Smart Structures and Systems, *Vol. 29, No. 1 (2022) 1-16*
https://doi.org/10.12989/sss.2022.29.1.001

A modified U-net for crack segmentation by Self-Attention-Self-Adaption neuron and random elastic deformation

Jin Zhao [1,2,3†], Fangqiao Hu [1,2,3†], Weidong Qiao [1,2,3], Weida Zhai [1,2,3], Yang Xu [*1,2,3], Yuequan Bao [1,2,3] and Hui Li [1,2,3]

[1] *Key Lab of Smart Prevention and Mitigation of Civil Engineering Disasters of the Ministry of Industry and Information Technology, Harbin Institute of Technology, Harbin, 150090, China*
[2] *Key Lab of Structures Dynamics Behavior and Control of the Ministry of Education, Harbin Institute of Technology, Harbin, 150090, China*
[3] *School of Civil Engineering, Harbin Institute of Technology, Harbin, 150090, China*

(*Received March 6, 2021, Revised May 26, 2021, Accepted June 1, 2021*)

Abstract. Despite recent breakthroughs in deep learning and computer vision fields, the pixel-wise identification of tiny objects in high-resolution images with complex disturbances remains challenging. This study proposes a modified U-net for tiny crack segmentation in real-world steel-box-girder bridges. The modified U-net adopts the common U-net framework and a novel Self-Attention-Self-Adaption (SASA) neuron as the fundamental computing element. The Self-Attention module applies softmax and gate operations to obtain the attention vector. It enables the neuron to focus on the most significant receptive fields when processing large-scale feature maps. The Self-Adaption module consists of a multiplayer perceptron subnet and achieves deeper feature extraction inside a single neuron. For data augmentation, a grid-based crack random elastic deformation (CRED) algorithm is designed to enrich the diversities and irregular shapes of distributed cracks. Grid-based uniform control nodes are first set on both input images and binary labels, random offsets are then employed on these control nodes, and bilinear interpolation is performed for the rest pixels. The proposed SASA neuron and CRED algorithm are simultaneously deployed to train the modified U-net. 200 raw images with a high resolution of 4928 × 3264 are collected, 160 for training and the rest 40 for the test. 512 × 512 patches are generated from the original images by a sliding window with an overlap of 256 as inputs. Results show that the average IoU between the recognized and ground-truth cracks reaches 0.409, which is 29.8% higher than the regular U-net. A five-fold cross-validation study is performed to verify that the proposed method is robust to different training and test images. Ablation experiments further demonstrate the effectiveness of the proposed SASA neuron and CRED algorithm. Promotions of the average IoU individually utilizing the SASA and CRED modules add up to the final promotion of the full model, indicating that the SASA and CRED modules contribute to the different stages of model and data in the training process.

Keywords: modified U-net; random elastic deformation; Self-Attention-Self-Adaption neuron; semantic segmentation; steel crack identification

1. Introduction

Fatigue cracks often inevitably occur inside the steel box girder of long-span bridges due to coupled effects of initial material defects and dynamic vehicle loads. The indulgent crack propagations significantly cause the deterioration of structural safety and service performance. Although structural health monitoring (SHM) systems have been implemented on bridges with various sensors, distributed tiny fatigue cracks could not be identified efficiently and effectively for lack of appropriate approaches. At present, manual inspection is the typical way of bridge crack detection, but it is labor-time-consuming and inaccurate. Besides, it highly relies on prior experiences and subjective judgments of human inspectors. A novel computer-vision-assisted technique is vital to achieving autonomous crack recognition by intelligent algorithms and can be integrated with novel robotics to enhance or replace manual inspection.

Conventional vision-based techniques have been widely employed for structural crack recognition since a decade ago. For example, Abdel-Qader *et al*. (2003) used edge detection filters to detect bridge cracks. Jahanshahi *et al*. (2009) conducted a survey of various potential image processing algorithms for crack detection and further applied morphological feature extraction filters to detect cracks with different thicknesses (Jahanshahi and Masri 2013, Jahanshahi *et al*. 2013a). A multiple sequential image filtering method was proposed by Nishikawa *et al*. (2012) to recognize structural damages. Lim *et al*. (2014) proposed a Laplacian of Gaussian (LoG) algorithm to detect cracks on bridge decks, and a global crack map was obtained by camera calibration and localization. Haar-like wavelets were also employed to extract features from different color channels of the input image and classify cracks by Yeum and Dyke (2015). Shi *et al*. (2016) proposed a road crack

*Corresponding author, Ph.D.,
E-mail: xyce@hit.edu.cn
† Co-first authors: Equal Contributions

ISSN: 1738-1584 (Print), 1738-1991 (Online)

detection framework based on random decision forests and pre-designed crack descriptors. These methods utilized classical image processing algorithms, including edge detection, threshold segmentation, morphological operations (dilation, erosion, opening, and closing), and image filtering. A few limitations still exist, for example, high dependence on manually selected parameters and filters and lacking generalization under different scenarios.

Recently, artificial intelligence and computer vision have made considerable progress. AlexNet, including a series of convolution layers with different convolutional kernels, was first proposed for deep feature extraction and won the ImageNet recognition competition (Krizhevsky *et al*. 2012), which significantly reduced the image classification error rate to 11% and was superior to other conventional methods. Afterward, computer vision has entered the era of deep learning. The wide application of convolutional neural networks (CNNs) is one of the most critical factors for the success of deep learning. CNN can extract hierarchical features from the original input image by multiple layers with different operations, obtain context semantics and enable scene understanding. GoogLeNet was then designed with the Inception module to simultaneously reduce network parameters and improve efficiency (Szegedy *et al*. 2016). ResNet used a residual connection module to improve the feature extraction capability and suppress gradient vanishing by establishing shortcuts between nonadjacent layers (He *et al*. 2016). Furthermore, DenseNet added shortcuts between all convolution layers (Huang *et al*. 2017). U-net was proposed in 2015 using a downsampling encoder and a sequential upsampling decoder with short cuts to output the pixel-level classification map successfully (Ronneberger *et al*. 2015). Fully connection network (FCN) was proposed for pixel-wise dense prediction using deconvolution layers and multiscale fusion with different strides (Long *et al*. 2015). Then, SegN*et al*so utilized a similar encoder-decoder paradigm to classify pixels using the transferred pool indices from its encoder to produce a sparse feature map (Badrinarayanan *et al*. 2017). Afterward, DeepLab designed atrous convolution (dilated convolution) for semantic segmentation to suppress feature loss in downsampling and pooling layers and expand the receptive field without increasing the size of convolutional kernels (Chen *et al*. 2017b).

Benefiting from the consistent improvements of CNN, deep-learning-based methods have been gradually developed for structural crack recognition, which mainly includes image classification, object detection, and semantic segmentation. A series of literature reviews about vision-based crack detection have been systematically summarized (Bao and Li 2020, Dong and Catbas 2020, Bao *et al*. 2019, Spencer *et al*. 2019). For classification-based methods, the entire original image containing cracks is divided into a large number of small patches, and these patches are then individually classified as the crack and non-crack patches. For example, Soukup and Huber-Mörk (2014) trained a CNN classifier using photometric stereo images of metal surfaces to detect rail defects. Makantasis *et al*. (2015) and Zhang *et al*. (2016) established a multilayer CNN classification model to detect tunnel and road cracks, respectively. Cha *et al*. (2017) employed a CNN classification model with a Softmax classifier to recognize concrete cracks. Chen *et al*. (2017a, Chen and Jahanshahi 2018) used combinations of CNN and Naïve Bayes data fusion to detect cracks on nuclear buildings. Bilateral filtering and adaptive thresholding were also used as additional postprocessing to enhance patch-based classification models (Fan *et al*. 2019). For surface damage identification on steel structures, deep learning models based on restricted Boltzmann machine (Xu *et al*. 2018) and deep fusion CNN networks (Xu *et al*. 2019a) were successively established for crack recognition in steel box girders. Kong and Li (2018) proposed a vision-based fatigue crack detection method of steel structures using video feature tracking. For these classification methods, the entire original image needs to be divided into many patches relatively smaller to produce a crack prediction map, which has limitations of existing crack gaps and unclear crack boundaries with low computational efficiency.

In addition to visible images, the depth information was also considered as inputs to enhance damage detection (Yang *et al*. 2018, Beckman *et al*. 2019), and depth sensors were used to detect and quantify pavement defects (Jahanshahi *et al*. 2013b). CNN is also used for post-earthquake reconnaissance to classify the building conditions (Yeum *et al*. 2018). Auto-encoder and one-class support vector machine were combined to recognize structural damages (Wang and Cha 2020). Object detection-based damage localization has been investigated for multi-type damage region detection, multi-type seismic damage localization, and crack recognition (Cha *et al*. 2018, Xu *et al*. 2019b, Kang *et al*. 2020). Moreover, transformed deep CNN models for android applications were employed to perform real-time crack detection and assessment using unmanned aerial devices and smartphones (Jiang and Zhang 2020).

Furthermore, semantic segmentation methods have been increasingly investigated to classify the pixel-level category of the input image. The goal of semantic segmentation is to classify each pixel of the target image, while image classification generates a classification label for an individual patch. Therefore, semantic segmentation can produce better pixel-level accuracy than classification-based methods. Recently, a series of semantic segmentation models have been developed for pixel-level crack recognition tasks. Zhang *et al*. (2019) applied SegNet to detect concrete cracks. Li *et al*. (2019) employed DenseNet as the base model for crack detection on concrete structures. SDDNet was proposed by Choi and Cha (2020) for crack segmentation similar to DenseNet. Ni *et al*. (2019) employed GoogleNet to recognize structural cracks. An encoder-decoder model was proposed for pixel-level crack recognition to obtain geometric features of cracks (Bang *et al*. 2019). Huang *et al*. (2018) established a defect-recognition framework in tunnels using a fully convolutional network (FCN) for semantic segmentation. A CNN for crack segmentation was established based on TernausNet applying transfer learning to recognize cracks from planking patterns (Benz *et al*. 2019). A deep learning-based crack detection-segmentation integrated algorithm was developed to detect and segment the fatigue cracks in

U-rib-to-deck weld seams (Wang *et al*. 2020). Afterward, a feature extraction procedure based on image processing is explored to obtain the morphological features involving the crack area, length, and width. Li *et al*. (2020) further proposed a postprocessing algorithm for concrete cracks after semantic segmentation.

As conventional CNNs could only perform feature extraction in small local regions and may not model the long-range dependencies, the attention mechanism was first proposed to draw global dependencies of inputs in machine translation (Vaswani *et al*. 2017). Afterward, the self-attention mechanism was extended to scene segmentation, and the spatial attention and channel attention modules were designed to model the contextual dependencies in spatial dimension and the interdependencies between channels (Fu *et al*. 2019). Recently, a series of attention-based crack segmentation models have been established aggregating the self-attention mechanism onto the feature maps after convolution. For example, the attention mechanism was applied on the outputs from each encoding layer in skip connection of U-net, and the full attention U-net was proposed for crack semantic segmentation (Lin *et al*. 2019). A spatial-channel hierarchical deep learning network was developed for crack detection comprising feature extraction, spatial attention, and pixel attention modules (Pan *et al*. 2020). An encoder-decoder architecture named CrackResAttentionNet was proposed for pavement crack detection by connecting the position attention and channel attention modules after each encoder (Wan *et al*. 2021).

Generally, the above brief review shows that the main improvement of using deep CNN architectures for structural crack recognition usually lies in skillful manipulations of novel architecture based on hierarchical feature extraction, multilayer feature fusion, concatenation, and combination. It also implies that the underlying fundamental building block of these deep CNN models remains the same mathematical convolution operation. Therefore, it is promising to modify the convolution operation and enhance the deep model with more powerful feature extraction and fusion capacities. For example, the spiking neuron model (Gerstner and Kistler 2002) and pulsed neural networks (VanRullen *et al*. 2005) have been proposed to model the spiking behaviors of neurons and explore the bio-plausible formulations with neuronal plasticity. Biological voltages of neurons could be persistently strengthening or weakening according to recent memories of learning activities. Shao and Zhou proposed a novel neuron model, Flexible Transmitter, to simulate this neural learning behavior by a two-variable two-valued function (Zhang and Zhou 2020).

On the other hand, identifying pixels of minor objects such as tiny cracks from a large image with a complicated background is still challenging. First, the vast disparity between pixel numbers of crack and background brings the extreme category imbalance problem. Second, massive complex background disturbances significantly cause false alarms. Furthermore, the large aspect ratio of crack brings inevitable challenges for feature extraction and recombination using conventional CNN-based models with small square kernels. The most significant difference among steel, concrete, and pavement cracks is that steel cracks are frequently very small in width when initiated and challenging to detect at an early stage. For example, the steel crack opening was quantified within 2 pixels and less than 0.5 mm (Kong and Li 2018); the extracted widths of U-rib-to-deck fatigue cracks were reported less than 0.7 mm (Wang *et al*. 2020); the reported concrete crack widths were within 1 mm (Nishikawa *et al*. 2012); the mean width of pavement crack increased from 3.62 mm for good to 3.98 mm for intermediate and 4.06 mm for poor conditions (Mokhtari *et al*. 2017). In addition, the investigated fatigue crack images were in high resolution and wide range and acquired in steel box girders with poor illumination, including many structural edges and disturbances in the complex background. It brought particular challenges different from other research when the crack images were often shot at a close distance and in good illumination, and the background was relatively simple. Therefore, developing a novel neuron model considering specific characteristics of tiny objects is supposed to benefit the recognition from large images with complex backgrounds.

In this study, a novel Self-Attention-Self-Adaption (SASA) neuron and crack random elastic deformation (CRED) algorithm for image augmentation are proposed to solve the above problems. Both of them are integrated into a modified U-net framework for minor fatigue crack segmentation. The remainder of this paper is arranged as follows. Section 2 describes the proposed methodology, including (1) a new Self-Attention-Self-Adaption (SASA) neuron model as the basic building block, (2) the integration of SASA neuron into the U-net framework, and (3) the CRED algorithm for image augmentation to enrich the morphology diversity of tiny cracks. Section 3 introduces the implementation details of data arrangement and hyperparameter settings. The training and test results of the modified U-net for tiny crack segmentation are shown in Section 4. Ablation experiments are conducted to demonstrate further the effectiveness of the proposed SASA neuron and CRED algorithm for the modified U-net. In addition, three validation tests of using the SASA neuron for classification under Gaussian blur, motion blur, and random blocking scenarios are also performed. Section 5 concludes the paper.

2. Methodology

2.1 Methodology overview

This study proposes a modified U-net model for tiny crack segmentation comprising a novel convolution operation to overcome the vast challenges of recognizing tiny cracks from high-resolution images with complicated disturbances. The proposed method utilizes a newly designed Self-Attention-Self-Adaption (SASA) neuron model and crack random elastic deformation (CRED) algorithm. The proposed SASA neuron is established to focus on the regions of interest and achieve deeper feature extraction via a subnet. In addition to the underlying modification of the fundamental calculation block, a new data augmentation technique (CRED) is also proposed to

enrich the sample diversities and irregular shapes, which benefits the tiny crack segmentation from the data side. Details are introduced in the following subsections.

2.2 Self-Attention-Self-Adaption neuron

The conventional neuron computing model can be expressed as

$$x_j^{l+1} = \sigma\left(\sum_i^{N^l} w_{ij}^{l,l+1} x_i^l + b_j^{l+1}\right) \quad (1)$$

where x_i^l denotes the ith neuron in the lth layer, x_j^{l+1} denotes the jth neuron in the (l+1)th layer, N^l denotes the number of neurons in the lth layer, $w_{ij}^{l,l+1}$ denotes the connecting weight between x_i^l and x_j^{l+1}, b_j^{l+1} denotes the individual bias associated with x_j^{l+1}, and σ denotes the nonlinear activation function (e.g., ReLU).

It has been reported that a class of calcium-mediated dendritic action potentials (dCaAPs) have been discovered on pyramidal neurons of the human cerebral cortex (Gidon *et al*. 2020). The newfound dCaAPs only respond for threshold-level stimuli, which is different from typical all-or-none action potentials. Therefore, an individual pyramidal neuron with dCaAPs is supposed to possess a more powerful computing capacity than conventional neurons. Inspired by Gidon *et al*. (2020), a novel Self-Attention-Self-Adaption (SASA) neuron computing model is proposed in this study, which is shown in Fig. 1 and can be expressed as

$$x_j^{l+1} = \sigma\left(\sum_i^{N} w_{ij}^{l,l+1} x_i^l + b_j^{l+1} + m_j^{l+1}\left(\boldsymbol{\alpha}^l \cdot \boldsymbol{X}^l, \theta_j^{l+1}\right)\right) \quad (2)$$

where $\boldsymbol{X}^l$ denotes the neurons in the previous lth layer, $\boldsymbol{\alpha}^l$ denotes the significance vector in the Self-Attention module, and m_j^{l+1} denotes the subnet of multilayer perceptron in the Self-Adaption module associated with the jth neuron in the (l+1) layer and parameterized with θ_j^{l+1}. Details are explained below.

The proposed SASA neuron is designed based on the typical neuron from Eq. (1), and two functional modules (i.e., Self-Attention and Self-Adaption) are added. The first Self-Attention module applies softmax and gate operations to obtain the attention vector. It enables the neuron to focus on the most significant receptive fields when processing large-scale feature maps. The second Self-Adaption module consists of a multiplayer perceptron subnet and achieves deeper feature extraction inside a single neuron. Details are explained as follows.

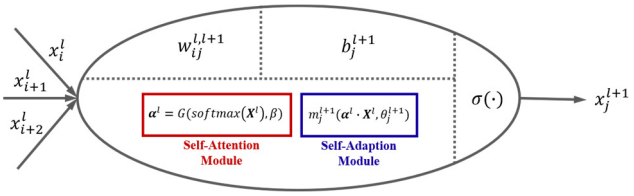


Fig. 1 Schematic of the proposed Self-Attention-Self-Adaption neuron computing model

The proposed Self-Attention module focuses on the saliency of interior neurons inside one layer. It does not introduce additional trainable parameters, which is the most significant difference from the self-attention mechanism (Vaswani *et al*. 2017). The proposed Self-Attention module comprises a gate function G and a nested softmax operation using neurons in the previous layer $\boldsymbol{X}^l = \{x_i^l | i = 1, \dots, N^l\}$ as inputs. The softmax function is defined as usual

$$softmax(\boldsymbol{X}^l)_i = \frac{e^{x_i}}{\sum_j^{N^l} e^{x_j}}; i = 1, \dots, N^l \quad (3)$$

which normalizes elements of $\boldsymbol{X}^l$ between 0-1 and adds up to 1. The gate function G reserves the top β elements and assigns the others to zero.

The output vector $\boldsymbol{\alpha}^l$ represents different significances of corresponding elements in $\boldsymbol{X}^l$ (shown in Fig. 2). Only the top β elements are retained, and thus additional computation costs can be controlled

$$\boldsymbol{\alpha}^l = G(softmax(\boldsymbol{X}^l), \beta) \quad (4)$$

The Self-Adaption module is designed as a subnet of multilayer perceptron m_j^{l+1} shown in Fig. 3 and implemented by a standard neural network with k equal hidden layers. The nonlinear activation function inside the subnet uses ReLU as well. The interior subnet structure is controlled by the number of hidden layers k and the number of neurons h in each hidden layer. For the consistency with the exterior neuron network, the number of neurons h in the hidden layer of the subnet is set proportionate to the number of neurons N^{l+1} in the current layer with a default coefficient γ. The Self-Adaption module takes the dot

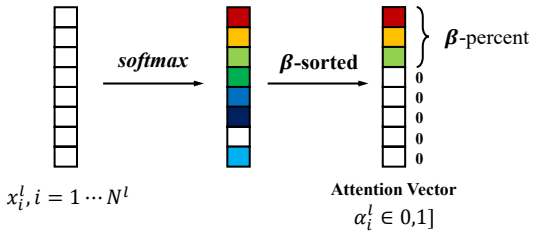


Fig. 2 Schematic of the Self-Attention module by softmax and gate functions

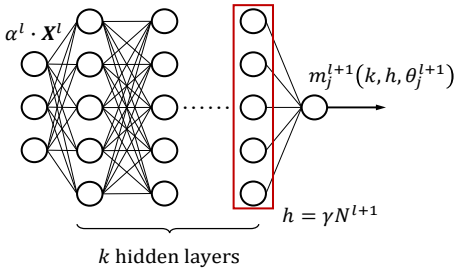


Fig. 3 Schematic of the Self-Adaption module: a multilayer perceptron subnet

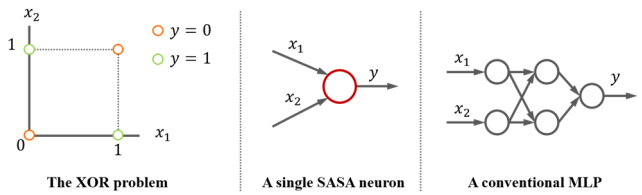


Fig. 4 Illustration of solving XOR problem by one SASA neuron or a conventional neuron network with two hidden layers

product between the output significance vector of the Self-Attention module $\boldsymbol{\alpha}^l$ and neurons $\boldsymbol{X}^l$ in the previous layer as input. θ_j^{l+1} represents parameters inside the subnet to be trained.

Compared with the conventional neuron computing model, $m_j^{l+1}(\boldsymbol{\alpha}^l \cdot \boldsymbol{X}^l, \theta_j^{l+1}))$ both applies attention mechanism to the input features and enhances deeper feature extraction. For example, a single SASA neuron can solve the linearly inseparable "exclusive or" (XOR) problem, which requires at least two hidden layers for a conventional neural network (shown in Fig. 4). Therefore, using the proposed SASA neuron can improve the feature extraction ability on large-scale images and classify linearly inseparable inputs with only a few layers instead of very deep layers in conventional neural networks.

2.3 Integration of SASA neuron in U-net

The proposed SASA model in Section 2.1 is in the form of neuron computing. For the CNN architecture, the convolution operation is modified following Eq. (2) as

$$y_{i'j'd'} = \sigma\left[b_{d'} + \sum_{i=1}^{H}\sum_{j=1}^{W}\sum_{d=1}^{D}\left(f_{ijd} \times x_{i'+i-1,j'+j-1,d,d'} + m_{\theta_{d'}}(\alpha_{ijd} \times x_{i'+i-1,j'+j-1,d,d'})\right)\right] \quad (5)$$

where x, f, y denote the input image, convolutional filter, and output feature map, respectively. H, W, and D denote the height, width, and channel of the convolutional filter. d' represents the index for the number of filters. $m_{\theta_{d'}}$ denotes that one group of convolutional filters sharing the same subnet, i.e., the subnet number is equal to the number of filters. $\theta_{d'}$ denotes the parameters inside the subnet. σ denotes the nonlinear activation function, and ReLU is used in this study. Except for the convolution (Conv) operation is modified using the SASA model, batch normalization (BN) and pooling are set as usual. Therefore, The SASA neuron model allows for "plug and play" of arbitrary conventional neural networks. In this study, U-net is selected as the baseline model for semantic segmentation shown in Fig. 5.

Details about the network layouts and parameters are shown in Table 1. Skip connection and 2x upsampling are used in the decoder stages. For example, in the first upsampling stage Up1, feature maps before and after the downsampling layer are concatenated in the channel dimension, and the channel size of the concatenated feature

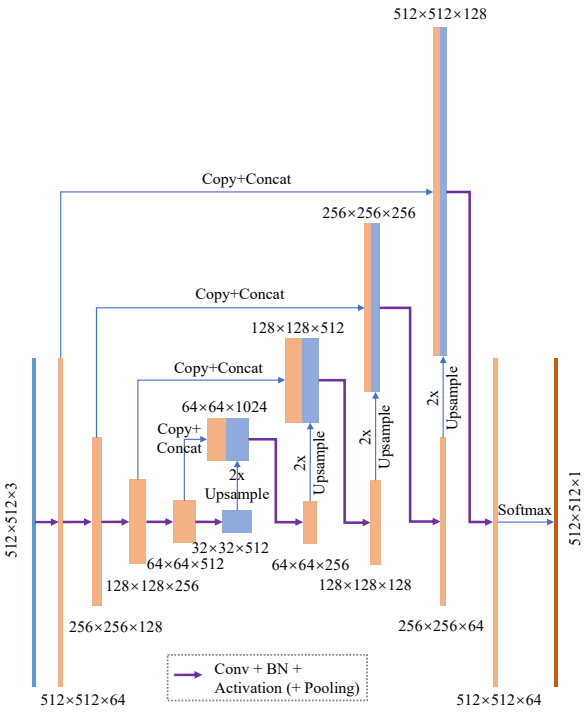


Fig. 5 Schematic of the baseline U-net model

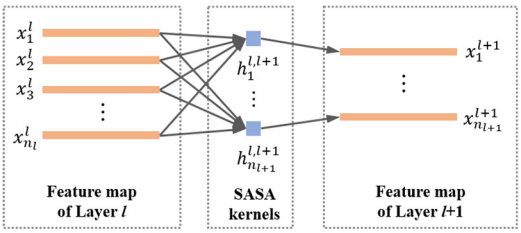


Fig. 6 Implementation of SASA neurons in U-net

maps is halved using 1 × 1 convolution. The proposed SASA neuron is embedded in U-net by replacing regular convolutional kernels with SASA kernels as Eq. (5), which is shown in Fig. 6.

The binary cross-entropy loss function is used as

$$L(p,q) = -\frac{1}{nHW}\sum_{i=1}^{n}\sum_{j=1}^{H\times W}\begin{bmatrix} w_{pos} \cdot q_j^i \cdot ln(p_j^i) \\ +(1-q_j^i) \cdot ln(1-p_j^i) \end{bmatrix} \quad (6)$$

where H and W denote the output size and are equal to 512, $p \in (0,1)$ and $q \in \{0,1\}$ are predicted and ground-truth pixel values, respectively. The positive weighting coefficient w_{pos} is supposed to affect the precision and recall of the trained model. If w_{pos} is larger, better recall and worse precision can be obtained, and vice versa. The Root Mean Square Prop (RMSprop) algorithm (Mukkamala and Hein 2017) is employed for training as

$$v_t = \rho v_{t-1} + (1-\rho) \times \left(\frac{\partial L}{\partial \omega}\right)^2$$
$$\Delta\omega_t = -\frac{\eta}{\sqrt{v_t + \varepsilon}} \times \frac{\partial L}{\partial \omega}, \omega_{t+1} = \omega_t + \Delta\omega_t \quad (7)$$

Table 1 Details about the network layouts and parameters of the modified U-net

Stage	Layer	Input size (W×H×C)	Output size (W×H×C)	Kernel	Stride	Total padding
	Input	512 × 512 × 3	512 × 512 × 64	3 × 3	1	2
Encoder	Down1	512 × 512 × 64	256 × 256 × 128	3 × 3	2	1
	Down2	256 × 256 × 128	128 × 128 × 256	3 × 3	2	1
	Down3	128 × 128 × 256	64 × 64 × 512	3 × 3	2	1
	Down4	64 × 64 × 512	32 × 32 × 512	3 × 3	2	1
Decoder	Up1	64 × 64 × 1024	64 × 64 × 256	3 × 3	1	2
	Up2	128 × 128 × 512	128 × 128 × 128	3 × 3	1	2
	Up3	256 × 256 × 256	256 × 256 × 64	3 × 3	1	2
	Up4	512 × 512 × 128	512 × 512 × 64	3 × 3	1	2
Pixel classification	Softmax	512 × 512 × 64	512 × 512 × 1	3 × 3	1	2

where η denotes the initial learning rate, v_t denotes the exponential average of squares of gradients, ρ denotes the decay rate, and ε is a small positive constant.

2.4 Crack random elastic deformation algorithm for image augmentation

Most image semantic segmentation tasks apply data augmentation to enhance the diversity of training data and improve the model robustness, including random rotation, translation, flipping, resize, cropping, illumination variation, and adding random noise. In this study, tiny fatigue cracks in steel box girders are typically in irregular morphological shapes, which is the most significant difference from the ordinary objects in other tasks. Therefore, a CRED algorithm is proposed in this study to obtain extra crack examples and enrich the diversity of crack morphology.

The proposed CRED algorithm is modified from the original elastic distortion method (Simard *et al*. 2003) for data augmentation to enrich the diversity of crack morphology. The original elastic distortion method (Simard *et al*. 2003) was created by generating random displacement fields, convolving with a Gaussian kernel, and multiplying by a scaling factor to control the deformation intensity. It was applied for image classification to recognize the MNIST digits, in which the distortion was only conducted on the input image, and was further widely used in the semantic segmentation task of medical images (Castro *et al*. 2018, Bloice *et al*. 2019). A python package *torchIO* was developed for the 3D medical image augmentation. In this study, the proposed CRED algorithm was modified by selecting grid control nodes in the original image, applying random offsets on these control nodes, obtaining offsets of other pixels by two-dimensional cubic spline interpolation, and calculating sub-pixels by bilinear interpolation with the surrounding pixels. It was simultaneously performed the corresponding random deformation on the input image and the segmentation mask.

The task of crack segmentation is to classify all the crack pixels within an image I. The binary segmentation mask $K \in \mathbb{R}^{h \times w}$ comprises zeros and ones, denoting background and crack pixels, respectively. The CRED function $W(\cdot)$ is simultaneously applied to each pair of training sample and binary mask as

$$\begin{aligned} \tilde{I}(x,y) &= W(I) = I(x+\Delta x, y+\Delta y) \\ \tilde{K}(x,y) &= W(K) = K(x+\Delta x, y+\Delta y) \end{aligned} \tag{8}$$

where the CRED function $W(\cdot)$ attaches an offset $(\Delta x, \Delta y)$ to each pixel in the original image I and is calculated from the following steps:

Step 1: $c \times c$ control nodes are equidistantly set in the original image. Random offsets $(\Delta x, \Delta y)$ of these control nodes are given following a uniform distribution in the range of $-\max(h,w)/20 < \Delta x, \Delta y < \max(h,w)/20$.

Step 2: Offsets of other pixels are interpolated from the control nodes using two-dimensional cubic spline interpolation.

Step 3: The sub-pixel value of $I(x+\Delta x, y+\Delta y)$ in Eq. (8) is calculated by bilinear interpolation with the surrounding pixel as

$$\begin{aligned} I(x+\Delta x, y+\Delta y) = &[x_2 - x - \Delta x \quad x + \Delta x - x_1] \\ &\begin{bmatrix} I(x_1,y_1) & I(x_1,y_2) \\ I(x_2,y_1) & I(x_2,y_2) \end{bmatrix} \begin{bmatrix} y_2 - y - \Delta y \\ y + \Delta y - y_1 \end{bmatrix} \end{aligned} \tag{9}$$

where x_1, x_2 and y_1, y_2 are adjacent bounding integers with $x_1 < x + \Delta x < x_2, y_1 < y + \Delta y < y_2$. Fig. 7 shows the schematic of the proposed CRED algorithm on mesh grids. Red points represent the control nodes $(c = 7)$, and

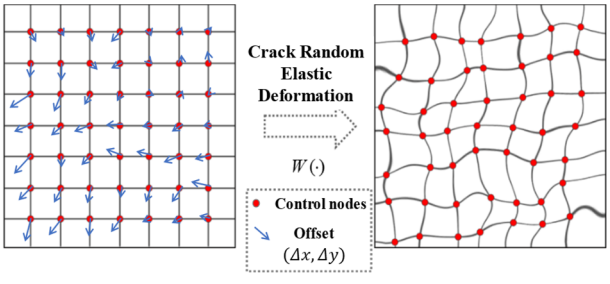


Fig. 7 Schematic of crack random elastic deformation on mesh grids

Fig. 8 A few representative original images and binary annotations for tiny cracks

the blue arrays represent the corresponding offsets of control nodes.

3. Implementation details

3.1 Image dataset

200 original images containing tiny cracks of steel box girder with a resolution of 4928 × 3264 are used in this study from the 1st International Project Competition of Structural Health Monitoring (IPC-SHM 2020, Bao *et al*. 2021). The training and test datasets use the first 160 images (No. 001-160) and the rest 40 images (No. 161-200), respectively. A few representative original images and binary annotations are shown in Fig. 8. They are cut into patches with a size of 512 × 512 pixels and an overlap of 256 in width and height. Consequently, a total of 26605 background and 3315 crack patches are obtained for training. The number of patches with only background pixels is approximately eight times that containing crack pixels. Although numerous patches are obtained, crack patches only take a proportion of 11% and crack pixels only account for 0.11%, which brings a highly imbalanced data problem. Only half of the background patches (13302) are randomly selected for training to handle this problem.

A series of conventional data augmentation operations are conducted, including random horizontal and vertical translations, flips, crops, rotation, resize, illumination variation, adding random noises, and Gaussian blurring. These operations were used as online methods during the training process. Besides, the CRED algorithm is applied to

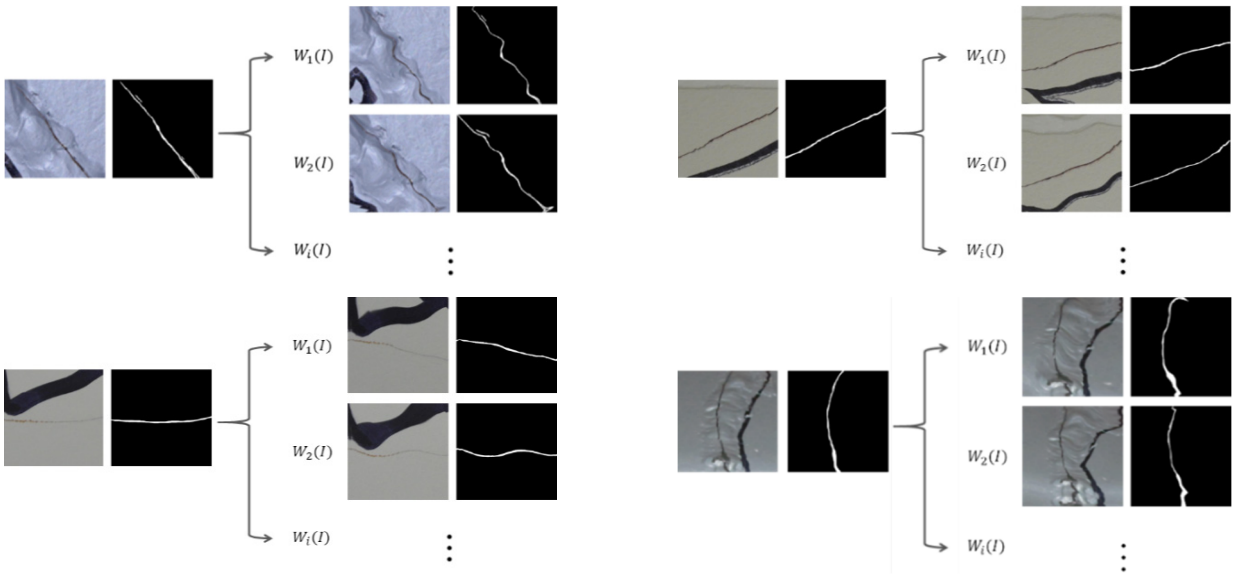


Fig. 9 Transformed crack patches and the corresponding binary labels by CRED

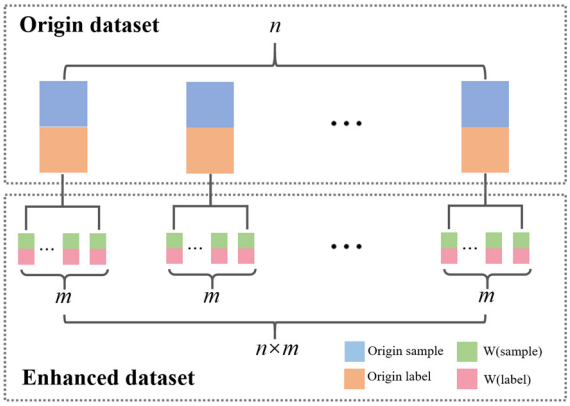


Fig. 10 Augmented training dataset generated from n original examples and m times of CRED

enrich the morphological diversity of cracks in the training set, as shown in Fig. 9.

The motivation of developing CRED is to increase the diversity of crack morphology and increase the sample space. Cracks after CRED may have considerable differences from the original ones. Therefore, the image dataset includes the original cracks and CRED-augmented cracks. For each training sample, CRED is applied for m times. Thus, the final training dataset includes the original n examples and $n \times m$ transformed examples after CRED are shown in Fig. 10. Other than image augmentation before training (offline), the CRED module is also employed during training (online), which varies training samples in every epoch to enhance the model robustness. From this aspect, the proposed CRED algorithm can enrich the sample space of crack pixels. The core code of image deformation function is released at *https://github.com/ZhaoJinHA/CRED*. Python packages, numpy, scipy, and cv2, are used in the implementation process of the proposed CRED method.

3.2 Hyper-parameter setting

In this study, hyperparameters include parameters of the SASA neuron model (the gate coefficient β, the number of hidden layers k, the ratio of the number of neurons in each hidden layer γ), m times of CRED for each training sample, training hyperparameters (the initial learning rate η, decay rate ρ, small positive constant ε, and mini-batch size n), positive weighting coefficient w_{pos} in the cross-entropy loss function, and the weight decay λ.

Only the top β elements are retained in the Self-Attention module, and others are set to zeros. One assumption is that layers with more neurons also need wider hidden layers in the interior subnet for feature extraction. Therefore, the number of neurons in the hidden layer of the subnet is set proportionate to the number of neurons with a coefficient γ. Larger β and γ can establish a more complex model, whereas the computational cost will significantly increase. An appropriate configuration of β = 0.2, γ = 0.2 is adopted after a few trials. Another common hypothesis is that a 2-hidden-layer fully connected neural network with certain width could simulate an arbitrary continuous function. In this study, k is set as 2 considering the tradeoff between the capacity of approximating arbitrary nonlinear functions and the computational cost of the interior subnet.

Generally, it is frequently tedious to configure the optimal hyperparameters for model training. The initial learning rate, the decay rate of the learning rate, and the weight decay are set up with significant variations in the literature on CNN-based crack detection. The setups of these hyperparameters in this study are selected according to practical recommendations for gradient-based training algorithms (Bengio 2012). These hyperparameter settings are arranged as follows. CRED transformation is performed 4 times (m = 4). An initial learning rate of 0.01 is chosen with a decreasing rate of 0.5 after each training epoch. In the RMSprop algorithm, $\rho = 0.9$, $\varepsilon = 10^{-8}$. The weight decay λ is set as $5e^{-3}$. The number of total training epochs is set to 10 by a few trials to ensure that the training process generally reaches convergence. A mini-batch size of 8 is used due to the memory size of the graphic processing unit. The positive weighting coefficient w_{pos} is set to 2 considering the unbalanced number of crack samples.

It is definitely possible to optimize hyperparameters using grid search and random search techniques. If the hyperparameters and structures are optimal, it is likely to obtain better results, although this process is time-consuming. For example, the grid search is simply an exhaustive search through all the combinations, which is unfortunately exponential in the number of hyper-

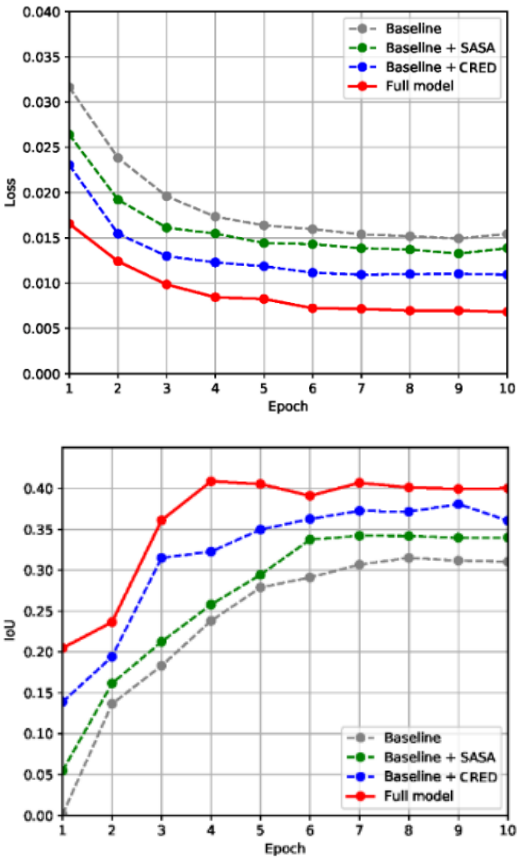


Fig. 11 Training loss and test IoU of the proposed approach

Table 2 Evaluation metrics of average precision, recall, and IoU for different models on all the test images

Model	Average precision	Average recall	Average IoU	IoU promotion
Baseline model	0.531	0.521	0.315	-
Baseline + SASA	0.539	0.565	0.342	0.027
Baseline + CRED	0.556	0.623	0.381	0.066
Full model	0.569	0.658	0.409	0.094

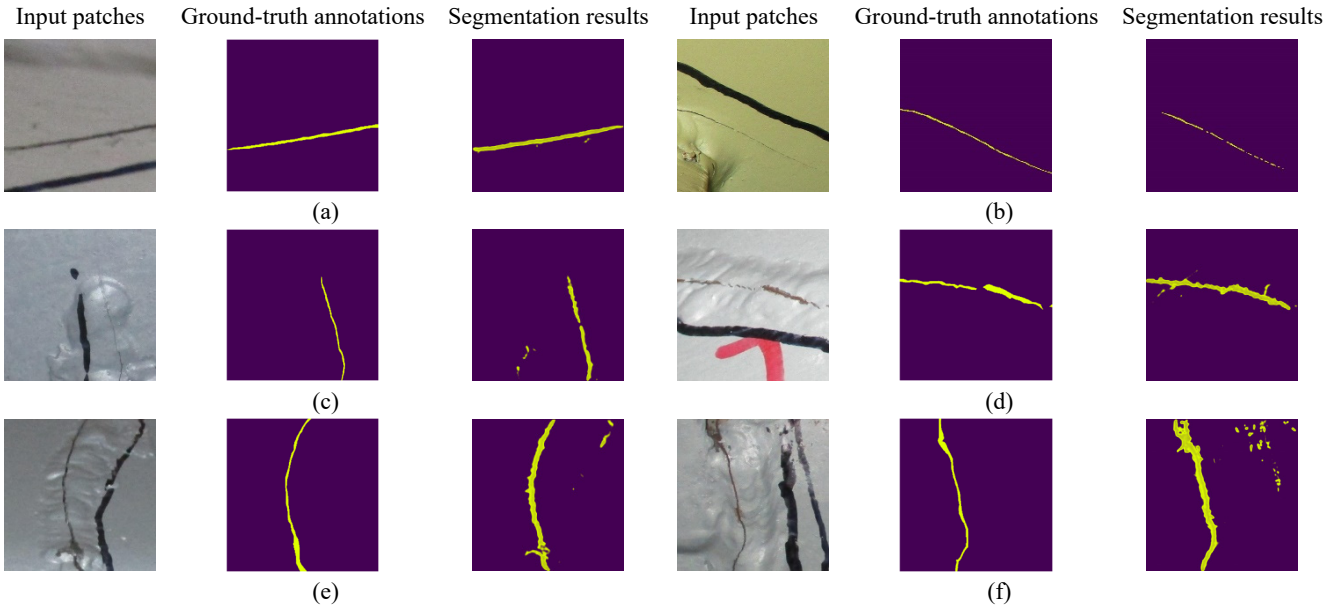


Fig. 12 Some recognition results of 512 × 512 patches containing crack segments

parameters. In this work, we focus on the original contributions of proposing SASA and CRED modules for CNN-based semantic segmentation and therefore do not evaluate different trials to find the best hyperparameters.

4. Results and discussions

4.1 Training curves and evaluation metrics

In this section, several experiments are conducted to demonstrate the performance of the modified U-net fusion with the proposed SASA neuron and CRED algorithm for tiny crack segmentation. Ablation experiments are employed to demonstrate the effectiveness of the proposed method: applying both the SASA and CRED modules (Full model), the individual SASA module (Baseline + SASA), the individual CRED module (Baseline + CRED), and none of them (Baseline Original U-net). Note that online data augmentation methods mentioned in Section 3.1 are used to process all the above models. All hyperparameters of the network structure, model training, and image datasets are the same in the ablation experiments for fair comparisons, as described in Section 3.2. 40 original images with a resolution of 4928 × 3264 are used for the test. Like the training process, 512 × 512 sub-images are first cut from these test images, and the predicted instances are then

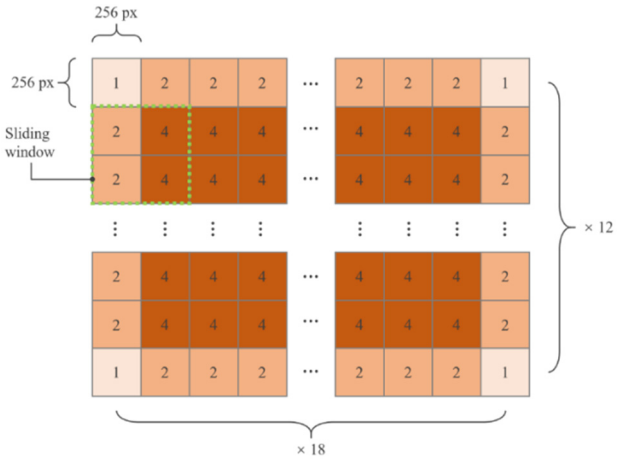


Fig. 13 Illustration of handling conflicting predictions for overlapping patches in an entire image

assembled to the original size. An overlap of 256 between adjacent sliding windows is used to avoid missing individual cracks.

The curves of training loss descending and test IoU increasing between the predicted and ground-truth crack labels are shown in Fig. 11, respectively. The corresponding average IoU is calculated under these scenarios, respectively. It indicates that the training process generally

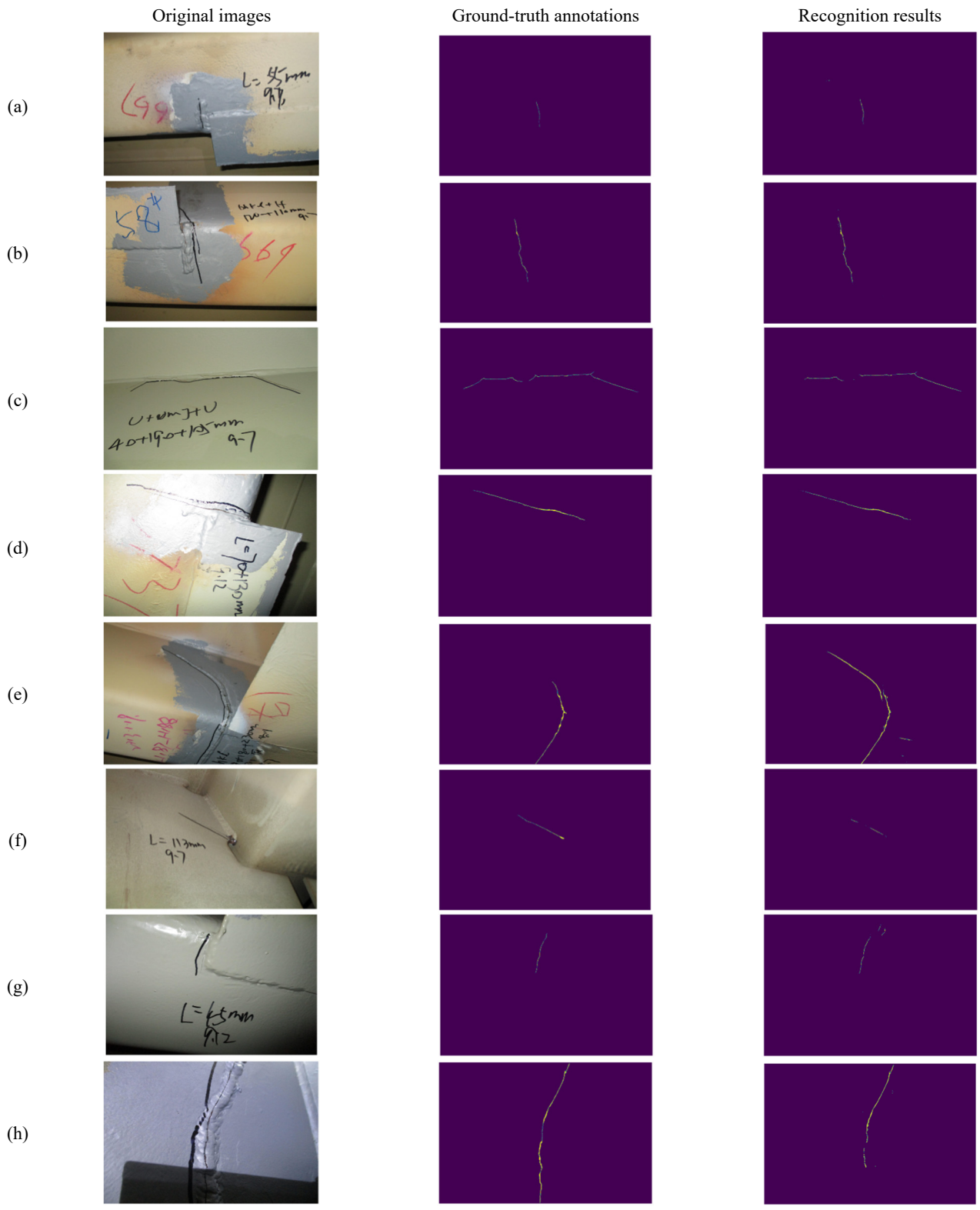


Fig. 14 Some representative crack segmentation results of the whole image

reaches convergence and converges after ten epochs and that the integration of SASA and CRED modules significantly promotes the original U-net model.

Table 2 shows evaluation metrics of average precision, recall, and IoU for different models on all the test images. It indicates that the full model using both SASA and CRED modules outperforms the baseline U-net model by 29.8% (IoU increases from 0.315 to 0.409). The ablation experiments show that using CRED could achieve a significant IoU promotion over the baseline model from 0.315 to 0.381. It demonstrates the necessity of the implementation of CRED. One possible explanation is that

the relationships between crack pixels were better learned with CRED-enhanced images. In addition, promotions in average IoU of individually utilizing the SASA and CRED module approximately add up to the final promotion of the full model, which indicates that the proposed SASA and CRED modules act on the model and data in different stages of the training process, respectively. It further suggests that the proposed SASA and CRED modules can be linearly combined. Therefore, it is validated that the proposed SASA neuron model and CRED image augmentation algorithm significantly contribute to the pixel-wise tiny crack segmentation.

4.2 Illustration of crack segmentation results

Fig. 12 shows some recognition results of 512 × 512 patches containing crack segments. The left, middle, and right columns represent the input, ground truth annotation, and segmentation results. It shows that the labeled cracks can be generally well identified with a good IoU, and the trained model can distinguish tiny cracks from handwriting scripts, even if there are certain morphological similarities between them. In addition, the recognized crack is somewhat more expansive than the ground truth, leading to a higher recall and a lower precision rate.

A sliding window of 512 × 512 with an overlap of 256 is used to generate patches from the original high-resolution image for prediction, as shown in Fig. 13. A probability map of [0,1] is achieved for each 512 × 512 patch. Moreover, each local region of 256 × 256 in the corner, edge, and core parts is predicted for one, two, and four times, respectively. Therefore, conflicting predictions may exist for these overlapping patches. The average classification probability for each pixel is utilized, and a threshold of 0.5 is used in this study to determine whether it should be classified into crack or background and form the final binary prediction map.

Fig. 14 shows some representative recognition results of the whole images using the proposed approach (full model). The left, middle, and right columns represent the original image, ground truth annotation, and crack segmentation results. Figs. 14(a)-(d) show that most crack pixels can be ideally identified without false alarms. Despite that most tiny crack pixels can be successfully recognized, misrecognition still occurs under various complicated circumstances. Fig. 14(e) indicates that the welding line is recognized as crack because of its crack-like feature, while crack pixels at the top are correctly recognized, although they have been mislabeled as background. Figs. 14(f) and 14(g) show that extremely tiny cracks are difficult to be entirely identified and that a small fraction of background edges are inevitably recognized as cracks. Fig. 14(h) shows that shadow regions and dark illumination conditions also lead to misjudgments of black crack pixels.

4.3 Five-fold cross-validation study

A five-fold cross-validation study is performed to verify the sensitivity of the network to the selected images and ensure the reliable performance of the proposed method. The entire image dataset is divided into five folds, and each group includes 40 images in sequence, namely, No. 001-040 for Fold 1, No. 041-080 for Fold 2, No. 081-120 for Fold 3, No. 121-160 for Fold 4, and No. 161-200 for Fold 5. Each fold is used as the test set, and the other four folds are for training. The overall schematic of the five-fold cross-validation study is shown in Fig. 15. All hyperparameter and network setups are the same in the cross-validation study; the average test precision, recall, and IoU for each model are shown in Table 3. The results show that the proposed method is robust to different setups of training and test images, and the average IoU varies between 0.409 and 0.459. It indicates that the reported results using the proposed method are reliable.

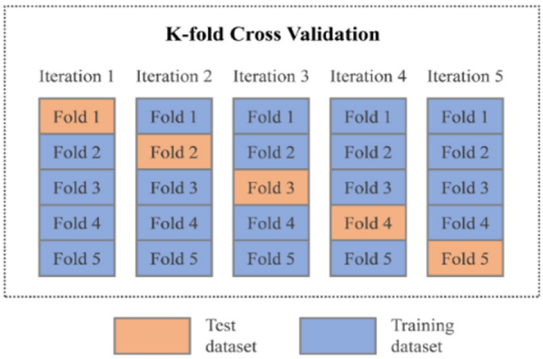


Fig. 15 Overall schematic of cross-validation study

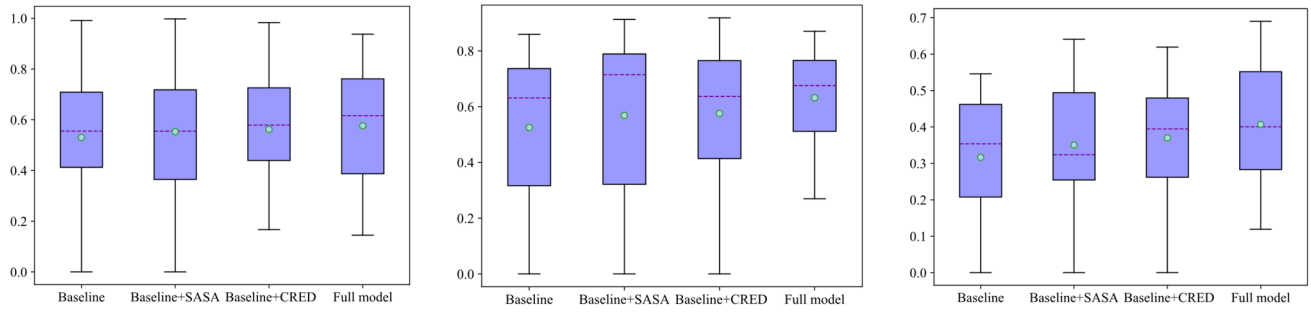
Table 3 Evaluation metrics in the cross-validation study

Iteration	Average precision	Average recall	Average IoU
Iteration 1	0.589	0.732	0.455
Iteration 2	0.525	0.823	0.445
Iteration 3	0.581	0.739	0.459
Iteration 4	0.516	0.668	0.409
Iteration 5	0.569	0.658	0.409

4.4 Demonstration of SASA and CRED modules by comparisons of ablation experiments

The 40 test images with a resolution of 4928 × 3264 are also used to verify model performances in ablation experiments. Using the same evaluation metrics in Table 2, Fig. 16 shows the box plots of precision, recall, and globalIoU (dash line and circle marker represent the median and average values for all the test images, respectively). Comparison results show that the full model using SASA and CRED possesses the best performance of average precision, recall, and global IoU in general. Besides, the diversity (the gap between maximum and minimum) of the full model is also the least, demonstrating the proposed approach has both good recognition accuracy and robustness.

Fig. 17 compares some representative crack segmentation results of the Baseline model, Baseline + SASA model, and Baseline + CRED model. They correspond to the same eight test images in Fig. 14. Without the proposed SASA and CRED modules, the tiny cracks



(a) Precision (b) Recall (c) Global IoU

Fig. 16 Boxplots of evaluation metrics in ablation experiments

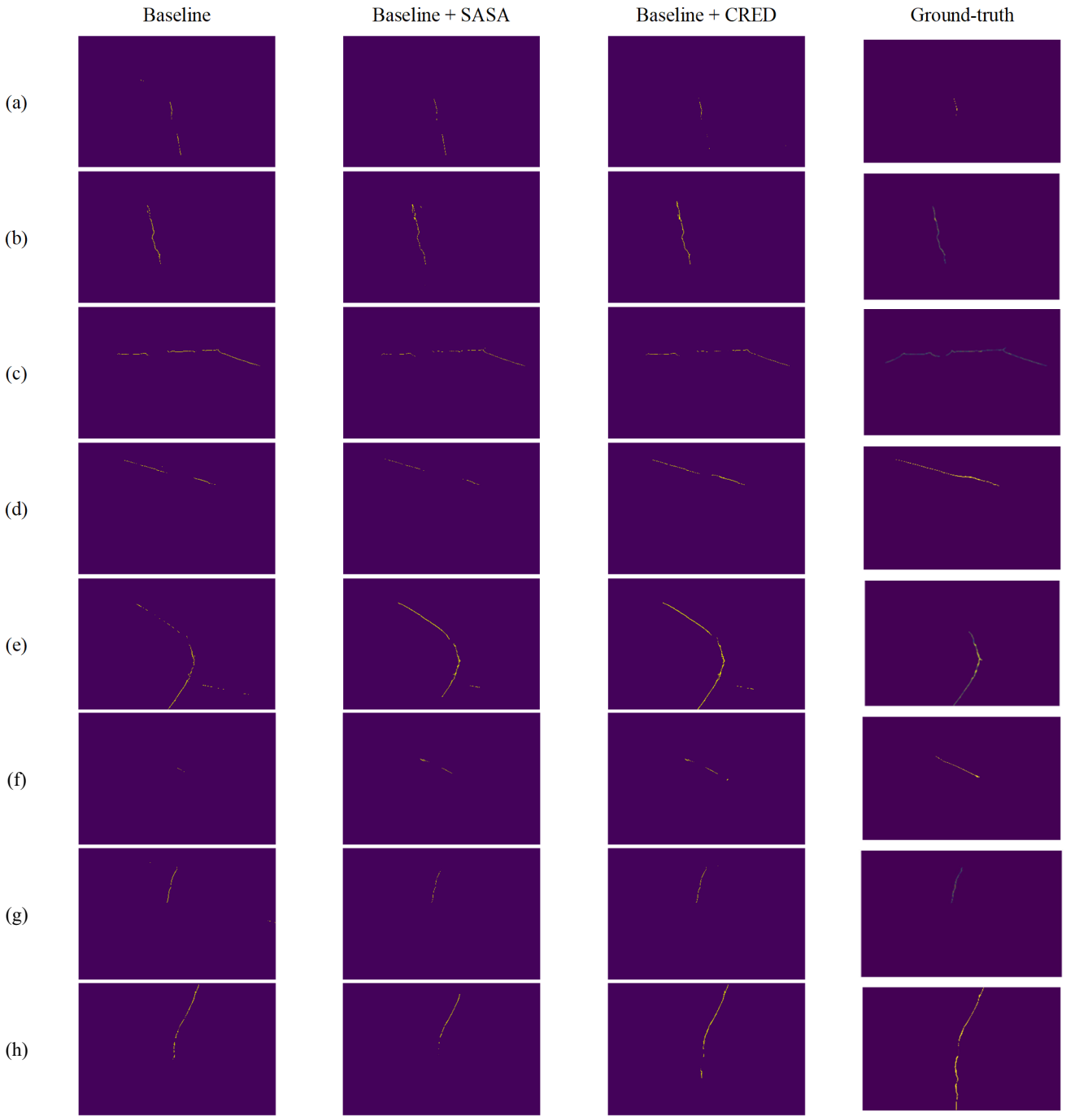


Fig. 17 Comparisons of representative crack segmentation results in ablation experiments

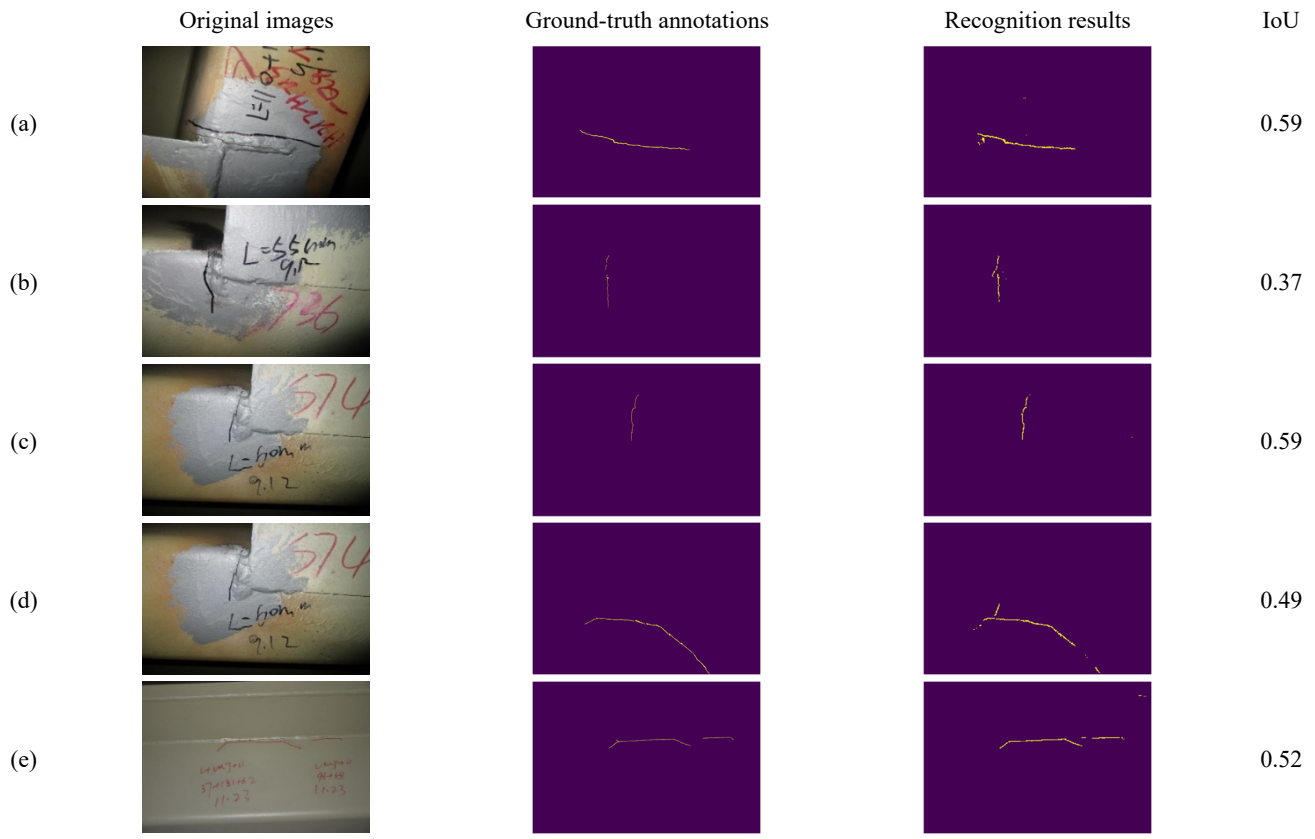


Fig. 18 Some representative test results using input size of 256 × 256

may be identified with more discontinuities, indicating that the proposed SASA neuron and CRED algorithm can assist in focusing on the local regions of interest and enriching the morphological diversity of cracks. Result comparisons show that the proposed method can accomplish pixel-level semantic segmentation of tiny cracks with complex background interferences under different circumstances with the false alarms and crack gaps suppressed.

4.5 Discussions on convolutional kernel and input sizes

Previous studies have shown that the convolutional kernel size affects the accuracy of CNN models to a certain degree. Small kernel size often leads to instability of model accuracy, and large kernel size makes it difficult for the model to converge compared to small ones (Agrawal and Mittal 2020). Besides, large kernels also lead to the explosion of computational costs and are not conducive to increase the model depth. According to the classical network structures of VGG (Simonyan and Zisserman 2014), GoogLeNet (Szegedy *et al*. 2015), and Inception architecture (Szegedy *et al*. 2016), two 3 × 3 kernels perform better than one 5 × 5 kernel with larger model capacity owing to using deeper layers and more nonlinear activation functions, and the number of parameters is also reduced. Therefore, the convolutional kernel size has remained as 3 × 3 in this study.

An additional investigation using the input size of 256 × 256 was conducted. The original image in 5152 × 3864 was downsampled to 768 × 512. A sliding window of 256 × 256 with an overlap of 128 was used to generate sub-image patches. The input size of the modified U-net decreased to 256 × 256 accordingly, and other model parameters remained the same. The model was re-trained using the same hyperparameters as reported in the manuscript. Some representative test results are shown in Fig. 18. The results show that the test average IoU is promoted to 0.46 following this configuration.

5. Conclusions

In this study, a modified U-net is proposed for deep-learning-based image segmentation of tiny fatigue cracks in real-world high-resolution images with complicated background interferences inside steel-box-girder bridges. The main conclusions are summarized as follows:

- A novel Self-Attention-Self-Adaption (SASA) neuron is proposed with two-fold advantages. The Self-Attention module enables the neuron to focus on the most significant receptive fields by softmax and gate functions when facing large-scale latent feature maps. The Self-Adaption module enhances deeper feature extraction using a multilayer perceptron subnet inside the neuron. The SASA neuron model allows for "plug and play" of arbitrary conventional neural networks and is employed in the regular U-net framework for tiny crack segmentation.

- A crack random elastic deformation (CRED) algorithm module is proposed for data augmentation to expand the diversity of limited crack morphologies and styles. Grid-based uniform control nodes are first set on both input images and binary labels, random offsets are then employed on these control nodes, and bilinear interpolation is performed for the rest pixels. Compared with the original images, augmented cracks possess various morphological changes, which is the most significant difference from affine transformation.
- 160 raw images with resolutions of 4928 × 3264 are used for training, and the rest 40 images are for the test. Results show that the proposed method can automatically classify crack pixels from complicated backgrounds at an average IoU of 0.409. A five-fold cross-validation study is performed to verify that the proposed method is robust to different training and test images. Compared with the regular U-net model, the proposed crack segmentation approach integrated with the SASA neuron and CRED data augmentation algorithm gains a 29.8% increase on IoU, which convincingly demonstrates the effectiveness of the proposed approach.

Acknowledgments

Financial support for this study was provided by the National Natural Science Foundation of China [Grant Nos. 52008138, 51638007, U1711265, and 51921006], National Key R&D Program of China [Grant No. 2019YFC1511102], China Postdoctoral Science Foundation [Grant Nos. BX20190102 and 2019M661286], and Heilongjiang Postdoctoral Funding [Grant Nos. LBH-TZ2016 and LBH-Z19064].

References

Abdel-Qader, I., Abudayyeh, O. and Kelly, M.E. (2003), "Analysis of edge-detection techniques for crack identification in bridges", *J. Comput. Civil Eng.*, **17**(4), 255-263. https://doi.org/10.1061/(ASCE)0887-3801(2003)17:4(255)

Agrawal, A. and Mittal, N. (2020), "Using CNN for facial expression recognition: a study of the effects of kernel size and number of filters on accuracy", *Visual Comput.*, **36**(2), 405-412. https://doi.org/10.1007/s00371-019-01630-9

Badrinarayanan, V., Kendall, A. and Cipolla, R. (2017), "Segnet: a deep convolutional encoder-decoder architecture for image segmentation", *IEEE Transact. Pattern Anal. Mach. Intell.*, **39**(12), 2481-2495. https://doi.org/10.1109/TPAMI.2016.2644615

Bang, S., Park, S., Kim, H. and Kim, H. (2019), "Encoder-decoder network for pixel-level road crack detection in black-box images", *Comput.-Aided Civil Infrastr. Eng.*, **34**(8), 713-727. https://doi.org/10.1111/mice.12440

Bao, Y. and Li, H. (2020), "Machine learning paradigm for structural health monitoring", *Struct. Health Monitor.*, 1475921720972416. https://doi.org/10.1177/1475921720972416

Bao, Y., Chen, Z., Wei, S., Tang, Z., Xu, Y. and Li, H. (2019), "The state of the art of data science and engineering in structural health monitoring", *Eng.*, **5**(2), 234-242. https://doi.org/10.1016/j.eng.2018.11.027

Bao, Y., Li, J., Nagayama, T., Xu, Y., Spencer Jr, B.F. and Li, H. (2021), "The 1st international project competition for structural health monitoring (IPC-SHM, 2020): a summary and benchmark problem", *Struct. Health Monitor.*, 14759217211006485. https://doi.org/10.1177/14759217211006485

Beckman, G.H., Polyzois, D. and Cha, Y. (2019), "Deep learning-based automatic volumetric damage quantification using depth camera", *Automat. Constr.*, **99**, 114-124. https://doi.org/10.1016/j.autcon.2018.12.006

Bengio, Y. (2012), "Practical recommendations for gradient-based training of deep architectures", *Neural Networks*: *Tricks of the Trade*, Springer, Berlin, Heidelberg.

Benz, C., Debus, P., Ha, H.K. and Rodehorst, V. (2019), "Crack segmentation on UAS-based imagery using transfer learning", *Proceedings of 2019 International Conference on Image and Vision Computing New Zealand* (*IVCNZ*), pp. 1-6. https://doi.org/10.1109/IVCNZ48456.2019.8960998

Bloice, M.D., Roth, P.M. and Holzinger, A. (2019), "Biomedical image augmentation using Augmentor", *Bioinformatics*, **35**(21), 4522-4524. https://doi.org/10.1093/bioinformatics/btz259

Castro, E., Cardoso, J.S. and Pereira, J.C. (2018), "Elastic deformations for data augmentation in breast cancer mass detection", *Proceedings of IEEE EMBS International Conference on Biomedical & Health Informatics* (*BHI*), Las Vegas, NV, USA, March, pp. 230-234. https://doi.org/10.1109/BHI.2018.8333411

Cha, Y., Choi, W. and Büyüköztürk, O. (2017), "Deep learning-based crack damage detection using convolutional neural networks", *Comput.-Aided Civil Infrastr. Eng.*, **32**(5), 361-378. https://doi.org/10.1111/mice.12263

Cha, Y.J., Choi, W., Suh, G., Mahmoudkhani, S. and Büyüköztürk, O. (2018), "Autonomous structural visual inspection using region-based deep learning for detecting multiple damage types", *Comput.-Aided Civil Infrastr. Eng.*, **33**(9), 731-747. https://doi.org/10.1111/mice.12334

Chen, F. and Jahanshahi, M.R. (2018), "NB-CNN: deep learning-based crack detection using convolutional neural network and naïve Bayes data fusion", *IEEE Transact. Indust. Electro.*, **65**(5), 4392-4400. https://doi.org/10.1109/TIE.2017.2764844

Chen, F., Jahanshahi, M.R., Wu, R. and Joffe, C. (2017a), "A texture-based video processing methodology using Bayesian data fusion for autonomous crack detection on metallic surfaces", *Comput.-Aided Civil Infrastr. Eng.*, **32**(4), 271-287. https://doi.org/10.1111/mice.12256

Chen, L., Papandreou, G., Schroff, F. and Adam, H. (2017b), "Rethinking atrous convolution for semantic image segmentation", *arXiv preprint*, http://arxiv.org/abs/1706.05587.

Choi, W. and Cha, Y. (2020), "SDDNet: Real-time crack segmentation", *IEEE Transact. Indust. Electro.*, **67**(9), 8016-8025. https://doi.org/10.1109/TIE.2019.2945265

Dong, C.Z. and Catbas, F.N. (2020), "A review of computer vision-based structural health monitoring at local and global levels", *Struct. Health Monitor.*, **20**(2), 692-743. https://doi.org/10.1177/1475921720935585

Fan, R., Bocus, M.J., Zhu, Y., Jiao, J., Wang, L., Ma, F., Cheng, S. and Liu, M. (2019), "Road crack detection using deep convolutional neural network and adaptive thresholding", *IEEE Intelligent Vehicles Symposium* (*IV*), Paris, France, June, pp. 474-479. https://doi.org/10.1109/IVS.2019.8814000

Fu, J., Liu, J., Tian, H., Li, Y., Bao, Y., Fang, Z. and Lu, H. (2019), "Dual attention network for scene segmentation", *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3146-3154.

Gerstner, W. and Kistler, W.M. (2002), *Spiking Neuron Models*: *Single Neurons*, *Populations*, *Plasticity*, Cambridge University

Press.
Gidon, A., Zolnik, T.A., Fidzinski, P., Bolduan, F., Papoutsi, A., Poirazi, P., Holtkamp, M., Vida, I. and Larkum, M.E. (2020), "Dendritic action potentials and computation in human layer 2/3 cortical neurons", *Science*, **367**(6473), 83-87. https://doi.org/10.1126/science.aax6239
He, K., Zhang, X., Ren, S. and Sun, J. (2016), "Deep residual learning for image recognition", *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, December, pp. 770-778.
Huang, G., Liu, Z., Van Der Maaten, L. and Weinberger, K.Q. (2017), "Densely connected convolutional networks", *Proceedings of 30th IEEE Conference on Computer Vision and Pattern Recognition CVPR 2017*, January, pp. 2261-2269.
Huang, H., Li, Q. and Zhang, D. (2018), "Deep learning based image recognition for crack and leakage defects of metro shield tunnel", *Tunnel. Undergr. Space Technol.*, **77**, 166-176. https://doi.org/10.1016/j.tust.2018.04.002
IPC-SHM (2020), http://www.schm.org.cn/#/IPC-SHM,2020/dataDownload
Jahanshahi, M.R. and Masri, S.F. (2013), "A new methodology for non-contact accurate crack width measurement through photogrammetry for automated structural safety evaluation", *Smart Mater. Struct.*, **22**(3), 035019. https://doi.org/10.1088/0964-1726/22/3/035019
Jahanshahi, M.R., Kelly, J.S., Masri, S.F. and Sukhatme, G.S. (2009), "A survey and evaluation of promising approaches for automatic image-based defect detection of bridge structures", *Struct. Infrastr. Eng.*, **5**(6), 455-486. https://doi.org/10.1080/15732470801945930
Jahanshahi, M.R., Masri, S.F., Padgett, C.W. and Sukhatme, G.S. (2013a), "An innovative methodology for detection and quantification of cracks through incorporation of depth perception", *Mach. Vision Applicat.*, **24**(2), 227-241. https://doi.org/10.1007/s00138-011-0394-0
Jahanshahi, M.R., Jazizadeh, F., Masri, S.F. and Becerik-Gerber, B. (2013b), "Unsupervised approach for autonomous pavement-defect detection and quantification using an inexpensive depth sensor", *J. Comput. Civil Eng.*, **27**(6), 743-754. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000245
Jiang, S. and Zhang, J. (2020), "Real-time crack assessment using deep neural networks with wall-climbing unmanned aerial system", *Comput.-Aided Civil Infrastr. Eng.*, **35**(6), 549-564. https://doi.org/10.1111/mice.12519
Kang, D., Benipal, S.S., Gopal, D.L. and Cha, Y.J. (2020), "Hybrid pixel-level concrete crack segmentation and quantification across complex backgrounds using deep learning", *Automat. Constr.*, **118**, 103291. https://doi.org/10.1016/j.autcon.2020.103291
Kong, X. and Li, J. (2018), "Vision-based fatigue crack detection of steel structures using video feature tracking", *Comput.-Aided Civil Infrastr. Eng.*, **33**(9), 783-799. https://doi.org/10.1111/mice.12353
Krizhevsky, A., Sutskever, I. and Hinton, G.E. (2012), "ImageNet classification with deep convolutional neural networks", *Proceedings of International Conference on Neural Information Processing Systems*, pp. 1097-1105.
Li, S., Zhao, X. and Zhou, G. (2019), "Automatic pixel-level multiple damage detection of concrete structure using fully convolutional network", *Comput.-Aided Civil Infrastr. Eng.*, **34**(7), 616-634. https://doi.org/10.1111/mice.12433
Li, G., Ren, X., Qiao, W., Ma, B. and Li, Y. (2020), "Automatic bridge crack identification from concrete surface using resnext with postprocessing", *Struct. Control Health Monitor.*, **27**(11), e2620. https://doi.org/10.1002/stc.2620
Lim, R.S., La, H.M. and Sheng, W. (2014), "A robotic crack inspection and mapping system for bridge deck maintenance", *IEEE Transactions on Automation Science and Engineering*, **11**(2), 367-378. https://doi.org/10.1109/TASE.2013.2294687
Lin, F., Yang, J., Shu, J. and Scherer, R.J. (2019), "Crack semantic segmentation using the U-net with full attention strategy", arXiv preprint arXiv:2104.14586.
Long, J., Shelhamer, E. and Darrell, T. (2015), "Fully convolutional networks for semantic segmentation", *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3431-3440.
Makantasis, K., Protopapadakis, E., Doulamis, A., Doulamis, N. and Loupos, C. (2015), "Deep convolutional neural networks for efficient vision based tunnel inspection", *Proceedings of 2015 IEEE 11th International Conference on Intelligent Computer Communication and Processing ICCP 2015*, Cluj-Napoca, Romania, September, pp. 335-342. https://doi.org/10.1109/ICCP.2015.7312681
Mokhtari, S., Wu, L. and Yun, H.B. (2017), "Statistical selection and interpretation of imagery features for computer vision-based pavement crack-detection systems", *J. Perform. Constr. Facil.*, **31**(5), 04017054. https://doi.org/10.1061/(ASCE)CF.1943-5509.0001006
Mukkamala, M.C. and Hein, M. (2017), "Variants of RMSprop and Adagrad with logarithmic regret bounds", *Proceedings of International Conference on Machine Learning*, pp. 2545-2553.
Ni, F., Zhang, J. and Chen, Z. (2019), "Pixel-level crack delineation in images with convolutional feature fusion", *Struct. Control Health Monitor.*, **26**(1), 1-18. https://doi.org/10.1002/stc.2286
Nishikawa, T., Yoshida, J., Sugiyama, T. and Fujino, Y. (2012), "Concrete crack detection by multiple sequential image filtering", *Comput.-Aided Civil Infrastr. Eng.*, **27**(1), 29-47. https://doi.org/10.1111/j.1467-8667.2011.00716.x
Pan, Y., Zhang, G. and Zhang, L. (2020), "A spatial-channel hierarchical deep learning network for pixel-level automated crack detection", *Automat. Constr.*, **119**, 103357. https://doi.org/10.1016/j.autcon.2020.103357
Ronneberger, O., Fischer, P. and Brox, T. (2015), "U-net: convolutional networks for biomedical image segmentation", *Proceedings of International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 234-241. https://doi.org/10.1007/978-3-319-24574-4_28
Shi, Y., Cui, L., Qi, Z., Meng, F. and Chen, Z. (2016), "Automatic road crack detection using random structured forests", *IEEE Transactions on Intelligent Transportation Systems*, **17**(12), 3434-3445. https://doi.org/10.1109/TITS.2016.2552248
Simard, P.Y., Steinkraus, D. and Platt, J.C. (2003), "Best practices for convolutional neural networks applied to visual document analysis", *Proceedings of the International Conference on Document Analysis and Recognition ICDAR*, January, pp. 958-963.
Simonyan, K. and Zisserman, A. (2014), "Very deep convolutional networks for large-scale image recognition", *arXiv preprint*, arXiv: 1409.1556.
Soukup, D. and Huber-Mörk, R. (2014), "Convolutional neural networks for steel surface defect detection from photometric stereo images", (Bebis G. *et al.* eds.), In: *Advances in Visual Computing*, pp. 668-677.
Spencer Jr, B.F., Hoskere, V. and Narazaki, Y. (2019), "Advances in computer vision-based civil infrastructure inspection and monitoring", *Engineering*, **5**(2), 199-222. https://doi.org/10.1016/j.eng.2018.11.030
Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V. and Rabinovich, A. (2015), "Going deeper with convolutions", *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1-9.
Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J. and Wojna, Z. (2016), "Rethinking the inception architecture for computer

vision", *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, December, pp. 2818-2826.

VanRullen, R., Guyonneau, R. and Thorpe, S.J. (2005), "Spike times make sense", *Trends Neurosci*., **28**(1), 1-4. https://doi.org/10.1016/j.tins.2004.10.010

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L. and Polosukhin, I. (2017), "Attention is all you need", In: *Advances in Neural Information Processing Systems*, pp. 5998-6008.

Wan, H., Gao, L., Su, M., Sun, Q. and Huang, L. (2021), "Attention-based convolutional neural network for pavement crack detection", *Adv*. *Mater*. *Sci*. *Eng*. https://doi.org/10.1155/2021/5520515

Wang, Z. and Cha, Y. (2020), "Unsupervised deep learning approach using a deep auto-encoder with an one-class support vector machine to detect structural damage", *Struct*. *Health Monitor*., **20**(1), 406-425. https://doi.org/10.1177/1475921720934051

Wang, D., Dong, Y., Pan, Y. and Ma, R. (2020), "Machine vision-based monitoring methodology for the fatigue cracks in U-rib-to-deck weld seams", *IEEE Access*, **8**, 94204-94219. https://doi.org/10.1109/ACCESS.2020.2995276

Xu, Y., Li, S., Zhang, D., Jin, Y., Zhang, F., Li, N. and Li, H. (2018), "Identification framework for cracks on a steel structure surface by a restricted Boltzmann machines algorithm based on consumer-grade camera images", *Struct*. *Control Health Monitor*., **25**(2), e2075. https://doi.org/10.1002/stc.2075

Xu, Y., Bao, Y., Chen, J., Zuo, W. and Li, H. (2019a), "Surface fatigue crack identification in steel box girder of bridges by a deep fusion convolutional neural network based on consumer-grade camera images", *Struct*. *Health Monitor*., **18**(3), 653-674. https://doi.org/10.1177/1475921718764873

Xu, Y., Wei, S., Bao, Y. and Li, H. (2019b), "Automatic seismic damage identification of reinforced concrete columns from images by a region-based deep convolutional neural network", *Struct*. *Control Health Monitor*., **26**(3), e2313. https://doi.org/10.1002/stc.2313

Yang, X., Li, H., Yu, Y., Luo, X., Huang, T. and Yang, X. (2018), "Automatic pixel-level crack detection and measurement using fully convolutional network", *Comput*.*-Aided Civil Infrastr*. *Eng*., **33**(12), 1090-1109. https://doi.org/10.1111/mice.12412

Yeum, C.M. and Dyke, S.J. (2015), "Vision-based automated crack detection for bridge inspection", *Comput*.*-Aided Civil Infrastr*. *Eng*., **30**(10), 759-770. https://doi.org/10.1111/mice.12141

Yeum, C.M., Dyke, S.J. and Ramirez, J. (2018), "Visual data classification in post-event building reconnaissance", *Eng*. *Struct*., **155**, 16-24. https://doi.org/10.1016/j.engstruct.2017.10.057

Zhang, S.Q. and Zhou, Z.H. (2020), "Flexible transmitter network", *arXiv preprint*, arXiv:2004.03839.

Zhang, L., Yang, F., Daniel Zhang, Y. and Zhu, Y.J. (2016), "Road crack detection using deep convolutional neural network", *Proceedings of International Conference on Image Processing ICIP*, August, pp. 3708-3712. https://doi.org/10.1109/ICIP.2016.7533052

Zhang, X., Rajan, D. and Story, B. (2019), "Concrete crack detection using context-aware deep semantic segmentation network", *Comput*.*-Aided Civil Infrastr*. *Eng*., **34**(11), 951-971. https://doi.org/10.1111/mice.12477