Smart Structures and Systems, *Vol. 29, No. 1 (2022) 237-250*
https://doi.org/10.12989/sss.2022.29.1.237

Synthetic data augmentation for pixel-wise steel fatigue crack identification using fully convolutional networks

Guanghao Zhai*[1], Yasutaka Narazaki [2a], Shuo Wang [1b], Shaik Althaf V. Shajihan [1c] and Billie F. Spencer Jr. [1d]

[1] *Department of Civil and Environmental Engineering, University of Illinois at Urbana-Champaign, Urbana, IL, 61801, USA*
[2] *Zhejiang University - University of Illinois at Urbana-Champaign Institute, Zhejiang University, Haining, Zhejiang 314400, China*

(*Received May 9, 2021, Revised June 24, 2021, Accepted September 28, 2021*)

Abstract. Structural health monitoring (SHM) plays an important role in ensuring the safety and functionality of critical civil infrastructure. In recent years, numerous researchers have conducted studies to develop computer vision and machine learning techniques for SHM purposes, offering the potential to reduce the laborious nature and improve the effectiveness of field inspections. However, high-quality vision data from various types of damaged structures is relatively difficult to obtain, because of the rare occurrence of damaged structures. The lack of data is particularly acute for fatigue crack in steel bridge girder. As a result, the lack of data for training purposes is one of the main issues that hinders wider application of these powerful techniques for SHM. To address this problem, the use of synthetic data is proposed in this article to augment real-world datasets used for training neural networks that can identify fatigue cracks in steel structures. First, random textures representing the surface of steel structures with fatigue cracks are created and mapped onto a 3D graphics model. Subsequently, this model is used to generate synthetic images for various lighting conditions and camera angles. A fully convolutional network is then trained for two cases: (1) using only real-word data, and (2) using both synthetic and real-word data. By employing synthetic data augmentation in the training process, the crack identification performance of the neural network for the test dataset is seen to improve from 35% to 40% and 49% to 62% for intersection over union (IoU) and precision, respectively, demonstrating the efficacy of the proposed approach.

Keywords: fully convolutional networks; semantic segmentation; steel fatigue crack; synthetic data

1. Introduction

Civil infrastructure is critical to the wellbeing and economic development of modern societies. However, much of this infrastructure is inadequately maintained. For example, the transportation system, which is an essential component of the nation's infrastructure in the United States, has more than 617,000 bridges; approximately 7.5% of these bridges are classified as in poor condition (ASCE's 2021 infrastructure report card). To properly maintain civil infrastructure and ensure its safety and functionality, appropriate inspection and condition assessment is required. Structural health monitoring (SHM) offers an alternative to traditional inspection and assessment methods that can provide improved real-time information about the condition of a structure with reduced cost (Farrar and Worden 2012).

One of the significant outward signs of damage in a structure is cracking. In both concrete and steel structures, excessive and cyclic loading can cause cracking damage that is often seen on the surface of structural members (Mohan and Poobal 2018). These cracks can be easily overlooked, potentially leading to catastrophic failures and loss of life. For example, the Alexander L. Kielland, a Norwegian semi-submersible drilling rig, capsized in the Ekofisk oil field in March 1980, owing to a fatigue crack in one of its six bracings, killing 123 people. Therefore, researchers have devoted considerable effort to detect and monitor cracks in structures. For concrete structures, visual inspections are widely used for surface crack detections. Although visual inspection is easy and effective, the involvement of human inspectors and the existence of hard-to-reach regions can pose safety risks to the inspectors, as well as being subjective and inefficient. For steel structures, dye penetrants and magnetic particles can be applied as simple indicators of cracks, but these methods are tedious and surface pretreatment is necessary. Eddy current testing is an effective non-contact method for steel crack detection, but extensive experience and professional insights are needed for the interpretation of collected data. Phased array ultrasonic testing (PAUT) employs an array of individual transducers to generate a composite ultrasound beam, enabling the creation of ultrasonic images. However, PAUT is expensive because of the utilization of multiple sensors. In conclusion, the approaches mentioned above are limited in applicability and can have one or more of the following drawbacks: 1) the approach is labor intensive and time

*Corresponding author, Ph.D. Candidate,
E-mail: gzhai4@illinois.edu
[a] Assistant Professor
[b] Ph.D. Candidate
[c] Ph.D. Candidate
[d] Professor

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

consuming, 2) the approach needs close access to the target structure, 3) the approach requires sensors to be installed on the structure, which can be expensive and inconvenient, 4) the approach requires specific expertise in data collection and/or interpretation, and 5) the approach only provides localized assessment. More information on non-destructive crack detection methods can be found in the book written by Bray and Stanley (1996).

Computer vision (CV) and machine learning (ML) techniques offer enormous potential for structural health monitoring of civil infrastructure. Images and videos are two major data formats used to proceed with noncontact monitoring (Narazaki *et al*. 2020a). Graphic data can capture the actual structural condition in the field of view. The data can be collected from various pre-determined positions such that it covers the details of the entire structure. Subsequently, arranging it along the time-axis captures the three-dimensional change, performing in a manner similar to the human eyes in decoding and extracting visual information (Spencer *et al*. 2019). The critical concept in moving towards condition assessment relies upon feature extraction from crack patterns captured as RGB pixels. These approaches offer a convenient, precise, and visualized manner to identify damage. CV and ML have also been applied to detect and localize cracks in structures.

Jahanshahi *et al*. (2009) provides an extensive review of the early use of CV methods for concrete crack detection. Several of the prominent methods cited for crack detection in concrete structures include: the application of edge detection filters using the canny method (Abdel-Qader *et al*. 2003), and the top-hat method (Giakoumis *et al*. 2005). In later research, Liu *et al*. (2016) projects the 2D crack images to 3d reconstruction scene to avoid the restriction during data collection. Field applications of automated crack detection have also been reported for inspection of concrete bridges (Prasanna *et al*. 2014), and (Adhikari *et al*. 2014). In addition, Yeum and Dyke (2015) explored the potential of CV for fatigue crack detection in steel structures. The damage is pre-located by the damage sensitive area (e.g., bolt holes), which overcomes some difficulties for steel fatigue crack identification. These methods only use CV methods, which requires significant computational time and limits identification accuracy.

Recent developments in ML have enabled improved performance and accuracy in crack identification. For example, Zhang *et al*. (2017) proposed CrackNet for the semantic segmentation of pavement cracks. Hoskere *et al*. (2018) used fully convolutional neural networks (FCN) (Long *et al*. 2015) for concrete damage segmentation, including crack and spalling. Bao *et al*. (2019) reviewed crack identification approaches for steel surface, including the application of restricted boltzmann machines (RBMs) (Xu *et al*. 2018). Xu *et al*. (2019) applied a region-based convolutional neural network to identify fatigue crack in steel box girders. The studies indicate the promising potential offered by pixel-level segmentation-based approaches for fatigue crack identification. However, application of pixel-level segmentation-based crack identification in steel structures is still a challenging task, due primarily to the lack of adequate data to train the ML algorithms (Frid-Adar *et al*. 2018), which is further complicated by small width of cracks in metallic structures and the noisy and diverse steel textured environment (Jahanshahi *et al*. 2017).

The performance of convolutional neural networks (CNN) heavily relies on the size of the dataset. Shorten and Khoshgoftaar (2019) presented a survey of the existing data augmentation approaches, where data augmentation methods are classified into basic image manipulations and deep learning approaches. The basic image manipulations are mainly conducted by a series of geometric transformations, such as flipping, rotating, and cropping. Although those geometric transformations increase the apparent size of the dataset, the diversity of the augmented dataset is not significantly improved. Therefore, several deep learning approaches have been proposed to augment data based on the features, such as the feature maps of a CNN. For example, Bowles *et al*. (2018) introduced the implementation of generative adversarial networks (GANs) (Goodfellow *et al*. 2014) on feature space augmentation, which generates images using a generator network and evaluate those images using discriminator network. However, producing high-resolution images needed for structural damage assessment is not straightforward. Moreover, large dataset is needed to train GANs successfully, hindering the application to structural damage (rare events).

Some researchers have proposed the use of synthetic data to supplement real-world images. For example, Ros *et al*. (2016) proposed such synthetic data augmentation for autonomous navigation purposes. By combining the synthetic and real-world data together for training, improved accuracy was achieved. Narazaki *et al*. (2021) also employed synthetic data augmentation to improve the identification of concrete damage in Japanese high-speed railway viaducts. One of the advantages of this approach is that the annotation of synthetic data is automatically rendered using masks for different classes which show their location on the image, thus avoiding the laborious manual image labelling process of images. The existing work about synthetic data augmentation offers significant potential to address the lack of data problem in ML; however, the approach has not been investigated for steel fatigue cracks with different visual properties than those of concrete damage, requiring a new formulation for data generation. Moreover, quantitative discussions of the effectiveness of such synthetic data augmentation are limited in the literature.

In this paper, synthetic data augmentation is proposed to achieve effective pixel-wise identification of fatigue cracks in steel structures by leveraging fully convolutional neural networks (FCN). Based on images from an open dataset of crack bridge girders provided by 1st International Project Competition for Structural Health Monitoring (IPC-SHM 2020), which is designated herein as the real-world dataset, a procedure is proposed for generating synthetic textures that mimic the observed damage patterns. Textures are mapped to an object in the 3D synthetic environment that produce images and associated precise ground truth label

for the structure. The synthetic images are leveraged to augment the original real-world dataset (designated herein as the augmented dataset) to develop diversity in the training images. The FCN is then applied to identify the unique features of fatigue cracks. The algorithm produces pixel-wise labels, rather than image-wise labels or bounding boxes. The proposed approach is applied to the problem of identifying fatigue cracks in steel box girders using images from the open dataset (IPC-SHM 2020). First, an FCN is trained on the real-world data. Then, 616 synthetic images of fatigue cracks are generated and rendered at a resolution of 4928×3264, and a second FCN is trained on the real-world data, augmented with the synthetic data (designated herein as FCN-A). The performance of FCN-A offers 5% and 13% improvement for the intersection over union (IoU) and precision, respectively, as compared to FCN-R (i.e., the case using only the real-world data). The proposed method shows promises for improving the accuracy and performance of crack detection in field conditions for SHM, particularly when limited data is available.

2. Synthetic data generation

The synthetic data is based on observations from the real-world dataset containing images of fatigue cracks in steel box girders (IPC-SHM 2020). A sample image is shown in Fig. 1. The main components of the image data of the box girders are: (i) fatigue cracks, (ii) welds, (iii) pen marks, and (iv) paint. The cracks and welds are mostly represented by the geometric information. The pen marks, including numbers and letters, can be found in red, black, and blue in the real-world dataset. The paint on the surface is observed to be either yellow or silver. The markings and painted surfaces contribute to the differences in the color map.

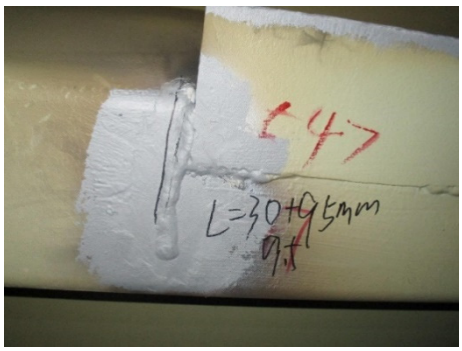


Fig. 1 Example of real-world data

To generate synthetic data possessing these attributes, four parts are required: (1) mesh generation, (2) mesh texturing, (3) texture generation, and (4) data collection. The entire modeling process is implemented using Blender, an open-source computer graphics modeling software, and automated using the Blender-Python API. Details of each aspect of the generation of the synthetic data are presented in the following subsections.

2.1 Mesh generation

In the synthetic environment, a 3D mesh is first created to represent the geometry of each object. In this research, the objects to be considered are: (i) the steel box girder, which is represented as a rectangular prism with beveled corners, and (ii) a steel plate placed behind the prism, which serves as a simplified model of the background. A point light source is placed above the models to simulate the lighting conditions and shadows in the real-world inspection environment. An example of the mesh is shown in Fig. 2(b). To increase the diversity of synthetic data, the location, dimension, and orientation of the prism relative to the plate are randomized and follow a uniform distribution. Generation of the 3D mesh is implemented through the Blender-Python API.

2.2 Mesh texturing

The mesh is textured using a physically based rendering approach, in which various visual effects in the mesh surface are controlled by defining associated texture maps, such as RGB color maps, normal maps, roughness maps, and metallic maps (Burley & Disney, 2012). A color map, also called albedo, is an RGB image that determines the amount of light reflected at each part of the surface. For example, humans perceive the color red if the reflected light has a wavelength that is primarily around 700 nm (Szeliski 2010); therefore, parts of the surface that will reflect this wavelength of light are modeled as red in the color map. The normal map contains the height information of a surface. The intensity in the normal map represents the direction of the vector that is normal to the surface at each point. Height changing and realistic shading are the primaries uses of this normal vector. The roughness and metallic maps play an important role in making the texture

(a) Objects (b) Mesh of the cube

Fig. 2 The 3D mesh in synthetic environment

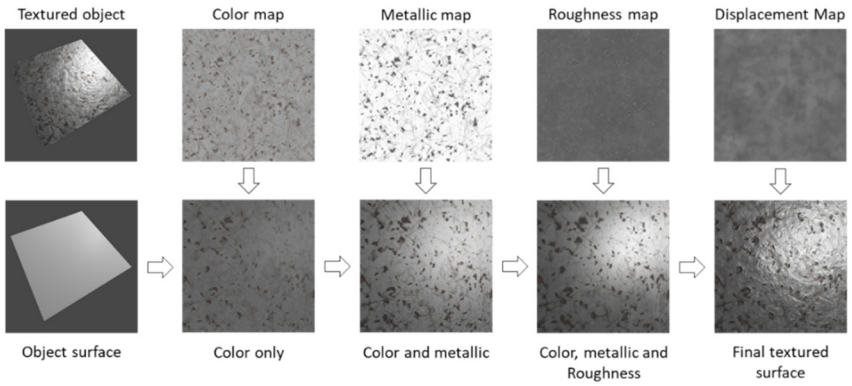


Fig. 3 Texture influence

more realistic. The roughness map controls a statistical model of light reflection caused by fine surface variations that cannot be modeled by the normal map. The roughness ranges from 0 to 1, where a value of 0 implies a perfectly smooth surface. The metallic factor controls the specular reflection of the surface (0: diffuse surface, 1: metallic surface). In Fig. 3, the influences of different texture maps are displayed by an example of a damaged metal surface.

The texture maps are applied to the mesh surface using the Blender Principle bidirectional scattering distribution function (BSDF) node through the Blender Shader Editor. In this paper, displacement maps are used to define the height intensity at each point directly instead of normal maps, which can be transferred to 3D mesh displacement by the Displacement node to render the realistic shading. All texture maps for the final layout are assembled as shown in Fig. 4. By combining all the texture maps using this approach, the desired visual effects of the surface can be realized.

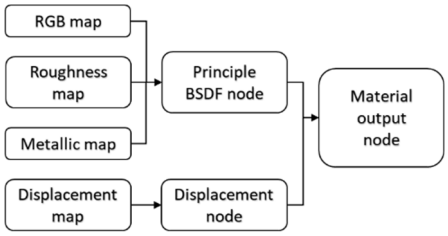


Fig. 4 Texture nodes in Shader editor

2.3 Texture map generation

This section describes a method for generating realistic random texture maps mentioned in the previous section to represent the surface of steel structure with fatigue cracks and welds. The method is an extension of the existing approach for creating random concrete damage textures, which characterizes a realistic random texture by three components: (1) random geometry, (2) random displacement, and (3) texture discontinuity (Narazaki *et al.* 2021). However, steel surfaces which potentially have cracks in them have unique characteristics that make texture development difficult. For example, in contrast with the jagged patterns exhibited by large area of concrete damage, steel fatigue cracks are thin with relatively smooth patterns, and fatigue cracks typically start and propagate along a weld with only minor deviations. Moreover, crack damage patterns coexist with non-damage patterns, such as different paint colors, drawings, and uneven material surfaces, allow of which can potentially mislead a network seeking to recognize cracks in steel structures. Therefore, the synthetic steel fatigue crack has strict requirement on the similarity for texture details such as the shape and location of cracks. To develop an effective synthetic dataset for this research, new procedure and parameterization for the textures must be developed that can realize these characteristics efficiently, as discussed in the remainder of this section.

The random geometric maps of damage are generated using simplex noise (Perlin 2001), which are the superposition of several octaves of noise maps with different spatial frequencies and levels of details. Therefore, simplex noise can represent large-amplitude low-frequency changes (e.g., mountain elevation), as well as fine details (e.g., profiles of rocks and hills). Simplex noise maps are an improvement over Perlin noise (Perlin 1985), which generate random yet smooth patterns, which are particularly useful for game and movie design to create terrain and sea surface profiles. The simplex noise maintains the key features of Perlin noise, with additional advantages of lower computation cost and better continuous gradients. Therefore, simplex noise can represent large-amplitude low-frequency changes (e.g., mountain elevation), as well as fine details (e.g., profiles of rocks and hills).

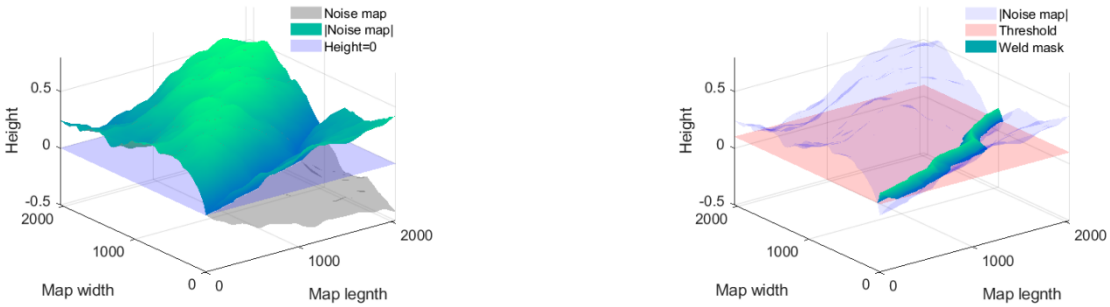
In this paper, a simplex noise texture with resolution of 2048×2048 is generated. The geometric profiles (displacement maps) of cracks and welds are generated synthetically by transforming the simplex noise as follows

$$\mathbf{P} = \text{abs}\left(\boldsymbol{N}\left(freq_x, freq_y, octaves\right)\right) \tag{1}$$

$$\mathbf{M} = 1_{\{P<threshold\}} \tag{2}$$

$$\begin{aligned} &Displacement\ map \\ &= A \times \text{normalize}(\mathbf{P} - threshold)^n \circ \mathbf{M} \end{aligned} \tag{3}$$

where $\boldsymbol{N}\left(freq_x, freq_y, octaves\right)$ is the 2048×2048 simplex noise map that takes values in the range $[-1, 1]$. $\mathbf{P}$ is the absolute noise map generated from simplex noise map



(a) Absolute value of noise map (b) After threshold and flip

Fig. 5 Crack and weld displacement generation

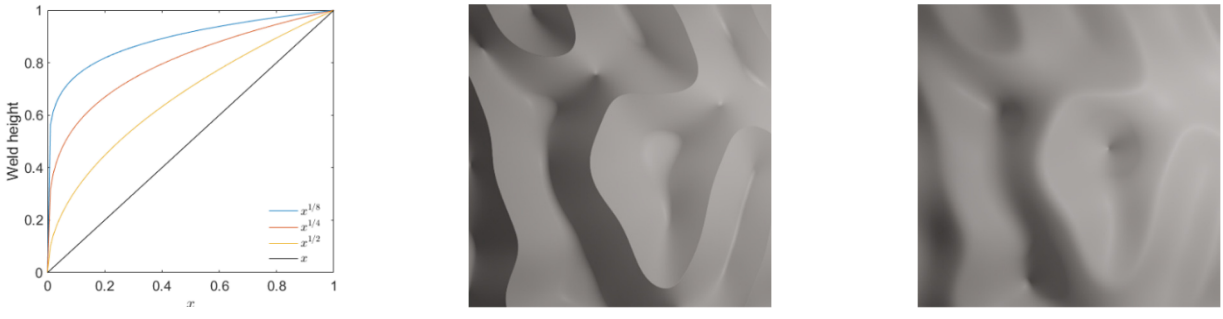
of the noise map, proportional to the number of noise c $\boldsymbol{N}$. $freq_x$, $freq_y$ denote the spatial frequency parameters ycles in an image. $octaves$ defines the number of texture maps that are superimposed, contributing to the complexity and details of the results. **M** denotes a mask that extracts image regions that satisfy the condition defined by the threshold ($1_{\{\cdot\}}$ is an indicator function). In the third equation, the threshold is applied to the noise map, which is then normalized, raised to the nth power, re-scaled by A, and multiplied elementwise by **M** (° denotes element-wise multiplication). Each of these processing steps is illustrated in Fig. 5.

The selection of parameters $freq_x$, $freq_y$, and $octaves$ is a key to realize the thin and smooth patterns representing steel fatigue cracks. To ensure that only one single curve of crack is contained per image, the texture map only cover a single slope of a noise cycle, restricting the frequency parameters not to be too large. In addition, the number of octaves needs to be adjusted so that features such as welds and cracks can be represented adequately. A large number of octaves are used to create the fine details on the smooth outline. To provide sufficient complexity to the synthetic textures, 12 octaves are used for cracks and 8 octaves are used for welds. The influence of octaves on the shape of welds and cracks is shown in Fig. 6.

To obtain the realistic weld profiles, the power function and the gaussian filter are employed. Those operations reduce the sharpness of the ridge of the original pattern generated by Eqs. (1)-(3). The power function is employed to change the large-scale geometry. If the height of the weld is normalized to the interval $[0,1]$, the flat ridges can be obtained by reducing the exponent. The geometrical influence of the power function is shown in Fig. 7(a). The gaussian filter is convolved with the displacement map,

(a) 2 octaves (b) 4 octaves (c) 8 octaves

Fig. 6 Influence of the number of octaves on the shape of welds and cracks



(a) Power function (b) Before smoothing, n = 1 (c) After smoothing, n = 20

Fig. 7 Scaling effect of power function

(a) Same frequency (b) Varied frequency

Fig. 8 Frequency control on weld-crack relative position

after applying the power function to remove fine discontinuities. The displacement of the mesh surface before and after smoothing is shown in Figs. 7(b) and (c).

To ensure that the fatigue crack starts and propagates along the weld, the noise textures that generate crack and weld geometric profiles share the same random seed. Deviations of cracks from weld centerlines are realized by using different spatial frequency parameters ($freq_x$, $freq_y$). The effect of changing the frequency parameters is illustrated in Fig. 8.

Non-crack textures of the steel box girder can mislead the damage identification process. In this paper, different colors of paints, label-like drawings, and uneven surfaces are simulated to create typical textures found in the images of box girders with cracks. To this end, simplex noise textures are used with larger spatial frequency parameters. The resulting curves are used to create color maps that approximate these observed visual effects, including the pen marks on the surfaces. However, generated labels only need to fit manual label features at the local level, because the original image will be cropped before being fed into FCN. Therefore, continuous curves are kept in the final texture to

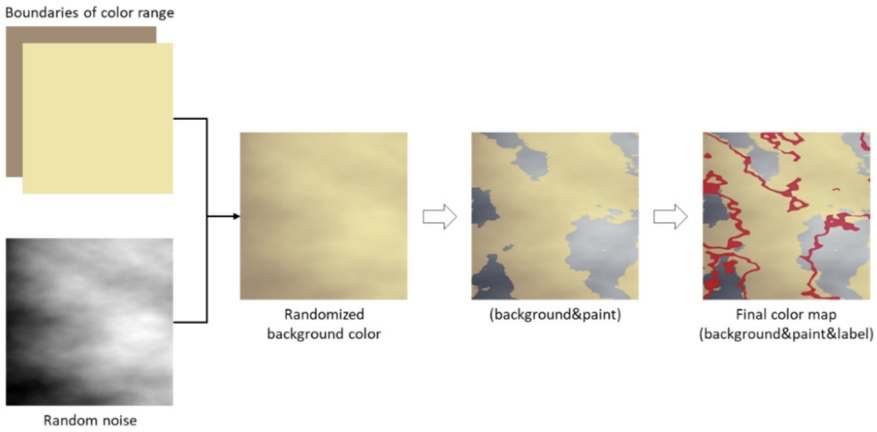


Fig. 9 Example of color map generation

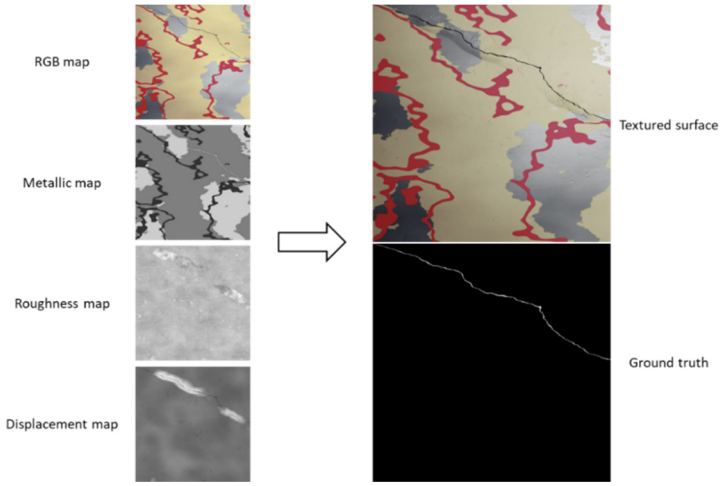


Fig. 10 Example of random texture generation procedure

represent the labels and drawings. The uneven surface also casts shadows that are often misleading for crack detection algorithms. Such effects are mainly caused by scratches, imperfect connection of plates, and varying thickness of coatings and paints. This research uses displacement maps downloaded from a database of 3D scan textures (Textures.com, CC0 Textures) to enable accurate modeling of those textures. These steps yield texture maps that leads to the realistic representation of the non-damage surface.

Texture discontinuity occurs when surfaces with different states (damage, scratches, paint, etc.) are modeled by texture maps. For example, the surface of a steel member may be painted or weathered, while the cracks expose the bare steel texture underneath the surface. This research expresses such texture discontinuity by the RGB color map, roughness map, and metallic map. Color maps are generated based on the images from the image dataset of fatigue cracks found in steel box girders. These surfaces have areas with yellow and silver paint, as well as pen marks in red, blue, and black. For each color, RGB values are obtained manually from the darkest and the brightest parts of these images, as shown in Table 1. To enhance the realism, the associated color map is generated by scaling the noise map, so that the RGB value lie between those extreme colors. The procedure for generating the color map is shown in Fig. 9. Roughness and metallic maps for object surfaces are collected from online databases (Textures.com and CC0 Textures) to represent the detailed and sharp color variation caused by such discontinuities. The approach presented herein is implemented using Python, so that random textures with appropriate values of parameters are created automatically.

Table 1 Color range

Color name:	R channel	G channel	B channel
Yellow	160 - 237	140 - 229	116 - 170
Silver	70 - 205	74 - 209	86 - 210
Red	170 - 185	38 - 70	38 - 90
Blue	19 - 49	47 - 81	95 - 130
Black	23 - 50	23 - 50	28 - 57

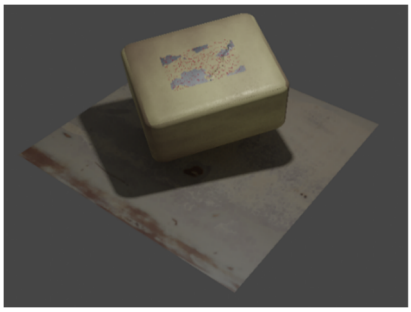
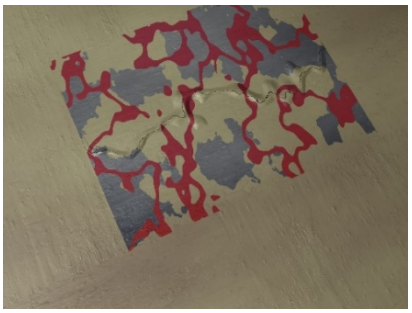
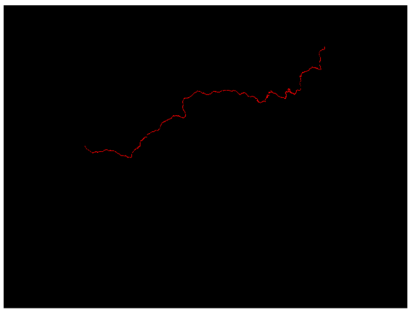


Fig. 11 Example of textured mesh in Blender

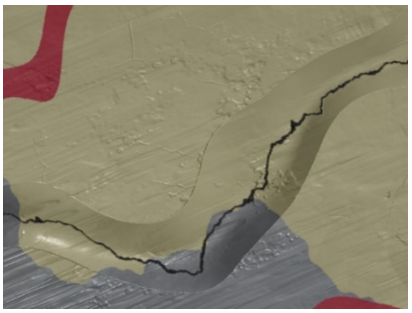
Finally, all maps are combined to generate photo-realistic textures that closely replicate the key characteristics of the original images of fatigue cracks in steel box girders using the Blender Shader Editor, introduced in section 2.2. An example of the effects of combining different maps obtained in this section is shown in Fig. 10. The figures in the left column show four types of texture maps that contain different information about the steel surface. The RGB map represents steel surface, painted surface drawing, and bare steel exposed by cracking. The metallic map shows different specular reflections for associated textures mentioned above. The roughness map combines two texture maps from a real-world dataset for steel and weld surfaces. The displacement map not only



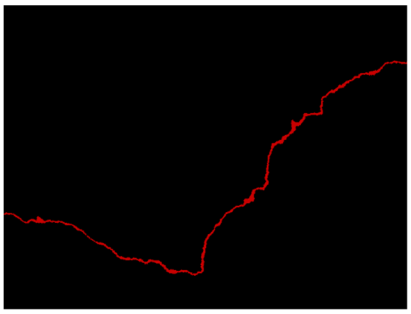
(a) Synthetic texture



(b) Synthetic groundtruth



(c) Detail of synthetic texture



(d) Detail of synthetic groundtruth

Fig. 12 Example of synthetic image pairs

includes the large height changes of welds and cracks introduced in this section, but also contains the fine discontinuities seen on the surfaces, corresponding to roughness maps obtained using 3D scans of steel surfaces. The right column shows the photo-realistic steel surface with all the properties defined in this section, as well as the associated ground truth.

2.4 Rendering synthetic images

The synthetic environment is now generated by superposing the textures from Sections 2.2 and 2.3 onto the mesh created in Section 2. The example of the textured synthetic environment is shown in Fig. 11, based on the mesh in Fig. 12. The images are rendered and stored automatically based on a virtual camera whose position is defined in the Blender-Python API. The optical axis of the cameras aligns with the center of the damaged surface of the steel box girder. The camera distance is variable and is sampled uniformly, so that the width of the crack in the rendered images range from 1 to 5 pixels. After the data collection process, the dataset contains 120 original images and 616 synthetic images, along with the associated ground truth images. The example of synthetic data pairs is shown in Fig. 12.

3. Semantic segmentation based on fully convolutional networks

This section details the implementation of the FCN (Long *et al*. 2015) used herein for steel fatigue crack identification. The FCN is selected to provide pixel-wise identification owing to its advantages in accuracy and simplicity, which has been shown in several field applications (Hoskere *et al*. 2020, Narazaki *et al*. 2020b). The architecture, additional techniques, and the datasets are described in the remainder of this section.

3.1 Network architecture

The FCN employed herein enables end-to-end training and pixel-to-pixel segmentation, with the output of the network being evaluated directly by the ground truth. Considering the size of the dataset, 15 convolutional layers are used in the FCN, with an encoder-decoder architecture. An encoder is built to extract feature maps at multiple resolutions.

The encoder consists of six convolutional layers with filter size of 3×3 The convolution layer is followed by a ReLU activation function, which adds non-linearities to the operation. A pooling layer with a stride of 2×2 is placed after every three convolutional layers to downsample the layer output maps (feature maps). After the last convolutional layer, an average pooling layer reduces the image resolution to one eighth of original size and passes the output into the decoder. The decoder in the FCN replaces the traditional fully connected layers in the standard convolutional neural networks (CNNs). The decoder first applies convolutional layers of 1×1 to the two-dimensional feature maps, leading to the spatial prediction map, which is then upsampled and merged with the feature maps at higher resolutions. Dropout layers are used in the first three convolutional layers of the decoder to suppress the effects of overfitting. This decoder provides a learnable way for upsampling the features and allows pixel-wise prediction. The architecture of the proposed network is shown in Fig. 13. Details of each layer are listed in Table 2.

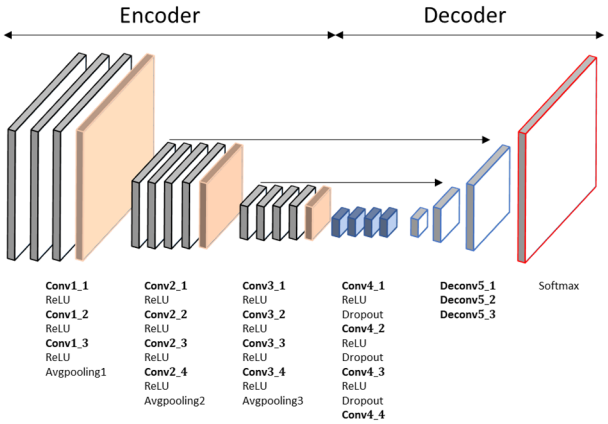


Fig. 13 Network architecture

Table 2 Details of convolutional layers in the network architecture

Layer name:	Filter size	Number of filters	Stride
Conv1_1	$3 \times 3 \times 3$	64	1×1
Conv1_2-3	$3 \times 3 \times 64$	64	1×1
Conv2_1	$3 \times 3 \times 64$	128	1×1
Conv2_2-4	$3 \times 3 \times 128$	128	1×1
Conv3_1	$3 \times 3 \times 128$	256	1×1
Conv3_2-4	$3 \times 3 \times 256$	256	1×1
Conv4_1	$1 \times 1 \times 256$	256	1×1
Conv4_2-3	$1 \times 1 \times 256$	256	1×1
Conv4_4	$1 \times 1 \times 256$	2	1×1
Deconv5_1	$4 \times 4 \times 2$	128	1×1
Deconv5_2	$4 \times 4 \times 128$	64	1×1
Deconv5_3	$4 \times 4 \times 64$	2	1×1

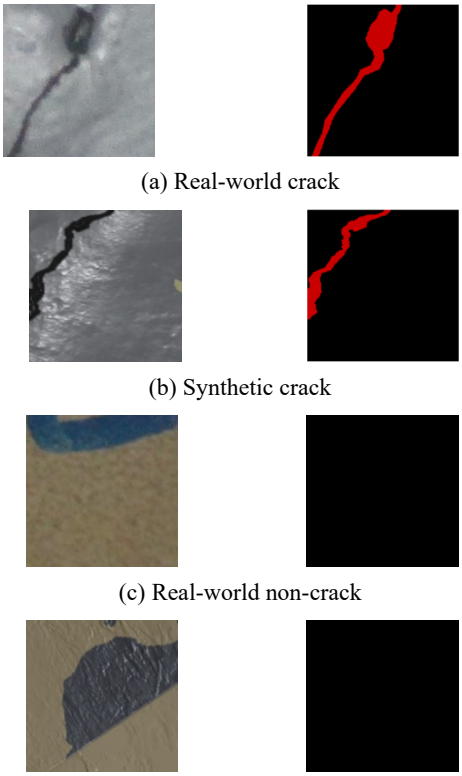
3.2 Additional training considerations

A crack can be described as a narrow, continuous curve, typically covering only a small percentage of the images being considered, but may still span the entire image. To fully capture these features of a crack, two techniques are implemented in this paper: (1) median frequency balancing (Margineantu 2000), (2) image cropping.

Median frequency balancing assigns weights to different classes while calculating the loss by cross entropy. In the image with crack, the ratio of cracks to non-crack pixels is about 1/64, due to which the effect on the loss function on crack pixels is minimal. Therefore, the prediction can converge to the non-crack class for all pixels, because the loss can remain at a reasonably low-level with no crack being predicted; this imbalance of crack pixels will always

(a) Before reweighting (b) After reweighting

Fig. 14 Crack prediction before/after reweighting



(a) Real-world crack

(b) Synthetic crack

(c) Real-world non-crack

(d) Synthetic non-crack

Fig. 15 Cropped image and ground truth

mislead the network. To prioritize the influence of the crack pixels, median frequency balancing is implemented, where the weight for class c is given by $weight(c) = median_{freq}/sum(c)$.

The weighted cross entropy loss given by $L = -\sum_c w(c)p(c)\log\left(\hat{p}(c)\right)$ is employed to place a higher weight on the crack class. Here $p(c)$ denotes the ground truth for class c, $\hat{p}(c)$ defines the probability of class c, and $w(c)$ denotes the weight assigned for class c. With this emphasis on the class cracks, the crack label cannot be ignored, and the expected probability for crack identification is significantly improved during the training process. However, the final prediction score for the class crack is overestimated by median frequency balancing during training, leading to a dramatic increase in false positives, as illustrated in Fig. 14. Therefore, the prediction score should be reweighted by the inverse of the balancing weight applied to the entropy loss calculation for the associated classes.

In addition, because cracks are quite narrow, a great portion of the information is lost during the downsampling process. To address this problem, the original image is cropped into 330 sub-images, each being 224×224 pixels, which are used as the input to the network. This approach not only retains the original crack information, but also allows the network to put more attention to local textures. An example of cropped images and the ground truth are shown in Fig. 15.

3.3 Crack image dataset

The dataset used in this study (designated herein as the real-world dataset) is based on 120 real world images of steel-fatigue cracks of size 4928×3264 and 5152×3864 with labels provided by the 1st International Project Competition for structural health monitoring (IPC-SHM 2020). The real-world images with ground truth are randomly split into training and test datasets. These real-world images are cropped into sub-images with resolution of 224×224 pixels and randomly flipped and rotated to augment the dataset. Subsequently, 616 synthetic images are created and combined with the real-world dataset to create an augmented dataset, which will be used in the training, validation, and testing. The images in the augmented dataset are similarly cropped into sub-images with resolution of 224×224 pixels. FCN-R is trained on the sub-images from the real-world dataset, whereas FCN-A is trained on the augmented dataset. The details of the real-world and augmented dataset are given in Table 3.

3.4 Network training schemes

After image cropping, all the sub-images are classified as either crack or non-crack images. The crack and non-crack are annotated into either 0 or 1, respectively.

Because the number of synthetic images is significantly larger than the number of real-world images, the augmented dataset must be balanced prior to training the FCN; if balancing is not performed, the FCN predictions will be biased toward the synthetic images. To address this problem, the real-world crack sub-images are copied several times to create a 1:1 balance between the real-world crack

Table 3 Details of real-world and augmented datasets

Dataset	Training		Validation		Testing
	Real-world	Synthetic	Real-world	Synthetic	Real-world
Real-world dataset	70	0	10	0	40
Augmented dataset	70	496	10	50	40

sub-images and the sub-synthetic crack images. Specifically, the synthetic dataset has 9971 sub-images that contain cracks, whereas only 2658 sub-images have cracks in the real-world dataset. Therefore, the 2658 real crack images are repeated three times for a total of $2658 \times 4 = 10632$ sub-images, which is comparable to the 9971 synthetic crack sub-images. After this balancing, the number of training epochs for FCN-A is reduced to 1/4 of the epochs for FCN-R, so that both network models will go through and learn from the entire real-world crack data for same number of epochs.

In addition, the number of non-crack sub-images is much higher than the number of crack sub-images, which greatly reduces the efficiency of the training for identifying cracks. Therefore, the ratio of crack and non-crack sub-images is selected to be 1:2 in the training dataset. For the FCN-A discussed herein, the real-world non-crack sub-images are used to form the non-crack sub-images in the training dataset; if the non-crack images in a real-world dataset are insufficient to achieve the desired ratio of crack to non-crack sub-images, then the synthetic non-crack images could be used to supplement the training dataset. Finally, this dataset, termed herein as the balanced augmented dataset, will be used as input to the FCN.

Both the real-world and augmented datasets are then used to train the network. The training process first uses 50 epochs with learning rate of 0.0001 and then 10 epochs with learning rate of 0.00001. The 10 epochs using the lower learning rate help the network converge further based on the main direction determined by the higher learning rate. In each epoch, the network goes through the entire dataset using a batch size of 8 sample images. Because the real-world crack images are repeated four times in the augmented dataset, the number of epochs is reduced to $50/4 = 12.5$ epochs; thus, the number of iterations, as well as all other parameters in training process, is the same for both FCN-R and FCN-A. After the networks are updated based on a batch of data samples, the network is evaluated using the loss of training and validation set to monitor overfitting

4. Results and discussion

The crack identification results for both FCN-R and FCN-A are evaluated using three metrics: 1) precision $\left(\mathrm{TP}/(\mathrm{TP}+\mathrm{FP})\right)$, 2) recall $\left(\mathrm{TP}/(\mathrm{TP}+\mathrm{FN})\right)$, and 3) intersection over union (IoU) $\left(\mathrm{TP}/(\mathrm{TP}+\mathrm{FP}+\mathrm{FN})\right)$, where TP is true positive, FP is false positive, TN is true negative, and FN is false negative. The performance of FCN-A is first evaluated pixel-wise on 70 synthetic that were not used for training. As shown in Table 4, FCN-A achieves accurate identification for the synthetic testing dataset in all the metrices, demonstrating that FCN-A has been influenced significantly by the synthetic images and may be more robust in detecting diverse crack images.

Next, the performance of both FCN-R and FCN-A is evaluated pixel-wise on the real-world testing dataset, the results of which are given in Table 5. FCN-R identifies 95% of cracks at the image-level. The IoU, precision, and recall of FCN-R show promising results for pixel-wise fatigue crack identification, which reach 35%, 49%, and 56%, respectively. FCN-R can meet the requirements of the basic crack identification, though the accuracy has room for improvement. FCN-A further improves the performance of real-world crack identification. The IoU, precision, and recall achieve 40%, 62%, and 52%, respectively. The result shows that synthetic data can augment a limited dataset and enhance the overall accuracy, as shown through 5% increase of IoU value. In addition, the precision is significantly increased, which indicates that the augmented prediction overcomes some difficulties of incorrect identification of the crack-like features. The relatively small reduction in recall is caused by the imperfect representation of synthetic cracks used in this paper, such as limited types of rust colors for the inner cracks and the lack of motion blur. Therefore, recall can be further improved by updating the rust color of the inner cracks and blurring the image to mimic motion blur in the future.

To investigate the bias from synthetic data, another augmented dataset, in which the number of synthetic crack sub-images are twice as many as the number of real-world crack sub-images, is built to train the third FCN (FCN-A2).

Table 4 Pixel-wise performance on 70 synthetic testing images

Training dataset	Augmented (FCN-A)		
	Crack	No crack	Mean
Precision	87%	99%	94%
Recall	93%	99%	96%
IoU	82%	99%	91%

Table 5 Pixel-wise performance on real-world testing dataset

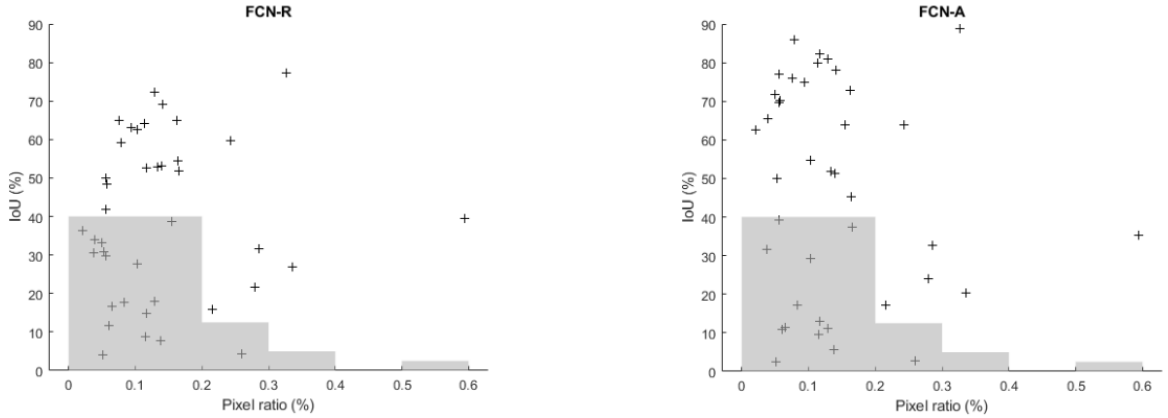
Network	FCN-R			FCN-A		
	Crack	No crack	Mean	Crack	No crack	Mean
Precision	49%	99%	74%	62%	99%	81%
Recall	56%	99%	78%	52%	99%	76%
IoU	35%	99%	68%	40%	99%	70%

Table 6 Pixel-wise performance of FCN-A2 on real-world testing dataset

Training dataset	Augmented (FCN-A2)		
	Crack	No crack	Mean
Precision	51%	99%	76%
Recall	54%	99%	77%
IoU	36%	99%	68%

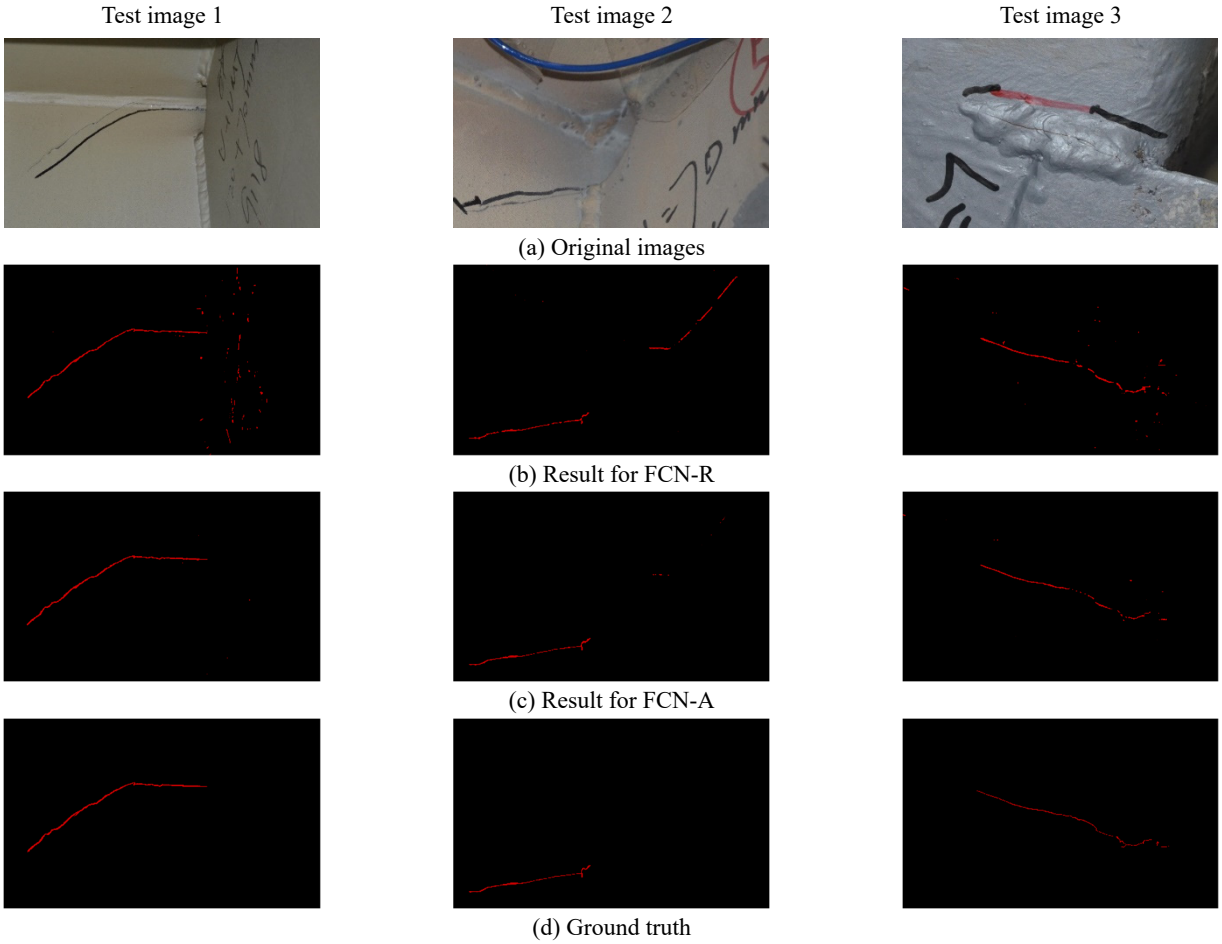
The evaluation of FCN-A2 is shown in Table 6. For FCN-A,the augmented dataset of 1:1 ratio between real-world and synthetic crack sub-images significantly enhances the performance on crack identification. However, compared with FCN-A, excessive ratio of synthetic crack sub-images misleads FCN-A2, resulting in the reduction by 4% of IoU and 11% of precision. Therefore, the best model, FCN-A, is discussed hereafter.

Fig. 16 further presents the variance of the image-wise



(a) FCN-R (b) FCN-A

Fig. 16 Image-wise IoU for testing images with associated pixel ratio of fatigue cracks



(a) Original images

(b) Result for FCN-R

(c) Result for FCN-A

(d) Ground truth

Fig. 17 Example of crack identification result on real-world testing set

Table 7 Image-wise crack identification criteria of example testing results

Network	FCN-R			FCN-A		
	Precision	Recall	IoU	Precision	Recall	IoU
Image 1	63%	76%	53%	93%	88%	82%
Image 2	35%	72%	31%	71%	63%	50%
Image 3	32%	84%	30%	46%	73%	39%

IoU on the testing dataset for both FCN-R and FCN-A. The IoU for each image is plotted against the pixel ratio, which indicates the proportion of crack pixels in the entire image. Comparing the Figs. 16(a) and (b), overall IoU with low pixel ratio is raised by the synthetic data augmentation, except for some extreme cases with low accuracy. The cracks in those cases are rarely found and not sufficiently trained even with augmented dataset.

Fig. 17 provides an example comparing prediction of both FCN-R and FCN-A. Fig. 17(a) shows that FCN-R can clearly identify the fatigue crack on the steel box girder, but a large region of false positives is also observed in the right half of the image 1 and 2. The black marks and shadow are classified as cracks because FCN-R is vulnerable to the misleading feature. In contrast, as demonstrated in Fig. 17(b), FCN-A recognizes the crack-like features correctly, which significantly increase the accuracy of the identification. For test image 3, the effect of synthetic augmentation is weakened with extremely fine crack. The performance of FCN-R and FCN-A are similar for the other images in the testing dataset.

Two types of errors can be observed from the segmentation result in Fig. 17. First, some finer cracks are not detected which make the prediction discontinuous. For example, the image 3 in Fig. 17 indicates the discontinuity effects caused by finer cracks that are narrower than four pixels. Second, certain crack-like features are labelled incorrectly at discrete spots. These issues may be attributed to the relatively simple architecture proposed in this paper; with the application of a network of higher complexity, such issues may be resolved to some degrees.

Overall, FCN-A is more robust in identifying cracks in the presence of diverse features such as different colors, lighting, and shapes. The advantages of the use of synthetic images for augmentation are summarized as follows: (1) more diverse and comprehensive training images, (2) ease of adjustment based on target for identification, (3) automatic labeling, and (4) precise annotations. Future work includes enhancing the reality of the synthetic dataset, so that the dataset bias is reduced, and fatigue crack identification accuracy improves further.

5. Conclusions

A synthetic data augmentation approach has been proposed to improve the identification of fatigue cracks in steel box girders typically found in long-span bridges using a fully convolutional network (FCN). The scarcity of datasets for fatigue crack in steel bridge girders and the very-fine nature of these cracks has hindered the application ML techniques for its identification. To overcome such challenges the use of synthetic images has been proposed to augment real-world datasets used for training neural networks that can identify fatigue cracks. 3D scanned random textures depicting the damaged surface of steel structures were created and mapped onto a graphical steel box girder model. The process was automated to simulate various lighting conditions and camera angles using a Python-Blender API. The efficacy of the approach was evaluated by comparing the performance of the proposed FCN trained for two cases: (1) using only real-word data (FCN-R), and (2) using real-word data augmented with synthetic data (FCN-A). FCN-A has shown an increase in precision by 13% (49% to 62%), although as a trade-off, the recall fell by 4%. Moreover, the results show an overall increase in the crack identification performance by 5% (35% to 40%) based on the intersection over union (IoU). The proposed approach has been demonstrated to be an effective and practical tool for fatigue crack identification in steel box girders. The approach also offers promising potential for identifying other types of damage in structures to strengthen existing datasets.

The complexity of the task of identifying cracks with only a limited dataset still has room for improvement. For example, (a) employing a more complex network based on the recent progress in this field, such as DeeplabV3 proposed by Chen *et al*. (2017), (b) improvement of synthetic image generation process by considering RGB maps that combine real textures from the open-source online libraries and the expected features such as pen marks and paint, rather than use of RGB map generated with random noise for the steel surface in the current work, and (c) refinement of the present damage simulation process by incorporating more features of fatigue crack such as updating the rust color of the inner cracks and blurring the image to mimic motion blur during actual image inspection. These extensions could help enhance the performance of steel fatigue crack identification, as well as to improve the application of synthetic data augmentation for other type of structural damage.

Acknowledgments

The authors would like to express their sincere thanks to Jau Yu Chou and Vedhus Hoskere for providing comments and valuable suggestions during the course of this research. In addition, the first and third author were supported in part by the China Scholarship Council under grants No. 201908040012 and No. 201706320312, respectively.

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)

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

ASCE's 2021 infrastructure report card (2021), Bridges; American Society of Civil Engineers, USA. https://infrastructurereportcard.org/cat-item/bridges/

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

Bao, Y.Q., Chen, Z.C., Wei, S.Y., Xu, Y., Tang, Z.Y. 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.Q., 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), 14759217211006485. https://doi.org/10.1177/14759217211006485

Blender (n.d.), https://www.blender.org/

Blender API Documentation. (n.d.), https://docs.blender.org/api/2.79/

Bowles, C., Chen, L., Guerrero, R., Bentley, P., Gunn, R., Hammers, A., Dickie, D.A., Valdés Hernández, M., Wardlaw, J. and Rueckert, D. (2018), "Gan augmentation: Augmenting training data using generative adversarial networks", arXiv preprint arXiv:1810.10863.

Bray, D.E. and Stanley, R.K. (1996), Nondestructive Evaluation: A Tool in Design, Manufacturing and Service, CRC Press.

Burley, B. and Studios, W.D.A. (2012), "Physically-based shading at Disney", ACM SIGGRAPH, 2012, 1-7.

CC0 Textures - Free Public Domain PBR Materials. https://www.sharetextures.com/

Chen, L.C., Papandreou, G., Schroff, F. and Adam, H. (2017), "Rethinking atrous convolution for semantic image segmentation", arXiv preprint arXiv:1706.05587.

Farrar, C.R. and Worden, K. (2012), *Structural Health Monitoring: A Machine Learning Perspective*, John Wiley & Sons.

Fisher, J.W. and Yuceoglu, U. (1978), "A survey of localized cracking in steel bridges", 1978, p. 334.

Frid-Adar, M., Klang, E., Amitai, M., Goldberger, J. and Greenspan, H. (2018), "Synthetic data augmentation using GAN for improved liver lesion classification", *Proceedings of 2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)*, Washington, DC, USA, April, pp. 289-293. https://doi.org/10.1109/ISBI.2018.8363576

Giakoumis, I., Nikolaidis, N. and Pitas, I. (2005), "Digital image processing techniques for the detection and removal of cracks in digitized paintings", *IEEE Transact. Image Process.*, **15**(1), 178-188. https://doi.org/10.1109/TIP.2005.860311

Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. and Bengio, Y. (2014), "Generative adversarial networks", arXiv preprint arXiv: 1406.2661.

Han, Q.H., Xu, J., Carpinteri, A. and Lacidogna, G. (2015), "Localization of acoustic emission sources in structural health monitoring of masonry bridge", *Struct. Control Health Monitor.*, **22**(2), 314-329. https://doi.org/10.1002/stc.1675

Hoskere V., Narazaki, Y., Hoang, T.A. and Spencer Jr., B.F. (2018), "Towards automated post-earthquake inspections with deep learning-based condition-aware models", arXiv:1809.09195.

Hoskere, V., Narazaki, Y., Hoang, T.A. and Spencer Jr., B.F. (2020), "MaDnet: multi-task semantic segmentation of multiple types of structural materials and damage in images of civil infrastructure", *J. Civil Struct. Health Monitor.*, **10**, 757-773. https://doi.org/10.1007/s13349-020-00409-0

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., Chen, F.C., Joffe, C. and Masri, S.F. (2017), "Vision-based quantitative assessment of microcracks on reactor internal components of nuclear power plants", *Struct. Infrastr. Eng.*, **13**(8), 1013-1026. https://doi.org/10.1080/15732479.2016.1231207

Liu, Y.F., Cho, S., Spencer Jr., B.F. and Fan, J.S. (2016), "Concrete crack assessment using digital image processing and 3D scene reconstruction", *J. Comput. Civil Eng.*, **30**(1), 04014124. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000446

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.

Margineantu, D.D. (2000), "When does imbalanced data require cost-sensitive learning?", Report No. WS-00-05; AAAI Workshop.

Mohan, A. and Poobal, S. (2018), "Crack detection using image processing: A critical review and analysis", *Alexandria Eng. J.*, **57**(2), 787-798. https://doi.org/10.1016/j.aej.2017.01.020

Narazaki, Y., Gomez, F., Hoskere, V., Smith, M.D. and Spencer Jr., B.F. (2020a), "Efficient development of vision-based dense three-dimensional displacement measurement algorithms using physics-based graphics models", *Struct. Health Monitor.*, 1475921720939522. https://doi.org/10.1177/1475921720939522

Narazaki, Y., Hoskere, V., Hoang, T.A., Fujino, Y., Sakurai, A. and Spencer Jr., B.F. (2020b), "Vision-based automated bridge component recognition with high-level scene consistency", *Comput.-Aided Civil Infrastr Eng.*, **35**(5), 465-482. https://doi.org/10.1111/mice.12505

Narazaki, Y., Hoskere, V., Yoshida, K., Spencer Jr., B.F. and Fujino, Y. (2021), "Synthetic environments for vision-based structural condition assessment of Japanese high-speed railway viaducts", *Mech. Syst. Signal Process.*, **160**, 107850. https://doi.org/10.1016/j.ymssp.2021.107850

Perlin, K. (1985), "An image synthesizer", *ACM Siggraph Computer Graphics*, **19**(3), 287-296. https://doi.org/10.1145/325165.325247

Perlin, K. (2001), "Noise hardware. In Real-Time Shading", SIGGRAPH Course Notes.

Prasanna, P., Dana, K.J., Gucunski, N., Basily, B.B., La, H.M., Lim, R.S. and Parvardeh, H. (2014), "Automated crack detection on concrete bridges", *IEEE Transact. Automat. Sci. Eng.*, **13**(2), 591-599. https://doi.org/10.1109/TASE.2014.2354314

Ros, G., Sellart, L., Materzynska, J., Vazquez, D. and Lopez, A.M. (2016), "The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes", *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3234-3243.

Sadler, D.J. and Ahn, C.H. (2001), "On-chip eddy current sensor for proximity sensing and crack detection", *Sensors Actuators A: Phys.*, **91**(3), 340-345. https://doi.org/10.1016/S0924-4247(01)00605-7

Shader Editor - Blender Manual. https://docs.blender.org/manual/en/latest/editors/shader_editor.html

Shorten, C. and Khoshgoftaar, T.M. (2019), "A survey on image data augmentation for deep learning", *J. Big Data*, **6**(1), 1-48. https://doi.org/10.1186/s40537-019-0197-0

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

Szeliski, R. (2010), *Computer Vision*: *Algorithms and Applications*, Springer Science & Business Media.

Textures for 3D, graphic design and Photoshop! (n.d.). https://www.textures.com/

Xu, J.L., Dong, Y.K., Zhang, Z.H., Li, S.L., He, S.Y. and Li, H. (2016), "Full scale strain monitoring of a suspension bridge using high performance distributed fiber optic sensors", *Measure. Sci. Technol.*, **27**(12), 124017. https://doi.org/10.1088/0957-0233/27/12/124017

Xu, Y., Li, S.L., Zhang, D.Y., Jin, Y., Zhang, F.J., 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.Q., Chen, J.H., Zuo, W.M. 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

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

Zhang, A., Wang, K.C.P., Li, B.X., Yang, E.H., Dai, X.X., Peng, Y., Fei, Y., Liu, Y., Li, J.Q. and Chen, C. (2017), "Automated pixel-level pavement crack detection on 3D asphalt surfaces using a deep-learning network", *Comput.-Aided Civil Infrastr. Eng.*, **32**(10), 805-819. https://doi.org/10.1111/mice.12297