Smart Structures and Systems, *Vol. 30, No. 1 (2022) 17-34*
https://doi.org/10.12989/sss.2022.30.1.017

Semantic crack-image identification framework for steel structures using atrous convolution-based Deeplabv3+ Network

Quoc-Bao Ta [1a], Ngoc-Loi Dang [2b], Yoon-Chul Kim [3c], Hyeon-Dong Kam [1d] and Jeong-Tae Kim*[1]

[1] *Department of Ocean Eng., Pukyong National University, Nam-gu, Busan 48513, Korea*
[2] *Urban Infrastructure Faculty, Mien Tay Construction University, Vinh Long 890000, Vietnam*
[3] *Department of Civil and Environmental Eng., Yonsei University, Seodaemun-gu, Seoul 03722, Korea*

(*Received August 11, 2021, Revised November 12, 2021, Accepted March 21, 2022*)

Abstract. For steel structures, fatigue cracks are critical damage induced by long-term cycle loading and distortion effects. Vision-based crack detection can be a solution to ensure structural integrity and performance by continuous monitoring and non-destructive assessment. A critical issue is to distinguish cracks from other features in captured images which possibly consist of complex backgrounds such as handwritings and marks, which were made to record crack patterns and lengths during periodic visual inspections. This study presents a parametric study on image-based crack identification for orthotropic steel bridge decks using captured images with complicated backgrounds. Firstly, a framework for vision-based crack segmentation using the atrous convolution-based Deeplapv3+ network (ACDN) is designed. Secondly, features on crack images are labeled to build three databanks by consideration of objects in the backgrounds. Thirdly, evaluation metrics computed from the trained ACDN models are utilized to evaluate the effects of obstacles on crack detection results. Finally, various training parameters, including image sizes, hyper-parameters, and the number of training images, are optimized for the ACDN model of crack detection. The result demonstrated that fatigue cracks could be identified by the trained ACDN models, and the accuracy of the crack-detection result was improved by optimizing the training parameters. It enables the applicability of the vision-based technique for early detecting tiny fatigue cracks in steel structures.

Keywords: atrous convolution; Deeplabv3+ network; fatigue crack; image processing technique; semantic segmentation; steel structures

1. Introduction

Orthotropic steel bridge decks have been preferred for the construction of long-span steel bridges due to their advances, such as rapid construction speed, low self-weight, and life-cycle costing (Connor 2012). Welding is commonly employed to combine bridge decks, ribs, and diagrams to form a structural unit. Fatigue cracks in steel box bridges are prone to appear at weld joints of rib-to-deck, rib-to-diaphragm, or rib splice induced by long-term cycle loading, particularly under overloading and unavoidable manufacturing defects (Ya *et al*. 2011, Sim and Uang 2012, Li *et al*. 2018, Di *et al*. 2021). Moreover, the effect of traffic loading and crack formation is random and stochastic, respectively; thus, it will be the challenges for bridge inspectors. Besides, fatigue cracks in severe levels can significantly decrease structural performance and service-life bridges (Fasl *et al*. 2016). To monitor fatigue cracks, visual inspection is commonly used by trained inspectors (Campbell *et al*. 2020). Although the inspection result could provide much information on inspected members, the method is time-consuming, inevitably labor, and cost-expensive. Moreover, inspection results mainly rely on the experiences of inspectors.

To overcome the issue, non-destructive testing methods have been developed using advanced sensing technologies based on contact-based sensors. Lee *et al*. (2020) utilized features of guided waves to identify fatigue cracks in steel joints. Due to sensing capacity, the method requires an array of actuators and sensors to detect fatigue crack propagation in a single steel joint. Strain-based methods can be considered as a technical way to detect fatigue cracks based on the well-defined relationship between stress-strain of steel materials. Ghahremani *et al*. (2013) adopted foil strain gages to evaluate distortion-induced fatigue crack in a large-scale girder bridge. Due to the relatively small footprint of strain gages, a great number of sensors are demanded to monitor defects on a large structural surface. To overcome the mentioned issue, the concept of skin sensor using soft elastomeric capacitor was proposed for identifying fatigue cracks in steel structures (Kong *et al*. 2018). Moreover, impedance-based techniques have been implemented to monitor fatigue cracks in steel structures (Soh and Lim 2009, Huynh *et al*. 2017) and assessed the residual fatigue life of bolted joints (Bhalla *et al*. 2012). Also, other research efforts have been made by utilizing

*Corresponding author, Ph.D., Professor,
E-mail: idis@pknu.ac.kr
[a] Ph.D. Student
[b] Ph.D.
[c] Undergraduate Student (Summer Internship)
[d] Master Student

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

vibration characteristics to predict fatigue damages in metallic structures (Papadimitriou *et al*. 2011, Habtour *et al*. 2016, Huynh *et al*. 2021). In general, most contact-based sensor techniques demands high-precision data acquisition systems, which are embedded with algorithms to compensate for external effects (e.g., temperature changes) (Bastani *et al*. 2011, Huynh *et al*. 2015, 2018, Sun *et al*. 2019). Consequently, the methods are expensive and problematic to be employed for popular large-size steel bridges.

Recently, vision-based structural health monitoring (SHM) techniques have emerged as the alternative to sensor-based methods (Park *et al*. 2015, Ye *et al*. 2019, Dong *et al*. 2020, Kim *et al*. 2020, Pham *et al*. 2020, Ta and Kim 2020). Extracted features from digital images or videos captured by digital cameras are used to track structural vibration (Khuc and Catbas 2016, Dong and Catbas 2020b, Erdogan and Ada 2020) or to detect local damages or changes of critical members in structural bridges (Huynh *et al*. 2019, Zhao *et al*. 2019, Huynh 2021). The advantages of vision-based methods include low-cost, non-contact sensing, and time-saving. Thanks to advances in cloud-based computation, transfer learning, and computer hardware improvement, moreover, trained networks can be embedded into a tiny platform (e.g., Raspberry Pi) to autonomously monitor and assess structural conditions (Jeong *et al*. 2020, Jin *et al*. 2021), thus enabling the reduced long-term monitoring cost.

Vision-based crack detection can be a solution to ensure structural integrity and performance by continuous monitoring and non-destructive assessment. Based on the nature of fatigue cracking, it can be classified as motion-based and image-based methods (Xu *et al*. 2018a, Dellenbaugh *et al*. 2020, Dong *et al*. 2021). Dellenbaugh *et al*. (2020) captured web-gap motions of a steel beam using three-dimensional digital image correlation (DIC). The method provides crack initialization and crack-development region by using strain map measurement, mainly induced by repeated cycle loading. Meanwhile, the image-based methods are appropriate to detect existing fatigue cracks. For the image-based detection, features such as cracks and marks should be extracted from the taken images by implementing pre-trained computer algorithms that enable minimizing collections of training datasets (Cha *et al*. 2017, Dung and Anh 2019).

For orthotropic steel bridges, captured images possibly consist of complex backgrounds such as handwritings and marks, which were made to record crack patterns and lengths during periodic visual inspections. Therefore, a critical issue is to distinguish cracks from other background features in captured images. Compared with published datasets of concrete or pavement cracks (Dung and Anh 2019, Yao *et al*. 2020), the fatigue-crack images had more complex backgrounds. The obstacles could have effects on a crack detection result using computer vision algorithms. So far, Xu *et al*. (2018b) have proposed a deep CNN (convolutional neural network) for crack segmentation. Image patches having a size of 64 × 64 × 3 pixels were input to the trained CNN network to classify whether the patches belong to crack or background labels. It is known that the accuracies of vision-based damage detection mainly rely on training datasets and machine learning algorithms (Barbedo 2018, Spencer *et al*. 2019). Moreover, the 1st International Project Competition for Structural Health Monitoring (IPC-SHM, 2020) was held to encourage the development of SHM worldwide using recent the development of deep learning for autonomous crack monitoring (Bao *et al*. 2021). Another effort was also made on IPC-SHM 2020 data by adopting the encoder-decoder network for segmenting fatigue cracks from the complicated backgrounds (Dong *et al*. 2021). The original images were cropped into image patch with a size of 512 × 512 × 3 pixels to train the modified Unet network. The approach could produce a better IoU value due to keeping the original image resolutions. However, the effect of obstacle in the complex background of crack images (e.g., ruler and handwriting) has not been achieved so far. Although recent research efforts using the recent advancement of deep learning could yield better fatigue-crack detection results, the research on fatigue cracks identification in orthotropic steel bridges using atrous convolution-based Deeplabv3+ network (ACDN) is still essential based on the following reasons: (1) considering effects of obstacles (e.g., ruler, handwriting, and weld line) on the crack detection result, (2) identifying optimal training parameters for the image-based ACDN model, and (3) identifying number of training images needed for training the ACDN-based crack segmentation model.

This study presents a parametric study on image-based crack identification for orthotropic bridges using taken images with complicated backgrounds. To achieve the objective, firstly, a framework for vision-based crack segmentation using the modern ACDN is designed. Secondly, features on crack images are labeled to build three databanks by consideration of objects in the backgrounds. Thirdly, evaluation metrics estimated from the trained ACDN models are utilized to evaluate the effects of the obstacles on the crack-detection result. Fourthly, various image sizes and hyper-parameters consisting of the learning rule, learning rate, and epoch number are optimized for the image-based crack detection via the ACDN. Lastly, a number of training images for the proposed crack detector are also suggested.

2. Semantic crack-image segmentation framework using Deeplabv3+ network

2.1 Framework of proposed method

Fig. 1 shows the two-phase framework of image-based crack segmentation using captured images of orthotropic steel bridges. Phase I is to select an optimal model for image-based crack segmentation (as illustrated in Sections 3-4). Firstly, fatigue crack images with complicated backgrounds are labeled to build three databanks by considering different objects on images. Secondly, the labeled images and raw images are resized to build training datasets. Thirdly, these datasets are used to train ACDN models for object segmentation. Finally, three trained

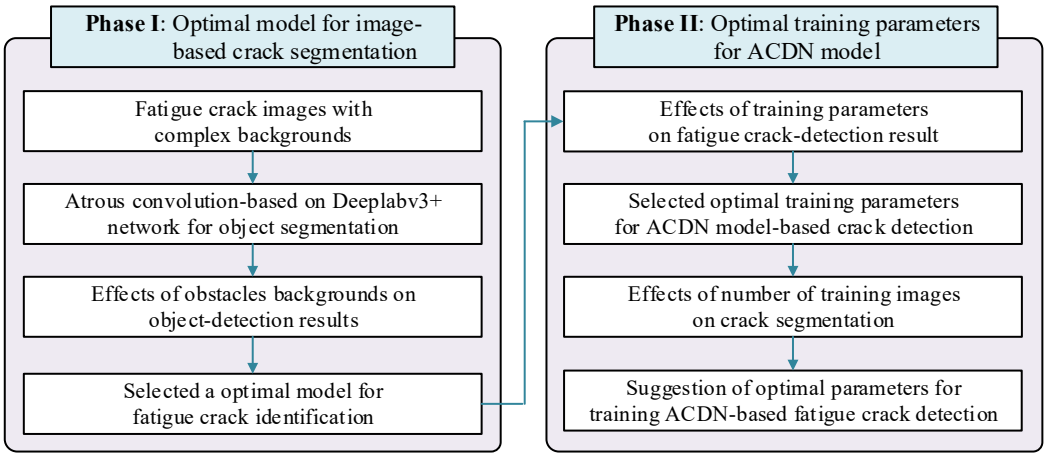


Fig. 1 Proposed framework for semantic crack-image identification

ACDN models are evaluated via evaluation metrics computed from 10% untrained and 10% training images to analyze the effects of the obstacles on crack-identification results. An optimal model of crack segmentation is suggested based on the evaluation.

Phase II is to optimize training parameters for the ACDN model of crack segmentation (as illustrated in Section 5). Training parameters include image sizes, CNN networks, and hyper-parameters (e.g., learning rate and learning rule). Firstly, effects of training parameters are quantified to select optimal training factors for crack identification. Secondly, the number of training images is examined by using the optimal parameters to quantify the accuracy of crack segmentation. Lastly, optimal training parameters of the ACDN model are recommended for image-based crack identification.

It is notable that raw images of orthogonal steel bridges had a very high resolution (e.g., 4928 × 3264 × 3 pixels) (Bao *et al*. 2021). To train an ACDN model using a normal computer configuration, two approaches can be employed: (1) raw images and their labels are split into sub-images (e.g., 512 × 512 × 3 pixels (Dong *et al*. 2021)), and (2) the images and labels are resized with an appropriate size (e.g., 480 × 640 × 3 pixels). Although the first approach can yield a higher detection result due to keeping the original pixel resolution, it requires a higher computation cost, and professional programming skills. Our proposed approach is towards low computational cost and simplicity in implementation; thus, for the parametric analysis of image-based object segmentation, the second approach (i.e., resizing images) was chosen.

2.2 DeepLabv3+ network with encoder and decoder architecture

With the development of deep learning and great pre-labeled datasets, semantic object segmentation can be achieved at pixel-level classification. This study aims to build an end-to-end deep learning model based on the Deeplabv3+ network for segmenting semantic tinny-crack images captured from the complex background of steel bridges. Compared to other deep-learning networks, such as fully convolutional network (Yang *et al*. 2018), U-net (Ronneberger *et al*. 2015 and Ly *et al*. 2021), or Deeplab networks (Chen *et al*. 2014, 2017, and 2018a), the Deeplabv3+ network (Chen *et al*. 2018b) was constructed based on two encoder and decoder modules, which enable to recognize irregular distributions of tinny cracks and crack characteristics (Sun *et al*. 2021). Also, the atrous separable convolution makes the deeplabv3+ faster (i.e., significant reduction of the computational complexity) and stronger in feature learning (Chen *et al*. 2018b).

This study selected DeeLabv3+ based on the Resnet50 network for the sematic crack segmentation, as illustrated in Fig. 2. The input was the fatigue crack images with complex backgrounds, and the output was a recognition result. The result shows pixel labels of segmented objects (e.g., crack, handwriting, and ruler) and their locations. The Resnet50 network was adopted for the Deeplabv3+. It is noted that the Resnet50 network can achieve more accuracy and fast computation speed (He *et al*. 2016). The encoder and decoder modules of Deeplabv3+ are briefly described in Fig. 2.

2.2.1 Encoder module

The encoder module aims to obtain deeply hidden vital crack features by down-sampling image size and increasing the receptive field. These feature maps extracted from the ResNet50 network are applied atrous SPP (spatial pyramid pooling). The atrous SPP technique has been improved based on the SPP used in the region-based CNN object detection (He *et al*. 2015), in which features could be accurately classified from convolutions of multiple sizes and any regions. As shown in Fig. 3, the atrous SPP comprises four atrous convolutions with different rates (i.e., rates of 18, 12, 6, and 1) to extract deeply hidden features. Moreover, the use of the atrous SPP aims to diminish spatial-hierarchical information losses and to resolve a small object information issue, which is unable to reconstruct.

The ResNet50 module, the backbone of the Deeplabv3+ network, was used to extract feature maps of crack images from input datasets. It is noted that ResNet50 network (He *et al*. 2016) was developed for image-based object

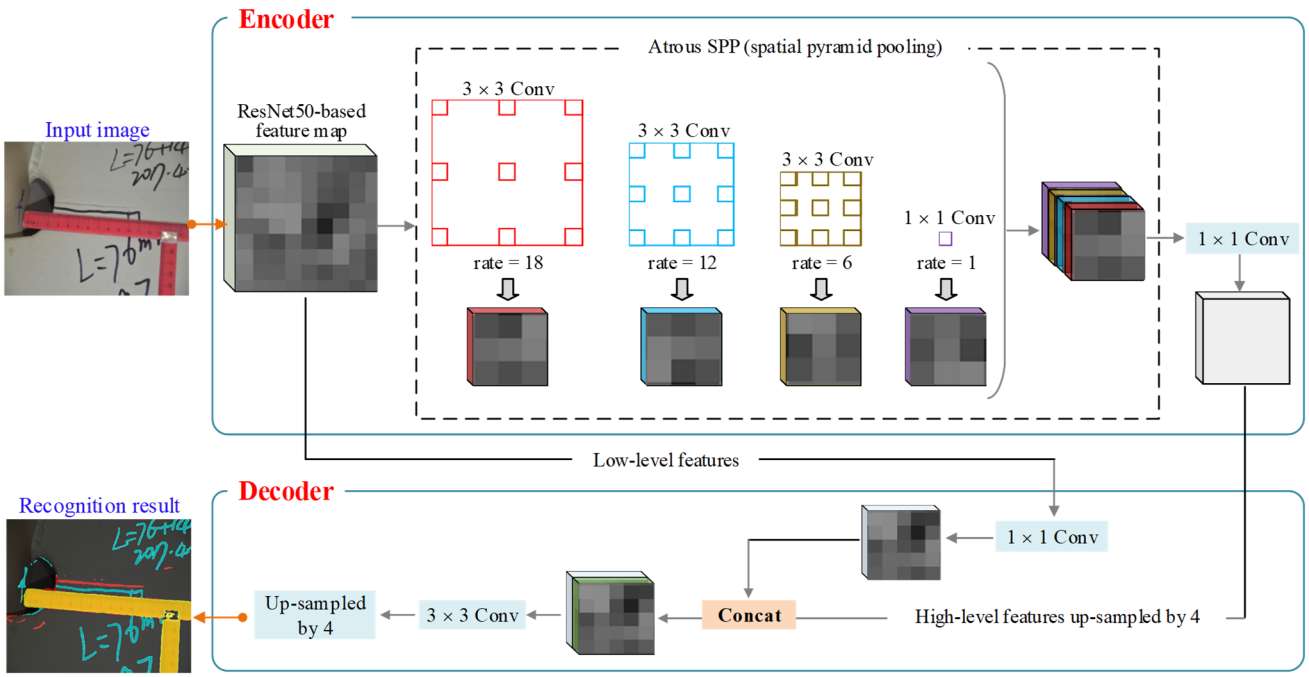


Fig. 2 Overall of Deeplabv3+ architecture based on Resnet50 network for object segmentation

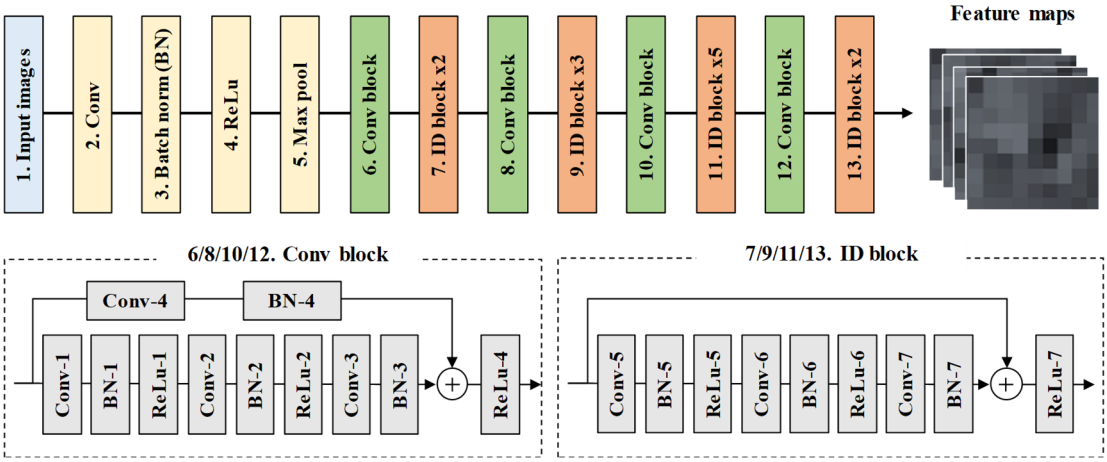


Fig. 3 Architecture of Resnet50 network for features extraction from training images

classification with multiple objects (e.g., people, cars, and animals). Using the pre-trained network (i.e., ResNet50) can be an effective solution for training CNN models with a limited number of images. To employ the ResNet50 for feature extraction, the output layer of the original network is removed and switched to the atrous SPP in the encoder module of Deeplabv3+. Fig. 3 shows 13 components of the Resnet50 network for feature extraction of crack images.

Detailed layers and operators of the modified ResNet50 network are outlined in Table 1. The network comprises an input image layer (Input), convolution layers (Conv), a batch normalization layer (Batch norm), an activation function layer (ReLu), max-pooling layers (Max pool), four convolution (Conv) blocks, and twelve identity (ID) blocks. The convolution blocks were used to skip connection in case of un-matching between the input and output. The last layers of the ResNet50 network were connected to the atrous SPP layers.

2.2.2 Decoder module

The decoder module is used to regenerate the concise and sharp context information (e.g., crack) given from the encoder module. The low-level crack features exacted from Restnet50 and high-level crack features computed from the atrous SPP the encoder module are inputted into the decoder. The high-level crack features are up-sampled four times and then concatenated with low-level crack features after applying some 1 × 1 convolution. Several 3 × 3 convolutions are used to classify object properties. Lastly, an up-sampling procedure (scale factor of 4) is conducted to expand the region to form output images.

Table 1 Detailed layers and operators of modified Resnet50 network

No	Type	Depth	Filter size	Stride	Padding	Output image size
1	Input	3	-	-	-	[w × h](*)
2	Conv	64	7×7	2	3	[w × h]
3	Batch norm	-	-	-	-	[w × h]
4	ReLu	-	-	-	-	[w × h]
5	Max pool	64	3×3	2	2	[w/2×h/2]
6/8/10/12. Conv block	Conv-1	64/128/256/512	1×1	1/2/2/1	0/0/2/0	
	BN-1	-	-	-	-	
	ReLu-1	-	-	-	-	
	Conv-2	64/128/256/512	3×3	1	1	
	BN-2	-	-	-	-	[w/2 × h/2] for block #6 [w/4 × h/4] for block #8 [w/8 × h/8] for block #10 [w/16 × h/16] for block #12
	ReLu-2	-	-	-	-	
	Conv-3	256/512/1024/2048	1×1	1	0	
	BN-3	-	-	-	-	
	ReLu-4	-	-	-	-	
	Conv-4	256/512/1024/2048	1×1	1/2/2/1	0/0/2/0	
	BN-4	-	-	-	-	
7/9/11/13. ID Block	Conv-5	64/128/256/512	1×1	1	0	
	BN-5	-	-	-	-	
	ReLu-5	-	-	-	-	
	Conv-6	64/128/256/512	3×3	1	1	[w/2 × h/2] for block #7 [w/4 × h/4] for block #9 [w/8 × h/8] for block #11 [w/16 × h/16] for block #13
	BN-6	-	-	-	-	
	ReLu-6	-	-	-	-	
	Conv-7	256/512/1024/2048	1×1	1	0	
	BN-7	-	-	-	-	
	ReLu-7	-	-	-	-	

(*) w and *h* are the width and height of an image

2.3 Evaluation metrics for image-based object segmentation

To evaluate a pixel-level object segmentation model, the mean accuracy (mAcc) of predicted pixel-labels and mean intersection over union (mIoU) are commonly used (Csurka *et al*. 2013). Fig. 4 shows an explanatory diagram for calculating two evaluation metrics (i.e., mAcc and mIoU), wherein P_{ij} (so-called false positive, FP) is the sum of pixels in class *i* that belong to class *j*. Likewise, P_{ii} and P_{ji} denote true positive (TP) and false negative (FN) of predicted pixel-labels.

The mAcc and mIoU metrics calculated for 0 ~ *w*+1 (including background) were shown in Eqs. (1) and (2), respectively. The mAcc expresses the ratio of precise pixels calculated according to each defined class (e.g., crack or ruler) and then averaged. Meanwhile, the mIoU considers the equality the number of TP over the sum of FN, FP, and TP. In other words, mIoU is superimposed by the number of classes and then averaged.

$$mAcc = \frac{1}{w+1}\sum_{i=0}^{w} P_{ii} \Big/ \sum_{j=0}^{w} P_{ij} \tag{1}$$

$$mIoU = \frac{1}{w+1}\sum_{i=0}^{w} P_{ii} \Big/ \left(\sum_{j=0}^{w} P_{ij} + \sum_{j=0}^{w} P_{ji} - P_{ii} \right) \tag{2}$$

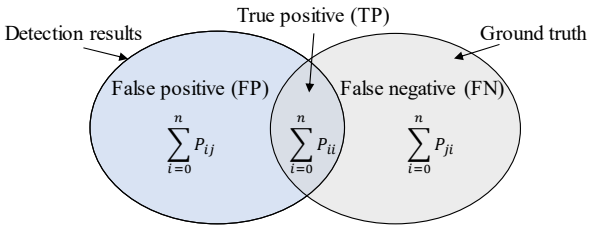
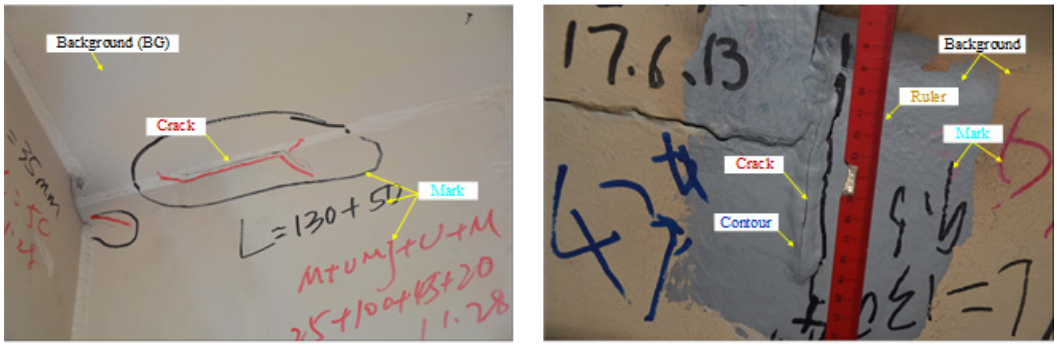
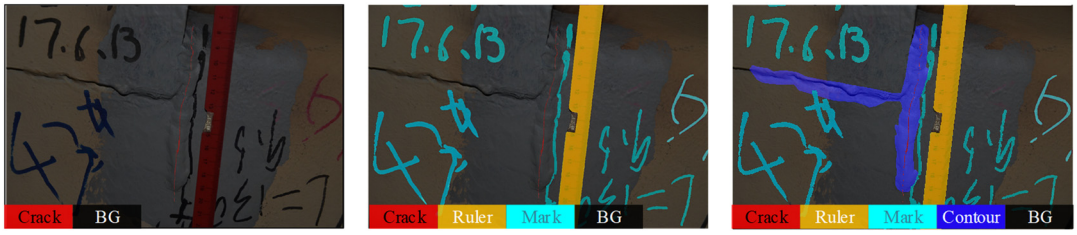


Fig. 4 Relational diagram for calculating evaluation metrics



(a) Image #32 with 3 classes (b) Image #178 with 5 classes

Fig. 5 Fatigue crack images of orthotropic steel bridge decks with complex background



(a) Labeled dataset D1 (b) Labeled dataset D2 (c) Labeled dataset D3

Fig. 6 Ground-truth definition for training three ACDN models

3. Training datasets for atrous convolution-based Deeplabv3+ network (ACDN)

3.1 Description of fatigue crack image from orthotropic steel bridges

Total 200 images of steel bridges with fatigue cracks were granted by the committee of the 1st International Project Competition for Structural Health Monitoring (IPC-SHM 2020) (Hou *et al*. 2013, Bao *et al*. 2021). The images were captured by resolution sizes of 3264 × 4928 × 3 or 3864 × 5152 × 3 pixels which were taken by many inspectors at various distances and perspective angles. Fig. 5 shows two representative images in the dataset. The backgrounds include rulers and handwriting marks made during periodic inspections to keep records on the crack length and crack propagation.

As observed in the Fig. 5, the width (thickness) of fatigue crack was very small as compared to the image size. Also, the crack line was coincidental with the crack-indicating mark and the weld contour. The effects of obstacles (e.g., ruler, handwriting, and weld line) should be quantified to improve the accuracy of the crack detection result. Overall, five main features (so-called classes) were visualized in the provided dataset, including crack, marks with three different (black, blue, and red) colors, ruler, welding contour, and remaining background. To be segmented as distinct classes, these five features were named as Crack, Mark, Ruler, Contour, and BG (background) classes.

3.2 Image labeling and division of training dataset

3.2.1 Image labeling to build three databank

Three labeled datasets, namely D1, D2, and D3 with different classes, were analyzed for the effects of obstacles on crack detection results. As outlined in Table 2, the three labeled datasets (D1, D2, and D3) were produced from total 200 images of the raw dataset (namely as D0). In the dataset D1 (see Fig. 6(a)), two classes were assigned for pixels of the crack line (Crack) and all other features (BG). In the dataset D2 (see Fig. 6(b)), four classes were assigned for pixels of the crack line (Crack), the three-colored mark (Mark), the ruler (Ruler), and remaining features (BG). In the dataset D3 (see Fig. 6(c)), five classes were assigned for pixels of the crack line (Crack), the marks (Mark), the ruler

Table 2 Three labeled datasets (D1-D3) for training three ACDN models (M1-M3)

Dataset	Labeled dataset			Trained ACDN models
	Name	Number of class	Name of classes	
D0	D1	2	Crack and BG (background)	M1 (using D0-D1)
	D2	4	Crack, Mark, Ruler, and BG	M2 (using D0-D2)
	D3	5	Crack, Mark, Contour, Ruler, and BG	M3 (using D0-D3)

(Ruler), the welding (Contour), and remaining features (BG). Three labeled datasets D1-D3 (200 images for individual datasets) with the raw data D0 were used to train three corresponding ACDN models (namely M1-M3), as listed in Table 2.

For each of ACDN models, crack detection can be affected by the different number of classes: (1) the model M1 by one class (BG), (2) the model M2 by three classes (Mark, Ruler, and BG), and (3) the model M3 by four classes (Mark, Contour, Ruler, and BG). It is noted that images in the provided dataset (Bao *et al*. 2021) had black, blue, and red marks. The features of cracks and black marks (see Fig. 5(a)) owned the quite relevant colors. To examine the effects of black marks on crack detection results, the other ACDN model, so-called M4, with three labels of Crack, BM (black mark), and BG was tested. It is found that the training accuracy (not presented in the manuscript) was not improved since the number of pixels of red and blue marks in the whole dataset was quite small compared to those of the black mark. To make the paper concise, all marks were assigned as "Mark".

3.2.2 Selection of initial training parameters

As described previously, the raw images (i.e., dataset D0) were very high resolutions, which were too large for training or testing CNN models unless a high-performance computer unit (e.g., workstation) was used. To enable training and testing the ACDN models using a desktop computer, the images were automatically resized to 720 × 960 × 3 pixels resolutions. Also, assigned label pixels in datasets D1-D3, which were made using the same size of the raw images, were also interpolated to near features using build-in algorithms supported by Matlab software.

For training ACDN models (M1-M3), initial training parameters were selected as: image size = 720 × 960, learning rule = SGDM, learning rate = 10^{-3}, maximum epoch = 10, CNN network = Resnet50 (as described previously), mini-batch size = 1, and momentum = 0.9. It is noted that mini-batch size expresses the number of images passing through the network for every iteration. A small value of mini-batch size (e.g., 1 for this analysis) is suitable for a low configuration computer, and features from images can also be extracted better by computer algorithms.

The computation hardware and software were selected as follows: a desktop computer with i9-9900 @ 3.6 GHz CPU, an 11 GB memory NVIDIA RTX2080Ti graphics processing unit (GPU), 64 GB of RAM, and Matlab 2020a software with deep learning modules.

3.2.3 Dataset division for training, validation, and testing

To train, validate, and test the ACDN models (M1-M3), the prepared datasets were divided into training (80%), validation (10%), and testing (10%). To analyze object segmentation results, the evaluation metrics (see Eqs. (1) and (2)) were computed by using 10% of untrained images and 10% of training images. In order to compare the accuracy of crack segmentation of the three models M1-M3, the same images were selected for training, validation, and testing of the models (see Table 3).

Specifically, the 160 images (except images 5:10:200 and 3:10:200 with an interval of 10) were utilized for the training process. The 20 images (3, 13, …, 193) were used for validation during training models. Meanwhile, the 20 images (1, 11, …, 191) and 20 images (5, 15, …, 195) were used for testing on the trained and untrained datasets, respectively. It is notable that the testing on trained and untrained images aims to confirm the feature learning capability of the ACDN models.

As shown in Fig. 7, the six data argumentation techniques including reflection, translation, rotation, shear, scaling, and combination, were adopted to enlarge the diversity of training datasets and to minimize the over-fitting issue (Shorten and Khoshgoftaar 2019). During the training process, training images were randomly reflected, translated from -20~20 pixels along two directions, rotated

Table 3 Selection of images for training, validation, and testing ACDN models M1-M3

Parameters		Number	Order of selected images
Training (80%)		160	except 5:10:200 and 3:10:200
Validation (10%)		20	3:10:200 (interval of 10)
Testing	Trained images (10%)	20	1:10:200
	Untrained images (10%)	20	5:10:200

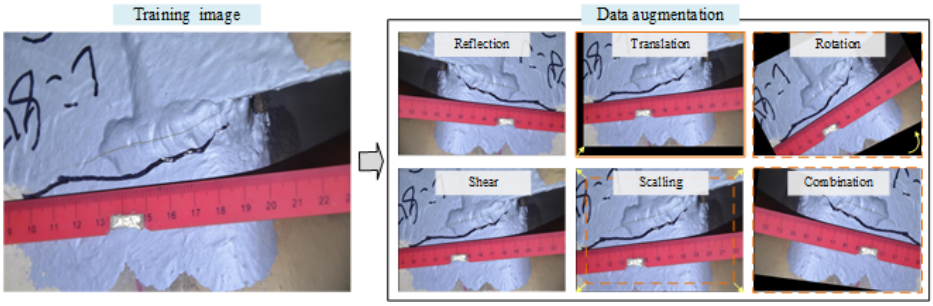
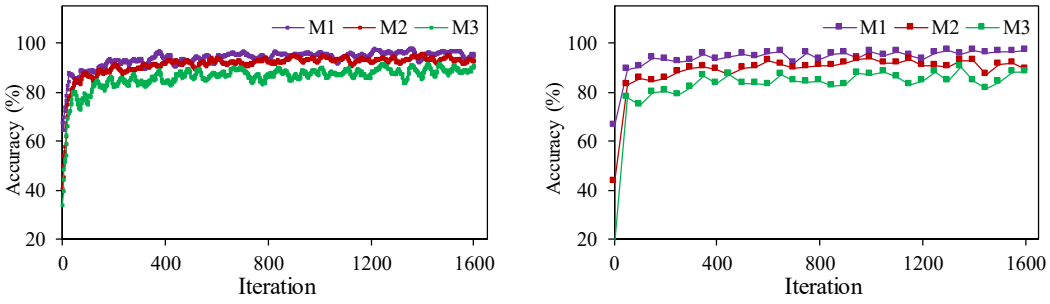
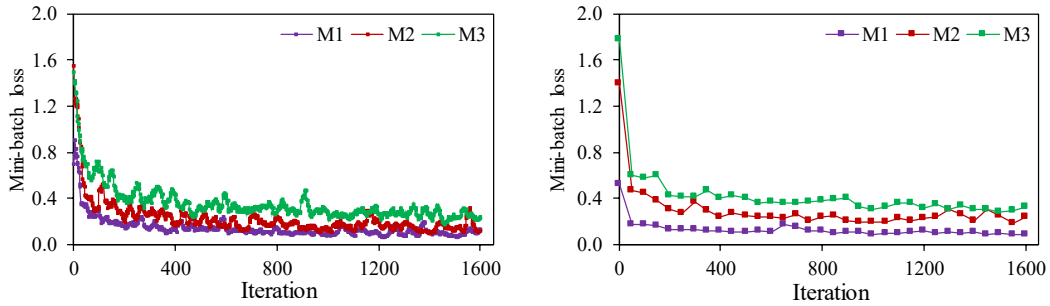


Fig. 7 Illustration of six data argumentation techniques for model training



(a) Training accuracy (b) Validation accuracy

Fig. 8 Accuracies of models M1-M3 during training process



(a) Training loss (b) Validation loss

Fig. 9 Mini-batch losses of models M1-M3 during training process

from -30 (to the left) ~30 (to the right) degrees, sheared in the range -30~30 degrees, and scaled with a factor 0.75-1.5 times. Furthermore, a combination of those five techniques was also utilized as shown in Fig. 7. During the model training, the data augmentation was performed using a built-in training algorithm. The augmented data were automatically discarded after the computing process.

4. Object segmentation using Trained ACDN models

4.1 Training accuracy of ACDN models

As shown in Figs. 8(a)-(b), the accuracies of the training and validation processes were examined for the three ACDN models (M1-M3) under 1600 iterations. Among the three models, the model M1 recorded the highest performance (about 98% accuracy) of the learning features. The computer learning algorithm worked relatively better for the training model M1 with two classes (i.e., Crack and BG). As shown in Figs. 9(a)-(b), the total losses of the training and validation processes were examined for the models M1-M3. In terms of convergence speed and segmenting effectiveness, the models M2 and M3 show relatively lower performances than the model M1.

Both the accuracy and the loss of the training were insignificantly changed after the 800th iteration. The training and validation results of the three ACDN models were converged during the training process. The initial input parameters (described in the previous section) were feasible for training the ACDN models.

4.2 Object segmentation using trained ACDN models

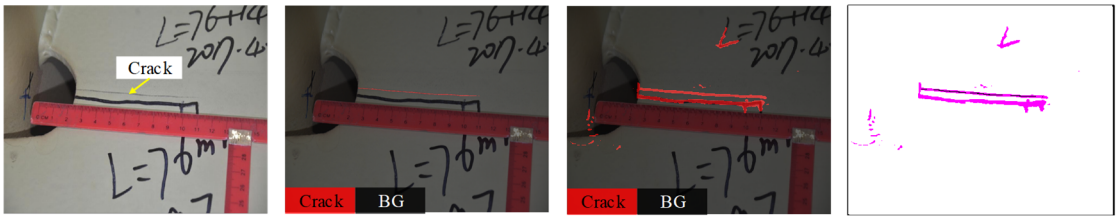
The accuracy of the trained models M1-M3 for object segmentation was assessed by using 10% of trained images and 10% of untrained images (see Table 3). The evaluation metrics, Eqs. (1) and (2), were computed for the images. Then, a pair of a trained image (#191) and an untrained image (#195) were plotted for the models M1-M3 to visualize the effectiveness of object segmentation.

4.2.1 Model M1 with two classes (Crack and BG)

For a trained image #191, the object segmentation was tested by using the trained model M1 (see Fig. 10): a raw image with real crack (see Fig. 10(a)) and a pixel-level crack prediction (Fig. 10(c)). As shown in Fig. 10(d), the overlapping between the ground-truth (see Fig. 10(b)) and the predicted pixel labels (see Fig. 10(c)) was used to emphasize the crack estimation result. The crack was well-segmented, although the false-positive estimate occurred along with the crack zone.

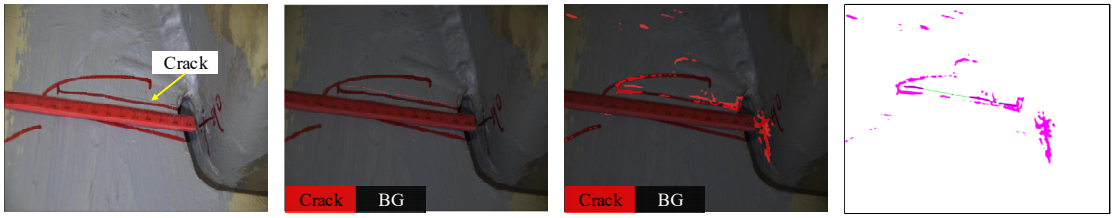
For an untrained image #195, the crack segmentation was evaluated by using the trained model M1 (see Fig. 11). The crack pixel-label (see Fig. 11(c)) was well detected, but the false-positive estimation was also found near the crack zone. The overlapping between the ground-truth and the predicted pixel labels was also examined, as shown in Fig. 11(d).

As shown in Fig. 12(a), the accuracy of predicted classes (Crack and BG) was calculated from the 10% trained images and the 10% untrained images. As shown in Fig. 12(b), the accuracy of $mAcc_2$ and $mIoU_2$ was also



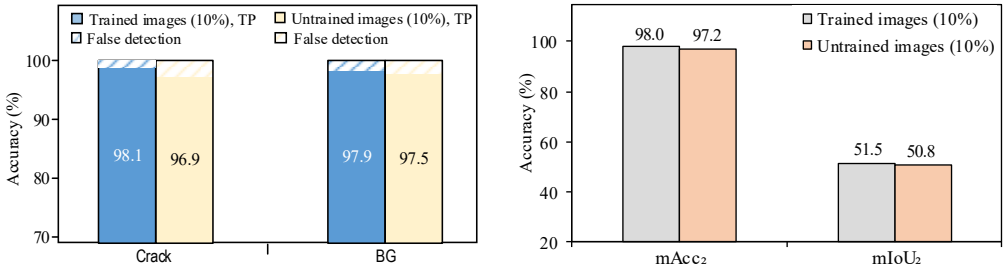
(a) Raw image (b) Ground truth (c) Prediction (d) overlapping

Fig. 10 Object segmentation using model M1: trained image #191



(a) Raw image (b) Ground truth (c) Prediction (d) overlapping

Fig. 11 Object segmentation using model M1: untrained image #195



(a) Accuracy of predicted classes (b) mAcc and mIoU

Fig. 12 Evaluation metrics for object segmentation using model M1

estimated from the same trained and untrained images. For the two classes (Crack and BG), the true pixel-label prediction (TP) and the false detection in the bar chart were estimated from the confusion matrix. The trained dataset yielded a better prediction of pixel-level objects than the untrained one, thus indicating the ACDN model was trained properly. For the observation made by the trained dataset (see Fig. 12(a)), the prediction accuracy of Crack (98.1%) was higher than that of BG (97.9%). The mean accuracy $mAcc_2$ of Crack was 98.0%, and the $mIoU_2$ of Crack was 51.5% (see Fig. 12(b)).

Ideally, the mIoU value can reach up to 100% when predicted pixel labels are matched to labeled ones. As described previously, the crack width was relatively small compared to the crack length. Although the crack pixel-label was well predicted, the false estimation surrounding the crack length led to the relatively low value of mIoU.

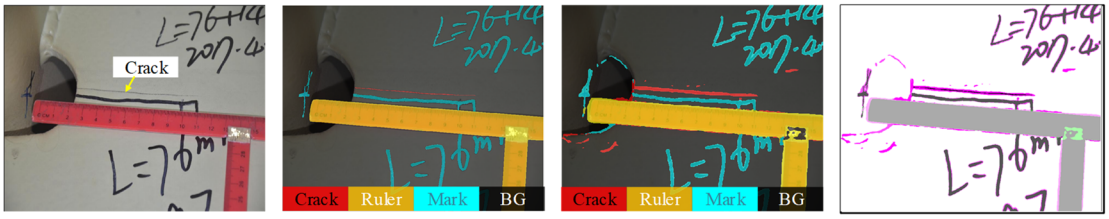
4.2.2 Model M2 with four classes (Crack, Ruler, Mark, and BG)

For the trained image #191, the object segmentation was tested by using the trained M2 model (see Fig. 13). All objects (i.e., Crack, Ruler, Mark, and BG) were well-segmented via the pixel-level prediction (see Fig. 13(c)). The overlapping between the ground-truth and the predicted pixel labels of four objects was also examined, as shown in Fig. 13(d).

For the untrained image #195, the object segmentation was evaluated by using the trained M2 model (see Fig. 13). The two-pixel classes of Ruler and Mark were well detected, and the crack pixel-label was also identified with false-positive pixels (see Fig. 14(c)). Also, the overlapping between the ground-truth and the predicted pixels of the four classes was illustrated in Fig. 14(d). Thin-dark marks and tiny cracks were not accurately distinguished; meanwhile, the ruler was clearly detected.

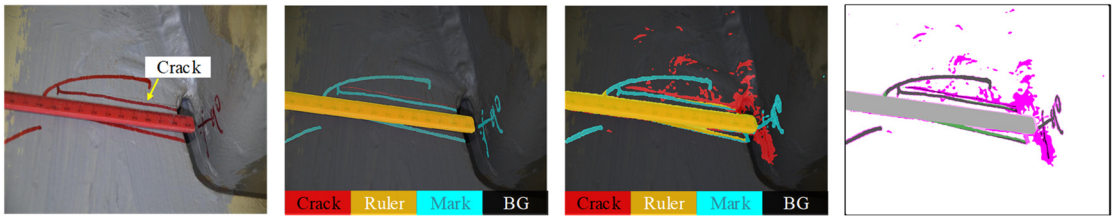
As shown in Fig. 15(a), the accuracy of the four predicted classes was computed from the 10% untrained images. Overall, the accuracy of all classes was higher than 90%. Among the four classes, the Ruler pixel-level was most accurate as 99.7%, and the Crack pixel-label was least accurate as 92.3%. Fig. 15(b) shows the evaluation metrics (i.e., mAcc and mIoU) calculated for Crack and BG (namely $mAcc_2$ and $mIoU_2$) and all classes (namely $mAcc_4$ and $mIoU_4$). Similar to the model M1, the trained dataset produced a more accurate prediction of labels than the untrained one. It is notable that $mIoU_2$ (49.1% for trained images) was smaller than $mIoU_4$ (62.9%). It is noted that

the $mIoU_4$ considered the contribution of true pixel labels of four classes, in which the ruler pixel-label had the highest accuracy (99.7% see Fig. 15(a)).



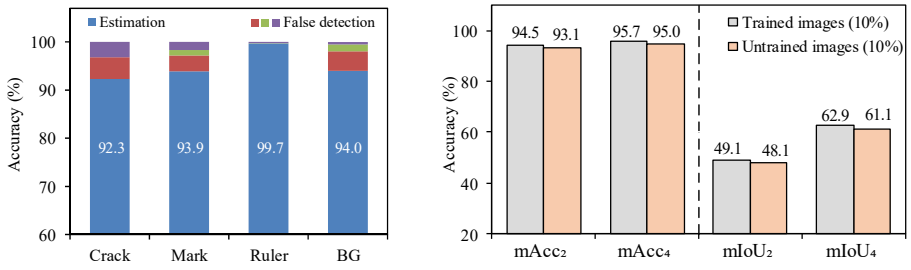
(a) Raw image (b) Ground truth (c) Prediction (d) overlapping

Fig. 13 Object segmentation using model M2: trained image #191



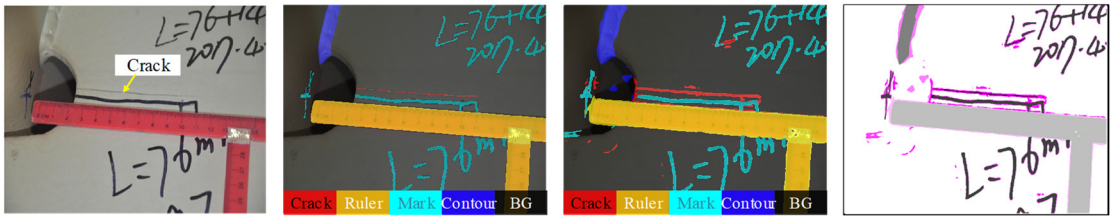
(a) Raw image (b) Ground truth (c) Prediction (d) overlapping

Fig. 14 Object segmentation using model M2: untrained image #195



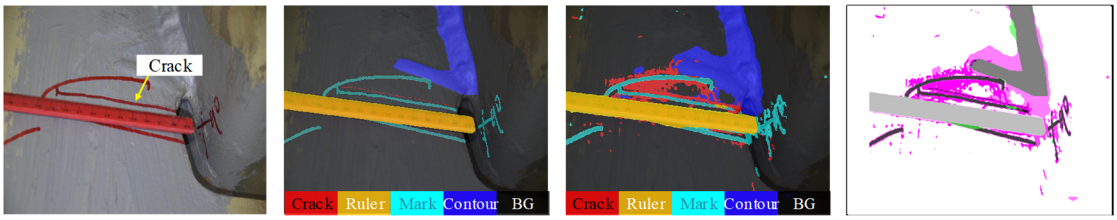
(a) Accuracy of predicted classes (10% untrained) (b) mAcc and mIoU

Fig. 15 Evaluation metrics for object segmentation using model M2



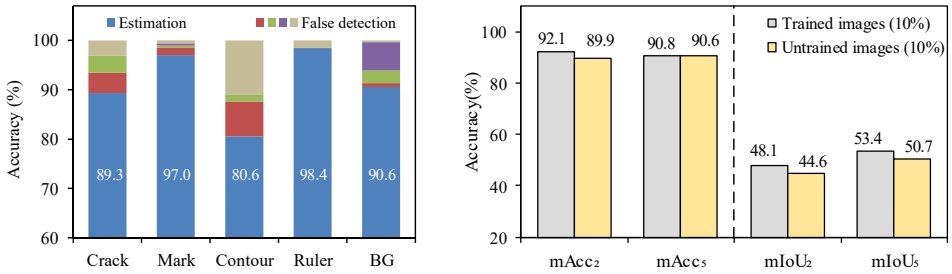
(a) Raw image (b) Ground truth (c) Prediction (d) overlapping

Fig. 16 Object segmentation using model M3: trained image 191



(a) Raw image (b) Ground truth (c) Prediction (d) overlapping

Fig. 17 Object segmentation using model M3: untrained image #195



(a) Accuracy of predicted classes (10% untrained) (b) mAcc and mIoU

Fig. 18 Evaluation metrics for object segmentation using Model M3

4.2.3 Model M3 with five classes (Crack, Mark, Contour, Ruler, and BG)

Fig. 16 shows an example of the segmentation result of Crack, Mark, Contour, Ruler, and BG tested for the trained image #191 using model M3. Generally, five objects were well-segmented (see Fig. 16(c)). The overlapping between the ground truth and the predicted pixel labels was also examined, as shown in Fig. 16(d). For the untrained image, Ruler and Contour classes were well-segmented (see Fig. 17(c)). Crack and Mark classes were detected with positive-estimation errors. Also, the overlapping between the ground truth-predicted pixels was plotted for comparison (see Fig. 17(d)).

Fig. 18(a) shows the accuracy of the five predicted objects computed from the 10% untrained images. Among them, the predicted pixel-label of Contour was the lowest (80.6%), and the pixel-label of Ruler exhibited the highest precision (98.4%). Fig. 18(b) shows the evaluation metrics evaluated for Crack and BG classes (namely $mAcc_2$ and $mIoU_2$) and all five classes (namely $mAcc_5$ and $mIoU_5$). Similar to the models M1 and M2, the evaluation metrics of the trained dataset were slightly higher than those of the untrained one.

4.3 Effects of complicated background features on crack-detection results

As observed in Fig. 8(a), the training accuracy of model M1 (two classes) kept almost stable (~95%) after the first 800 iterations. Meanwhile, the training accuracy of model M3 (five classes) varied about 86%. As the number of classes (i.e., background features) increased, the training images in the dataset should be increased to achieve higher accuracies.

For the three trained models M1-M3, all objects were well segmented by the pixel-level estimation with the accuracies higher than 80%. The increased number of pixel-labels decreased the accuracies of the pixel-level object estimation (see Fig. 12(a) versus Fig. 15(a)). This observation is consistent with the training accuracy of the model M1 versus the model M3 (see Fig. 8(a)).

The differences between the evaluation metrics (i.e., mAcc or mIoU) computed from the trained and the untrained datasets were small, thereby indicating that the distribution of features (e.g., Crack and Mark) was insignificant between the two tested datasets. In the image, the object was easy to recognize (e.g., Ruler), the higher predicted accuracy was achieved, thus demonstrating the feasibility of the built ACDN models for the object segmentation.

For overall values of the mIoUs, the model M2 had the highest value ($mIoU_4$ = 61.1%, see Fig. 15(b)) as compared to the model M3 ($mIoU_5$ = 50.7%, see Fig. 18(b)) and the model M1 ($mIoU_2$ = 50.8%, see Fig. 12(b)). The performance of the model M2 was good for the detection of cracks and previous inspection marks. Comparing $mIoU_2$ values calculated from Crack and BG, moreover, the $mIoU_2$ of the model M1 (50.8% for the untrained dataset, see Fig. 12(b)) shows higher than that of model M2 (48.1%, see Fig. 15(b)) or model M3 (44.6%, see Fig. 18(b)).

The analysis revealed that the obstacles affected the crack-estimation results. Based on the values of $mIoU_2$ and $mAcc_2$ of two classes (Crack and BG), model M1 yielded the best indicator for crack segmentation. Thus, model M1 with two classes was selected for the optimal training parameters, as discussed in the next section.

5. Optimal ACDN model for fatigue crack detection

5.1 Effects of training parameters on crack detection result

5.1.1 Selection of training parameters

The accuracy of vision-based damage detection mainly relies on training datasets and machine learning algorithms (Barbedo 2018, Spencer *et al*. 2019). The model M1 described in the previous section was utilized to analyze the effect of training parameters on crack segmentation. Five parameters were investigated as listed in Table 4.

The first parameter was image size ($M1_{1i}$). The effect of image sizes were examined from 360 × 480 ~ 720 × 960 pixels with a scale factor of 1.5. The second parameter was learning rule ($M1_{2j}$). Three popular learning rules were investigated for training CNN networks, including SGDM (stochastic gradient descent with momentum), ADAM (adaptive moment estimation), and RMSPROP (root mean square propagation). The third parameter was learning rate ($M1_{3i}$), which were analyzed by four different rates in the range 10^{-2}~10^{-5} with an interval of 0.1. The fourth parameter was epoch number ($M1_{4i}$), which was analyzed by four different cases in the range 10~40. The final parameter was the CNN network ($M1_{5j}$). Four popular networks were examined for object segmentation, including

Table 4 Effects of training parameters on crack identification result of Model M1

Parameters, i	Sub-parameter, j			
	1	2	3	4
Fixed initial parameters, $M1_{00}$: Image size = 720 × 960, learning rule = SGDM, learning rate = 0.001, Epoch = 10, CNN network = Resnet50.				
1. Image size (pixels), $M1_{1j}$	360 × 480	480 × 640	720 × 960	
2. Learning rule, $M1_{2j}$	SGDM	ADAM	RMSPROP	
3. Learning rate, $M1_{3j}$	10^{-2}	10^{-3}	10^{-4}	10^{-5}
4. Epoch number, $M1_{4j}$	10	20	30	40
5. CNN, $M1_{5j}$	Resnet18	Resnet50	Inceptionresnetv2	Xception

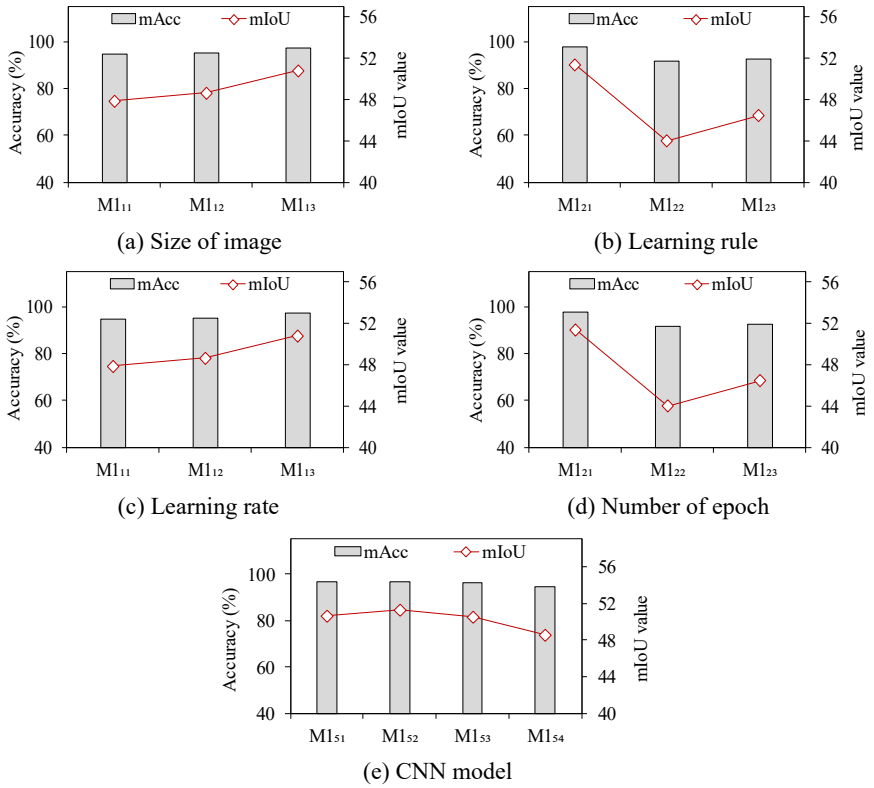
Resnet18 (18 layers depth), Resnet50 (50 layers depth), Inceptionresnetv2 (164 layers depth), and Xception (171 layers depth).

The effect of training parameters on crack detection was estimated as follows. Firstly, the initial model, namely $M1_{00}$ (i = 0, j = 0), was set with initial training parameters listed in Table 4. Secondly, three cases of image sizes ($M1_{11}$~$M1_{13}$) were analyzed by changing image sizes in the initial model (i.e., $M1_{00}$) from 360 × 480 to 720 × 960 pixels. Thirdly, three cases of learning rule ($M1_{21}$~$M1_{23}$) were studied by replacing the learning rules SGDM, ADAM, and RMSPROP in the initial model. Separate analyses were made on four cases of learning rate ($M1_{31}$~$M1_{34}$), four cases of the number of epoch ($M1_{41}$~$M1_{24}$), and five cases of CNN ($M1_{51}$~$M1_{55}$). Total 19 ACDN models were established to estimate optimal training parameters. Note that the dataset division for training, validation, and testing was described in the previous section.

5.1.2 Effects of training parameters on crack segmentation

Two evaluation metrics, mAcc and mIoU, were utilized to evaluate optimal training parameters for the ACDN models. Figs. 19(a)-(e) show the values of mAcc and mIoU calculated from the 10% untrained images for different cases of image sizes, learning rule, learning rate, epoch number, and CNN networks, respectively. As illustrated in the figures, mAcc values increased along with mIoU values, and vice versa. The mIoU index was selected to evaluate optimal training parameters since it has been commonly used for pixel-level crack prediction (Csurka *et al*. 2013).

Firstly, the effect of image size on the crack detection accuracy was calculated as shown in Fig. 19(a). $M1_{13}$ (720

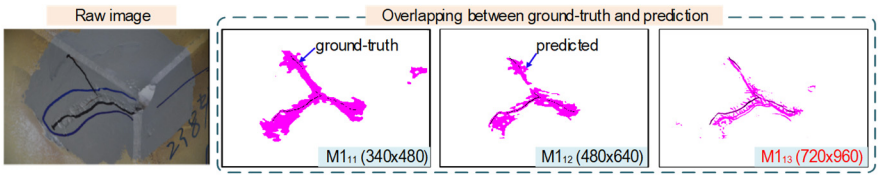


(a) Size of image (b) Learning rule

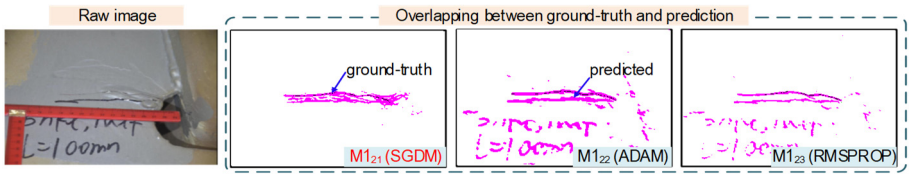
(c) Learning rate (d) Number of epoch

(e) CNN model

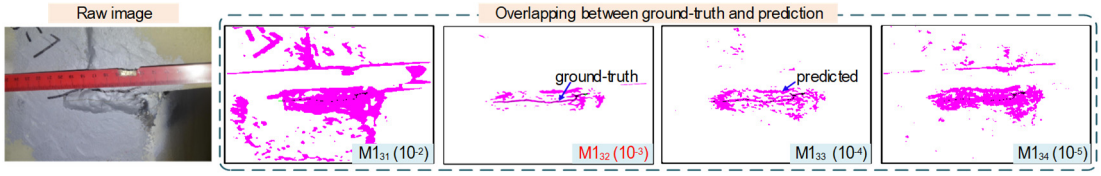
Fig. 19 Effects of training parameters on crack detection accuracy



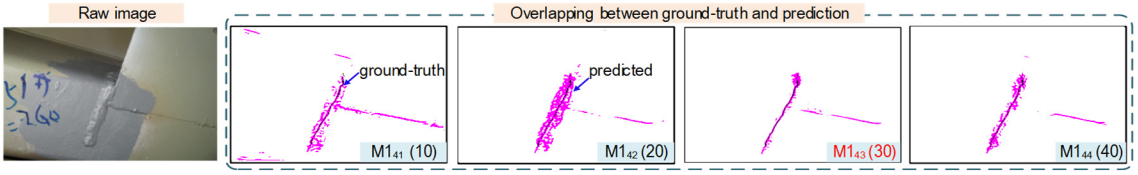
(a) Effect of image size on image #51



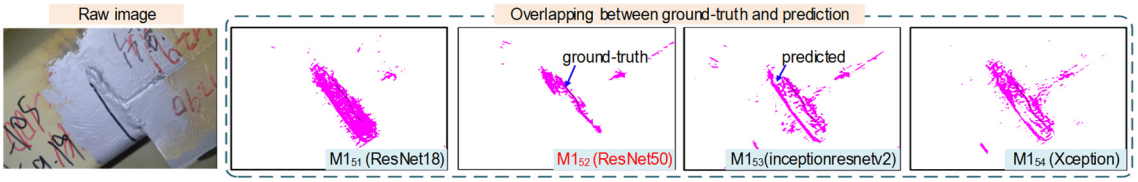
(b) Effect of learning rule on image #131



(c) Effect of learning rate on image #121



(d) Effect of epoch number on image #111



(e) Effect of CNN network on image #1

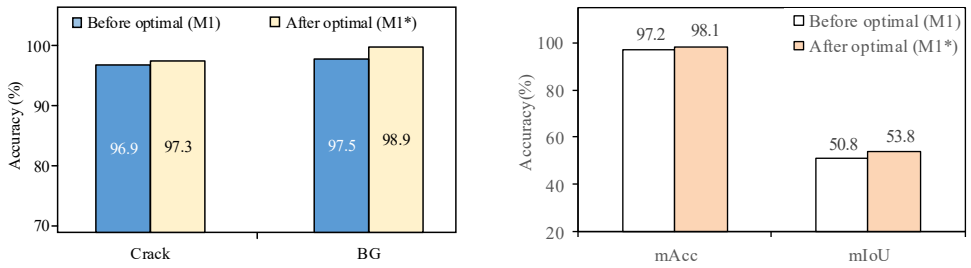
Fig. 20 Effects of training parameters on crack detection accuracy

× 960) yielded the highest mIoU value. Increasing image sizes, a higher computer configuration was required for training and testing of the ACDN models. Secondly, the effect of learning rule was estimated, as shown in Fig. 19(b). The SGDM (M1$_{21}$) produced the best algorithm for the training, and the ADAM (M1$_{22}$) yielded the lowest mIoU value. Thirdly, the effect of learning rate was estimated as shown in Fig. 19(c). The appropriate learning rate was 10^{-3} (M1$_{32}$) among the four trial cases. Fourthly, the effect of epoch number was estimated as shown in Fig. 19(d). The mIoU value was slightly improved as the increasing number of training epochs from 10 to 30, and it was deceased for 40 epochs. The result suggested that 40 epochs were considered as the final epochs for the parametric study, and the suitable number of epoch was 30 (M1$_{43}$). Finally, the effect of CNN networks was calculated as shown in Fig. 19(e). The Resnet50 (M1$_{52}$) produced the highest mIoU metric for crack segmentation.

The effects of the five training parameters (i.e., image size, learning rule, learning rate, epoch number, and CNN network) on the accuracy of crack detection was examined as shown in Fig. 20. The overlapping between the ground-truth and the crack pixel-label prediction was used to compare the crack estimation result. It is obvious that the red-marked training parameters (see Figs. 20(a)-(e)) produced better crack detection results. For the ACDN model, the five optimal training parameters were selected as follows: 720 × 960 pixels (image size), SGDM (learning rule), 0.001 (learning rate), 30 (epoch number), and Resnet50 (CNN network). It is observed that the optimal parameters were the same ones as used for training the models M1-M3, except the number of epoch. Hereafter, the M1 model with the optimal parameters was namely as model M1*.

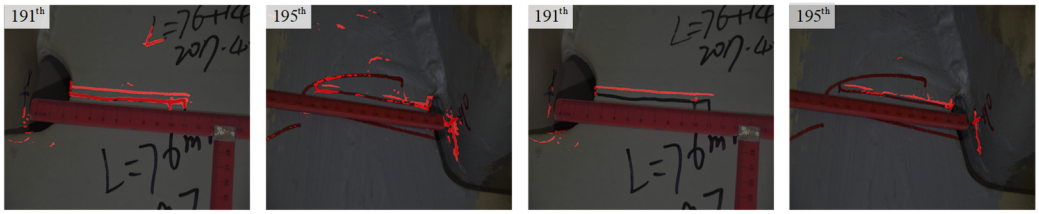
5.1.3 Crack segmentation results by optimal training parameters

The accuracy of crack segmentation was evaluated for the model M1*, which had the optimal training parameters. For the pre-trained model M1 and the optimal model M1*, evaluation metrics were calculated from the 10% untrained images. Fig. 21(a) shows the comparison of the predicted



(a) Accuracy of predicted classes (b) mAcc and mIoU

Fig. 21 Evaluation metrics calculated from 10% untrained images with optimal training parameters



(a) Before optimization (M1) (b) After optimization (M1*)

Fig. 22 Crack segmentation results of model M1 and optimal model M1*

class accuracies. Before optimizing, the accuracy of crack pixel-label prediction was 96.9% (see also Fig. 12(a)), and it was slightly increased (97.3%) after optimizing the training parameters. A similar observation was made for the accuracy of the BG class. Fig. 21(b) shows slight increases of mAcc and mIoU values after optimizing training parameters (model M1*) evaluated on the 10% untrained images. Specifically, the mIoU value was increased from 50.8% (model M1) to 53.7% (model M1*), and the mAcc was also increased 97.2% to 98.1%. It indicated that the accuracy of crack segmentation was improved by the optimal model M1*.

Figs. 22(a)-(b) show the comparison of crack segmentation results calculated for the trained image #191 and untrained image #195 by using the models M1 and M1*, respectively. Before optimization, the M1 model produced the false-positive prediction on crack pixel-label close to the crack line (see Fig. 22(a)). After optimization, the M1* model produced crack pixel-label prediction improved for both the trained image #191 and the untrained image #195 (see Fig. 22(b)).

5.2 Effects of training images on crack segmentation

5.2.1 Number of training images

The number of images for training the ACDN model was analyzed for the four cases (namely C1-C4) as listed in Table 5. The number of images was reduced from 200 to 50 with an interval of 50. Corresponding to the image number, the order of selected images was selected as noted in Table 5. The model M1* (i.e., M1 with optimal training parameters) was utilized for this analysis. For each case, the division of dataset was randomly selected for training (80%), testing (10%), and validation (10%).

5.2.2 Effects of training images on crack segmentation results

Table 6 shows the effects of the number of training images on crack segmentation results experimented for the four cases (C1-C4). As observed in the table, the accuracies of training and validation were slightly reduced when the number of images decreased. Also, the training time was significantly reduced (41.2-10.2 minutes for C1-C4). For the evaluation metrics, the mAcc and mIoU values calculated from the 10% untrained images were also reduced. Notably, in case C2 (150 images), the variation of mIoU was 0.3% compared to C1. Meanwhile, the variation of mIoU values was reduced to 2.7% in C3 (100 images) or 6.6% in C4 (50 images). The result suggested that 150 images are feasible for training the ACDN model.

Fig. 23 shows crack-segmentation results tested on the untrained image #1 for the different training images. Although fatigue crack was well detected in four simulated

Table 5 Data selection for analyzing effects of training images on crack segmentation

Case	Number of images	Selected images	Data division
C1	200	1,2,3…199,200 (1:1:200)	Training: 80% Testing: 10% Validation: 10% Order: random
C2	150	1,2,3, 5,6,7, …,197,198,199	
C3	100	1,3,5,…,197,199 (1:2:200)	
C4	50	1,5,9,…,193,197 (1:4:200)	

Table 6 Effects of training images on crack-identification results

Case	Accuracy (%)		Evaluation metrics (%)		(*) Training time (min.)	Variation (%)	
	Training	Validation	mAcc	mIoU		DmAcc	DmIoU
C1	98.0	97.5	98.8	53.7	41.2	-	-
C2	98.9	97.1	96.4	53.5	28.5	-2.4	-0.3
C3	98.0	96.9	93.4	52.2	19.7	-5.5	-2.7
C4	96.2	94.7	93.5	50.1	10.2	-5.3	-6.6

(*) Training time is based on computer configuration: i9-9900 CPU, 64GB of memory, and 11GB of GPU

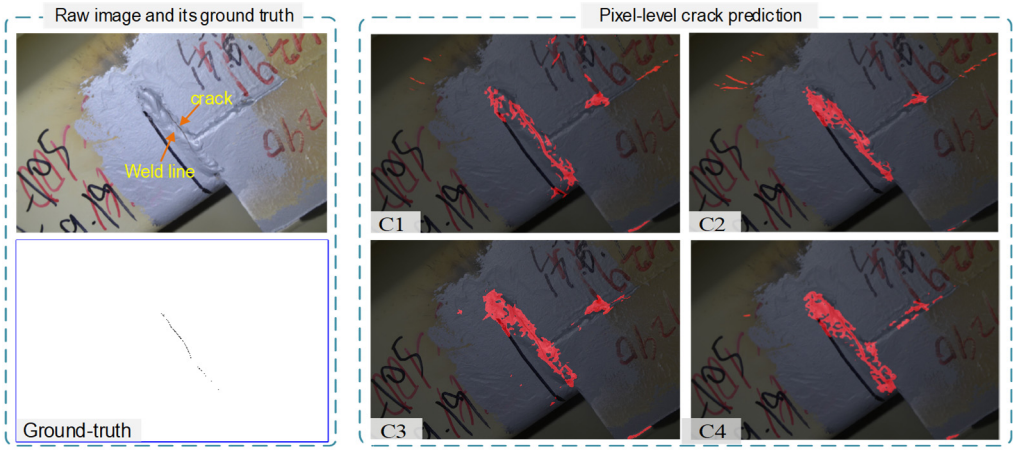


Fig. 23 Effects of training images on crack segmentation result: image #1

cases, false-positive predicted crack pixel-label exists surrounding the weld line. Among four cases, C1 and C2 had fewer crack pixel-label errors than C3 and C4 did. Moreover, the raw image shows that the fatigue crack occurred along with the weld line, which was a rough surface. Moreover, the crack width was comparatively small compared with its length. Tiny crack width can cause a challenge for computer vision-based damage detection when the image resolution is not high enough to reflect crack properties (e.g., crack width), which demands further study.

6. Conclusions

In this study, the parametric analysis on vision-based fatigue crack identification using captured images of orthotropic steel bridges was conducted. At first, the framework for vision-based crack segmentation based on the modern Deeplapv3+ network was designed. Then, the three ACDN models were investigated to examine the obstacle effects in the backgrounds on crack segmentation results. Finally, the various training parameters, including the image sizes, the hyper-parameters, and the number of training images, were examined to analyze their effects on the ACDN model-based crack identification.

From the parametric study, the following remarks can be drawn. Firstly, the vision-based object segmentation algorithm using the ACDN model could be used to identify tiny fatigue cracks in orthotropic bridge decks. Secondly, the obstacles in image backgrounds showed significant effects on the training accuracy and crack-segmentation results. The ACDN model with the two classes yielded the higher performance of the crack detection. Thirdly, The optimal training parameters for the ACDN model-crack detection were SGDM for learning rule, 10^{-3} for learning rate, 30 epochs, pre-strained Resnet50, and 150 fatigue images.

In comparison with the recently proposed framework for crack detection (Dong *et al*. 2021), the accuracy of our framework yielded 53.5% of mIoU for 30 epochs, which was higher than that of fully connected network-based framework (51.9% of mIoU for 45 epoch). Also, it was lower than the accuracy of Unet-based Framework (62.4% of mIoU for 45 epochs). Moreover, the data argumentation techniques enable to reduce the training time and the number of the training images. Instead of using 180 training images, the optimal framework shows that 135 images with data argumentation methods could reduce about 31% training time for the ACDN model.

Despite those advances, the proposed framework has some limitations. The effect of the capture angle distortion and the light intensity on the accuracy of the crack detector has not been considered. Also, although the training accuracy of crack detection with the optimal training parameters gained up to 96.4% the mAcc value achieved only 53.5%, which was lower than that of previous works on the similar dataset (Xu *et al*. 2018b or Dong *et al*. 2021).

In future studies, more standardized crack images from steel structures with various environmental conditions (e.g., light intensity) should be considered. Because crack-pixel labels occupied a relatively small compared with the image

size, class-pixel label balancing approach needs to be examined to improve the feature-learning process. In addition, adaptive image-cutting techniques and the estimation of geometric cracks (e.g., crack length) are also recommended for the next study.

Acknowledgments

This work was supported by a grant (21CTAP-C163708-01) from Technology Advancement Research Program funded by Korea Agency for Infrastructure Technology Advancement (KAIA). The datasets used in this paper were granted by the committee of the 1st International Project Competition for Structural Health Monitoring (IPC-SHM 2020). The authors would like to thank for the opportunity provided by IPC-SHM 2020.

Author contributions

Quoc-Bao Ta and Jeong-Tae Kim developed the methodology; Quoc-Bao Ta performed the framework design; Quoc-Bao Ta and Yoon-Chul Kim performed the simulation; Ngoc-Loi Dang designed the logics of the manuscript; Yoon-Chul Kim analyzed labeling images for two models (M2 and M3); Hyeon-Dong Kam analyzed labeling images for a model (M1); Jeong-Tae Kim revised the manuscript and supervised the whole work. All authors have read and agreed to the submitted version of the manuscript.

References

Bao, Y., Li, J., Nagayama, T., Xu, Y., Spencer, 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

Barbedo, J.G.A. (2018), "Impact of dataset size and variety on the effectiveness of deep learning and transfer learning for plant disease classification", *Comput. Electron. Agricult.*, **153**, 46-53. https://doi.org/10.1016/j.compag.2018.08.013

Bastani, A., Amindavar, H., Shamshirsaz, M. and Sepehry, N. (2011), "Identification of temperature variation and vibration disturbance in impedance-based structural health monitoring using piezoelectric sensor array method", *Struct. Health Monitor., Int. J.*, **11**(3), 305-314. https://doi.org/10.1177/1475921711427486

Bhalla, S., Vittal, P.A. and Veljkovic, M. (2012), "Piezo-impedance transducers for residual fatigue life assessment of bolted steel joints", *Struct. Health Monitor., Int. J.*, **11**(6), 733-750. https://doi.org/10.1177/1475921712458708

Campbell, L.E., Connor, R.J., Whitehead, J.M. and Washer, G.A. (2020), "Benchmark for evaluating performance in visual inspection of fatigue cracking in steel bridges", *J. Bridge Eng.*, **25**(1), 04019128. https://doi.org/10.1061/(ASCE)BE.1943-5592.0001507

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

Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K. and Yuille, A.L. (2014), "Semantic image segmentation with deep convolutional nets and fully connected crfs", *Computer Vision and Pattern Recognition*, *arXiv preprint arXiv:1412.7062*. https://doi.org/10.48550/arXiv.1412.7062

Chen, L.C., Papandreou, G., Schroff, F. and Adam, H. (2017), "Rethinking atrous convolution for semantic image segmentation", *arXiv preprint arXiv:1706.05587*. https://doi.org/10.48550/arXiv.1706.05587

Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K. and Yuille, A.L. (2018a), "DeepLab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs", *IEEE Transact. Pattern Anal. Machine Intell.*, **40**(4), 834-848. https://doi.org/10.1109/TPAMI.2017.2699184

Chen, L.C., Zhu, Y., Papandreou, G., Schroff, F. and Adam, H. (2018b), "Encoder-decoder with atrous separable convolution for semantic image segmentation", *Proceedings of the European Conference on Computer Vision* (*ECCV*). https://doi.org/10.48550/arXiv.1802.02611

Chollet, F. (2017), "Xception: Deep learning with depthwise separable convolutions", *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (*CVPR*). https://doi.org/10.48550/arXiv.1610.02357

Connor, R.J. (2012), "Manual for design, construction, and maintenance of orthotropic steel deck bridges", No. FHWA-IF-12-027; United States, Federal Highway Administration. https://rosap.ntl.bts.gov/view/dot/41395

Csurka, G., Larlus, D., Perronnin, F. and Meylan, F. (2013), "What is a good evaluation measure for semantic segmentation?", In: *Bmvc*, Vol. 27, pp. 10-5244. http://dx.doi.org/10.5244/C.27.32

Dellenbaugh, L., Kong, X., Al-Salih, H., Collins, W., Bennett, C., Li, J. and Sutley, E.J. (2020), "Development of a distortion-induced fatigue crack characterization methodology using digital image correlation", *J. Bridge Eng.*, **25**(9), 04020063. https://doi.org/10.1061/(ASCE)BE.1943-5592.0001598

Dhivya, J.J. and Ramaswami, M. (2018), "A perusal analysis on hybrid spectrum handoff schemes in cognitive radio networks", *International Conference on Intelligent Systems Design and Applications*, pp. 312-321. https://doi.org/10.1007/978-3-030-16660-1_31

Di, J., Ruan, X., Zhou, X., Wang, J. and Peng, X. (2021), "Fatigue assessment of orthotropic steel bridge decks based on strain monitoring data", *Eng. Struct.*, **228**, 111437. https://doi.org/10.1016/j.engstruct.2020.111437

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

Dong, C.Z., Bas, S. and Catbas, F.N. (2020), "Investigation of vibration serviceability of a footbridge using computer vision-based methods", *Eng. Struct.*, **224**, 111224. https://doi.org/10.1016/j.engstruct.2020.111224

Dong, C., Li, L., Yan, J., Zhang, Z., Pan, H. and Catbas, F.N. (2021), "Pixel-level fatigue crack segmentation in large-scale images of steel structures using an encoder–decoder network", *Sensors*, **21**(12), 4135. https://doi.org/10.3390/s21124135

Dung, C.V. and Anh, L.D. (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

Erdogan, Y.S. and Ada, M. (2020), "A computer-vision based vibration transducer scheme for structural health monitoring applications", *Smart Mater. Struct.*, **29**(8), 085007. https://doi.org/10.1088/1361-665X/ab9062

Fasl, J., Helwig, T. and Wood, S.L. (2016), "Fatigue response of a fracture-critical bridge at the end of service life", *J. Perform. Constr. Facil*, **30**(5), 04016019. https://doi.org/10.1061/(ASCE)CF.1943-5509.0000871

Ghahremani, K., Sadhu, A., Walbridge, S. and Narasimhan, S. (2013), "Fatigue testing and structural health monitoring of retrofitted web stiffeners on steel highway bridges", *Transport. Res. Record*: *J. Transport. Res. Board*, **2360**(1), 27-35. https://doi.org/10.3141/2360-04

Habtour, E., Cole, D.P., Riddick, J.C., Weiss, V., Robeson, M., Sridharan, R. and Dasgupta, A. (2016), "Detection of fatigue damage precursor using a nonlinear vibration approach", *Struct. Control Health Monitor.*, **23**(12), 1442-1463. https://doi.org/10.1002/stc.1844

He, K., Zhang, X., Ren, S. and Sun, J. (2015), "Spatial pyramid pooling in deep convolutional networks for visual recognition", *IEEE Transact. Pattern Anal. Mach. Intell.*, **37**(9), 1904-1916. https://doi.org/10.1109/TPAMI.2015.2389824

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* (*CVPR*). https://doi.org/10.1109/CVPR.2016.90

Hou, Y., Yue, P., Xin, Q., Pauli, T., Sun, W. and Wang, L. (2013), "Fracture failure of asphalt binder in mixed mode (Modes I and II) by using phase-field model", *Road Mater. Pave. Des.*, **15**(1), 167-181. https://doi.org/10.1080/14680629.2013.866155

Huynh, T.C. (2021), "Vision-based autonomous bolt-looseness detection method for splice connections: Design, lab-scale evaluation, and field application", *Automat. Constr*, **124**, 103591. https://doi.org/10.1016/j.autcon.2021.103591

Huynh, T.C., Park, Y.H., Park, J.H., Hong, D.S. and Kim, J.T. (2015), "Effect of temperature variation on vibration monitoring of prestressed concrete girders", *Shock Vib.*, 1-9. https://doi.org/10.1155/2015/741618

Huynh, T.C., Dang, N.L. and Kim, J.T. (2017), "Advances and challenges in impedance-based structural health monitoring", *Struct. Monitor. Maint., Int. J.*, **4**(4), 301-329. https://doi.org/10.12989/smm.2017.4.4.301

Huynh, T.C., Dang, N.L. and Kim, J.T. (2018), "PCA-based filtering of temperature effect on impedance monitoring in prestressed tendon anchorage", *Smart Struct. Syst., Int. J.*, **22**(1), 57-70. https://doi.org/10.12989/sss.2018.22.1.057

Huynh, T.C., Park, J.H., Jung, H.J. and Kim, J.T. (2019), "Quasi-autonomous bolt-loosening detection method using vision-based deep learning and image processing", *Automat. Constr.*, **105**, 102844. https://doi.org/10.1016/j.autcon.2019.102844

Huynh, T.C., Nguyen, T.T., Kim, J.T., Ta, Q.B., Ho, D.D. and Phan, T.T.V. (2021), "Deep learning-based functional assessment of piezoelectric-based smart interface under various degradations", *Smart Struct. Syst., Int. J.*, **28**(1), 69-87. https://doi.org/10.12989/sss.2021.28.1.069

Jeong, S., Kim, H., Lee, J. and Sim, S.H. (2020), "Automated wireless monitoring system for cable tension forces using deep learning", *Struct. Health Monitor.*, **20**(4), 1805-1821. https://doi.org/10.1177/1475921720935837

Jin, S.S., Jeong, S., Sim, S.H., Seo, D.W. and Park, Y.S. (2021), "Fully automated peak-picking method for an autonomous stay-cable monitoring system in cable-stayed bridges", *Automat. Constr.*, **126**, 103628. https://doi.org/10.1016/j.autcon.2021.103628

Khuc, T. and Catbas, F.N. (2016), "Computer vision-based displacement and vibration monitoring without using physical target on structures", *Struct. Infrastr. Eng.*, **13**(4), 505-516. https://doi.org/10.1080/15732479.2016.1164729

Kim, H., Yoon, J. and Sim, S.H. (2020), "Automated bridge component recognition from point clouds using deep learning", *Struct. Control Health Monitor*, **27**(9). https://doi.org/10.1002/stc.2591

Kong, X., Li, J., Collins, W., Bennett, C., Laflamme, S. and Jo, H. (2018), "Sensing distortion-induced fatigue cracks in steel bridges with capacitive skin sensor arrays", *Smart Mater. Struct*, **27**(11), 115008. https://doi.org/10.1088/1361-665X/aadbfb

Lee, Y.F., Lu, Y. and Guan, R. (2020), "Nonlinear guided waves for fatigue crack evaluation in steel joints with digital image correlation validation", *Smart Mater. Struct.*, **29**(3), 035031. https://doi.org/10.1088/1361-665X/ab6fe7

Li, M., Suzuki, Y., Hashimoto, K. and Sugiura, K. (2018), "Experimental study on fatigue resistance of rib-to-deck joint in orthotropic steel bridge deck", *J. Bridge Eng.*, **23**(2), 04017128. https://doi.org/10.1061/(ASCE)BE.1943-5592.0001175

Ly, C.D., Vo, T.H., Mondal, S., Park, S., Choi, J., Vu, T.T.H., Kim, C.S. and Oh, J. (2021), "Full-view in vivo skin and blood vessels profile segmentation in photoacoustic imaging based on deep learning", *Photoacoustics*, **25**, 100310. https://doi.org/10.1016/j.pacs.2021.100310

Papadimitriou, C., Fritzen, C.P., Kraemer, P. and Ntotsios, E. (2011), "Fatigue predictions in entire body of metallic structures from a limited number of vibration sensors using Kalman filtering", *Struct. Control Health Monitor.*, **18**(5), 554-573. https://doi.org/10.1002/stc.395

Park, J.-H., Huynh, T.-C., Choi, S.-H. and Kim, J.-T. (2015), "Vision-based technique for bolt-loosening detection in wind turbine tower", *Wind Struct., Int. J.*, **21**(6), 709-726. https://doi.org/10.12989/was.2015.21.6.709

Pham, H.C., Ta, Q.B., Kim, J.T., Ho, D.D., Tran, X.L. and Huynh, T.C. (2020), "Bolt-loosening monitoring framework using an image-based deep learning and graphical model", *Sensors*, **20**(12). https://doi.org/10.3390/s20123382

Ronneberger, O., Fischer, P. and Brox, T. (2015), "U-net: convolutional networks for biomedical image segmentation", In: *International Conference on Medical image computing and Computer-assisted Intervention*, 9351. https://doi.org/10.1007/978-3-319-24574-4_28

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

Sim, H.B. and Uang, C.M. (2012), "Stress analyses and parametric study on full-scale fatigue tests of rib-to-deck welded joints in steel orthotropic decks", *J. Bridge Eng.*, **17**(5), 765-773. https://doi.org/10.1061/(ASCE)BE.1943-5592.0000307

Soh, C.K. and Lim, Y.Y. (2009), "Detection and characterization of fatigue induced damage using electromechanical impedance technique", *Adv. Mater. Res.*, **79-82**, 2031-2034. https://doi.org/10.4028/www.scientific.net/AMR.79-82.2031

Spencer, 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

Sun, L.M., Zhang, W. and Nagarajaiah, S. (2019), "Bridge real-time damage identification method using inclination and strain measurements in the presence of temperature variation", *J. Bridge Eng.*, **24**(2), 04018111. https://doi.org/10.1061/(ASCE)BE.1943-5592.0001325

Sun, Y., Yang, Y., Yao, G., Wei, F. and Wong, M. (2021), "Autonomous crack and bughole detection for concrete surface image based on deep learning", *IEEE Access*, **9**, 85709-85720. https://doi.org/10.1109/ACCESS.2021.3088292

Ta, Q.B. and Kim, J.T. (2020), "Monitoring of corroded and loosened bolts in steel structures via deep learning and hough transforms", *Sensors*, **20**(23). https://doi.org/10.3390/s20236888

Xu, Y., Bao, Y., Chen, J., Zuo, W. and Li, H. (2018a), "Surface fatigue crack identification in steel box girder of bridges by a deep fusion convolutional neural network based on consumer-grade camera images", *Struct. Health Monitor*, **18**(3), 653-674. https://doi.org/10.1177/1475921718764873

Xu, Y., Li, S., Zhang, D., Jin, Y., Zhang, F., Li, N. and Li, H. (2018b), "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), 2075. https://doi.org/10.1002/stc.2075

Ya, S., Yamada, K. and Ishikawa, T. (2011), "Fatigue evaluation of rib-to-deck welded joints of orthotropic steel bridge deck", *J. Bridge Eng.*, **16**(4), 492-499. https://doi.org/10.1061/(ASCE)BE.1943-5592.0000181

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

Yao, L., Dong, Q., Jiang, J. and Ni, F. (2020), "Deep reinforcement learning for long-term pavement maintenance planning", *Comput.-Aided Civil Infrastr. Eng.*, **35**(11), 1230-1245. https://doi.org/10.1111/mice.12558

Ye, X., Jin, T. and Yun, C. (2019), "A review on deep learning-based structural health monitoring of civil infrastructures", *Smart Struct. Syst.., Int. J.*, **24**(5), 567-586. https://doi.org/10.12989/sss.2019.24.5.567

Zhao, X., Zhang, Y. and Wang, N. (2019), "Bolt loosening angle detection technology using deep learning", *Struct. Control Health Monitor.*, **26**(1), 2292. https://doi.org/10.1002/stc.2292

HJ