Construction and Building Materials 234 (2020) 117367

Contents lists available at ScienceDirect

Construction and Building Materials

journal homepage: www.elsevier.com/locate/conbuildmat

Image-based concrete crack detection in tunnels using deep fully convolutional networks

Yupeng Ren [a,b], Jisheng Huang [b], Zhiyou Hong [c], Wei Lu [b], Jun Yin [b], Lejun Zou [a], Xiaohua Shen [a,*]

[a] School of Earth Sciences, Zhejiang University, Hangzhou 310027, Zhejiang, China
[b] Advanced Research Institute, Dahua Technology Co., Ltd, Hangzhou 310053, Zhejiang, China
[c] College of Electronic Science and Technology, Xiamen University, Xiamen 361005, Fujian, China

HIGHLIGHTS

- Deep learning-based automatic segmentation of concrete cracks in tunnels.
- A new end-to-end crack segmentation method based on fully convolutional networks.
- More efficiency and higher accuracy than the conventional and other deep learning-based crack segmentation methods.
- Use of dilated convolution, spatial pyramid pooling, skip connections, and an optimized loss function.

ARTICLE INFO

Article history:
Received 21 December 2018
Received in revised form 20 August 2019
Accepted 23 October 2019

Keywords:
Concrete
Crack detection
Deep learning
Convolutional neural network
Pixel-wise segmentation
Structural health monitoring

ABSTRACT

Automatic detection and segmentation of concrete cracks in tunnels remains a high-priority task for civil engineers. Image-based crack segmentation is an effective method for crack detection in tunnels. With the development of deep learning techniques, especially the development of image segmentation based on convolutional neural networks, new opportunities have been brought to crack detection. In this study, an improved deep fully convolutional neural network, named as CrackSegNet, is proposed to conduct dense pixel-wise crack segmentation. The proposed network consists of a backbone network, dilated convolution, spatial pyramid pooling, and skip connection modules. These modules can be used for efficient multiscale feature extraction, aggregation, and resolution reconstruction which greatly enhance the overall crack segmentation ability of the network. Compared to the conventional image processing and other deep learning-based crack segmentation methods, the proposed network shows significantly higher accuracy and generalization, making tunnel inspection and monitoring highly efficient, low cost, and eventually automatable.

1. Introduction

Concrete cracking is a common phenomenon in tunnels and has a great impact on structural stability of tunnels [1–6]. Cracking is one of the earliest indications of degradation in concrete structures, and therefore crack detection and analysis are crucial for maintaining the safety of tunnels [2,7]. Manual crack inspection in tunnels is often performed by trained human operators and is a labor-intensive and time-consuming process that may expose inspection personnel to hazardous environments. In addition, inspection results are highly dependent on human subjectivity, which may lead to inaccuracies, false and missed detections [1,4,5,8,9]. For these reasons, there is significant interest in the development of image-based, semiautomatic, and automatic structural health monitoring (SHM) methods to facilitate visual inspection in tunnels [10–13].

Relying on consumer-grade digital cameras and image processing techniques, these technologies enable rapid scanning of the structural surface for assessment of structural health [12]. Many image-based, automatic, or semiautomatic crack detection methods using the conventional digital image processing techniques have been proposed [1–5,9]. Tomoyuki and Shuji [1] introduced an efficient and high-speed crack detection method that employs percolation-based image processing. Arena et al. [8] proposed a

Abbreviations: SHM, structural health monitoring; CNN, convolutional neural network; FCN, fully convolutional network; PSPNet, Pyramid Scene Parsing Network; SPP, spatial pyramid pooling; ReLU, Rectified Linear Unit; PA, pixel accuracy; IoU, Intersection over union.

* Corresponding author.
E-mail address: shenxh@zju.edu.cn (X. Shen).

https://doi.org/10.1016/j.conbuildmat.2019.117367
0950-0618/

crack quantification method based on 2D images that employs binarization, morphological processing, separation, geometrical analysis, and filtering. Li et al. [14] developed a method to detect concrete cracks in which local binarization and connected component analysis were used to segment candidate cracks from the background. Geometrical properties were then extracted to filter noise and false results, and the remaining objects were considered as cracks and their geometrical attributes were recorded. Generally, conventional crack detection methods follow the extraction process of image graying, binarization, edge extraction, or morphological operations and filtering. Post-processing is also carried out to reduce errors and to estimate crack parameters [6]. Although conventional crack detection methods are quite effective compared with manual inspections, they are highly dependent on sophisticated classifiers and processing flows, and this dependence leads to low efficiency and weak generalizations [9,15].

With the development of Artificial Intelligence (AI), especially the breakthrough of deep learning techniques in computer vision, new opportunities are available for image-based crack detection. As one of the most effective supervised learning methods, deep convolutional neural networks (CNNs) are trainable on the end tasks of image classification and object detection, and have strong feature extraction and generalization capabilities [16–20]. Despite the need for high-quality labeled images for training purposes, the convolution layers in CNNs can automatically learn crack features from images by means of backward propagation of errors using gradient descent, allowing analysis of large changes in appearance or background among different crack images. CNN-based crack image classification and detection methods have been proposed in the past few years. Cha et al. [15] used CNNs to build a four-layer classifier to detect cracks in images. More than forty thousand labeled images with or without cracks were used to train the classifier, and consistent performance was obtained on test images under various conditions. Gopalakrishnan et al. [21] employed a pre-trained VGG-16 CNN and transferred learned features to detect cracks automatically in pavement images. The CNN architecture consisted of thirteen convolution layers and a new classifier. The pre-trained CNN was fine-tuned on a pavement crack detection dataset and showed good performance. Dorafshan et al. [9] compared the performance of CNN-based crack detection methods with six common edge detectors. The results show the superiority of the CNN architecture over edge detectors and the promise for future development of improved CNN-based crack detection [22]. Although CNN-based crack image classification and detection methods have achieved good performance, inferring image-wise labels such as crack or the absence of a crack to determine the image category and drawing bounding boxes to get the approximate location of cracks does not provide precise information about the crack shape and location. Thus, it is necessary to develop a dense pixel-level crack segmentation method to extract precise information and high-level features of cracks in an image, such as path, location, length, width, and density.

In recent years, pixel-level semantic segmentation in computer vision has rapidly improved with the development of the convolution technique [23–27]. Convolutional neural networks not only performed better in image classifications, but also made significant progress in segmentation. Fully Convolutional Networks (FCNs), proposed by Long et al. [26], extend the original CNN structure to enable intensive prediction without a fully connected layer. The structure allows the segmentation map to generate images of any size and has improved processing speed compared to the image block classification method. After this successful demonstration, almost all studies on semantic segmentation adopted the FCN structure. Based on FCN, a u-shaped network (U-net) is a semantic segmentation network proposed by Ronneberger et al. in 2015 [25]. This network has an elegant architecture, fast training speed, fewer requirements for data volume, and good performance, allowing its wide use in biomedical image segmentation. The Pyramid Scene Parsing Network (PSPNet) further improved the effect of semantic segmentation using a spatial pyramid pooling module that performs pooling operations at different scales in parallel, and utilizes a global scene category for clues to enhance the final predictions [27]. Several recent studies applied FCN and U-net methods to concrete crack segmentation. Dung et al. [28] proposed a detection method based on FCN for concrete crack segmentation. The FCN network exhibited a high precision in validation of a crack-labeled image dataset. Liu et al. [29] adopted a U-net approach to detect concrete cracks. The U-net based method showed high effectiveness, good robustness, and better accuracy than the previous FCN networks. Despite the recent introduction of these methods for crack detection, both FCN and U-net methods are widely used in the field of computer vision-based semantic segmentation, such as for scene parsing or biomedical image segmentation. The morphological characteristics, spatial and data distribution of concrete cracks are quite different from objects in scene parsing and biomedical images. Thus, the network architecture should be further improved based on the characteristics of crack datasets to meet the requirements for crack detection and inspection in structural health monitoring area.

Here, we proposed a new end-to-end pixel-wise crack segmentation network consisting of convolution feature extraction, dilated convolution receptive field expansion, multiscale max pooling, and skip connection of feature fusion. An optimized loss function is employed to address the class imbalance problem. This method adopts modular design concepts, it not only retains the advantages of FCN, U-net, and PSPNet, but also is improved based on crack dataset characteristics, which makes it robust, efficient, and highly generalizable. The model can be updated and optimized by the implementation of powerful architectures and additional training with more labeled images. With this better model, more effective segmentation of concrete cracks can be implemented using images acquired by cameras in tunnels to perform long-term crack inspection and monitoring.

2. Data

A total of 409 images, with resolution of 4032×3016 pixels, were obtained from a digital single lens reflex camera in a tunnel in Huzhou, Zhejiang Province, China, under different lighting conditions. Images were stored as JPGs, with an average file size of approximately 5 MB. The crack images were manually annotated with Photoshop, a commercial software package. The crack images were labeled: crack pixels were marked by 0 and background pixels were marked by 1, allowing storage of the image information in binary format (Fig. 1). Annotation of each 4032×3016 pixel image typically takes about 40–60 min for a skilled person. Although FCNs can input images of any size, training large images may lead to excessive GPU memory usage and cause the training to fail. To avoid this, large images were cropped to a size of 512×512, facilitating training with multiple batches of simultaneous network input. A total of 919 cropped images were obtained and randomly divided into a training set and a test set, at a ratio of 4:1. The training set was used to train the model parameters, while the test set was used to evaluate the accuracy of the final model.

Data augmentation was implemented due to the limited number of training images. The training images were processed by rotation, translation, scaling and shearing. The rotation angle was 0–20°, the translation range was 0–5% of the length and width of the image, the shear intensity was in the range of 0–0.2 in radians, and the scaling range was ±5%. The above data enhancement operations were used to comprehensively transform the training

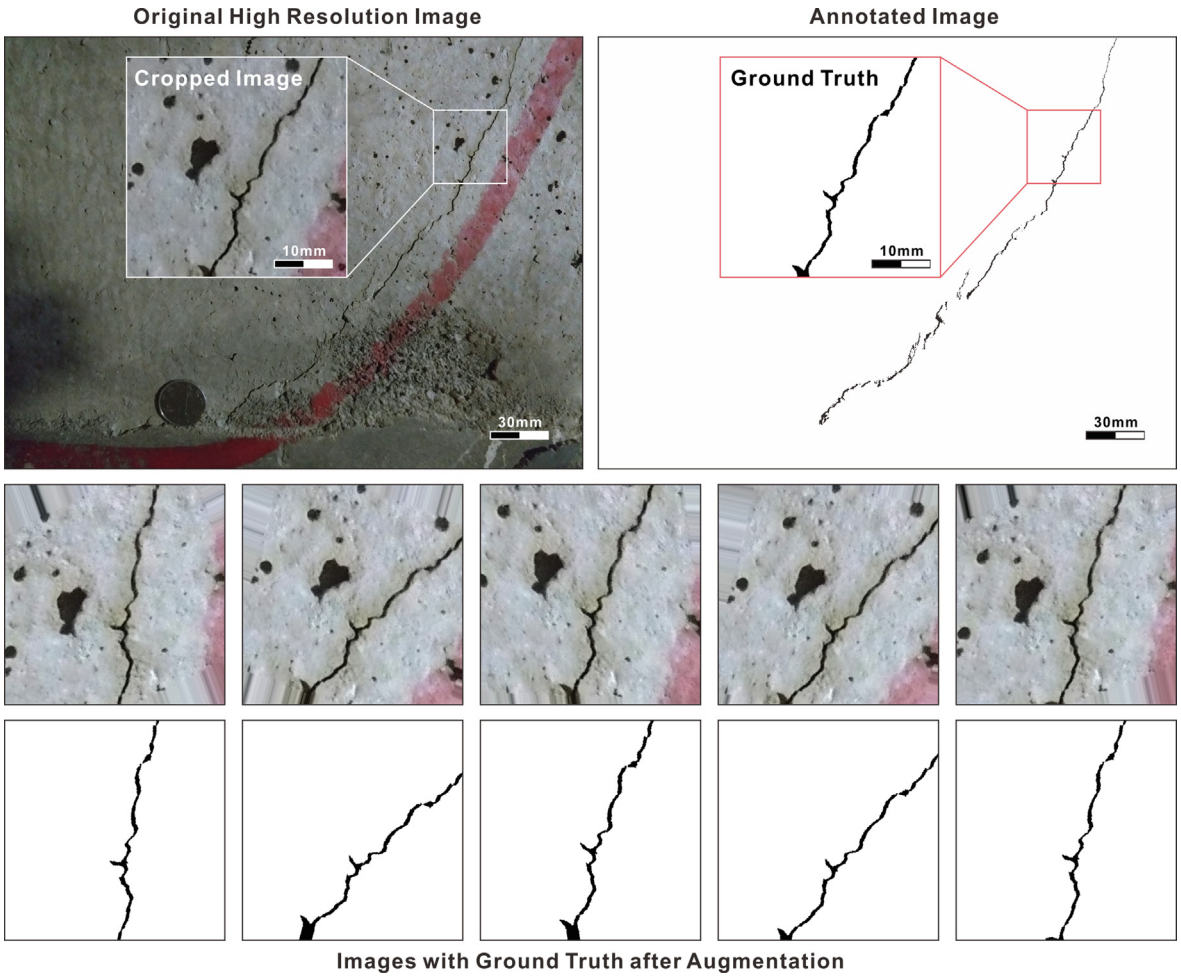

Fig. 1. Illustration of data annotation and augmentation.

data, randomly select the transformation parameters within the specified range of values, and construct a training sample generator with real-time data augmentation, which greatly expanded the number of training samples.

3. Methodology

3.1. Convolutional neural networks

The building blocks of the convolutional neural networks (CNNs) consist of convolution, activation and pooling layers. These layers extract image features, introduce nonlinearity, and reduce feature dimensionality to improve performance and generalization of neural networks [17,30].

The main purpose of the convolution layer in CNNs is feature extraction, which also preserves the spatial relationship between pixels in original images. The convolution kernel or filter slides the image by pixels, computes element-wise multiplication and adds the pixels together to obtain the output. The output matrix is called the feature map. A convolution kernel or filter in deep learning is a collection of 2D arrays of weights. Each 2D array of weights is applied onto all input channels of the previous layer to generate multiple channels, and then summed together to form one single output feature map. This process is repeated for all 2D arrays of weights to generate multiple feature maps of the output layer. Therefore, convolution kernels with different weights will produce different feature maps for the same input image, which means that different convolution kernels extract different features. The lower convolution layers mainly extract structural details and location information from an image, while feature maps of higher layers contain more semantic meaning and less location information. The convolution kernels with weights can be learned and optimized during the training process using an error backpropagation algorithm [31].

After the convolution process, an activation function is used to introduce nonlinearity in neural networks [32,33]. Common activation functions are Rectified Linear Units (abbreviated as ReLU): $f(x) = \max(0, x)$, tanh: $f(x) = \tanh(x)$ and sigmoid: $f(x) = (1 + e^{-x})^{-1}$. In most situations, ReLU is preferred because it trains the neural network several times faster than the other activation functions [17]. ReLU is also an element-wise operation and all the negative values are set to zero. The activation function makes sure that the feature map, after the linear convolution, can be applied with nonlinear operations because most of the data for learning are nonlinear.

The pooling layer is used to downsample the feature maps after convolution and activation to reduce their dimensionality, but important information or background information is still retained. The effect is to reduce the outputs of the pooling kernel at the feature map into a single value in the next layer, and hence expands the receptive field of each neuron. For example, the max-pooling layer uses the maximum value of each pooling kernel, and average pooling layer uses the average value of each pooling kernel from the prior feature map [34]. The pooling layer reduces the number

of model parameters and reduces overfitting, which improves overall neural network generalization and makes it resistant to small transformations, scaling, and distortions.

The fully connected layer is a conventional multilayer perceptron and follows a softmax function as a classifier [35]. This structure bridges the convolution module with conventional neural network classifiers. The fully connected layer uses the features extracted by the convolution layers to classify the input image into different classes. Every neuron in the fully connected layer is connected to all neurons in the previous layer. It is worth noting that a fully connected layer can be converted to a convolution layer because these layers have identical functional forms. The only difference between these layers is that a convolution kernel is connected to a local region of the input layer, while a fully connected kernel covers the entire input region.

Obviously, fully connected layers in CNNs lack location information of image pixels and include only category information. Although CNNs are powerful tools that yield hierarchies of features for image classification, resulting in fairly accurate crack images, the images lack spatial consistency, without precise delineation of cracks. Overall, the use of CNNs for fine crack segmentation based on images remains a significant challenge.

3.2. Fully convolutional networks

Long et al. [26] advanced the coarse-to-fine inference methods by making a prediction for every pixel in an image. This methods lacks fully connected layers and proposes a fully convolutional network (FCN), enabling a classification net to output a classification map with retention of the location information. Skip connections are applied to combine semantic information from deep, coarse layers with location information from shallow, fine layers to produce more accurate and detailed predictions. The FCNs can be trained end-to-end and pixel-to-pixel, requiring approximately 210 ms to analyze an image of 500 × 500 pixels for a single GPU of NVIDIA Tesla K40c.

3.2.1. Encoder-decoder and skip connections in U-net

Ronneberger et al. [25] improved the FCN architecture by proposing a U-net architecture. The U-net has symmetric contracting and expanding paths, and is also known as an encoder-decoder network (Fig. 2). The encoder network is identical to the typical convolutional networks of VGG-16 and the decoder network includes an upsampling layer and dense skip connections [18,25,26]. Like the FCN, high resolution features from the encoder network are combined with the upsampled feature maps by skip connections, allowing the convolution layers in the decoder network to output more precise results. An encoder-decoder network and dense skip connections are also applied in our CrackSegNet to obtain better predictions.

3.2.2. Spatial pyramid pooling

Labeling each pixel in an image by aggregating a small region is difficult. The category of a pixel depends not only on short-range information but also on long-range information [23]. Spatial pyramid pooling (SPP) is one of the most successful methods for multiscale image fusion in computer vision [37]. This method segments the image from coarse to fine levels, and aggregates multilevel features into the output for effective image classification and object detection. Zhao et al. [27] proposed PSPNet for dense semantic segmentation, using global average pooling with different kernel sizes to obtain multiscale context information (Fig. 2). Global pooling layers with context aggregation in different levels are robust to object deformations and therefore can improve the accuracy of crack segmentation.

In this study, global average and max pooling methods were compared after convolution feature extraction to determine which pooling method is better and more applicable for our network. As shown in Fig. 2, the spatial pyramid pooling layer downsampled feature maps according to the size of the pooling kernel; with 1, 1/2, 1/4, and 1/8 lengths and widths of feature maps, respectively. The output feature maps of 1 × 1, 2 × 2, 4 × 4, and 8 × 8 are then upsampled and concatenated.

3.2.3. Dilated convolutions

Dilated convolution, also known as atrous convolution, is a convolution kernel with defined gaps [36,38–40]. Holes (zeros) are inserted into the convolutional kernels to increase image resolution. During convolution, a dilation rate of $k = 1$ is a normal convolution, and $k = 2$ indicates that the convolution kernel skips one pixel per input, and so on. In dilated convolution, convolution can be applied using different dilation rates at different ranges. This method can be used to replace the pooling or subsampling layers of CNNs to expand the receptive field without loss of image resolution and convergence [36,40–42].

In this study, four dilated convolution layers with dilation rates of $k = 2$, 2, 4, and 4 were applied in cascades after the backbone network. This resulted in a deeper network, with better feature extraction, while maintaining the original spatial resolution of the resulting feature maps.

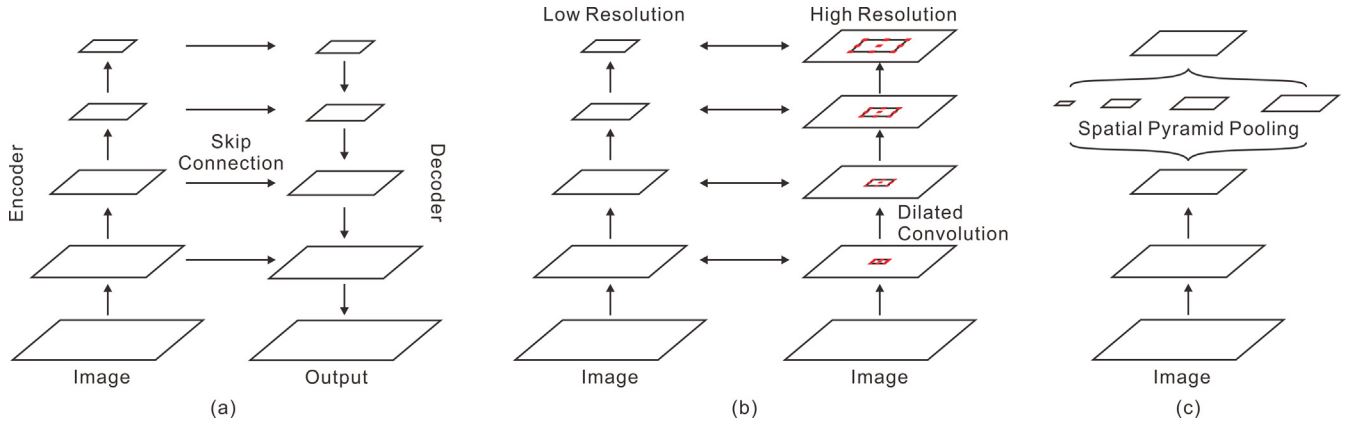

Fig. 2. Architecture of (a) U-net, (b) dilated convolutions, and (c) spatial pyramid pooling (Modified from Chen et al. [36]).

3.3. Network architecture of CrackSegNet

CrackSegNet was developed as a crack segmentation network, and adopted modular design concepts of an encoder path, a decoder path, dilated convolutions, spatial pyramid max pooling, and skip connections (Fig. 3). The backbone net in the encoder path is modified from the typical convolution network VGG-16, which includes a succession of two 3×3 convolutional layers and a 2×2 max pooling layer [18]. Each convolutional layer is followed by a ReLU activation function. After four stages of convolutional and max-pooling layers, the feature vector is input to four dilated convolution layers with dilation rates of 2, 2, 4 and 4 for additional feature extraction without down-sampling. Then, the decoder path applies spatial pyramid max pooling to harvest different sub-region representations, followed by up-sampling and concatenation to form features that include both local and global information. Feature maps after each convolution stage in the encoder path are up-sampled and concatenated with the feature map after spatial pyramid pooling by skip connections. Finally, the feature map is fed into three convolutional layers to obtain the final dense pixel-wise prediction.

3.4. Model training

The model is trained from data of the labeled crack images using backward propagation of errors, an algorithm for supervised learning of artificial neural networks using gradient descent [16,31,43]. This algorithm can be divided into the following steps:

a) Initialize random weights for the network,
b) Feed in an example and obtain a prediction,
c) Calculate the error between the prediction and ground truth for every node in the network, and
d) Update the weights with learning rate α according to

$$\omega^{t+1} = \omega^t - \alpha \frac{\partial E}{\partial \omega} \tag{1}$$

where E is a loss function that defines the error between the prediction and ground truth, and ω^t denotes the weights of the neural network for iteration t by gradient descent.

The commonly used binary classification loss function is the cross entropy loss function, which is defined as follows:

$$L_{ce} = -y\log\hat{y} - (1-y)\log(1-\hat{y}) \tag{2}$$

where $y \in \{0, 1\}$ is the ground-truth class probability and $\hat{y} \in [0, 1]$ is the predict class probability.

The cross entropy loss function is altered by introducing a weighting factor β and a tunable focusing parameter γ ($\gamma \geq 0$), to address class imbalance problems [44]. The function is as follows:

$$L_{fl} = -y\beta(1-\hat{y})^{\gamma}log\hat{y} - (1-y)(1-\beta)\hat{y}^{\gamma}\log(1-\hat{y}) \tag{3}$$

where β and γ are respectively set as 0.25 and 2.

The deep learning framework of Keras with TensorFlow is used as a backend to complete the training and prediction of CrackSegNet on a single GPU of NVIDIA GTX1080Ti [45,46]. Random weights in convolutional layers are initialized using built-in He normal initializers in Keras [46,47]. An Adam optimizer is used according to the default parameters provided by Kingma et al. [48] with an initial learning rate of 0.0001. Several rounds of adjustment of hyper-parameters and training are carried out for each model to maximize the convergence of loss function to the global optimum. An average of 40 training epochs were used, with 2000 training rounds per epoch. The batch-size was set at 2, with a total number of training round that is approximately 220 times the size of the training set. Models during the training process are saved after every epoch with monitoring of the minimum loss value. The accuracy of the model was then verified on the test set and the model with the highest accuracy was saved as the final model.

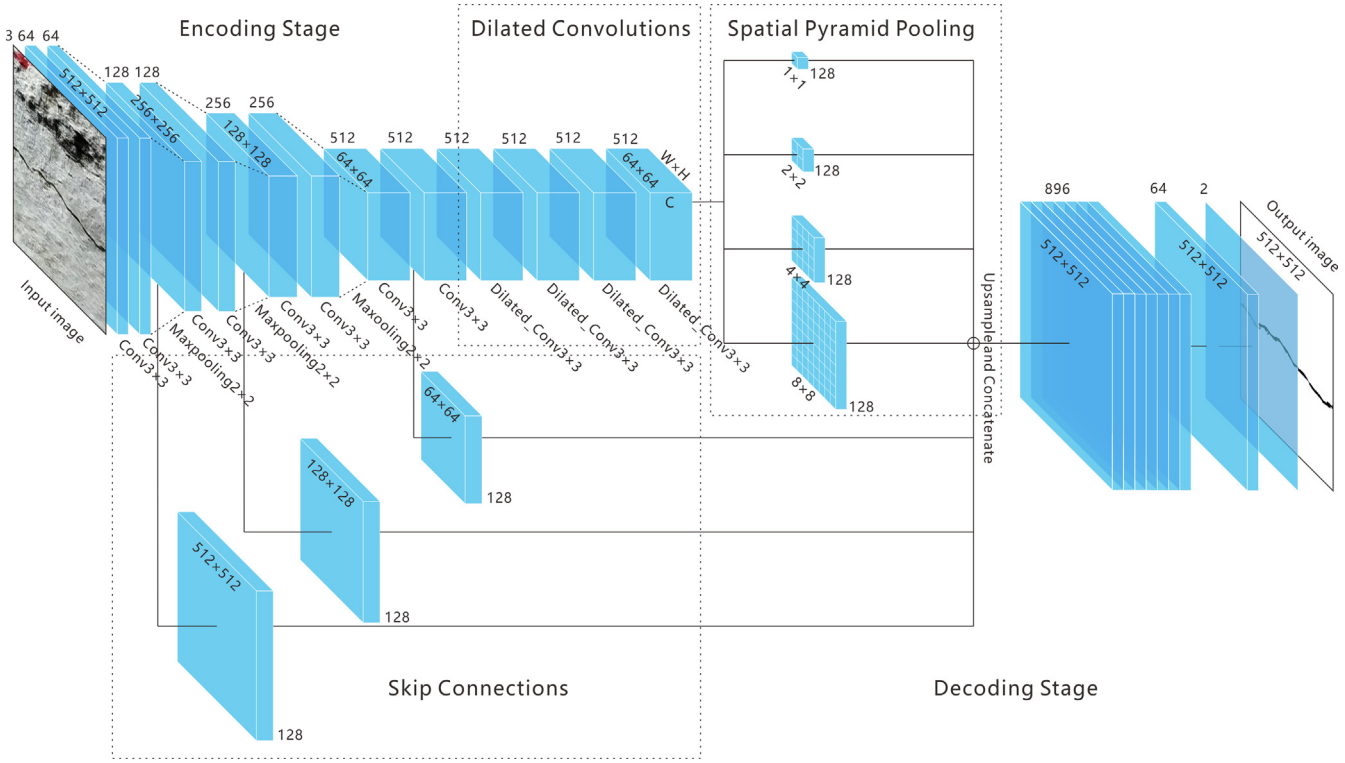

Fig. 3. Crack segmentation network.

The output of CrackSegNet is a probability map with pixel values that range from 0 to 1, in which a black color on the white background indicates that the pixel is more likely to be a crack pixel. The threshold of the probability map is set to 0.5 to obtain a binarized crack segmentation image.

3.5. Conventional methods

The ability of this method to detect cracks was compared to that of conventional digital image processing methods [14]. The conventional detection process is summarized in Fig. 4. First, the original RGB image is grayed and binarized. Local binarization is then used to process the resulted image, using a filtering window 50×50 pixels in size to perform adaptive threshold binarization based on the statistical average brightness value. The red paint in the original image (from the original marking when the image was obtained) is removed through the red channel threshold.

Next, median filtering with a size of 5×5 is used to reduce noise as a typical pre-processing step to improve the results for later processing. Morphological processing, including dilation, erosion, and thinning steps are applied. Dilation operation is used to rebuild the connected region of any cracks broken by image binarization. Erosion is applied to show the original width of the crack. Next, thinning processing using Zhang-Suen thinning algorithm is used to skeletonize the curves and convert their width to one pixel [49]. Next, branches, intersection points, and the endpoints of each curve are detected to separate individual cracks and to calculate the lengths and the minimum circumscribed circles of the cracks. The relevant algorithm in an open source computer vision library (OpenCV) is adopted to calculate the minimum circumscribed circle [50]. By calculating the factor of circle F_c, the candidate crack curves are then filtered, where the factor of circle F_c is defined as:

$$F_c = \frac{4A}{\pi l^2} \tag{4}$$

where A is the area of the extracted crack and l is the diameter of the minimum circumscribed circle. The value of F_c ranges from 0 to 1. An F_c value close to zero indicates that the candidate crack has a large aspect ratio, and a value close to one implies a circle shape [1,14]. The threshold of F_c is set to 0.17, and curves with an aspect ratio greater than this value are removed. Any remaining curves are considered cracks.

4. Accuracy evaluation

Several metrics are used for accuracy evaluation, including pixel accuracy (PA), Intersection over union (IoU), Precision, Recall, and F1 score [51,52]. These evaluation criteria can assess the accuracy of the semantic segmentation tasks. For binary classification they are defined as:

$$\mathrm{PA} = \frac{\sum_{i=0}^{k} p_{ii}}{\sum_{i=0}^{k}\sum_{j=0}^{k} p_{ij}} = \frac{TP + TN}{TP + FP + FN + TN} \tag{5}$$

$$\mathrm{IoU} = \sum_{i=0}^{k} \frac{p_{ii}}{\sum_{j=0}^{k} p_{ij} + \sum_{j=0}^{k} p_{ji} - p_{ii}} = \frac{TP}{TP + FP + FN} \tag{6}$$

$$\mathrm{Precision} = \frac{TP}{TP + FP} \tag{7}$$

$$\mathrm{Recall} = \frac{TP}{TP + FN} \tag{8}$$

and

$$\mathrm{F1} = \frac{2 \times P \times R}{P + R} = \frac{2 \times TP}{2 \times TP + FP + FN} \tag{9}$$

where the total class number including background is $k+1$, $k \in \{0, 1\}$, and p_{ij} represents the number of pixels of class i inferred to belong to class j. So, p_{ii}, p_{ij} and p_{ji} represent the pixel number of true positives (TP), false positives (FP) and false negatives (FN), respectively (Table 1). P and R in F1 represent Precision and Recall, respectively.

PA is the simplest of the evaluation metrics, and calculates the ratio between the amount of correctly classified pixels and the total number of pixels in an image. IoU is a standard metric to assess performance in semantic segmentation tasks.

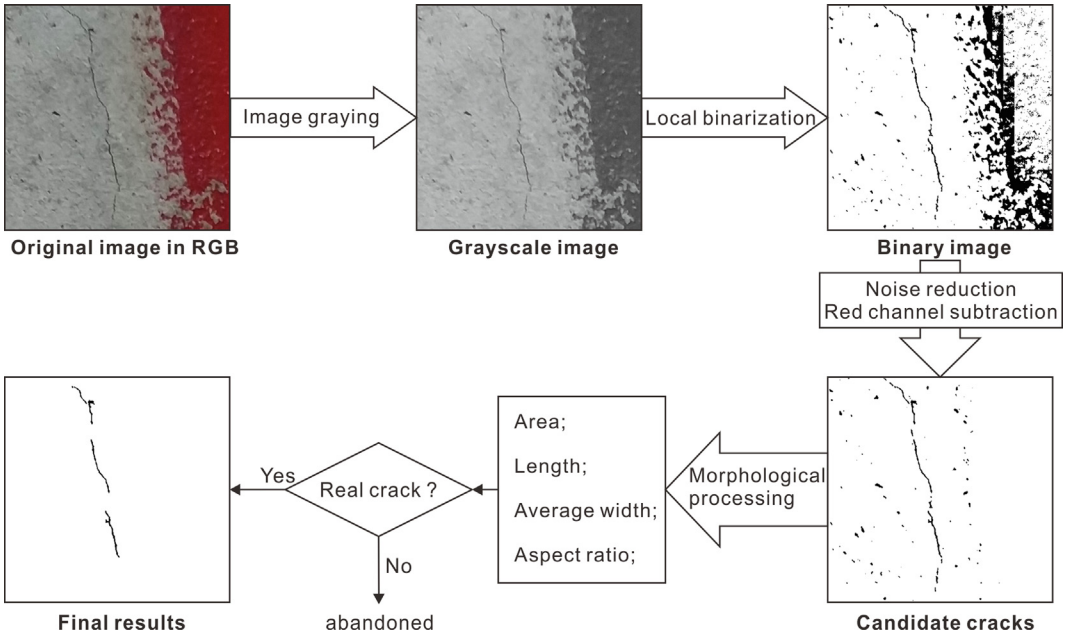

Fig. 4. Conventional crack detection method (modified from Li et al. [14]).

Table 1
Confusion matrix of binary classification.

Data	Classified as positive	Classified as negative
positive	True positive (TP)	False negative (FN)
negative	False positive (FP)	True negative (TN)

5. Results and discussion

5.1. Comparison of crack detection accuracy between CrackSegNet and other methods

We compared the accuracy of the conventional crack segmentation method and the fully convolutional network method for crack segmentation for the same test set. U-net was used as a baseline and CrackSegNet was evaluated with multiple modules such as backbone network, spatial pyramid average pooling, spatial pyramid max pooling, skip connections, and dilated convolutions. For CrackSegNet, focal loss is applied during the training with the final model after addition of the dilated convolution module. Each network includes one new module compared to the previous network, and then is retrained to compare their accuracy and running time. Evaluation metrics of PA, IoU, Precision, Recall, and F1 score were calculated for the different methods as listed in Table 2.

The accuracy evaluation results show significantly higher PA, IoU, Precision, Recall and F1 metrics of the CNN-based crack extraction methods compared to those of the conventional method. Both U-net and CrackSegNet perform well and achieve a high level of accuracy on the test set. The baseline network U-net achieves an IOU of 47.64% and an F1 score of 63.09%. CrackSegNet greatly surpasses U-net in both IOU and F1 metrics and has better generalization, with good performance for low illumination and fuzzy jitter images (Fig. 5). U-net and CrackSegNet provide similar segmentation results for single, simple scenarios (Fig. 5a and b). However, when the scene is complex or the crack width is very small (1 to 2 pixels), CrackSegNet obtains better segmentation results with more accurate location and morphology details (Fig. 5c–e), and is robust to disturbances such as stains in the background (Fig. 5f–j).

The inference time of U-net on NVIDIA GTX1080Ti is approximately 55 ms for an image with a size of 512×512 pixels. The computational efficiency of the CrackSegNet method is slightly lower than that of U-net, with inference time that is approximately 90 ms per image on the same GPU. All the FCNs are much faster than the conventional method, which takes an average time of about 2.67 s per image on 4-core CPU of Core i5 6500.

5.2. Improvements of CrackSegNet

Unlike the traditional hierarchical structure of CNNs, the architecture design of CrackSegNet adopts a modularization concept. Each module is designed independently, allowing the network to be customized and upgraded as necessary, and for parts to be swapped out and reused. This allows incremental adjustments to the network rather than complete replacement. With the transfer of learned features, the upgraded neural network can be fine-tuned and retrained, in a process that is usually much faster and easier than the de novo training of a network with randomly initialized weights. The modularization allows the rapid transfer of learned features or modules to a new architecture or dataset.

The convolutional network of VGG-16 was first modified as a network backbone resulting in IoU and F1 scores of 38.2% and 54.45% respectively. In the next step, spatial pyramid max pooling or spatial pyramid average pooling was added. Both models exhibited better accuracy than the U-net model. After feature extraction of the backbone network, the feature maps were then input to an SPP module to aggregate global and different-region-based features. The inclusion of the SPP module makes CrackSegNet invariant to global or large-scale geometric transformations, scaling, or distortions. The SPP and the max pooling layer in the backbone network make CrackSegNet invariant to scale and morphological changes of cracks from coarse to fine. As shown in Figs. 6 and 7, the network including the SPP module results in a larger TP and smaller FN than the backbone network. The SPP-containing network allows more precise extraction of crack paths. Both spatial pyramid average pooling and spatial pyramid max pooling were conducted to obtain the average and maximum value in the pooling kernel as output. The average pooling mainly extracts context information, and max pooling focuses on the texture information of the image. Considering the morphological characteristics and segmentation methods of cracks, obtaining texture and edge information is essential. The results show that SPP methods combined with either pooling methods provided higher accuracy than U-net, and the method that included max pooling was better than the average pooling method, with an F1 score of 66.52% (Table 2).

After the backbone network and SPP module, skip connections were added to combine the outputs from the higher and lower layers, resulting in a 0.48% improvement of the F1 score. The multi-scale feature hierarchies are 1/8 the size of the original input image, and after application of the SPP module, mainly contain high-level semantic and class meanings with less location information (Fig. 3). The directly upsampled, dense, pixel-wise predictions tend to have low resolution and spatial precision (Figs. 6c, d, 7c, and d). This problem can be addressed by employing skip connections that combine feature hierarchies with finer strides of lower convolutional layers in the backbone network, with higher ones after SPP. Feature hierarchies from the SPP module and skip connections are then upsampled and fused together by concatenation. The final feature map aggregates four SPP layers and three skip connection layers in parallel, which covers a large-scale range of the original image (Fig. 3). The network with skip connections showed significantly improved extraction accuracy of crack edges, and yielded a smaller FP (Figs. 6e and 7e). Dense pixel predictions with high resolution and spatial precision brought the F1 score of the model with skip connections to 67%, higher than the 66.52% of the previous model (Table 2).

Four dilated convolution layers were next added between the network backbone and the spatial pyramid pooling, resulting in performance that was improved by approximately 5.58%.As the

Table 2
Comparison of crack segmentation accuracy of the conventional digital image processing, U-net, and CrackSegNet methods with different modules.

Methods		PA	IoU	Precision	Recall	F1	Running time
Conventional method (Li et al. 2018)		97.17%	14.46%	25.27%	26.33%	25.79%	2670 ms
U-net (Liu et al. 2019)		98.88%	47.64%	66.49%	60.02%	63.09%	55 ms
CrackSegNet	Backbone	98.77%	38.2%	63.85%	47.46%	54.45%	49 ms
	Spatial Pyramid Pooling (Average)	98.98%	49.1%	69.56%	61.47%	65.38%	55 ms
	Spatial Pyramid Pooling (Max)	98.99%	50.64%	71.38%	62.28%	66.52%	55 ms
	Skip Connections	98.99%	51.43%	70.41%	63.91%	67%	83 ms
	Dilated Convolution	99.12%	57.28%	74.84%	70.46%	72.58%	90 ms
	Focal Loss	99.01%	59.06%	66.07%	85.54%	74.55%	90 ms

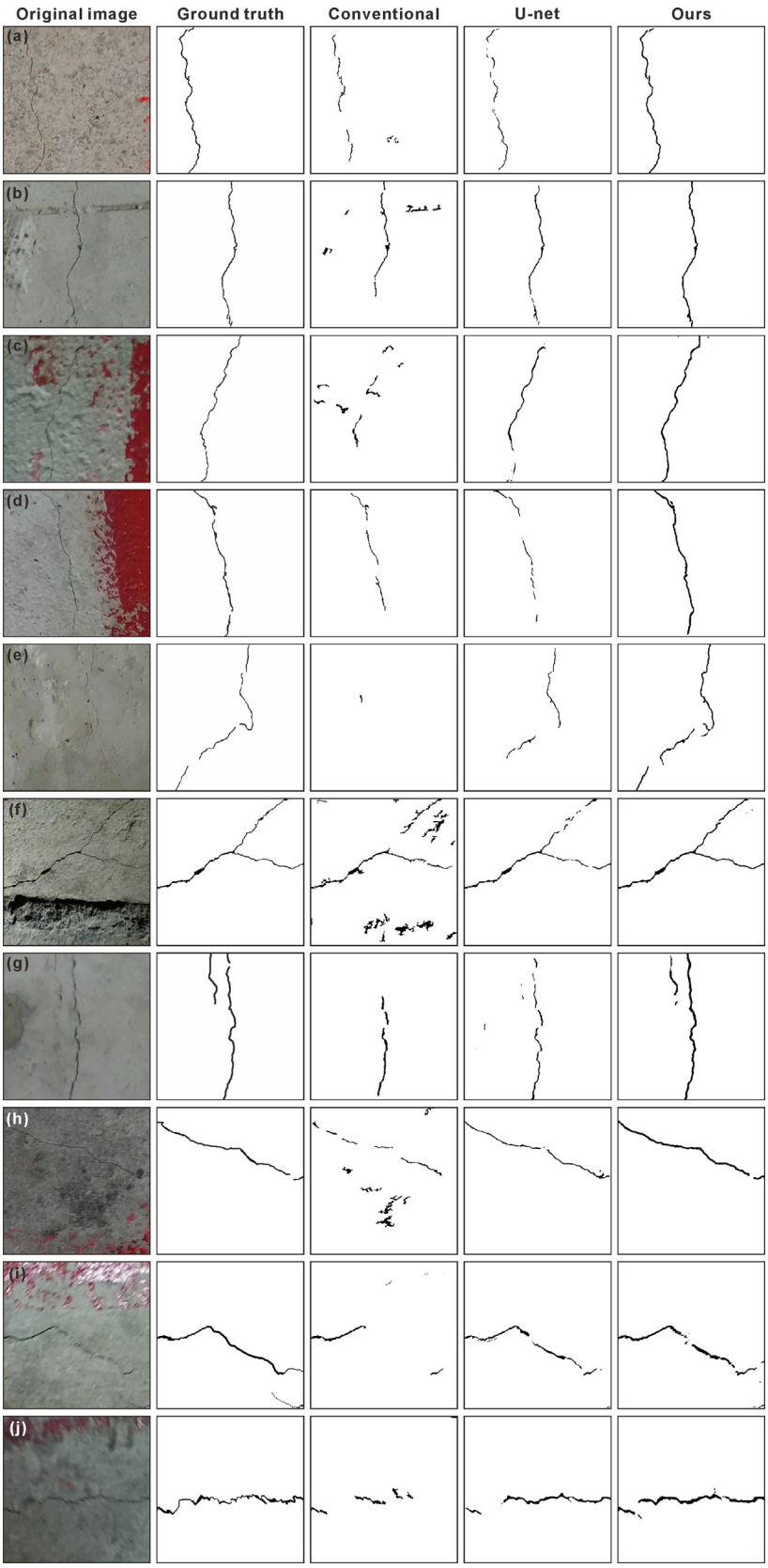

Fig. 5. Examples of crack segmentation results on test set. Original image and ground truth are provided. The CrackSegNet shows superior performance with its accurate shape segmentation and boundary delineation capabilities. U-net is more efficient with less inference time but its precision is lower than that of CrackSegNet, particularly in complex scenarios. Of the tested models, the conventional method is time-consuming and shows the weakest performance.

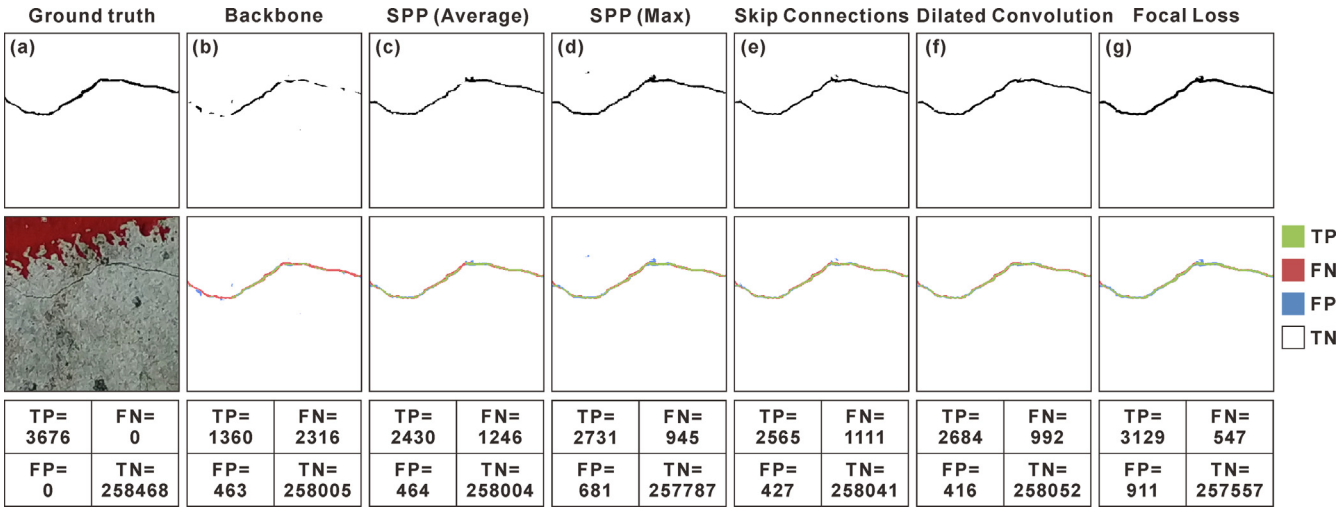

Fig. 6. Crack segmentation results of networks with different modules of fine images in the first row. TP, FN, FP and TN are visualized in different colors in the second row. Detailed values are in the third row.

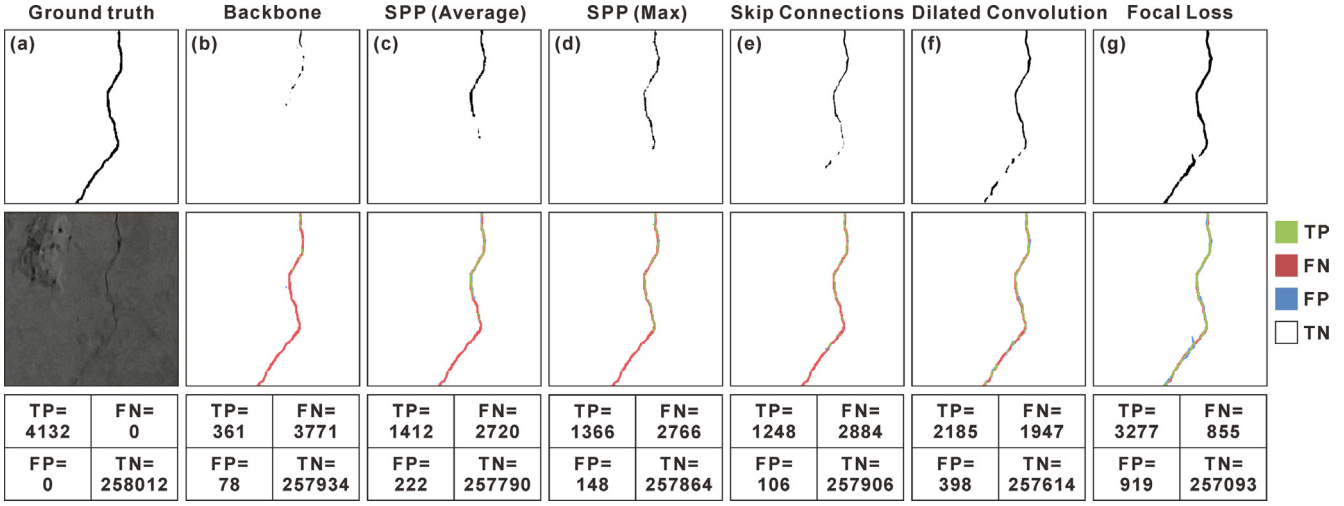

Fig. 7. Crack segmentation results of networks with different modules of low-contrast images in the first row. TP, FN, FP and TN are visualized in different colors in the second row. Detailed values are in the third row.

backbone network goes deeper, the resolution of high-level features becomes lower. The downsampling of the feature maps is mainly due to the max pooling layers. The incorporation of dilated convolution was next tested in an effort to maintain the resolution and receptive field of the network without using max pooling layers, enabling feature extraction in deeper convolution networks. Four dilated convolution layers were added after the backbone network to extract better features and enlarge the field of convolutional kernels. This brings more accurate category prediction for each pixel, especially for a dark or low-contrast image. Fig. 7 indicates that inclusion of dilated convolution layers resulted in the correct classification of more crack pixels, leading to an increase of TP and reduction of FN. The experiment results show that the F1 score of the network increased from 67% to 72.58% for the model containing the dilated convolution layers compared to the model without these layers (Table 2).

Loss functions were next optimized to address class imbalances, which yielded a higher F1 score of approximately 1.97% over the previous model. The recall of the final model was greatly improved at the expense of precision and PA degradation. The final predictions of crack pixels were much better than those of the baseline model (Fig. 5). The cross entropy loss function was reshaped to address class imbalances, downweighting the loss of the easy examples and focusing on the hard examples. Obviously, the number of crack pixels in a single image is much lower than the number of background pixels. Vast numbers of easy background examples comprise the majority of the loss, and dominate the gradient, leading to inefficient training of the network and even degeneration of the model. Optimized loss functions focused training on a sparse set of crack pixels, and the final results show that both recall and the F1 score of the model on the test set increased, and precision decreased (Table 2). Furthermore, focusing on the hard examples of crack pixels resulted in higher TP and lower FN values, essential to determine the paths of fine cracks in low quality images taken in poor lightning or with a lack of focus (Fig. 7g). From the perspective of engineering applications, tools must effectively detect all cracks. Therefore, it is more important to improve the recall rate of crack segmentation during tunnel inspection even at the expense of precision.

5.3. Advantages of CrackSegNet over the conventional method

The results of this study indicate that CrackSegNet achieved better performance than the conventional method. CrackSegNet

can extract the cracks from complex scenes, effectively remove the influence of shadows, stains, and other interferences, and perform end-to-end crack segmentation. Feature extractors based on CNNs can learn crack features robustly, comprehensively evaluate these features from multiscale, edge, and background information, and successfully predict the crack probability of each pixel. The conventional crack segmentation method is unable to achieve high accuracy, especially under complex conditions. Designing feature extractors to use for crack segmentation is time-consuming and requires expert knowledge.

With increased number of samples, CrackSegNet will learn more robust features, reduce over-fitting, and further enhance model generalization. These features can improve the accuracy of crack detection under complex situations. However, the robustness or accuracy of conventional crack segmentation methods cannot be improved as data volumes increase. Large amounts of data may even have negative effects on the final extraction results. Manually designed feature extractors must be tuned before application to a larger data set or more complex situations. Moreover, an increase of data volume or scene complexity often requires the design of more sophisticated feature extractors, requiring additional computational and storage and with insufficient accuracy and performance for real-time applications.

The results of CrackSegNet are very favorable for the next steps of analysis. For example, with crack parameter extraction and elimination of the complex processing process, the area, length, and average width of cracks can be directly measured after binarization and thinning.

5.4. Advantages of CrackSegNet over U-net

CrackSegNet has a deeper network and higher accuracy than U-net (Table 2 and Fig. 5). CrackSegNet not only includes advantages of U-net, such as the encoder-decoder path and skip connections, but also adds an SPP module and cascaded dilated convolutions. Addition of the SPP module allows CrackSegNet to capture cracks at multiple scales and makes it invariant to global and multiscale geometric transformations and distortions. Dilated convolutions enlarge the receptive field of convolution kernels to incorporate a multiscale context. In particular, the proposed dilated convolutions with various dilation rates in cascade further deepen the network and strengthen the capabilities of feature extraction without degrading the resolution of feature maps.

As shown in Fig. 5a–g, CrackSegNet provides robust feature extraction and better predictions of crack pixels to enhance the segmentation of the detailed morphology of cracks. CrackSegNet allowed greatly improved IoU, precision, and recall, with 99% PA. However, the deeper network structure lowers the efficiency of CrackSegNet, and the inference time of each image increased to 90 ms, higher than the 55 ms for U-net.

5.5. Implications for crack detection and monitoring

Deep learning or CNN-based methods provide powerful tools for crack detection and other SHM-related tasks, and have achieved high accuracy for various real-world situations. End-to-end deep learning techniques can automatically and robustly convert raw images into actionable information without the need for specialized features or multi-step feature extraction. Remote cameras, unmanned aerial vehicles (UAVs), and mobile robots can be used to obtain continuous high resolution crack images at a very low cost, allowing tunnel inspection and monitoring without the need for human inspectors.

An accurate and rapid crack image segmentation method can effectively improve the efficiency of crack detection and monitoring. CrackSegNet performs exceedingly well compared to the conventional method and other CNN-based methods, and can precisely delineate the shape and location of cracks in images. The algorithm efficiency is about 11 frames per second (fps) on the GPU side, which can easily meet real-time requirements by targeted speed optimizing or adopting lightweight backbone networks. In addition, with increased computational capabilities, the front-end equipment or back-end servers can perform crack detection more quickly, obtaining detection results in dense time dimension to further improve the accuracy of crack monitoring and change detection.

This crack detection algorithm can be integrated into surveillance cameras, intelligent unmanned autonomous systems, such as unmanned aerial vehicles (UAVs) and mobile robots, or integrated into a software platform and deployed on the server side. Indoor and remote crack detection and monitoring can be implemented continuously and automatically by processing crack images acquired on site, facilitating tunnel inspection and maintenance. Additionally, the entire workflow of data collection, image processing, and results analysis can be digitally rebuilt, making it paperless, low in labor costs, and convenient.

There remain several limitations of deep learning and CNN-based crack detection using images. As one of the most effective supervised learning methods, the CNN-based method requires an annotated dataset of cracks for training, yet obtaining high-quality labeled crack segmentation images is labor-intensive and time-consuming. In addition, crack pixels in the image coordinates cannot be converted to physical or world coordinates due to a lack of camera calibration and scale factor for 2D images [53]. It is difficult to obtain physical parameters of cracks, such as length, width, and area. However, accurate crack detection in images can promote the multi-dimensional data fusion of images with other high-precision data, such as radar, laser and GPS data, compensating for defects of different data types from multi-modal sensors and facilitating subsequent quantitative and refined detection of crack information [54].

6. Conclusions

This study proposed a new end-to-end crack detection method consisting of convolution feature extraction, dilated convolution receptive field expansion, multiscale max pooling, and skip connection of feature fusion modules. Additionally, an optimized loss function was employed to address the class imbalance problem. The resulting method has good generalization and low data requirements, and can be enhanced by the inclusion of more powerful architectures or additional training of labeled images. Compared to the conventional method, the accuracy and speed of the proposed crack segmentation method are greatly improved. Moreover, compared with the other fully convolution network structures such as U-net, CrackSegNet provides higher accuracy. The rapid advances in theoretical studies and applications of deep learning techniques in various fields provide increasing motivation for civil engineers to develop deep learning or CNN-based methods to improve monitoring and inspection of civil infrastructure. The feasibility of CNN-based crack detection during tunnel inspection and other SHM-related tasks will ultimately lead to safer, more efficient, less expensive civil infrastructure condition assessment, with the potential for eventual automation.

7. Computer code availability

- Name of code: CrackSegNet.
- Developer: Yupeng Ren*, Jisheng Huang.
- *Contact address: School of Earth Sciences, Zhejiang University, Hangzhou, 310027, Zhejiang, China.

- *Telephone number and e-mail: +8618158519696, renyupeng@zju.edu.cn.

- Year first available: 2018.
- Hardware required: CPU Intel Core i3 or higher, Memory > 4 GB, GPU GTX 750Ti or higher, Hard disk > 10 GB.
- Software required: Linux: Python, TensorFlow, Keras, Numpy, h5py, scikit-image; Windows 7: VS2010, OpenCV.
- Program language: Python & C++.
- Program size: Code ~ 100 KB, Model ~ 50–500 MB, Data ~ 300 MB.
- Details on how to access the source code:
  a) *Code available at.* https://github.com/Arenops/CrackSegNet.
  b) *Download link for trained models (CrackSegNet and U-net)*: https://pan.baidu.com/s/1gxmr93Y6A62xR1vaTnGXVw *Password*: dpmm.
  c) *Download link for dataset*: https://pan.baidu.com/s/10W01KQqMS8FFAoRpKl4-Qw *Password*: e9d2.
  d) *Download link for Traditional method and evaluation code*: https://pan.baidu.com/s/1T-Zqu8Dh6WyLbKwRi9Rj0w *Password*: pp69

Declaration of Competing Interest

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.

Acknowledgements

This study was funded by Chinese National Science and Technology Major Project (2017ZX05008-001), the National Natural Science Foundation of China (Grant NO. 41872214) and was partially funded by the Special Fund from Zhejiang Provincial Government, China (zjcx. 2011 No. 98). We also thank Professor Li Changjiang and Professor Lu Zhiming for their helpful discussions.

References

[1] T. Yamaguchi, S. Hashimoto, Fast crack detection method for large-size concrete surface images using percolation-based image processing, Mach. Vis. Appl. 21 (2010) 797–809, https://doi.org/10.1007/s00138-009-0189-8.

[2] T.T. Wang, Characterizing crack patterns on tunnel linings associated with shear deformation induced by instability of neighboring slopes, Eng. Geol. 115 (2010) 80–95, https://doi.org/10.1016/j.enggeo.2010.06.010.

[3] R.S. Adhikari, O. Moselhi, A. Bagchi, Image-based retrieval of concrete crack properties for bridge inspection, Autom. Constr. 39 (2014) 180–194, https://doi.org/10.1016/j.autcon.2013.06.011.

[4] B. Liu, T. Yang, Image analysis for detection of bugholes on concrete surface, Constr. Build. Mater. 137 (2017) 432–440, https://doi.org/10.1016/j.conbuildmat.2017.01.098.

[5] H. Nhat-Duc, Q.L. Nguyen, V.D. Tran, Automatic recognition of asphalt pavement cracks using metaheuristic optimized edge detection algorithms and convolution neural network, Autom. Constr. 94 (2018) 203–213, https://doi.org/10.1016/j.autcon.2018.07.008.

[6] A. Mohan, S. Poobal, Crack detection using image processing: a critical review and analysis, Alexandria Eng. J. 57 (2018) 787–798, https://doi.org/10.1016/j.aej.2017.01.020.

[7] E. Menendez, J.G. Victores, R. Montero, S. Martínez, C. Balaguer, Tunnel structural inspection and assessment using an autonomous robotic system, Autom. Constr. 87 (2018) 117–126, https://doi.org/10.1016/j.autcon.2017.12.001.

[8] A. Arena, C. Delle Piane, J. Sarout, A new computational approach to cracks quantification from 2D image analysis: application to micro-cracks description in rocks, Comput. Geosci. 66 (2014) 106–120, https://doi.org/10.1016/j.cageo.2014.01.007.

[9] S. Dorafshan, R.J. Thomas, M. Maguire, Comparison of deep convolutional neural networks and edge detectors for image-based crack detection in concrete, Constr. Build. Mater. 186 (2018) 1031–1045, https://doi.org/10.1016/j.conbuildmat.2018.08.011.

[10] Y. Liu, J. Yao, X. Lu, R. Xie, L. Li, DeepCrack: a deep hierarchical feature learning architecture for crack segmentation, Neurocomputing 338 (2019) 139–153, https://doi.org/10.1016/j.neucom.2019.01.036.

[11] C.V. Dung, H. Sekiya, S. Hirano, T. Okatani, C. Miki, A vision-based method for crack detection in gusset plate welded joints of steel bridges using deep convolutional neural networks, Autom. Constr. 102 (2019) 217–229, https://doi.org/10.1016/j.autcon.2019.02.013.

[12] B.F. Spencer, V. Hoskere, Y. Narazaki, Advances in computer vision-based civil infrastructure inspection and monitoring, Engineering 5 (2019) 199–222, https://doi.org/10.1016/j.eng.2018.11.030.

[13] L. Attard, C.J. Debono, G. Valentino, M. Di Castro, Tunnel inspection using photogrammetric techniques and image processing: a review, ISPRS J. Photogramm. Remote Sens. 144 (2018) 180–188, https://doi.org/10.1016/j.isprsjprs.2018.07.010.

[14] L. Li, Q. Wang, G. Zhang, L. Shi, J. Dong, P. Jia, A method of detecting the cracks of concrete undergo high-temperature, Constr. Build. Mater. 162 (2018) 345–358, https://doi.org/10.1016/j.conbuildmat.2017.12.010.

[15] Y. Cha, W. Choi, O. Büyüköztürk, Deep learning-based crack damage detection using convolutional neural networks, Comput. Civ. Infrastruct. Eng. 32 (2017) 361–378, https://doi.org/10.1111/mice.12263.

[16] Y. Lecun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, in: Proc. IEEE, 1998, pp. 2278–2324, https://doi.org/10.1109/5.726791.

[17] A. Krizhevsky, G.E. Hinton, ImageNet Classification with Deep Convolutional Neural Networks, in: F. Pereira, C.J.C. Burges, L. Bottou, K.Q.B.T.-A. in N.I.P.S. 25 (NIPS 2012) Weinberger (Eds.), Int. Conf. Neural Inf. Process. Syst., Curran Associates, Inc., 2012: pp. 1–9.

[18] K. Simonyan, A. Zisserman, Very Deep Convolutional Networks for Large-Scale Image Recognition. http://arxiv.org/abs/1409.1556, 2014.

[19] K. He, X. Zhang, S. Ren, J. Sun, Deep Residual learning for image recognition, in: IEEE Conf. Comput. Vis. Pattern Recognit, IEEE, Las Vegas, NV, USA, 2016, pp. 770–778, https://doi.org/10.1109/CVPR.2016.90.

[20] G. Huang, Z. Liu, L. van der Maaten, K.Q. Weinberger, Densely connected convolutional networks, 2017 IEEE Conf. Comput. Vis. Pattern Recognit., IEEE, Honolulu, HI, USA, 2017.

[21] K. Gopalakrishnan, S.K. Khaitan, A. Choudhary, A. Agrawal, Deep Convolutional Neural Networks with transfer learning for computer vision-based data-driven pavement distress detection, Constr. Build. Mater. 157 (2017) 322–330, https://doi.org/10.1016/j.conbuildmat.2017.09.110.

[22] K. Gopalakrishnan, Deep learning in data-driven pavement image analysis and automated distress detection: a review, Data 3 (2018) 28, https://doi.org/10.3390/data3030028.

[23] C. Farabet, C. Couprie, L. Najman, Y. Lecun, Learning hierarchical features for scene labeling, IEEE Trans. Pattern Anal. Mach. Intell. 35 (2013) 1915–1929, https://doi.org/10.1109/TPAMI.2012.231.

[24] B. Hariharan, P. Arbeláez, R. Girshick, J. Malik, Simultaneous detection and segmentation, in: Comput. Vis. – ECCV 2014, Springer, Cham, 2014, https://doi.org/10.1007/978-3-319-10584-0_20.

[25] O. Ronneberger, P. Fischer, T. Brox, P.FischerU-Net: Convolutional Networks for Biomedical Image Segmentation, T. Brox, U-Net: Convolutional Networks for Biomedical Image Segmentation, in: N. Navab, J. Hornegger, W.M. Wells, A.F. Frangi (Eds.), Med. Image Comput. Comput. Interv. - MICCAI 2015, Springer, Cham, 2015, pp. 234–241, https://doi.org/10.1007/978-3-319-24574-4_28.

[26] J. Long, E. Shelhamer, T. Darrell, Fully convolutional networks for semantic segmentation, in: IEEE Conf. Comput. Vis. Pattern Recognit, IEEE, Boston, MA, USA, 2015, pp. 3431–3440, https://doi.org/10.1109/CVPR.2015.7298965.

[27] H. Zhao, J. Shi, X. Qi, X. Wang, J. Jia, Pyramid scene parsing network, in: IEEE Conf. Comput. Vis. Pattern Recognit, IEEE, Honolulu, HI, USA, 2017, pp. 6230–6239, https://doi.org/10.1109/CVPR.2017.660.

[28] C.V. Dung, L.D. Anh, Autonomous concrete crack detection using deep fully convolutional neural network, Autom. Constr. 99 (2019) 52–58, https://doi.org/10.1016/j.autcon.2018.11.028.

[29] Z. Liu, Y. Cao, Y. Wang, W. Wang, Computer vision-based concrete crack detection using U-net fully convolutional networks, Autom. Constr. 104 (2019) 129–139, https://doi.org/10.1016/j.autcon.2019.04.005.

[30] M.D. Zeiler, R. Fergus, Visualizing and understanding convolutional networks, in: D. Fleet, T. Pajdla, B. Schiele, T. Tuytelaars (Eds.), Comput. Vis. – ECCV 2014, Springer, Cham, Zurich, Switzerland, 2014, pp. 818–833, https://doi.org/10.1007/978-3-319-10590-1_53.

[31] D.E. Rumelhart, G.E. Hinton, R.J. Williams, Learning representations by back-propagating errors, Nature 323 (1986) 533–536, https://doi.org/10.1038/323533a0.

[32] V. Nair, G. Hinton, Rectified linear units improve restricted Boltzmann machines, in: Proc. 27th Int. Conf. Mach. Learn., 2010, pp. 807–814.

[33] A.L. Maas, A.Y. Hannun, A.Y. Ng, Rectifier nonlinearities improve neural network acoustic models, in: Proc. Int. Conf. Mach. Learn., 2013, https://doi.org/10.1016/0010-0277(84)90022-2.

[34] D. Scherer, S. Behnke, Evaluation of pooling operations in convolutional architectures for object recognition, in: K. Diamantaras, W. Duch, L.S. Iliadis (Eds.), Artif. Neural Networks – ICANN 2010, Springer, Berlin, Heidelberg, Thessaloniki, Greece, 2010, pp. 92–101, https://doi.org/10.1007/978-3-642-15825-4_10.

[35] H. Simon, Neural network: a comprehensive foundation, 2nd ed., Prentice Hall, 1998.

[36] L.-C. Chen, G. Papandreou, F. Schroff, H. Adam, Rethinking Atrous Convolution for Semantic Image Segmentation. http://arxiv.org/abs/1706.05587, (2017).

[37] K. He, X. Zhang, S. Ren, J. Sun, Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition Technical report, Spat. Pyramid Pooling Deep

Convolutional Networks Vis. Recognit. 37 1904–1916. DOI:10.1109/TPAMI.2015.2389824, 2015.
[38] F. Yu, V. Koltun, Multi-scale context aggregation by dilated convolutions, 2016 Int. Conf. Learn. Represent., 2015. http://arxiv.org/abs/1511.07122.
[39] P. Wang, P. Chen, Y. Yuan, D. Liu, Z. Huang, X. Hou, G. Cottrell, Understanding convolution for semantic segmentation. 1451–1460. https://arxiv.org/abs/1702.08502, 2017.
[40] L.C. Chen, Y. Zhu, G. Papandreou, F. Schroff, H. Adam, Encoder-decoder with atrous separable convolution for semantic image segmentation, in: Comput. Vis. – ECCV 2018, 2018, https://doi.org/10.1007/978-3-030-01234-2_49.
[41] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, A.L. Yuille, Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected CRFs, Atrous Convolution, and Fully Connected CRFs, Iclr. 40 2014 1–14. DOI:10.1684/ejd.2011.1428.
[42] L.C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, A.L. Yuille, DeepLab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs, IEEE Trans. Pattern Anal. Mach. Intell. 40 (2018) 834–848, https://doi.org/10.1109/TPAMI.2017.2699184.
[43] Y. LeCun, B. Boser, J.S. Denker, D. Henderson, R.E. Howard, W. Hubbard, L.D. Jackel, Backpropagation applied to handwritten zip code recognition, Neural Comput. 1 (1989) 541–551, https://doi.org/10.1162/neco.1989.1.4.541.
[44] T.Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollar, Focal loss for dense object detection, in: IEEE Int. Conf. Comput. Vis, IEEE, Venice, Italy, 2017, pp. 2999–3007, https://doi.org/10.1109/ICCV.2017.324.
[45] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D.G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, X. Zheng, TensorFlow: A system for large-scale machine learning. http://arxiv.org/abs/1605.08695, 2016.
[46] Navin Kumar Manaswi, Understanding and working with keras, in: N.K. Manaswi (Ed.), Deep Learn. with Appl. Using Python, Apress, Berkeley, CA, 2018, pp. 31–43, https://doi.org/10.1007/978-1-4842-3516-4_2.
[47] K. He, X. Zhang, S. Ren, J. Sun, Delving deep into rectifiers: surpassing human-level performance on ImageNet classification, in: IEEE Int Conf. Comput. Vis., 2015, pp. 1026–1034, https://doi.org/10.1109/ICCV.2015.123.
[48] D.P. Kingma, J. Ba, Adam: a method for stochastic optimization San Diego, in: 3rd Int. Conf. Learn. Represent., 2015, https://doi.org/10.1145/1830483.1830503.
[49] T.Y. Zhang, C.Y. Suen, A fast parallel algorithm for thinning digital patterns, Commun. ACM 27 (1984) 236–239, https://doi.org/10.1145/357994.358023.
[50] OpenCV - Open Source Computer Vision Library. https://opencv.org, 2018.
[51] M. Sokolova, G. Lapalme, A systematic analysis of performance measures for classification tasks, Inf. Process. Manage. 45 (2009) 427–437, https://doi.org/10.1016/j.ipm.2009.03.002.
[52] A. Garcia-Garcia, S. Orts-Escolano, S. Oprea, V. Villena-Martinez, J. Garcia-Rodriguez, A Review on Deep Learning Techniques Applied to Semantic Segmentation. https://arxiv.org/abs/1704.06857, 2017.
[53] D. Feng, M.Q. Feng, Computer vision for SHM of civil infrastructure: from dynamic response measurement to damage detection – a review, Eng. Struct. 156 (2018) 105–117, https://doi.org/10.1016/j.engstruct.2017.11.018.
[54] J. Valença, I. Puente, E. Júlio, H. González-Jorge, P. Arias-Sánchez, Assessment of cracks on concrete bridges using image processing supported by laser scanning survey, Constr. Build. Mater. 146 (2017) 668–678, https://doi.org/10.1016/j.conbuildmat.2017.04.096.