Smart Structures and Systems, *Vol. 29, No. 1 (2022) 221-235*
https://doi.org/10.12989/sss.2022.29.1.221

Crack segmentation in high-resolution images using cascaded deep convolutional neural networks and Bayesian data fusion

Wen Tang [1], Rih-Teng Wu*[3] and Mohammad R. Jahanshahi [1,2a]

[1] *Lyles School of Civil Engineering, Purdue University, West Lafayette, IN 47906, USA*
[2] *Elmore Family School of Electrical and Computer Engineering (Courtesy), Purdue University, West Lafayette, IN 47907, USA*
[3] *Department of Civil Engineering, National Taiwan University, Taipei, Taiwan*

(*Received May 8, 2021, Revised August 19, 2021, Accepted September 9, 2021*)

Abstract. Manual inspection of steel box girders on long span bridges is time-consuming and labor-intensive. The quality of inspection relies on the subjective judgements of the inspectors. This study proposes an automated approach to detect and segment cracks in high-resolution images. An end-to-end cascaded framework is proposed to first detect the existence of cracks using a deep convolutional neural network (CNN) and then segment the crack using a modified U-Net encoder-decoder architecture. A Naïve Bayes data fusion scheme is proposed to reduce the false positives and false negatives effectively. To generate the binary crack mask, first, the original images are divided into 448 × 448 overlapping image patches where these image patches are classified as cracks versus non-cracks using a deep CNN. Next, a modified U-Net is trained from scratch using only the crack patches for segmentation. A customized loss function that consists of binary cross entropy loss and the Dice loss is introduced to enhance the segmentation performance. Additionally, a Naïve Bayes fusion strategy is employed to integrate the crack score maps from different overlapping crack patches and to decide whether a pixel is crack or not. Comprehensive experiments have demonstrated that the proposed approach achieves an 81.71% mean intersection over union (mIoU) score across 5 different training/test splits, which is 7.29% higher than the baseline reference implemented with the original U-Net.

Keywords: Bayesian data fusion; crack detection; deep learning; semantic segmentation; structural health monitoring

1. Introduction

1.1 Motivation

Structural defect detection is an important aspect of structural health monitoring (SHM). Among various types of surface defects, cracks are one of the most common damage that occurs at the early stage of structural failure. To this end, in time detection and quantification of cracks provide important information regarding the condition of the structure that can prevent the destructive events from happening. The current crack inspection process for steel box girders, which are the most common structural components in long span bridges, is time-consuming and labor-intensive. Most of the cracks on metallic surfaces are thin, that require the use of high-resolution images to be identified. However, these cracks are usually located at inaccessible regions that are difficult for the inspector to access. Also, the shadows induced from the artificial lighting source during the inspection process, the handwritings, marks, corrosion, and other types of noisy patterns, introduce more challenges for crack inspection. There is an urgent need to develop a reliable and autonomous system for the detection and quantification of cracks for bridges.

Although there are many existing crack detection approaches being developed, a majority of these methods use single images with a resolution up to 1024 × 512. This could be problematic since the detection result from one image may contain false positives or false negatives due to noisy patterns and shadows. To address this issue, this study proposes a cascaded framework to detect and segment cracks from high-resolution (i.e., 4K) images. The proposed framework consists of the detection of cracks using a deep CNN (LeCun *et al*. 2015), followed by a modified U-Net (Ronneberger *et al*. 2015) that is used to segment crack pixels in the images.

The effects of false positives and false negatives are further mitigated by fusing the segmentation results from different overlapping image patches by using a Naïve Bayes decision rule. In Section 1.2, a literature review for relevant works in crack detection and segmentation is provided. Section 1.3 describes the contribution and the scope of this study.

1.2 Related work

The recent advances in computer vision techniques have opened up lots of opportunities for applications within SHM (Spencer *et al*. 2019, Bao *et al*. 2019, Bao and Li 2021). By leveraging image analysis, the detection of surface defects

*Corresponding author, Ph.D., Assistant Professor,
E-mail: rihtengwu@ntu.edu.tw
[a] Associate Professor

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

can be quickly conducted using contactless sensors such as digital cameras, surveillance cameras or small unmanned aerial vehicles (UAVs). For instance, image-based crack detection was explored in civil engineering field (Oh *et al*. 2009, Lim *et al*. 2014). A review of using computer vision techniques in bridge structures has been investigated in (Jahanshahi *et al*. 2009). An advanced robotic system has been developed to monitor the structures (Lee *et al*. 2011).

The early developments of crack detection approaches are based on techniques including thresholding (Cheng *et al*. 1999, 2003), image percolation (Yamaguchi and Hashimoto 2010), edge detection (Abdel-Qader *et al*. 2003, Fujita and Hamamoto 2010), or morphological operations. However, these conventional methods only work when the background is clean, and the crack has a high contrast. Also, hand-crafted features are required in the abovementioned methods. To overcome these issues, deep learning provides an alternative solution to crack detection task. Compared to conventional approaches, a deep learning based algorithm can extract the useful features itself, and it is more robust against higher levels of background noise. With the development of deep learning, cracks are detected using a sliding window of relatively small step size (Cha *et al*. 2017, Chen and Jahanshahi 2017) and each window is passed to a deep CNN to determine whether that window contains cracks or not. The final prediction mask is a set of small bounding boxes that outlines the shape of the cracks. This kind of approach is computationally expensive since the number of sliding windows that need to be processed increases quadratically as the step size decreases. To alleviate the computational cost, object detection based algorithms are used for damage detection and localization (Cha *et al*. 2018, Xue and Li 2018, Maeda *et al*. 2018, Beckman *et al*. 2019, Deng *et al*. 2020). These approaches use Faster Region-based Convolutional Network (Ren *et al*. 2015) to come up with different sizes of bounding boxes. Despite the above-mentioned algorithms run faster and give a tighter bounding box compared to sliding window technique, the cracks are still detected using a collection of bounding boxes that contains a large amount of background.

It is clear that localizing the cracks using bounding boxes is not sufficient in some applications where the width of the crack has to be quantified. To this end, crack segmentation is a procedure where each pixel in an image is classified as crack or non-crack. Fully convolutional networks (FCNs) (Long *et al*. 2015) are proposed to tackle semantic segmentation tasks. FCN uses an encoder-decoder architecture that first extracts the features and then reconstructs the features to get the output binary mask. In particular, U-Net (Ronneberger *et al*. 2015) is a type of FCN that contains skip connections to get better performance. A skip connection allows the input of the convolution block to skip some layers and directly merge with the output of the convolutional block. By creating a skip connection, vanishing or exploding gradients can be mitigated so that the loss calculated at the end of the network can be properly backpropagated to the earlier layers. Similar to U-Net, SegNet (Badrinarayanan *et al*. 2017) is developed with an encoder network and a corresponding decoder network, followed by a final pixelwise classification layer. There are also other heavier and larger segmentation networks, such as DeepLab (Chen *et al*. 2017b, 2018).

The above-mentioned deep learning architectures have been adopted for crack segmentation tasks and the performances are reported in Liu *et al*. (2019a), Ji *et al*. (2020), Dung (2019), Zhang *et al*. (2019), Yang *et al*. (2018). However, these studies have been tested on concrete surface without any noisy backgrounds which could limit the effectiveness of these methods when applied to real world data with high background noise level. For Bang *et al*. (2019), the proposed approach was tested on complex background but the reported mIoU was around 54% with lots of crack missed. This might be due to the fact that the cracks are relatively small in high-resolution images, and it would be hard for the network to distinguish crack from other instances after several downsampling layers. In the study of Liu *et al*. (2019b), a U-Net based concrete crack segmentation network is proposed that can achieve a high precision, but the size of the input image is small (512 × 512) compared to the high-resolution images. In Liu *et al*. (2020), the authors propose a two-step pavement crack detection and segmentation method based on You Only Look Once 3rd version (YOLO v3) and modified U-Net. The proposed method achieves high precision score on detection and segmentation. However, the detection and segmentation are trained and tested on two different datasets so the effect of detection network on the followed segmentation is not discussed. In the study (Choi and Cha 2019), the authors propose a semantic damage detection network (SDDNet), and the trained network achieves a mIoU of 0.846 on the test set with complex background noise. However, again the input size is 1024 × 512, which is relatively small compared to the high-resolution images.

It is noted that a majority of the existing crack segmentation approaches mentioned above use single image to perform crack detection, and the maximum resolution of the image is around 1024 × 512. To ensure the quality of segmentation on high-resolution images, one approach is to crop a given high-resolution image into several image patches and then perform crack segmentation on those smaller image patches. This approach has been used in some studies in the field of remote image sensing. The image patches in Tasar *et al*. (2019), Ding *et al*. (2020) are non-overlapping, and the information from other adjacent patches are not leveraged. Despite some levels of overlapping between image patches are considered in Liu *et al*. (2018), the information from different image patches is simply added together without considering any data fusion strategies. Similar to this study, Chen and Jahanshahi (2019) try to segment the crack on metallic surface and reduce the false positives using Bayesian data fusion. However, the studies are based on crack detection bounding boxes rather than pixel-level segmentation. Therefore, the crack segmentation given is much wider than the actual crack whereas the proposed approach can produce segmentation masks that are very close to the actual width of the cracks. In addition, Chen and Jahanshahi (2019) fuse the information from sequential video frames whereas this

study uses non-sequential high-resolution images.

The high-resolution images and partial ground truth masks come from Xu *et al*. (2019) where CNN based network is utilized to identify potential 64 × 64 crack patches. The identified crack patches are then processed by the optimal entropy threshold method to segment out the crack pixels. The proposed method can generate binary pixel-level crack masks on high-resolution images, but the masks are noisy and the reported F1 score is low under noisy backgrounds.

Xu *et al*. (2018) proposed a framework based on the restricted Boltzmann machine to identify the cracks from steel box girder images. The proposed network can output crack identification maps consisting of small image patches of size 24 × 24 that contain the crack but the crack itself is not segmented out.

1.3 Contribution and scope

This study proposes a cascaded framework to segment the cracks in high-resolution images for steel girders on long span bridges. Instead of relying on single network to achieve crack segmentation, the proposed approach first employs a deep CNN-based crack classifier to detect the existence of cracks in overlapping patches in order to reduce false positives. Once a crack patch is identified, it is further processed by a segmentation network with modified U-Net architecture to generate the crack score map. The generated crack score maps are integrated using a Naïve Bayes decision rule to generate a complete binary mask of the original high-resolution image. The contributions of this study are summarized as follows:

- The proposed framework consists of a deep CNN crack classifier and a modified U-Net achieves an 81.71% mIoU for crack segmentation on high-resolution images.
- A customized loss function and a modified U-Net architecture are proposed to enhance the segmentation performance.
- A Naïve Bayes data fusion strategy is proposed based on the statistical distribution of the crack scores to reduce the false positives and false negatives.
- A total of 230 high-resolution images are manually annotated to further evaluate the performance of crack segmentation.

An important contribution of this paper is that the statistical properties of the crack scores for each pixel between overlapping crack patches are considered and Bayesian data fusion is used to reduce the false positives and the false negatives. Compared with non-overlapping patches where each crack is only observed once, overlapping image patches allow each crack to be observed multiple times, which makes it more robust to occasional mistakes. In addition, non-overlapping cropping might result in cracks that locate in the corners of image patches that makes it challenging for the crack classifier to detect. Whereas overlapping cropping can mitigate this issue by observing the same crack in different crack patches where the crack might be more obvious and easier for the crack classifier to detect.

The remainder of this paper is organized as follows. Section 2 describes the details of the proposed framework. Section 3 discusses the datasets used in this study. The experimental results and the associated discussions are provided in Section 4. Concluding remarks are addressed in Section 5.

2. Proposed pipeline

Fig. 1 shows the schematic workflow of the proposed framework. Given a high-resolution image, the analysis procedure starts with breaking the image into overlapped small image patches. The deep CNN classifies whether an image patch contains a crack or not. If not, a mask with all black pixels is created. If yes, a modified U-Net segmentation model performs semantic segmentation on the image patch and returns the crack score map of the patch. In the last stage, a Naïve Bayes fusion strategy is employed to integrate the crack score maps of all the overlapping image patches and output the final binary crack mask of the original high-resolution image.

2.1 Crack classifier

The high-resolution image is raster scanned with a sliding window of size 448 × 448 using a step size of 130 pixels. Then, the proposed deep CNN classifier decides whether a given image patch contains a crack or not. It should be noted that a smaller step size can potentially achieve a more robust Bayesian fusion result since there will be more overlapping patches being considered.

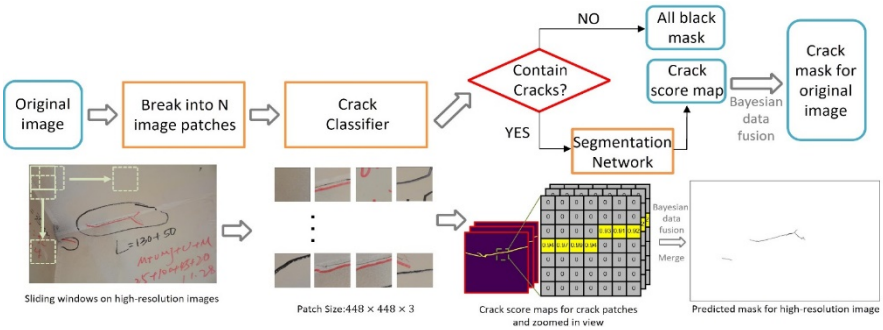
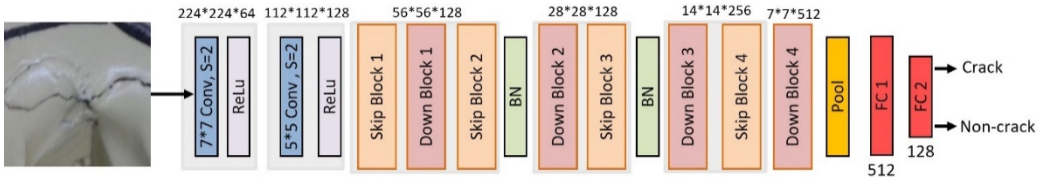
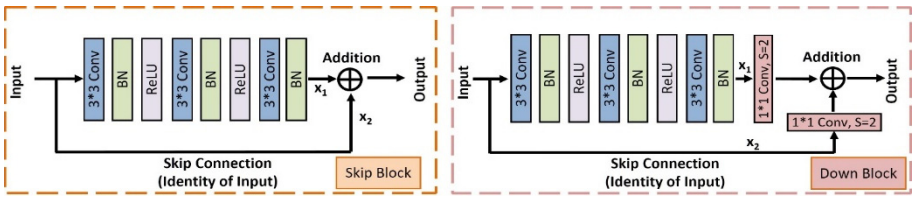


Fig. 1 Schematic workflow of the proposed approach



(a) The architecture of the proposed deep CNN classifier



(b) The architecture of Skip Block (c) The architecture of the Down Block

Fig. 2 Architecture of (a) the proposed deep CNN crack classifier, (b) Skip Block and (c) Down Block. The architecture of Down Block is similar to Skip Block, except X_1 and X_2 are downsampled with a 1×1 convolutional layer with stride 2. The numbers above the layers in (a) indicate the output tensor size. Conv: Convolutional layer; BN: Batch Normalization Layer; Pool: Global Average Pooling layer; ReLU: Rectified linear unit; FC: Fully Connected layer

However, the computational time will scale up rapidly as the step size decreases. Considering the dimension of the high-resolution image, which is around 4,000 pixels in horizontal and vertical directions, a reasonable step size of 130 pixels is chosen.

Fig. 2 presents the overall architecture of the proposed deep CNN crack classifier. The 448×448 RGB image patch is fed into the crack classifier and the final Softmax layer predicts whether the input patch contains a crack or not. Skip Block is a basic building block used in this architecture to prevent the effect of vanishing gradient during training. With the skip connections (He *et al*. 2016), the network includes shortcut pathways so that the loss calculated at the output layer can be properly backpropagated to the earlier layers of the network without vanishing gradient. As shown in Fig. 2, the Skip Block consists of three convolutional layers. Each convolutional layer is followed by a Batch Normalization layer (Ioffe and Szegedy 2015) and a rectified linear unit (ReLU) activation layer (Nair and Hinton 2010). Down Block is another basic building block inside the proposed architecture. The purpose of Down Block is to downsample the output tensors of the previous block along the spatial horizontal and vertical dimensions. The architecture of Down Block is similar to the architecture of Skip Block except that tensors X_1 and X_2 are fed into two separate 1×1 convolutional layers before the addition operation. Downsampling is achieved by sliding 1×1 convolutional kernels through tensors X_1 and X_2 with a stride of 2. Each time a tensor is passed into the Down Block, the height and width of the tensor is reduced by 2. Compared to the conventional pooling layers (Ciregan *et al*. 2012), the use of Down Block ensures that the network can conduct downsampling operations with learnable parameters during training. As shown in Fig. 2(a), the output tensor of Down Block 4 is passed through a Global Average Pooling layer (Lin *et al*. 2013) before feeding into the fully-connected layers. The crack classifier identifies the input as a crack patch if the Softmax score of being a crack is greater than a certain threshold θ_T, and a non-crack patch otherwise. The selection of threshold θ_T will be discussed in Section 4.3.

2.2 Segmentation network

Based on the classification result of the deep CNN crack classifier, each crack patch is fed into a segmentation network to obtain its corresponding crack score map *M*. The dimensions of the input RGB patch is $448 \times 448 \times 3$, while the dimensions of the resulting score map M are $448 \times 448 \times 1$. Each pixel in the score map is a real number $s^c \in [0,1]$, representing the possibility of that pixel being a crack pixel. The score maps of different overlapping crack patches are fused together using a Bayesian fusion strategy, which will be described in Section 2.3.

Inspired by the U-Net architecture (Ronneberger *et al*. 2015), a modified U-Net is implemented that consists of input, convolution (CONV), transposed convolution (T CONV), batch normalization (BN), ReLU activation, Max Pooling (POOL), dropout, and concatenation layers (CONCAT). The detailed architecture diagram of this network is shown in Fig. 3. The network consists of a contracting path and an expansive path, which give it the u-shaped architecture. The contracting path is a typical convolutional network that consists of repeated operations of CONV, BN, dropout, CONV, BN and POOL. During the contraction, the spatial information is reduced while feature information is extracted. The expansive pathway combines the feature and spatial information through a sequence of up-convolutions and concatenations with high-resolution features from the contracting path.

The pixel intensity of the input image patches will be clipped between -1 and 1. The CONCAT layer of U-Net appends the output of contraction layer with the new expansion layer input, which can preserve the multi-level and multi-scale features from different stages of convolutions. In addition, the network prevents vanishing gradient using skip connections indicated as blue arrows in Fig. 3. Details regarding configurations of the BN and the

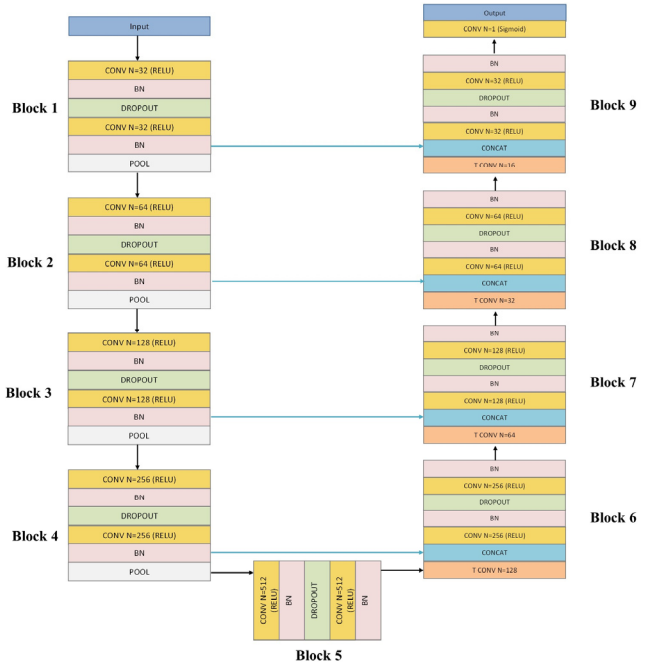


Fig. 3 Detailed architecture of the segmentation network. The blue arrows are skip connections.

dropout layers are discussed in Section 4.2.1.

Instead of conducting up-sampling with a fixed rule such as bilinear interpolation, the transposed convolution layers are used in the expansion path. This enables the network to upsample the tensor nonlinearly with trainable parameters which leads to more accurate crack score maps. Fig. 4 is a schematic presentation of the transposed convolution operation, where s is the stride, k is the kernel size and p is padding. All the convolution kernels in the modified U-Net are 3 × 3 with a stride of 1. The transposed convolution kernels are 2 × 2 with a stride of 2 in both spatial directions. All the kernels are initialized with He-Normal initializer (He *et al*. 2015), and the SAME padding method (Dumoulin and Visin 2016) is used.

In order to further improve the performance of the segmentation network, a customized loss function that combines Binary Cross Entropy loss (BCE) with the Dice loss is used. While BCE loss is commonly used to classify objects belonging to different classes, it can also be used in the semantic segmentation task to classify the crack and non-crack pixels. The gradients of the BCE loss are smooth and not prone to vanish, which is a desirable property for back-propagation during training. However, the dataset used in this study is highly unbalanced where the number of pixels counted as crack is significantly smaller than the background pixels. This highly unbalanced label distribution negatively affects the accuracy if only cross entropy loss is used, and one way to mitigate this issue is including the Dice loss in the loss function. Dice loss is calculated from the Dice coefficient, which is a statistical tool that can better gauge the similarity between two samples even if the label distribution is highly unbalanced. In addition, for segmentation tasks, usually the objective is to develop a network that maximizes mIoU. Therefore, using Dice coefficient in loss function allows us to directly optimize the objective function with respect to mIoU. Eq. (1) is the definition of IoU which is commonly used to evaluate the performance of semantic segmentation tasks. The mIoU can be calculated by average the IoU of crack and non-crack classes. Eq. (2) shows the equation for calculating the Dice coefficient, where *TP* stands for True Positive, *FP* stands for False Positive, and *FN* stands for False Negative. Eq. (3) shows the formula for BCE, where $y_{j,i}$ and $\hat{y}_{j,i}$ are the ground label and predicted label for i^{th} pixel in the j^{th} image in one batch, respectively. N is the total number of pixels in a single image patch and M is the total number of images in a single batch. Eq. (4) is the final combined loss function that is used for the modified U-Net during the training where the advantages of both the BCE loss and the Dice loss are leveraged at the same time. The mixture ratio α between these two losses is a hyperparameter which is tuned based on the nature of datasets. Effects of different values for mixture ratio α are discussed in Section 4.2.2. The final α value used in this study is 0.5.

$$IoU = \frac{TP}{TP + FP + FN} \tag{1}$$

$$L_D = \frac{2|X \cap Y|}{|X| + |Y|} = \frac{2 * TP}{2 * TP + FP + FN} \tag{2}$$

$$L_{BCE} = -\frac{1}{M}\sum_{j=1}^{M}\frac{1}{N}\sum_{i=1}^{N}[y_i log(\hat{y}_{j,i}) + (1 - y_{j,i})\log(1 - \hat{y}_{j,i})] \tag{3}$$

$$L_{total} = \alpha\,(-\log(L_D)) + (1 - \alpha)L_{BCE} \tag{4}$$

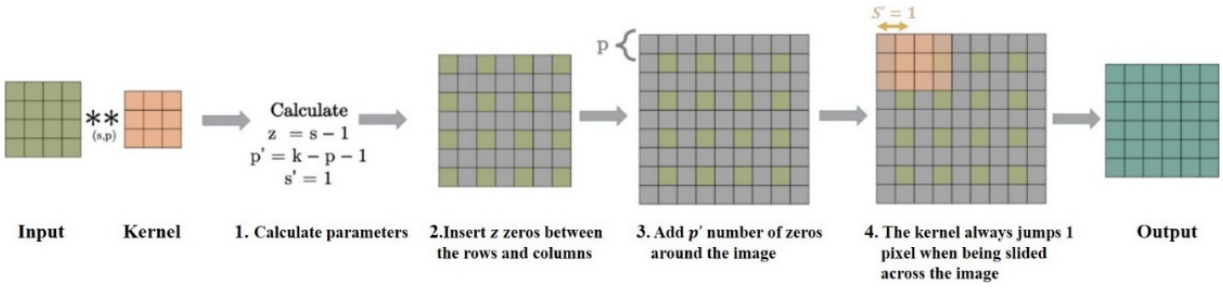


Fig. 4 Schematic diagram of the transposed convolution operation

2.3 Bayesian data fusion

One major advantage of using overlapping image patches to detect cracks is that the same crack would be presented in multiple adjacent image patches so that detection scores from multiple image patches can be fused to come up with a more robust detection result. Compared to the detection result purely obtained from one single image, fusing the crack score s^c from multiple image patches leads to a more robust detection result. The crack score s^c is the output of the sigmoid function indicated in Fig. 3.

To reduce the false positive and false negative rate, a data fusion method based on Bayes' theorem is proposed. After computing the crack score map for each crack image patch, the score maps are registered on the original high-resolution image and a list of crack scores for the same location is obtained. Given n overlapping patches, let $P(crack|s_1^c, s_2^c, \cdots, s_n^c)$ and $P(noncrack|s_1^c, s_2^c, \cdots, s_n^c)$ be the posterior probabilities of the pixel being crack and non-crack, respectively. The pixel is classified as crack if the ratio shown in Eq. (5) surpasses the threshold ε.

$$\frac{P(crack|s_1^c, s_2^c, s_3^c, \cdots, s_n^c)}{P(noncrack|s_1^c, s_2^c, s_3^c, \cdots, s_n^c)} > \varepsilon \tag{5}$$

where s_i^c is the pixel crack score given by the segmentation network for the i-th crack image patch. Assuming the crack scores s^c generated by segmentation network are identically and independently distributed, Eq. (5) can be written as Eq. (6) or Eq. (7)

$$\frac{P(crack)\prod_{i=1}^{n} P(s_i^c|crack)}{P(noncrack)\prod_{i=1}^{n} P(s_i^c|noncrack)} > \varepsilon \tag{6}$$

$$\frac{\prod_{i=1}^{n} P(s_i^c|crack)}{\prod_{i=1}^{n} P(s_i^c|noncrack)} > \frac{P(noncrack)}{P(crack)}\varepsilon \tag{7}$$

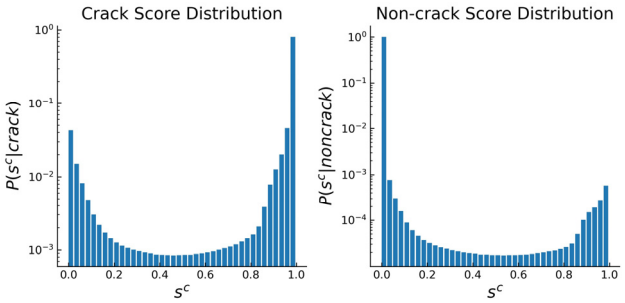
where $P(s_i^c|crack)$ is the conditional probability of getting a crack score s_i^c given that the pixel is truly a crack and $P(s_i^c|noncrack)$ is the conditional probability of getting a crack score s_i^c given that the pixel is truly a non-crack. The prior probabilities of the pixel being a crack and non-crack is denoted as $P(crack)$ and $P(noncrack)$ respectively. We can further simplify Eq. (7) by defining a new variable $\varphi_{NB}(s_i^c)$ shown below.

$$\varphi_{NB}(s_i^c) = P(s_i^c|crack)/P(s_i^c|noncrack) \tag{8}$$

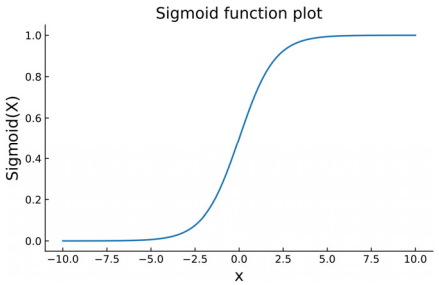
Therefore, the pixel would be considered as a crack if

$$\Phi^{NB}(s^c) = \prod_{i=1}^{n} \varphi_{NB}(s_i^c) > \theta_{NB} \tag{9}$$

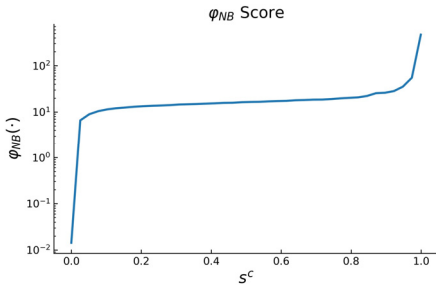
where $\theta_{NB} = \varepsilon\, P(noncrack)/P(crack)$. For a given list of crack scores $[s_1^c, s_2^c, s_3^c, \cdots, s_n^c]$ of the same pixel location, we say the pixel is crack if $\Phi^{NB}(s^c)$ surpasses the threshold θ_{NB}. The decision threshold θ_{NB} determines how sensitive the algorithm is to the false positives and false negatives. The prior probabilities $P(crack)$ and $P(noncrack)$ can be estimated by the percentage of crack pixels in the validation dataset.



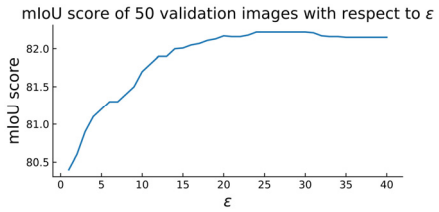
(a) Likelihood functions computed using segmentation network for crack and non-crack validation patches



(b) An illustration of Sigmoid function



(c) Value of $\varphi_{NB}(s_i^c)$ computed from Eq. (8). Logarithmic scale is used for y-axis



(d) Sensitivity test of mIoU with respect to ε

Fig. 5 Segmentation network statistics and sensitivity test

In this study, the values for $P(crack)$ and $P(noncrack)$ are 0.0145 and 0.9855, respectively. The value of ε is determined by running a parameter sensitivity test where the mIoU score of the 50 validation images are recorded with respect to different ε values. The validation images are randomly chosen from the high-resolution training images, which are different from the high-resolution test images. The optimal value $\varepsilon = 25$ is picked according to the result of the sensitivity test shown in Fig. 5(c), where the highest mIoU score is achieved when $\varepsilon = 25$. The conditional probabilities are estimated by feeding

the crack patches in validation dataset through segmentation network and plotted in Fig. 5(a). The value of $\varphi_{NB}(s_i^c)$ is calculated using Eq. (8) and plotted in Fig. 5(c).

From Fig. 5(a), it is observed that as s^c moves to zero, the crack score value gradually increases; also, in the non-crack score distribution, as s^c moves to one, the non-crack score value gradually increases. This interesting phenomenon may arise from the optimization process during the training of deep learning models. The proposed network uses Sigmoid activation function before the output layer to generate a clipped CNN score between [0, 1]. Notice that the neurons (i.e., nodes) before the Sigmoid function may output any arbitrary real number. When the inputs take values greater than 4 or smaller than -4, the Sigmoid function tends to flatten out and generate output values very close to 1 or 0, as shown in Fig. 5(b). This means that a lot of incorrectly classified samples will be "squeezed into" the opposite CNN score region (for crack patches, the opposite CNN score region would be 0; and for wrongly classified non-crack patches, the CNN score would be 1). Most importantly, the majority of the CNN score for crack patches are near 1. Fig. 5(a) in the manuscript is in log-scale so that the actual number of "crack score close to 1" is nearly 70 times larger than "crack score close to 0". For non-crack patches, the number of "crack score close 0" is nearly 5000 times larger than "crack score close to 1". Therefore, it is safe to say that the algorithm will output crack score close to 1 when it sees a crack pixel, and output crack score close to 0 when it sees a non-crack pixel.

To save some computational time, the summation $\Phi^{NB}(s^c)$ is calculated only if a pixel location has a crack score $s_i^c > 0.2$. After registering all the crack score maps of different image patches, the score map of the high-resolution image could be binarized with a threshold θ_{NB} and the final binary crack mask is obtained.

In this study, the Naïve Bayesian fusion is applied to the segmentation network, not the crack classifier. Segmentation tasks deal with pixel-level classification, which means that the same pixel would be observed multiple times from different image patches due to the overlapping sliding window schema. However, for the crack classifier, the exact same image patches can only be observed once despite of the overlapping regions between the patches. The CNN scores can be fused only when exactly the same element is observed multiple times in different scenarios. Therefore, the proposed Bayesian fusion is not applicable to the crack classifier in the context of segmentation unless overlapping video frames are used, which is not the case in the current study.

3. Database

In total, 350 high-resolution raw images were provided by the IPC-SHM 2020 competition (Bao *et al*. 2021) with a resolution of 3264 × 4928 or 3864 × 5152 pixels from the construction site inside the steel girder of a bridge (Xu *et al*. 2019). No preprocessing is applied to the images. Therefore, there are some noisy images in the dataset like blurred images that could affect the final segmentation result. 120 of the ground truth crack masks of the original high-resolution images are provided in (Xu *et al*. 2019, Bao *et al*. 2021) and the rest of the ground truth masks are manually annotated. Some of the provided ground truth masks are modified because thin black lines caused by the shadows are mislabeled as cracks. The high-resolution

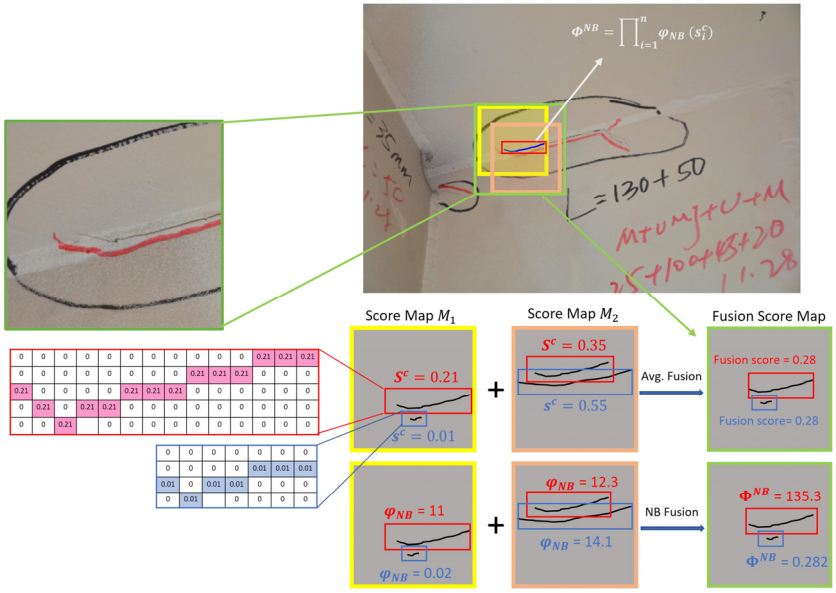


Fig. 6 Schematic plot of the proposed Bayesian data fusion method. The same true positive crack bounded by the red box is seen twice and the crack scores are fused. Cracks in the blue bounding box are false positives. It demonstrates a case where it is not possible to distinguish between false positive and true positive using simple averaging, but the proposed Naïve Bayes fusion is able to differentiate the false positives. For simplicity of visualization, assume that all the potential crack pixels have the same crack score inside each bounding box. Bayesian data fusion is performed on each pixel and bounding boxes are used only for visualization purposes.

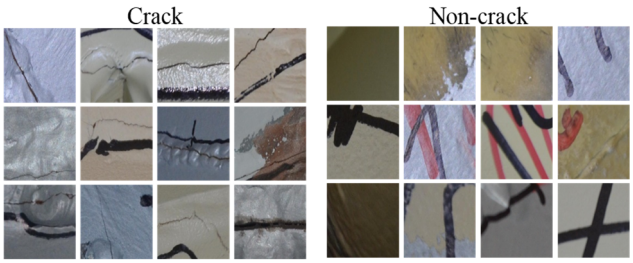


Fig. 7 Visualization of sample crack and non-crack image patches

images are then cropped down to overlapping image patches of size 448×448.

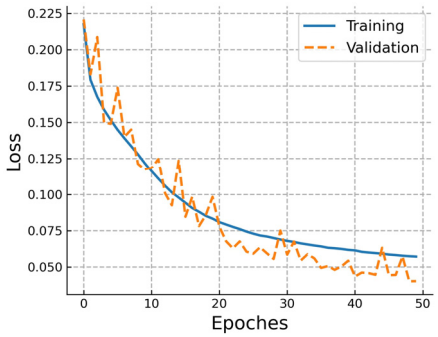
A total of 74,647 crack image patches are obtained from 300 high-resolution images. The rest of the 50 images are left out for testing. To construct a balanced dataset for crack classifier training, non-crack image patches were randomly cropped in the high-resolution images. The final dataset contains 74,647 and 75,000 crack and non-crack image patches, respectively. Fig. 7 illustrates samples of image patches. Due to the highly imbalanced distribution between crack and non-crack pixels, only crack patches are used during the training of the modified U-Net, whereas both crack and non-crack patches are used when training the crack classifier.

4. Experimental results

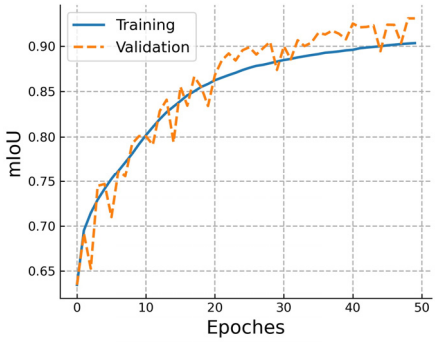
The performance of the proposed cascaded framework is evaluated using different experimental setups. Section 4.1 describes the details of the training process. In Section 4.2, the effects of batch normalization layers, dropout layers, and different loss function mixture ratios on segmentation network (i.e., modified U-Net) are discussed. In Section 4.3, parametric studies are conducted to decide the best configuration for both the crack classifier and the segmentation network. In Section 4.4, the performance of the proposed Naïve Bayes data fusion method is compared with a baseline reference implemented by average fusion. Section 4.5 discusses sample predictions on the test dataset.

4.1 Training and validation results

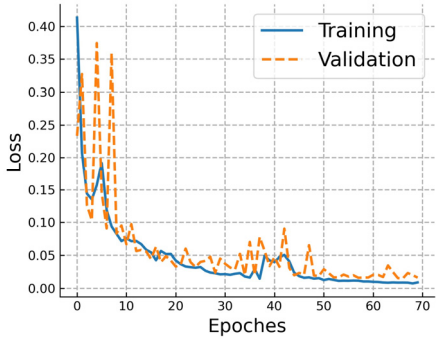
As discussed in Section 3, a total of 149,647 crack and non-crack image patches were used to train the crack classifier from scratch. For segmentation network, 74,647 crack patches with annotated ground truth binary crack masks are used during training. In both cases, 80% of the data was randomly selected and used to train the network, and the remaining 20% was used as the validation dataset to determine the appropriate timing to stop training in order to avoid overfitting. Adam optimizer (Kingma and Ba 2014) with a learning rate of 5×10^{-5} was used to back-propagate the error and update the learnable parameters inside the networks. The mini-batch size was set to 8 for both networks during training. Training history of the crack classifier and the segmentation network are shown in Fig. 8. According to Figs. 8(a) and 8(b), the lowest validation loss



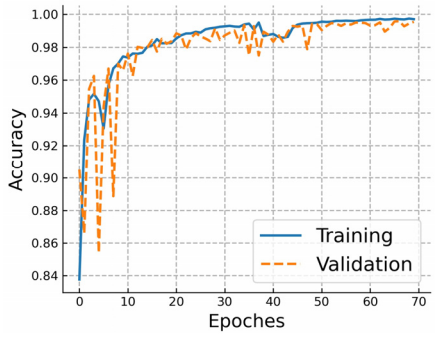
(a) Loss of the segmentation network



(b) mIoU of the segmentation network



(c) Loss of the crack classifier



(d) Classification accuracy of the crack classifier

Fig. 8 Training and validation results for crack classifier and segmentation networks

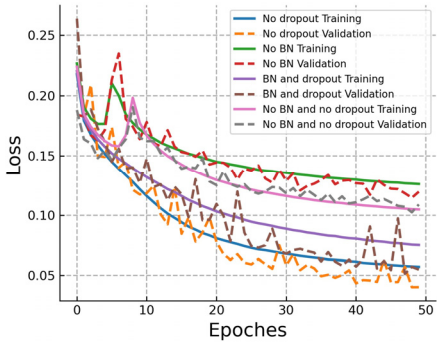
of the segmentation network (i.e., modified U-Net) is achieved at approximately the 47th epoch with an overall mIoU over 90%. It is noted that the mIoU plotted in Fig. 8 refers to the mIoU of the validation crack patches, not the mIoU of the high-resolution test images. As shown in Figs. 8(c) and 8(d), the lowest validation loss of crack classifier is

approximately achieved at the 65th epoch with an accuracy over 99%. It is also observed in Fig. 8(a) that the validation loss is lower than the training loss for segmentation network most of the time after the 25th epoch. This might due to the fact that heavy image augmentations including rotation, flipping, random contrast, zooming and shifting were used during training, which makes the prediction on the training dataset more challenging than the validation dataset. The training was conducted on a computation platform including a Linux server that consists of two Intel Xeon E5-2620 CPU, 256 GB DDR4 RAM and eight NVIDIA RTX Quadro 8000 GPU with 48 GB memory.

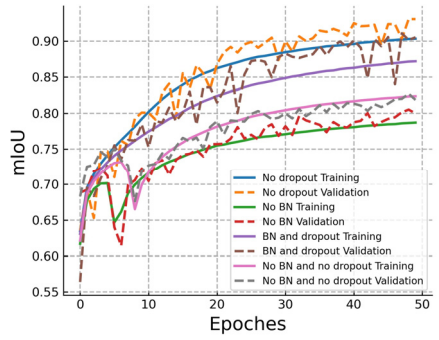
4.2 Comparison experiments of segmentation network architectures

4.2.1 Effects of batch normalization and dropout layers

The architecture of the segmentation network is defined in Fig. 3 where the BN and Dropout layers are attached to the end of every convolutional block. The use of BN layers can accelerate and benefit the training process with regularization. The dropout layers, in general, may reduce the effects of overfitting. However, recent studies (Li *et al.* 2019) show that BN and dropout layers shift the variance of a specific neural unit when the state of the network is transferred from training to testing. This phenomenon is called "variance shift" and therefore may cause unstable numerical problems. Li *et al.* (2019) also claim that the findings may vary depending on different network configurations and datasets, and therefore the use of BN and dropout layers should be investigated with respect to specific networks and datasets.



(a) Loss values of different configurations of segmentation network



(b) mIoU values of different configurations of segmentation network

Fig. 9 Training and validation performances for different configurations of the BN and dropout layers

In this section, three configurations of the network are studied: (1) apply the BN and dropout layers to all the convolution blocks, (2) apply only BN layers to all the convolutional blocks, (3) apply only dropout layers to all the convolutional blocks, and (4) neither BN nor dropout is applied to the network. The training and validation performances of the aforementioned configurations are plotted in Fig. 9 for comparison. According to Figs. 9(a) and 9(b), the lowest validation loss and the highest mIoU value are achieved when excluding all the dropout layers but keeping all the BN layers. Therefore, compared to the network architecture shown in Fig. 3, the optimal architecture adopted in this study excludes all the dropout layers during training.

4.2.2 Effects of different mixture ratio in loss function

To investigate the effect of mixture ratio α in the proposed loss function (Eq. (4)), the segmentation network is trained using α ranging from 0 to 1 with an interval of 0.25, and the corresponding mIoU on the validation crack patches are reported in Table 1. When the mixture ratio is set to $\alpha = 0$, Eq. (4) collapses into a pure BCE loss function. Similarly, when the mixture ratio is set to $\alpha = 1$, Eq. (4) changes to a pure Dice loss function.

It is observed that the mIoU on validation crack patches increases significantly when the Dice coefficient is included in the loss function, compared to pure BCE loss function. The reason is that BCE loss function is just a proxy loss function of the mIoU. However, the Dice coefficient can directly optimize the objective function with respect to mIoU. More detailed discussion regarding the advantages of including Dice coefficient in the loss function is included in Section 2.2. According to Table 1, the highest mIoU is achieved when $\alpha = 0.5$. Therefore, a mixture ratio of 0.5 is used in this study.

4.3 Parametric studies

The performance of the proposed framework was evaluated on 50 high-resolution validation images that were not used during training. Image patches cropped from the test images were fed into the classification network and the cracks score of an image patch was computed by the final Softmax layer in the network.

In Fig. 10, two of the six image patches marked with the red dashed square are crack image patches with scores lower than 0.5, meaning that they are classified as non-crack image patches if the default value of 0.5 is used as the

Table 1 Effect of different mixture ratios on mIoU of validation crack patches

	Mixture ratio (α)				
mIoU	0.00	0.25	0.5	0.75	1.0
	81.48%	91.79%	**92.43%**	92.16%	90.97%

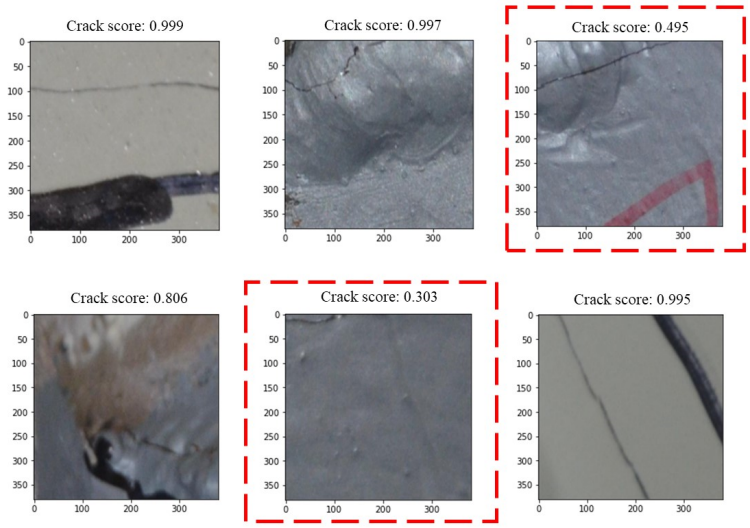


Fig. 10 Samples of Softmax scores (predicted possibilities) of crack patches. The crack patches that have a low Softmax score are marked with red dashed bounding boxes.

Table 2 Effect of classification threshold and Dice loss function on mIoU of high-resolution validation images

		Threshold (θ_T)				
		0.0	0.3	0.4	0.5	0.6
mIoU	With dice	79.43%	81.89%	**82.31%**	81.90%	81.13%
	W/O dice	77.13%	79.42%	80.09%	79.10%	79.4%

decision boundary of the Softmax layer. Therefore, these crack image patches are not fed into the segmentation network to perform the segmentation task. This leads to some missing crack segmentation in the final binary crack mask.

In order to capture more crack areas, we lower the detection threshold θ_T for the crack classifier. When $\theta_T = 0$, every image patch is passed directly to the segmentation network without classifying whether a patch contains cracks or not. Intuitively, this would reduce the false negatives i.e., (missing cracks) at the cost of increasing the false positives. To address this issue, a parametric study on θ_T is performed. Table 2 reports the mIoU on the validation data with different values of θ_T, and the performance of using only the BCE loss versus the proposed customized loss function that combines BCE and the Dice losses. The best mIoU of 82.31% and 80.09% is achieved when the threshold is set to 0.4 with and without Dice loss function, respectively. It is observed that the mIoU drops if every image patch is passed to the modified U-Net (i.e., $\theta_T = 0$), which is due to too many false positives in the final prediction. Therefore, it is necessary to screen the image patches before feeding them into the segmentation network. In addition, it is shown in Table 2 that on average, the mIoU increased by approximately 2%, when the Dice loss is included during training. This validates the effectiveness of the proposed loss function for crack segmentation.

According to Fig. 10, it can be observed that the area of the crack in the image patches is also an important factor that affects the score. The crack classifier does not perform well when a crack only takes up a small portion of an image patch. This can be mitigated by using the overlapping image patches, where the same crack might be observed in a more centered location and takes up more area from a different image patch and, therefore, the crack will be easier for the classifier to detect. This is another reason that overlapping image patches are adopted in the proposed framework.

4.4 Fusion scheme comparison

To illustrate the effectiveness of the proposed Naïve Bayes approach compared to other fusion methods, this study implements a baseline reference using the summation of the crack scores $[s_1^c, s_2^c, s_3^c, \cdots, s_n^c]$ from overlapping patches, and binarizes the score map of the high-resolution image using a threshold θ_{sum}. To this end, a pixel with different crack scores from different image patches is determined as a crack if

$$\sum_{i=1}^{n} s_i^c > \theta_{sum} \tag{10}$$

Table 3 lists the mIoU on the test dataset when using different data fusion schemes. The non-overlapping case represents the situation where non-overlapping image patches are cropped from the high-resolution image. Since

Table 3 mIoU and the corresponding optimized threshold for different data fusion schemes on 50 high-resolution validation images

	Non-overlapping	Overlapping	
		Sum of scores	Naïve Bayes
mIoU	78.81%	80.63%	**82.31%**
Threshold	N/A	3.5	0.37

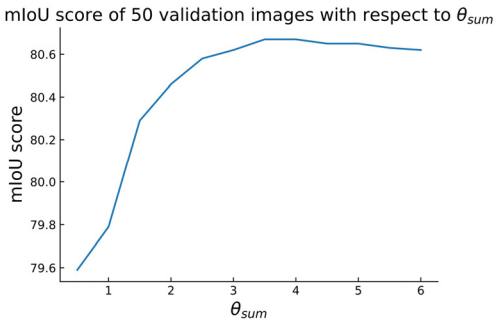
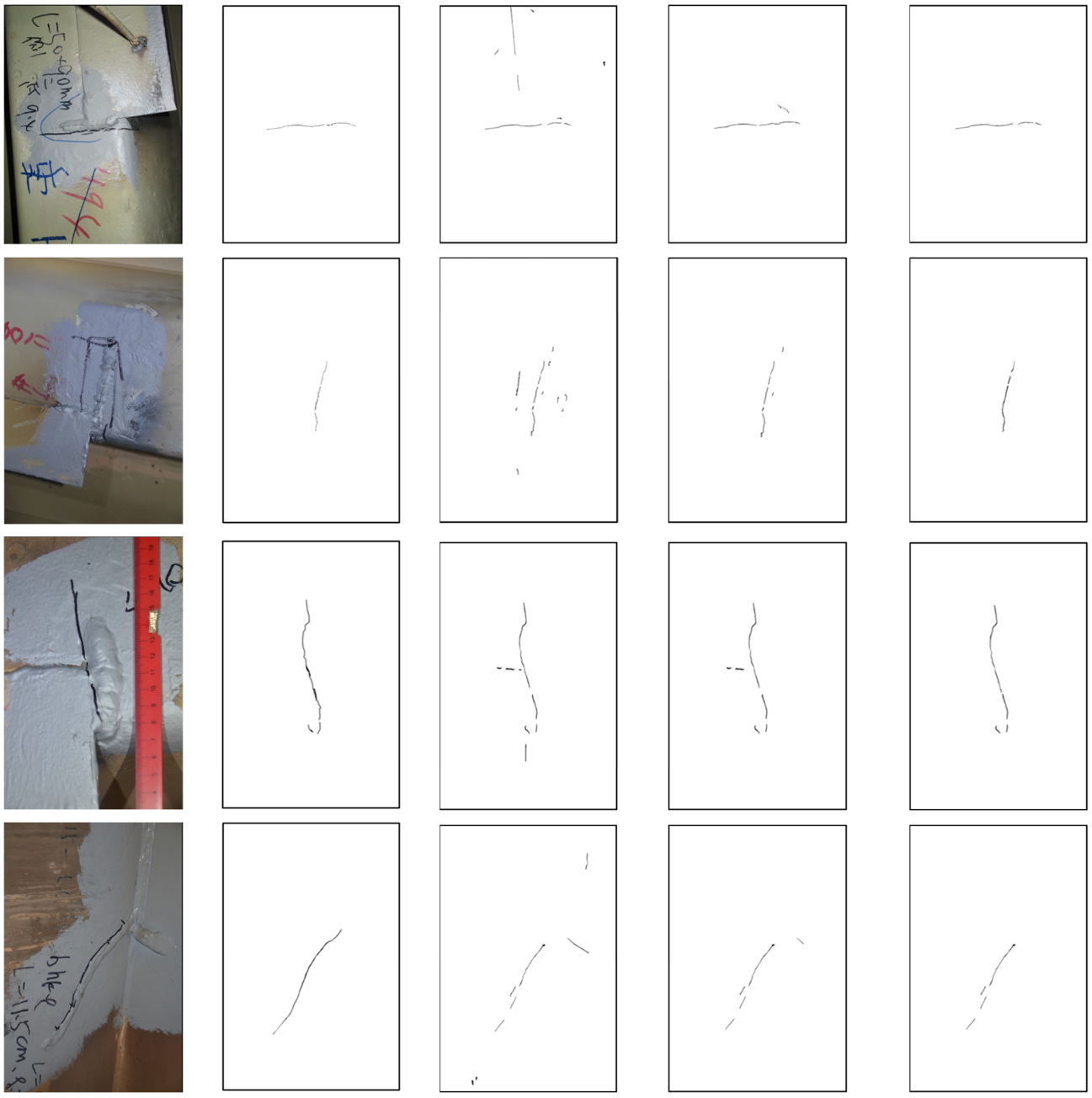


Fig. 11 Parameter sensitivity test: mIoU of 50 validation images with respect to different θ_{sum} values

there is no overlapping region, there is no data fusion incorporated in this case.

The optimal threshold θ_{sum} for the "sum of scores" method is decided by running a parameter scanning of $\theta_{sum} \in [0.5,6]$ at intervals of 0.5 on 50 high-resolution validation images. The validation images are randomly chosen from the high-resolution training images, which are different from the high-resolution test images. The maximum mIoU of the high-resolution test images is obtained at $\theta_{sum} = 3.5$ shown in Fig. 11. The optimal threshold θ_{NB} for Naïve Bayesian data fusion is discussed in Section 2.3. It is observed that the non-overlapping case has the lowest mIoU while the other methods that



(a) Input images (b) Ground Truth (c) Prediction w/o crack classifier (d) Prediction using "sum of scores" fusion (e) Prediction using Naïve Bayes fusion

Fig. 12 Sample data and comparison of the predicted crack masks of different configurations.

Table 4 mIoU for different configurations on high-resolution test images using different training/ validation splits

	U-Net					Ours
Crack classifier?			✓	✓	✓	✓
Dice loss?		✓		✓	✓	✓
Sum of score?					✓	
Naïve Bayes?						✓
mIoU (fold 1)	75.06%	76.34%	77.82%	78.66%	80.51%	**82.14%**
mIoU (fold 2)	73.98%	74.24%	77.22%	77.77%	79.83%	**80.91%**
mIoU (fold 3)	75.07%	76.27%	77.33%	78.86%	80.87%	**82.31%**
mIoU (fold 4)	73.72%	74.51%	75.91%	77.31%	78.66%	**80.78%**
mIoU (fold 5)	74.27%	76.52%	77.27%	78.96%	80.27%	**82.42%**
mIoU averaged across different folds	74.42%	75.58%	77.11%	78.31%	80.03%	**81.71%**

incorporate data fusion have significantly higher mIoU values. This observation justifies the hypothesis that classification of the same object using multiple image patches is more robust than using merely a single image patch. It is also observed that the proposed Naïve Bayes approach achieves better performance, with an improvement of 1.68% on mIoU, compared to the summation of crack scores.

4.5 Segmentation performance comparison

Fig. 12 shows some samples of the high-resolution input images, the corresponding ground truth annotations, and different predicted binary crack masks using different configurations of the proposed approach. According to Fig. 12(e), the proposed approach performs well, where the shape of segmented cracks is very close to the ground truth

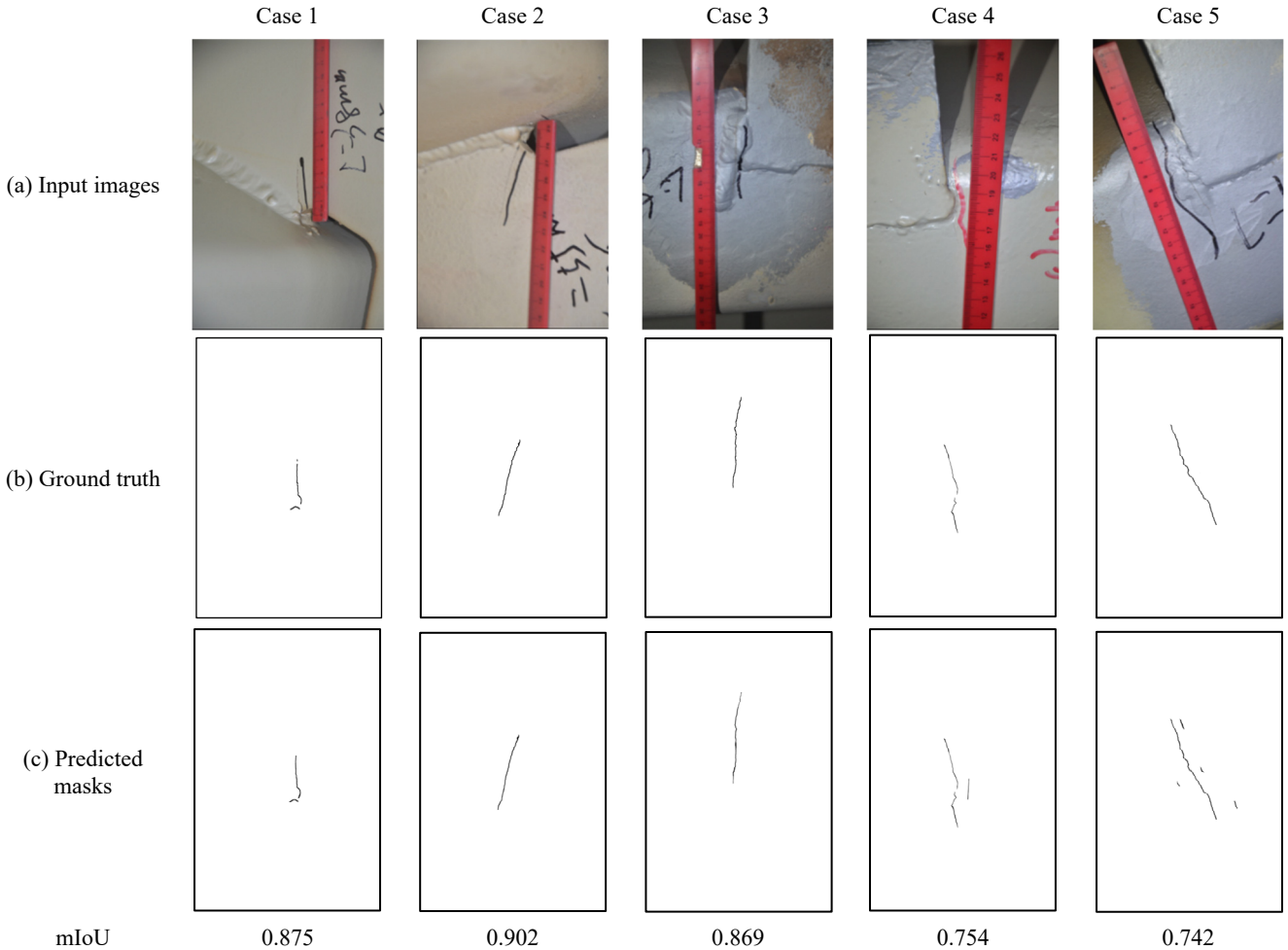


Fig. 13 Further illustrations of the segmentation results for the proposed approach on various environmental conditions

even though some of the cracks are very tiny and the background is noisy and includes stains and paint. The proposed approach rarely recognizes the handwriting and markers as the crack, despite these patterns are very similar to cracks.

It can be observed from Figs. 12(c), (d) and (e) that including a crack classifier before the segmentation network can effectively reduce the false positives. This is because the segmentation network is only trained on crack image patches, and feeding the segmentation network with non-crack patches that contains writings, shadows or paintings may result in false positives since those background noises and cracks sometimes have similar features. The crack classifier can filter those non-crack images out and reduce the false positives. Figs. 12(d) and (e) show that the Naïve Bayes removes more false positives compared to simple score summation. An intuitive explanation is that Bayesian data fusion puts more weights on the crack scores that the network is more confident about when aggregating all the scores from the overlapping patches. This means that the importance of each crack score does not linearly depend on its value s_i^c. Fig. 5(b) implicitly describes the relationship between the importance of each crack score and its value s_i^c: the closer a given score is to 0 or 1, the higher weight it is assigned to the score for decision making. Since the Naïve Bayes data fusion pays more attention to the "reliable" crack scores, the segmentation results are more robust against false positives and false negatives.

To further investigate the effect of different model configurations as well as the splits of training and testing data, five-fold cross validation test is implemented for each different configuration, and the mIoU values are reported for each fold. Table 4 shows how the mIoU performance changes when different configurations are made to enhance the crack segmentation. During the evaluation of different fold, the same θ_{NB}, θ_T and θ_{sum} are used, whose values are 0.37, 0.4 and 3.5 respectively. The U-Net refers to the vanilla U-Net originally proposed in Ronneberger *et al.* (2015), which serves as one of the baseline references used in this study. In the baseline setting, the high-resolution image is simply cropped down to non-overlapping image patches and each image patch is fed into the vanilla U-Net. For each image patch, a binary crack mask is predicted by the U-Net, then the small binary masks are stitched together without fusion to generate the binary crack mask of the high-resolution image. Compared with vanilla U-Net, the Dice loss and the crack classifier prior to the segmentation network gives a boost of 1.18% and 3.17%, respectively. Using Dice loss and crack classifier at the same time gives an improvement of 3.89% over the mIoU. When "sum of scores" data fusion is used together with Dice loss and the crack classifier, an improvement of 5.87% over the mIoU is achieved. Finally, when replacing "sum of scores" data fusion with Naïve Bayesian data fusion, the proposed approach achieves an 80.12% mIoU averaged across different training/test splits, which is 7.14% higher than the vanilla U-Net.

More examples of the proposed framework evaluated on images from various environmental conditions are illustrated in Fig. 13. Case 1 is related to the situation where the crack is thin and relatively small. Case 2 refers to the situation where the crack image is blurred in the images. Case 3 shows a situation where the welding crowns with crack-like features are close to the real crack. In Case 4 and 5, the proposed approach successfully extracts the cracks but also introduces some tiny false positives. These false positives mainly belong to the shadow of the ruler or the markers around the crack, which are hard to eliminate due to their crack-like features. It should be noted that, despite all the noisy background and blurriness arised from the complex scenes, the proposed approach performs well and successfully segments the cracks out of the images, with relatively high mIoU scores ranging from 0.742 to 0.902.

5. Conclusions

In this study, a cascaded framework that utilizes a crack classifier based on a deep CNN and a modified U-Net is proposed to segment the cracks in high-resolution inspection images collected from steel girders. To reduce the false positives and false negatives, a Naïve Bayes data fusion approach is proposed to aggregate the segmentation results from different overlapping image patches. The original high-resolution images are first cropped into small overlapping patches of size 448 × 448 and fed into the crack classifier to determine whether the patch contains a crack or not. Next, a modified U-Net is employed to generate the pixel-level crack score map for each crack image patch. At last, the Naïve Bayesian data fusion registers different overlapping image patches to the global coordinate of the high-resolution image, aggregates the crack scores for each pixel from overlapping patches, and computes the posterior probability of the pixel being a crack. The final crack pixels of the original high-resolution image are obtained if the posterior surpasses a certain threshold. Comprehensive experiments and parametric studies are conducted to discuss the effect of the incorporation of the deep CNN crack classifier, the decision threshold of the crack classifier, the design of the upsampling layer and the loss function for the segmentation network, as well as the effects of different data fusion schemes and the associated decision threshold for fusion. Results from a total of 50 high-resolution test images across different training and validation splits have demonstrated that the proposed approach achieves an mIoU of 81.71%, which is 7.29% higher than the baseline reference implemented with the vanilla U-Net. This shows the robustness of the proposed approach in crack segmentation with high-resolution images containing complex scenes and various noisy patterns.

One potential disadvantage is that the proposed approach is computationally more expensive due to the fact that overlapping image patches are used. The computational time for a single high-resolution image using non-overlapping image patches is 6.15 seconds on average with a step size of 448 pixels. When the proposed approach is used, the computational time increases to 50.23 seconds with a step size of 130 pixels. Since the step size of the proposed approach is approximately 3 times smaller than the non-overlapping case, the number of image patches that

needs to be processed using the proposed method is 9 times bigger than the non-overlapping case, which leads to the increasing computational time. It is noted that the proposed approach is not optimized for computational efficiency. Therefore, improving the computational efficiency using image patch batches and other techniques can be included as part of the future work.

Acknowledgments

The authors would like to thank the organization of the IPC-SHM 2020: Harbin Institute of Technology, University of Illinois at Urbana-Champaign and ANCRiSST for providing the valuable data used in this study.

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)

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. (2021), "Machine learning paradigm for structural health monitoring", *Struct. Health Monitor.*, **20**(4), 1353-1372. https://doi.org/10.1177/1475921720972416

Bao, Y., Chen, Z., Wei, S., Xu, Y., Tang, Z. and Li, H. (2019), "The state of the art of data science and engineering in structural health monitoring", *Engineering*, **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.*, **20**(4), 2229-2239. https://doi.org/10.1177/14759217211006485

Beckman, G.H., Polyzois, D. and Cha, Y.J. (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

Cha, Y.J., 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.C. and Jahanshahi, M.R. (2017), "NB-CNN: Deep learning-based crack detection using convolutional neural network and Naïve Bayes data fusion", *IEEE Transact. Indust. Electron.*, **65**(5), 4392-4400. https://doi.org/10.1109/TIE.2017.2764844

Chen, F.C. and Jahanshahi, M.R. (2019), "NB-FCN: Real-time accurate crack detection in inspection videos using deep fully convolutional network and parametric data fusion", *IEEE Transact. Instrument. Measur.*, **69**(8), 5325-5334. https://doi.org/10.1109/TIM.2019.2959292

Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K. and Yuille, A.L. (2017a), "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs", *IEEE Transact. Pattern Anal. Mach. Intell.*, **40**(4), 834-848. https://doi.org/10.1109/TPAMI.2017.2699184

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

Chen, L.C., Zhu, Y., Papandreou, G., Schroff, F. and Adam, H. (2018), "Encoder-decoder with atrous separable convolution for semantic image segmentation", *Proceedings of the European Conference on Computer Vision* (*ECCV*), Munich, Germany, September, pp. 801-818.

Cheng, H.D., Chen, J.R., Glazier, C. and Hu, Y.G. (1999), "Novel approach to pavement cracking detection based on fuzzy set theory", *J. Comput. Civil Eng.*, **13**(4), 270-280. https://doi.org/10.1061/(ASCE)0887-3801(1999)13:4(270)

Cheng, H.D., Shi, X.J. and Glazier, C. (2003), "Real-time image thresholding based on sample space reduction and interpolation approach", *J. Comput. Civil Eng.*, **17**(4), 264-272. https://doi.org/10.1061/(ASCE)0887-3801(2003)17:4(264)

Choi, W. and Cha, Y.J. (2019), "SDDNet: Real-time crack segmentation", *IEEE Transact. Industr. Electron.*, **67**(9), 8016-8025. https://doi.org/10.1109/TIE.2019.2945265

Ciregan, D., Meier, U. and Schmidhuber, J. (2012), "Multi-column deep neural networks for image classification", *IEEE Conference on Computer Vision and Pattern Recognition*, *Providence*, Rhode Island, USA, June, pp. 3642-3649.

Deng, J., Lu, Y. and Lee, V.C.S. (2020), "Concrete crack detection with handwriting script interferences using faster region-based convolutional neural network", *Comput.-Aided Civil Infrastr. Eng.*, **35**(4), 373-388. https://doi.org/10.1111/mice.12497

Ding, L., Zhang, J. and Bruzzone, L. (2020), "Semantic segmentation of large-size VHR remote sensing images using a two-stage multiscale training architecture", *IEEE Transact. Geosci. Remote Sensing*, **58**(8), 5367-5376. https://doi.org/10.1109/TGRS.2020.2964675

Dumoulin, V. and Visin, F. (2016), "A guide to convolution arithmetic for deep learning", arXiv preprint. https://arxiv.org/abs/1603.07285

Dung, C.V. (2019), "Autonomous concrete crack detection using deep fully convolutional neural network", *Automat. Constr.*, **99**, 52-58. https://doi.org/10.1016/j.autcon.2018.11.028

Fujita, Y. and Hamamoto, Y. (2011), "A robust automatic crack detection method from noisy concrete surfaces", *Mach. Vis. Applicat.*, **22**(2), 245-254. https://doi.org/10.1007/s00138-009-0244-5

He, K., Zhang, X., Ren, S. and Sun, J. (2015), "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification", *Proceedings of the IEEE International Conference on Computer Vision*, Boston, MA, USA, June, pp. 1026-1034.

He, K., Zhang, X., Ren, S. and Sun, J. (2016), "Deep residual learning for image recognition", *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, NA, USA, June, pp. 770-778.

Ioffe, S. and Szegedy, C. (2015), "Batch normalization: Accelerating deep network training by reducing internal covariate shift", *Proceedings of International Conference on Machine Learning*, Lille, France, June, pp. 448-456.

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. (2013), "An innovative methodology for detection and quantification of cracks through incorporation of depth perception", *Mach. Vis. Applicat.*, **24**(2), 227-241. https://doi.org/10.1007/s00138-011-0394-0

Ji, A., Xue, X., Wang, Y., Luo, X. and Xue, W. (2020), "An integrated approach to automatic pixel-level crack detection and quantification of asphalt pavement", *Automat. Constr.*, **114**, 103176. https://doi.org/10.1016/j.autcon.2020.103176

Kingma, D.P. and Ba, J. (2014), "Adam: A method for stochastic optimization", arXiv preprint. https://arxiv.org/abs/1412.6980

LeCun, Y., Bengio, Y. and Hinton, G. (2015), "Deep learning", *Nature*, **521**(7553), 436-444. https://doi.org/10.1038/nature14539

Lee, B.J., Shin, D.H., Seo, J.W., Jung, J.D. and Lee, J.Y. (2011), "Intelligent bridge inspection using remote controlled robot and image processing technique", *International Symposium on Automation and Robotics in Construction*, Seoul, Korea, June.

Li, X., Chen, S., Hu, X. and Yang, J. (2019), "Understanding the disharmony between dropout and batch normalization by variance shift", *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Long Beach, CA, USA, June, pp. 2682-2690.

Lim, R.S., La, H.M. and Sheng, W. (2014), "A robotic crack inspection and mapping system for bridge deck maintenance", *IEEE Transact. Automat. Sci. Eng.*, **11**(2), 367-378. https://doi.org/10.1109/TASE.2013.2294687

Lin, M., Chen, Q. and Yan, S. (2013), "Network in network", arXiv preprint. https://arxiv.org/abs/1312.4400

Lin, T.Y., Goyal, P., Girshick, R., He, K. and Dollár, P. (2017), "Focal loss for dense object detection", *Proceedings of the IEEE International Conference on Computer Vision*, Venice, Italy, October, pp. 2980-2988.

Liu, Y., Ren, Q., Geng, J., Ding, M. and Li, J. (2018), "Efficient patch-wise semantic segmentation for large-scale remote sensing images", *Sensors*, **18**(10), 3232. https://doi.org/10.3390/s18103232

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

Liu, Z., Cao, Y., Wang, Y. and Wang, W. (2019b), "Computer vision-based concrete crack detection using U-net fully convolutional networks", *Automat. Constr.*, **104**, 129-139. https://doi.org/10.1016/j.autcon.2019.04.005

Liu, J., Yang, X., Lau, S., Wang, X., Luo, S., Lee, V.C.S. and Ding, L. (2020), "Automated pavement crack detection and segmentation based on two-step convolutional neural network", *Comput.-Aided Civil Infrastr. Eng.*, **35**(11), 1291-1305. https://doi.org/10.1111/mice.12622

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*, Boston, MA, USA, June, pp. 3431-3440.

Maeda, H., Sekimoto, Y., Seto, T., Kashiyama, T. and Omata, H. (2018), "Road damage detection and classification using deep neural networks with smartphone images", *Comput.-Aided Civil Infrastr. Eng.*, **33**(12), 1127-1141. https://doi.org/10.1111/mice.12387

Nair, V. and Hinton, G.E. (2010), "Rectified linear units improve restricted boltzmann machines", *Proceedings of International Conference on Machine Learning*, Haifa, Israel, June.

Oh, J.K., Jang, G., Oh, S., Lee, J.H., Yi, B.J., Moon, Y.S., Lee, J.S. and Choi, Y. (2009), "Bridge inspection robot system with machine vision", *Automat. Constr.*, **18**(7), 929-941. https://doi.org/10.1016/j.autcon.2009.04.003

Ren, S., He, K., Girshick, R. and Sun, J. (2015), "Faster r-cnn: Towards real-time object detection with region proposal networks", *Adv. Neural Inform. Process. Syst.*, **28**, 91-99. https://proceedings.neurips.cc/paper/2015/file/14bfa6bb14875e45bba028a21ed38046-Paper.pdf

Ronneberger, O., Fischer, P. and Brox, T. (2015), "U-net: Convolutional networks for biomedical image segmentation", *International Conference on Medical Image Computing and Computer-Assisted Intervention*, October, pp. 234-241.

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

Tasar, O., Tarabalka, Y. and Alliez, P. (2019), "Incremental learning for semantic segmentation of large-scale remote sensing data", *IEEE J. Select. Topics Appl. Earth Observ. Remote Sensing*, **12**(9), 3524-3537. https://doi.org/10.1109/JSTARS.2019.2925416

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. (2019), "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

Xue, Y. and Li, Y. (2018), "A fast detection method via region-based fully convolutional neural networks for shield tunnel lining defects", *Comput.-Aided Civil Infrastr. Eng.*, **33**(8), 638-654. https://doi.org/10.1111/mice.12367

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

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

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