: 10th European Conference on Computer Vision, Marseille, France, October 12–18, 2008, Proceedings, Part IV 10. p 705-718

Weiss K, Khoshgoftaar TM, Wang D (2016) A survey of transfer learning. J Big Data 3(1):1–40

Xu B, Liu C (2022a) A 3D reconstruction method for buildings based on monocular vision. Comp-Aided Civil Infrastr Eng 37(3):354–369

Xu B, Liu C (2022b) Pavement crack detection algorithm based on generative adversarial network and convolutional neural network under small samples. Measurement 196:111219

Yang S, Kim Y, Kim Y, Kim C (2020) Combinational class activation maps for weakly supervised object localization. Proceedings of the IEEE/CVF Winter conference on applications of computer vision. p 2941-2949

Zakaria M, Karaaslan E, Catbas FN (2022) Advanced bridge visual inspection using real-time machine learning in edge devices. Adv Bridge Eng 3(1):1–18

Zhang M, Zhou Y, Zhao J, Man Y, Liu B, Yao R (2020) A survey of semi-and weakly supervised semantic segmentation of images. Artif Intell Rev 53:4259–4288

Zhang Y, Sohn K, Villegas R, Pan G, Lee H (2015) Improving object detection with deep convolutional networks via bayesian optimization and structured prediction. Paper presented at the proceedings of the IEEE conference on computer vision and pattern recognition. p 249–258

Zhao R, Zheng K, Wei X, Jia H, Li X, Zhang Q, Zhang F (2022) State-of-the-art and annual progress of bridge engineering in 2020. Adv Bridge Eng 3(1):1–71

Zhao R, Zheng K, Wei X, Jia H, Liao H, Li X, Xiao L (2021) State-of-the-art and annual progress of bridge engineering in 2020. Adv Bridge Eng 2:1–105

Publisher's Note

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