Structural Monitoring and Maintenance, *Vol. 9, No. 3 (2022) 289-303*
https://doi.org/10.12989/smm.2022.9.3.289

Pixel-based crack image segmentation in steel structures using atrous separable convolution neural network

Quoc-Bao Ta [1a], Quang-Quang Pham [1b], Yoon-Chul Kim [2c],
Hyeon-Dong Kam [1d] and Jeong-Tae Kim*[1]

[1] *Department of Ocean Engineering, Pukyong National University, Nam-gu, Busan 48513, Republic of Korea*
[2] *Department of Civil Engineering, Yonsei University, Seodaemun-gu, Seoul 03722, Republic of Korea*

(*Received July 9, 2022, Revised September 8, 2022, Accepted September 29, 2022*)

Abstract. In this study, the impact of assigned pixel labels on the accuracy of crack image identification of steel structures is examined by using an atrous separable convolution neural network (ASCNN). Firstly, images containing fatigue cracks collected from steel structures are classified into four datasets by assigning different pixel labels based on image features. Secondly, the DeepLab v3+ algorithm is used to determine optimal parameters of the ASCNN model by maximizing the average mean-intersection-over-union (mIoU) metric of the datasets. Thirdly, the ASCNN model is trained for various image sizes and hyper-parameters, such as the learning rule, learning rate, and epoch. The optimal parameters of the ASCNN model are determined based on the average mIoU metric. Finally, the trained ASCNN model is evaluated by using 10% untrained images. The result shows that the ASCNN model can segment cracks and other objects in the captured images with an average mIoU of 0.716.

Keywords: atrous convolution; crack identification; Deeplabv3+ network; semantic segmentation; steel structure; vision image

1. Introduction

During long-term services of steel bridges, fatigue cracks can be occurred at the connection details due to repeated vehicle loads and material flaws. Welding has been preferred for steel structures to form a structural unit which combines bridge deck, splice, and diaphragm. Due to intrinsic defects and repeated cyclic loadings, fatigue cracks are occurred at welded joints of the steel bridges (Battista and Pfeil 1999, Wang *et al.* 2019). The crack formation in the welded connection jeopardizes the structural safety, so it should be monitored periodically using reliable inspection techniques (Zhu *et al.* 2010, Lee *et al.* 2014, Mutlib *et al.* 2016, Wang *et al.* 2019).

Visual inspection is commonly used to monitor fatigue cracks in steel bridges. The inspection results mainly depend on the experiences of inspectors (Gallwey 1998a, b, See 2012, Campbell *et*

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

http://www.techno-press.org/?journal=smm&subpage=7 ISSN: 2288-6605 (Print), 2288-6613 (Online)

al. 2021). Moreover, the method is time-consuming and cost-expensive to cover the details of a long-span bridge. To overcome the issue, vision-based artificial intelligent methods have been developed using advanced vision sensing techniques and deep learning algorithms. The technique can provide efficient monitoring by acquiring structural information of many local points using digital images captured by cameras. The advantages of the approach include low-cost, non-contact sensing, and time-saving thanks to advances in cloud-based computation, transfer learning, and computer hardware (i.e., GPU) and software (i.e., Matlab, Python, cloud-based platforms).

Many researchers have worked on image-based methods to detect existing fatigue cracks (Dung *et al*. 2019, Dong *et al*. 2021, Ye *et al*. 2019). For the application of the image-based methods to steel bridges, image features of cracks and other non-damage features should be distinctly segmented from the captured raw images. Also, pre-trained computer algorithms should be implemented to minimize collections of training datasets (Cha *et al*. 2017, Dung *et al*. 2019). For orthotropic steel bridges, captured images possibly consist of complex backgrounds such as handwritings, marks, and others (ruler, contour) for periodic visual inspections. An 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 affect a crack detection result using computer vision algorithms. Thus, the segment of obstacle background (marks, ruler, contour) is to minimize the error in crack identification.

The effect of complex backgrounds (e.g., ruler and handwriting) on the accuracy of crack detection has not been examined comprehensively so far. Although recent research efforts yielded better fatigue-crack detection results, the crack segmentation by atrous convolution-based Deeplabv3+ network (ACDN) should overcome the following issues: (1) the effect of obstacles (e.g., ruler, handwriting, and weld line), and (2) the optimal parameters for training the ASCNN model.

Recent research efforts show that advanced deep learning techniques could yield better crack detection results. The accuracy of vision-based damage detection mainly relies on both training datasets and deep learning algorithms (Spencer *et al*. 2019, Barbedo 2018, Bailly *et al*. 2022). In this study, the atrous separable convolution neural network (ASCNN) is examined for crack identification in steel bridges by considering the effect of non-damage features (e.g., handwriting and odd marks) and the impact of assigned pixel labels for the ASCNN. Firstly, images containing fatigue cracks collected from steel structures are classified into four datasets by assigning different pixel labels based on image features. Secondly, the DeepLab v3+ algorithm is used to determine optimal parameters of the ASCNN model by maximizing the average mean-intersection-over-union (mIoU) metric of the datasets. Thirdly, ASCNN models are trained for various image sizes and hyper-parameters such as learning rule, learning rate, and epoch. The optimal parameters for the ASCNN model are determined based on the average mIoU metric. Finally, the performance model is evaluated by using 10% untrained images.

2. Training datasets for ASCNN

2.1 Raw images captured from steel bridge

The dataset includes 200 RGB images captured with various perspective angles and distances by different bridge inspectors (Bao *et al*. 2021). The images have two different resolutions of

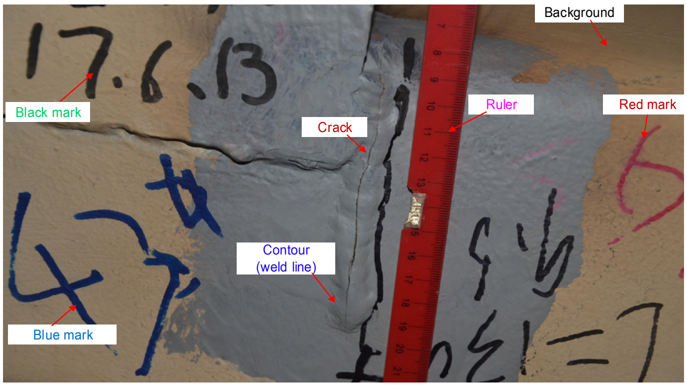


Fig. 1 Captured raw image of steel bridge deck

Table 1 Description of four datasets from labeling crack and other non-damage features

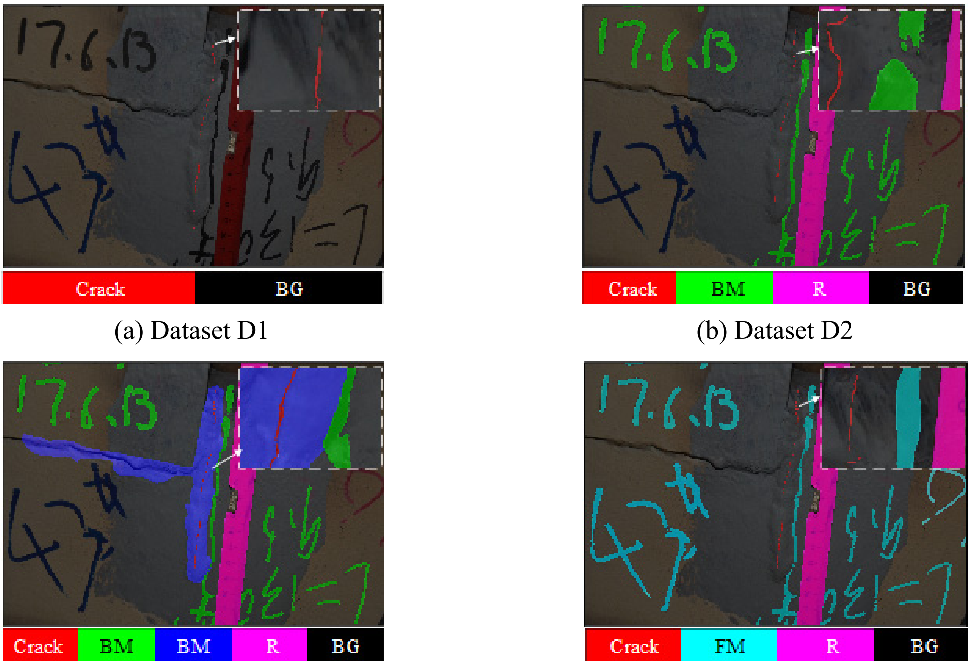
Dataset	Number of classes	Features
D1	2	Crack and Background (BG)
D2	4	Crack, Black Mark (BM), Ruler (R), and BG
D3	5	Crack, BM, Contour (C), R, and BG
D4	4	Crack, Full Mark (FM), R, and BG

height × width × channels pixel: $3264 \times 4928 \times 3$ and $3864 \times 5152 \times 3$. Each image contains cracks, contour (welding lines), handwriting with different colors, and background, and it could also have a ruler. As shown in Fig. 1, image features are classified as ‘crack’, ‘black mark (BM)’, ‘blue mark’, ‘red mark’, ‘ruler’, ‘contour’, and ‘background’.

To examine the accuracy of image-based crack identification using ASCNN, the image features were classified by labeling crack and other non-damage features. Based on their distinct characteristics, they were sorted into four different datasets D1-D4 (see Table 1). In the dataset D1, the features were labeled into two classes: crack and background (BG). In the dataset D2, the features were labeled into four classes: crack, black mark (BM), ruler (R), and background (BG). In the dataset D3, the features were labeled into five classes: crack, BM, contour, R, and BG. In the dataset D4, the features were labeled into four classes, which were crack, full mark (FM), ruler and BG. All datasets commonly had ‘crack’ and ‘background’ classes. Also, other classes such as marks, contour, and ruler were parametrically examined in the datasets D1-D4.

2.2 Labeling images of four datasets

The Image Labeler application in Matlab 2020a was utilized to assign pixel labels to each class and to generate databanks for the datasets D1-D4. As shown in Fig. 2, four datasets of the annotated ground truth of an image (e.g., image number 178) were built as follows: two classes for dataset D1 (see Fig. 2(a)), four classes for dataset D2 (see Fig. 2(b)) and D4 (see Fig. 2(d)), and five classes for dataset D3 (Fig. 2(c)). It is noted that colored class-labels represent class-domains. As shown in Fig. 1, the color pixels represent the corresponding learning areas in the original images.



(a) Dataset D1

(b) Dataset D2

(c) Dataset D3

(d) Dataset D4

Fig. 2 Labeled images of four datasets D1-D4

The 1st International Project Competition for Structural Health Monitoring (IPC-SHM 2020) released a total of 200 images of steel bridges which included 120 and 80 real images in sizes of $3264 \times 4928 \times 3$ and $3864 \times 5152 \times 3$, respectively. In this study, 180 images (90%) were randomly selected for training datasets, and the remaining 20 images (10%) were allocated for evaluating the trained datasets. By adopting data argumentation techniques, the horizontal reflection with 50% probability and the image translation randomly up to 10 pixels were utilized to increase the number of training samples and also to reduce the over-fitting issue.

It is noted that the selected features (Crack, BG) in D1 have been used to train the ASCNN model for deeply concentrating on crack identification. The datasets D2 and D3 were established to train the ASCNN models for observing the error levels in crack identification. Besides, the dataset D4 was established for the purpose of indicating the effects of color marks on crack identification.

3. Training datasets for ASCNN-based DeepLab v3+

3.1 ASCNN-based DeepLab v3+

The state-of-the-art architecture of DeepLab v3+ for the task of semantic segmentation was proposed by Chen *et al.* (2018). Compared to other deep-learning networks, such as fully convolutional network (Yang *et al.* 2018), U-net (Ronneberger *et al.* 2015), the Deeplabv3+ network (Chen *et al.* 2018) was constructed based on two encoder and decoder modules, which

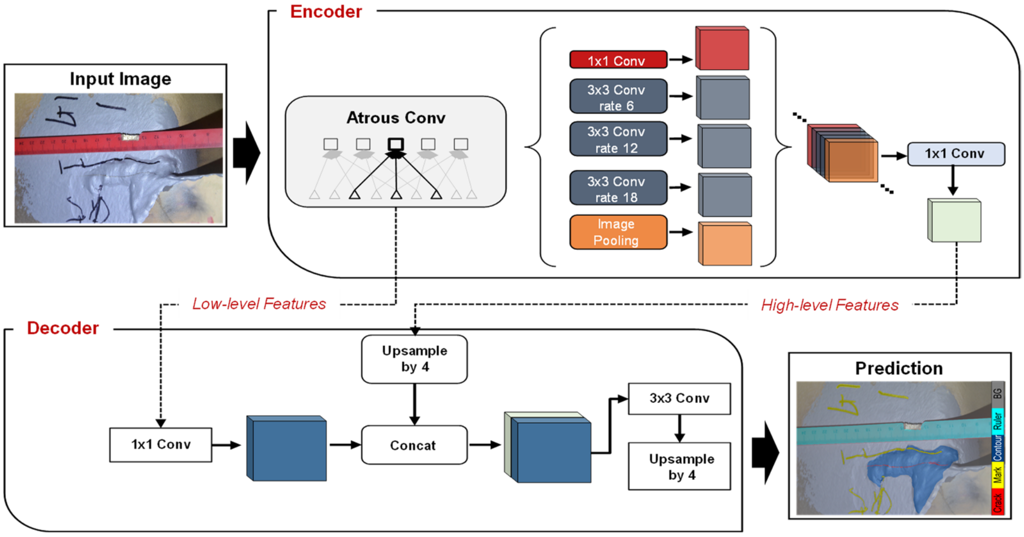


Fig. 3 Architecture of DeepLab v3+ for semantic segmentation of crack and other features

enable to recognize irregular distributions of tinny cracks and crack characteristics. Also, the atrous separable convolution makes the Deeplabv3+ faster (i.e., significant reduction of the computational complexity) and stronger in feature learning.

As shown in Fig. 3, an encoder-decoder architecture of ASCNN was designed for multi-class (e.g., crack and other features) semantic segmentation. For better performance, DeepLab v3+ adopted the spatial pyramid pooling module and the encoder-decoder architecture. The spatial pyramid pooling module can encode multi-scale object information through multiple ASCNNs with various rates. By the encoder-decoder architecture, the boundary of objects can be accurately estimated by gradually recovering spatial information.

Within the encoder, features can be drawn from the deep convolutional neural network (DCNN) to any resolution through atrous convolution. By inserting *Input Image* into the encoder, accurate semantic segmentation can be made by using atrous spatial pyramid pooling (ASPP) techniques. Atrous convolution runs at different rates in parallel with the feature map and then re-mix them. Also, DeepLab v3+ adds a process of 1x1 convolution to the encoder's final output and then bi-linear up-sampling to the concatenate. This allows effective management of object segmentation by reducing the channel during the decoder process

The encoder architecture robustly reduces image characteristics to capture higher semantic information. Meanwhile, the decoder part gradually recovers previous spatial image features. In order to segment a few features (e.g., crack, marks, ruler, and background), the ASCNN was fine-tuned for DeepLab v3+ architecture with the resnet50-based convolutional neural network (CNN) backbone (reference). The hardware capacity of the ASCNN model includes a desktop computer with i9-9900 @ 3.6 GHz CPU, 64 GB of RAM, and an 11 GB memory NVIDIA RTX2080Ti graphics processing unit (GPU) using Matlab 2020a.

The performance of the network was evaluated based on the average value of mIoU (mean-intersection-over-union) on the whole testing dataset. As described in Eq. 1, the average mIoU score can be defined from the concept of true-positive (TP), false-positive (FP), false-negative (FN) predictions, n class, and the sum of testing images N. A true positive result is estimated by the

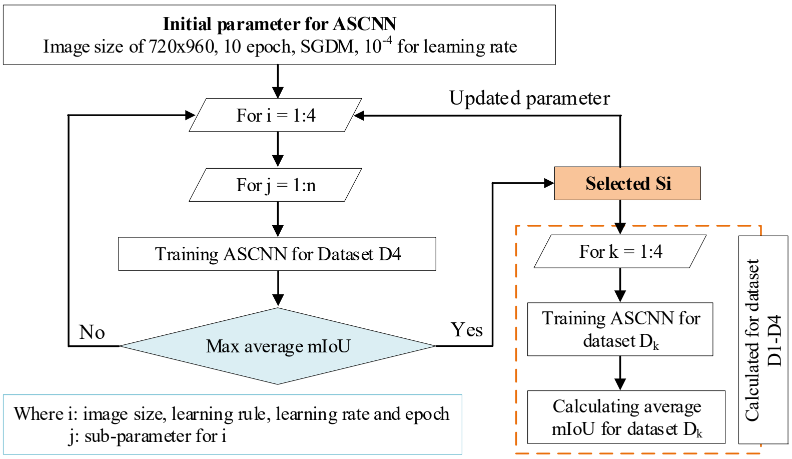

Initial parameter for ASCNN
Image size of 720x960, 10 epoch, SGDM, 10^{-4} for learning rate
For i = 1:4
Updated parameter
For j = 1:n
Selected Si
Training ASCNN for Dataset D4
For k = 1:4
No
Max average mIoU
Yes
Training ASCNN for dataset D$_k$
Calculating average mIoU for dataset D$_k$
Calculated for dataset D1-D4
Where i: image size, learning rule, learning rate and epoch
j: sub-parameter for i

Fig. 4 Parameter optimization process for ASCNN model

Table 2 Selected parameters of ASCNN model for datasets D1-D4

Parameters	Sub-Parameters			
	1	2	3	4
Image size (P1)	360×480 (P11)	720×960 (P12)	1080×1440 (P13)	
Learning rule (P2)	SGDM (P21)	ADAM (P22)	RMSPROP (P23)	
Learning rate (P3)	10^{-2} (P31)	10^{-3} (P32)	10^{-4} (P33)	10^{-5} (P34)
Epoch (P4)	10 (P41)	20 (P42)	30 (P43)	40 (P44)

intersection of false positive and false negative outputs. Here, the false positive (i.e., alarmed but not real) output is quantified based on the predicted pixels in class; meanwhile, the false negative (i.e., real but not alarmed) output is quantified based on the ground truth pixels in class.

$$average \quad mIoU = \frac{1}{N}\left(\frac{1}{n}\sum_{i=1}^{n}\frac{TP_i}{TP_i + FP_i + FN_i}\right) \tag{1}$$

3.2 Optimal parameters for ASCNN model

It is known that the input parameters for the architecture had significant effects on the accuracy of object detection (Zhao *et al*. 2019, Alzubaidi *et al*. 2021). The mIoU was selected as an objective function for the optimization process. As shown in Fig. 4, a workflow of parameter optimization was designed for ASCNN model to maximize mIoU values of the training datasets. The initial parameters for ASCNN model were selected as follows: epochs of 10, stochastic gradient descent method (SGDM) for learning rule, learning rate of 10^{-4}, learning rate drop period of 10, momentum of 0.9, mini-batch size of 1, and image size of 720×960 pixels. The best hyper-parameters for training datasets D1-D4 were examined from the selected parameters listed in Table 2.

The four examined parameters include image size (P1), learning rule (P2), learning rate (P3), and epoch (P4). The image size (P1) was tested for three sub-parameters P11-P13 (i.e., 360×480, 720×960, and 1080×1440 pixels) to find the maximum of average mIoU (in scenario S_1). The selected parameter S_1 was updated to the ASCNN's parameters and set as the basic parameter for the next step. The learning rule (P2) was examined for three sub-parameters P21-P23 (i.e., SGDM, ADAM, and RMSPROP) to search for the maximum of mIoU. The selected learning rule was updated into the ASCNN's parameters. Then, the learning rate (P3) and the epoch (P4) were analyzed for four sub-parameters, P31-P34 and P41-P44, respectively. Additionally, the optimal parameters for the four parameters (P1-P4) were utilized to compute an average mIoU for all datasets (D1-D4).

4. ASCNN-based crack identification

4.1 Training results

As illustrated in Fig. 5, training accuracies and loss values of the defined classes were estimated for the dataset D1-D4. The accuracy was defined as the proportion of the correct identification. The figure indicates that the loss decreased and the accuracy increased as the training iteration increased for all datasets. Also, insignificant changes in the accuracy were observed beyond the training iteration of 1200. The accuracies and loss values of the datasets D1-D4 were observed as follows: (1) Dataset D1 had the highest accuracy (97%) and also the lowest loss value (0.04); (2) Datasets D2 and D4 had the relatively high accuracy (95%) and the relatively high loss value (0.09); and (3) Dataset D3 also had the relatively high accuracy (92%) but the highest loss value (0.18).

As observed in Fig. 6, the overlapped bar chart and the line graph had a nonlinear correlation between the global accuracy (i.e., bar chart) and averages of mIoU (i.e., line graph). The global accuracy (gAcc) is defined as the ratio of the correctly classified pixels to the total number of pixels. The gAcc and mIoU are discordant, as follows: (1) Dataset D1 had the lowest mIoU (0.52), but the highest gAcc (98.7%); (2) Datasets D4 and D2 had similar gAcc values (i.e., 96.5% and 96.2%, respectively), but the mIoU of the dataset D4 (0.615) is higher than that of the dataset D2

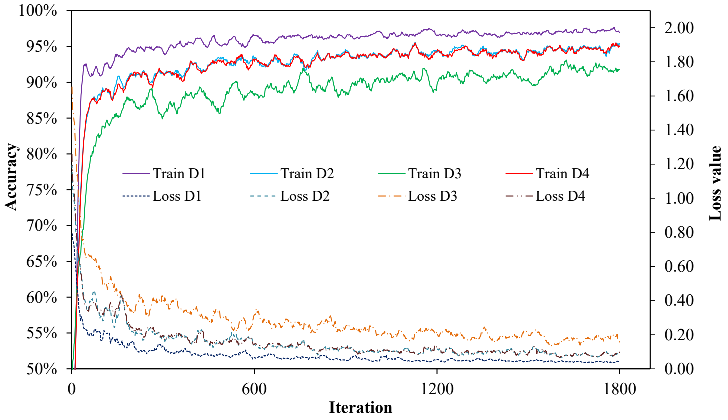


Fig. 5 Accuracy and loss value of the labeled datasets D1-D4 during training process

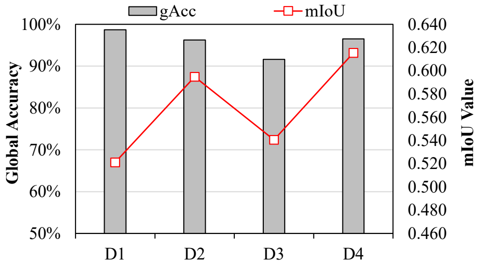


Fig. 6 Global accuracy and average mIoU of datasets D1-D4

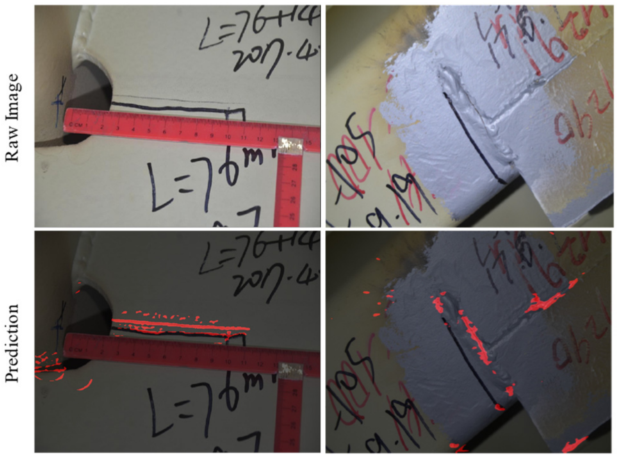


Fig. 7 Raw images and prediction results obtained from dataset D1

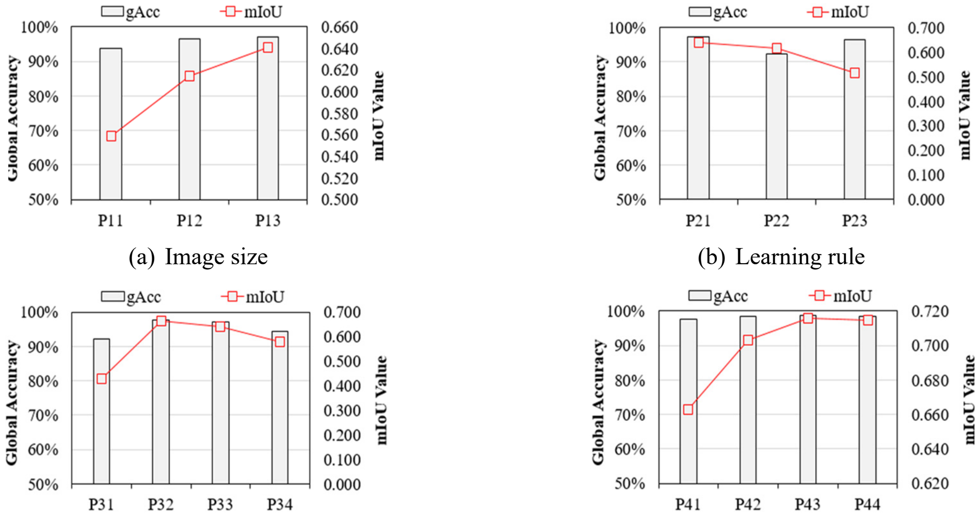
(0.595), and (3) Dataset D3 had the relatively low accuracy (91.6%) and the insignificant mIoU (0.54).

In the dataset D1, the mIoU value decreased significantly due to the unwanted recognition faults (i.e., false-positive errors) between the mark and the crack (see Fig. 7). Meanwhile, the mIoU obtained from the dataset D4 was the highest among the four datasets. The dataset D4 was selected to conduct optimization of ASCNN's parameters for image-based crack identification.

4.2 Optimal parameters for ASCNN-based crack identification

The iterative process was performed for four parameters which included image size, learning rule, learning rate, and epoch (see Table 2). As shown in Fig. 5, mIoU values were maximized for dataset D4 with workflow. The overlapped bar chart and line graph were utilized to illustrate the relationship between the gAcc (bar chart) and mIoU (line graph). Fig. 8 shows the optimal ASCNN parameters for dataset D4. As shown in Fig. 8(a), both gAcc and mIoU values were increased as image size was increased. The image size of 1080×1440 pixel resolutions (P13 in Table 2) was selected as the optimal one.

The difference in learning rules reflects the different behavior in the calculation and update weights and biases in the training process (Ruder 2016). As shown in Fig. 8(b), the SGDM method (P21 in Table 2) resulted in the highest mIoU. Then P21 was selected for the learning rules.



(a) Image size

(b) Learning rule

(c) Learning rate (Lr)

(d) Epoch numbers

Fig. 8 Effects of ASCNN parameters on global accuracy and mIoU values computed using dataset D4

Table 3 Average of mIoU with selected optimal parameters

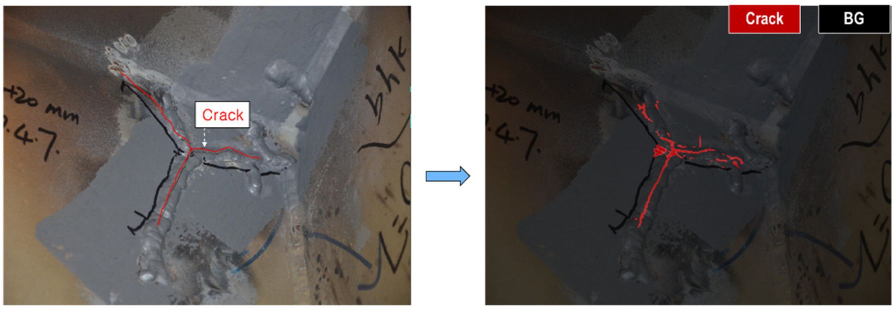
Parameter	**D1**	**D2**	**D3**	**D4**
Image size	0.540	0.626	0.566	0.641
Learning rule	0.540	0.626	0.566	0.641
Learning rate	0.553	0.675	0.604	0.663
Epoch	***0.596***	***0.708***	***0.642***	***0.716***

As shown in Fig. 8(c), the learning rate of 10^{-3} (P32 in Table 2) yielded the highest accuracy and highest mIoU among for examined parameters. Then P32 was selected and updated to ANCNN models. As shown in Fig. 8(d), the higher the mIoU was produced by the epoch of 30 (P43 in Table 2). Then P43 was selected for ASCNN models. It is observed that the mIoU value was decreased beyond the epoch of 30. It is also noted that the number of epochs had a significant effect on training time.

As schematized in Fig. 4, the selected hyper-parameters (image size, learning rule, learning rate, and epoch) were utilized to compute the mIoU values for all datasets. The mIoU values are listed in Table 3. For the dataset D4, the maximum mIoU value (0.716) was produced by the optimal parameters of ASCNN model: image size of 1080×1440 pixel, SGDM for learning rule, 10^{-3} for learning rate, and 30 epoch. Among the four parameters, the number of epochs had the most sensitive effect on the mIoU metric.

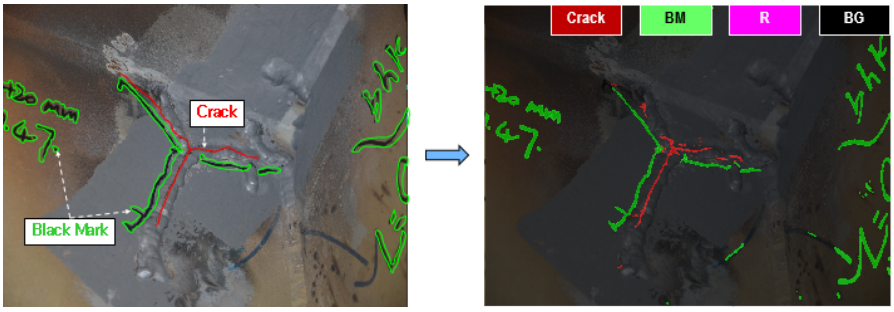
4.3 Crack segmentation results for four datasets

As shown in Figs. 9-12, the cracks and the other objects in the unstrained images (e.g., image number 57) were estimated using the trained models. Each class-pixel color followed the corresponding color in sub-session training and testing data generation. For the dataset D1, the labeling was made on two classes (crack and background), and the crack segmentation was estimated using the ASCNN model (see Fig. 9). The crack was detected with some false positives along the welding lines. For the dataset D2, the labeling was made on three classes (crack, black marks, and background), and the three objects were detected as shown in Fig. 10. All objects were detected, but the cracks were detected with lower accuracy as compared to the result of the dataset D1. For the dataset D3, the labeling was made on five classes (crack, black mark, contour, ruler,



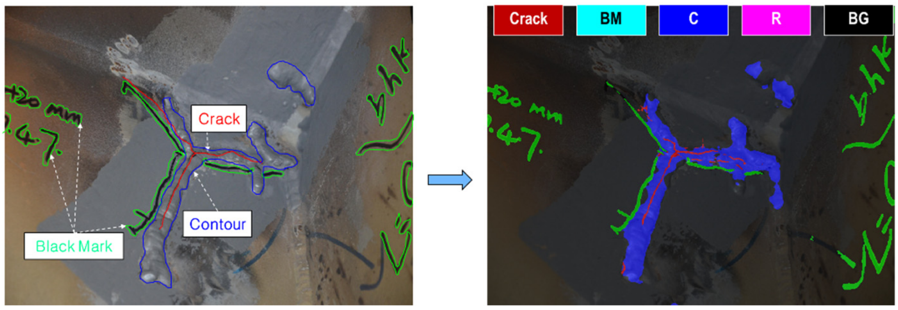
(a) Class labeling (b) Crack segmentation

Fig. 9 Crack estimation using untrained image from dataset D1



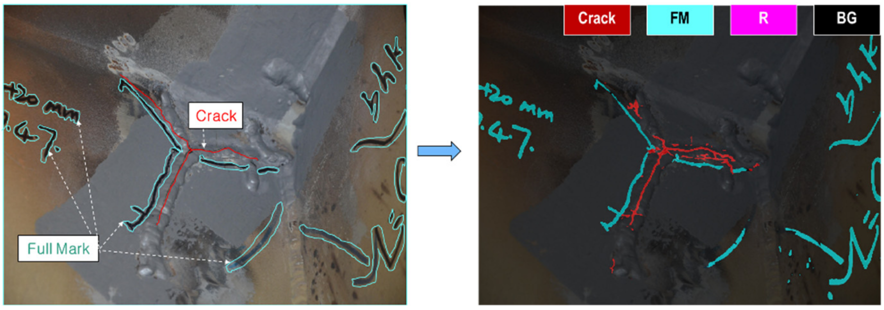
(a) Class labeling (b) Multi-class segmentations

Fig. 10 Crack estimation using untrained image from dataset D2



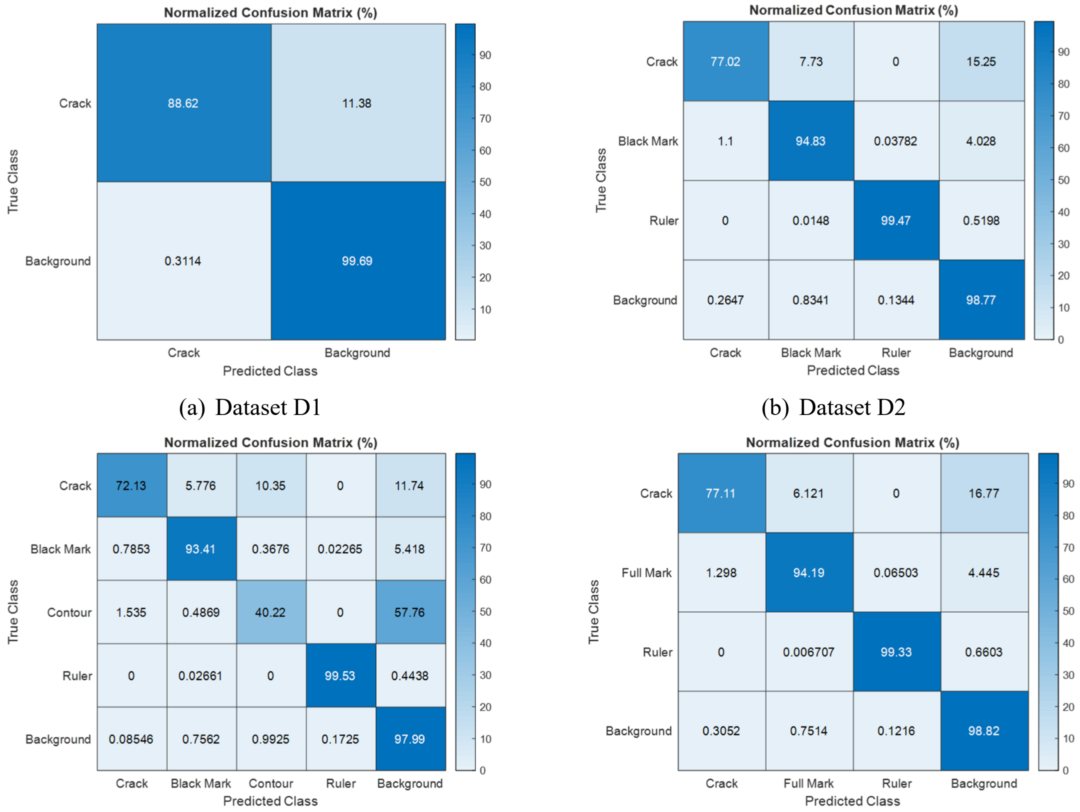
(a) Class labeling (b) Multi-class segmentations

Fig. 11 Crack estimation using untrained image from dataset D3



(a) Class labeling (b) Multi-class segmentations

Fig. 12 Crack estimation using untrained image from dataset D4



Normalized Confusion Matrix (%)

True Class \ Predicted Class	Crack	Background
Crack	88.62	11.38
Background	0.3114	99.69

(a) Dataset D1

Normalized Confusion Matrix (%)

True Class \ Predicted Class	Crack	Black Mark	Ruler	Background
Crack	77.02	7.73	0	15.25
Black Mark	1.1	94.83	0.03782	4.028
Ruler	0	0.0148	99.47	0.5198
Background	0.2647	0.8341	0.1344	98.77

(b) Dataset D2

Normalized Confusion Matrix (%)

True Class \ Predicted Class	Crack	Black Mark	Contour	Ruler	Background
Crack	72.13	5.776	10.35	0	11.74
Black Mark	0.7853	93.41	0.3676	0.02265	5.418
Contour	1.535	0.4869	40.22	0	57.76
Ruler	0	0.02661	0	99.53	0.4438
Background	0.08546	0.7562	0.9925	0.1725	97.99

(c) Dataset D3

Normalized Confusion Matrix (%)

True Class \ Predicted Class	Crack	Full Mark	Ruler	Background
Crack	77.11	6.121	0	16.77
Full Mark	1.298	94.19	0.06503	4.445
Ruler	0	0.006707	99.33	0.6603
Background	0.3052	0.7514	0.1216	98.82

(d) Dataset D4

Fig. 13 Confusion matrix obtained from scenario S4 for four datasets

and background), and the five objects were detected as shown in Fig. 11. All five objects were detected with various accuracies. For the last dataset D4, the labeling was made on four classes (crack, full marks, ruler, and background), and the four objects were detected, as shown in Fig. 12. All four objects were also detected, but the crack was detected with lower accuracy than those of other datasets.

As shown in Fig. 13, the normalized confusion matrices (i.e., the error matrix) of the dataset D1-D4 were estimated by using 10% of untrained images in order to assess the performance of the classification models. For the dataset D1, the crack was identified with an accuracy of 88.62%, which is identical to the true positive (TP) (see Fig. 13(a)). Also, the accuracy of the dataset D1 was the highest value among the four datasets. Meanwhile, the average of mIoU (0.596) was the smallest value. For the datasets D2-D4, the accuracies of the crack-pixel labels were 77.02%, 72.13%, and 77.11%, respectively. Meanwhile, the average mIoU of dataset D4 (0.716) yielded the highest value. Conclusively, the dataset D4 produced the highest average mIoU (0.716).

From the object detection results using the image-based ASCNN model, at least five observations were made as follows. Firstly, the result of object detection relied on the databank, especially labeling objects in the image. Secondly, each defined class in an image should be correctly and adequately labeled in order to improve the average mIoU. Thirdly, the global accuracy and the average of mIoU of the pixel-level crack classification model were improved by the high image resolution. The crack was thin as compared to other objects. The reduction of image resolution resulted in the loss of crack pixels, thus resulting in lower global accuracy or the mIoU metric. It is also noted that the increment of image resolution requires a stronger GPU and a longer computational time-consuming. Fourthly, the selection of training algorithms had little effect on image-based identification results. The different learning rules could produce slightly different mIoU values. Since each learning rule shows advantages for object detection algorithms, the learning rule should be tested and experienced by researchers. Finally, the increase in the training epochs could boost the model's accuracy.

5. Conclusions

In this study, the fatigue cracks of the steel box girder were segmented using an image-based ASCNN technique. Firstly, the features of images were classified into four groups based on their characteristics, and labeling of these features was conducted to build databanks. Secondly, the state-of-the-art ASCNN was selected for object segmentation. The algorithm was proposed to determine optimal parameters of the ASCNN model by maximizing the average mIoU metric of training datasets. Thirdly, the ASCNN models were trained with the input of various image sizes and hyper-parameters, including learning rule, learning rate, and epoch. The optimal parameters for the ASCNN model were determined based on the average mIoU metric. Finally, the performance model was evaluated by using 10% untrained images.

From the image-based object segmentation using ASCNN, the folowing concluding remarks can be drawn: (1) The ASCNN was successfully implemented for segmentation of the crack, ruler, handwriting, welding line, and background; (2) For a single computer (i9-9900, 11GB of GPU), the optimal parameters for the ASCNN model for object segmentation were selected as follows: image size of 1080×1440 pixel, SGDM for learning rule, 10-3 for learning rate and 30 epoch, and (3) The trained model with the labeling of crack, full mark, ruler and background yielded the highest average mIoU value. It is noted that the model trained in this study might give wrong predictions when the test image has a large difference from the training dataset (different surface texture, crack width, the human-made drawing, lighting conditions, etc.

As compared to the recent frameworks developed for vision-based crack detection (Dong *et al.* 2021), the ASCNN with trained D1 yielded 59.6 % of mIoU for 30 epochs. It was higher than that of fully connected network-based framework (51.9% of mIoU for 45 epochs). Also, it was lower than the accuracy of Unet-based Framework (62.4% of mIoU for 45 epochs).

Despite the promising result, the crack segmentation achieved lower 90% accuracy. In the future study, the performance of the ASCNN model (i.e., accuracy and computational cost) should be extensively examined as compared to other state-of-the-art methods. Furthermore, data argumentation techniques, including horizontal reflection and image translation for enlarging the diversity of the training dataset, should be employed to improve crack segmentation results.

Author contributions

Quoc-Bao Ta and Jeong-Tae Kim developed the methodology; Quoc-Bao Ta performed the framework design and simulation; Quang-Quang Pham designed the logic of the manuscript; Yoon-Chul Kim provided labeling images for datasets D2-D3; Hyeon-Dong Kam provided labeling images for datasets D1; Jeong-Tae Kim revised the manuscript and supervised the whole work. All authors have read and agreed to the submitted version of the manuscript.

Acknowledgments

This work was supported by the Basic Science Research Program through the National Research Foundation of Korea (2022R1A2C10038891161782064340101). 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.

References

Alzubaidi, L., Zhang, J., Humaidi, A.J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., Santamaría, J., Fadhel, M.A., Al-Amidie, M. and Farhan, L. (2021), "Review of deep learning: concepts, CNN architectures, challenges, applications, future directions", *J. Big Data*, **8** (1), 1-74. https://doi.org/10.1186/s40537-021-00444-8

Bailly, A., Blanc, C., Francis, E., Guillotin, T., Jamal, F., Wakim, B. and Roy, P. (2022), "Effects of dataset size and interactions on the prediction performance of logistic regression and deep learning models", *Comput. Methods Programs Biomed.*, **213**, 106504. https://doi.org/10.1016/j.cmpb.2021.106504

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 Agric.*, **153**, 46-53. https://doi.org/10.1016/j.compag.2018.08.013

Battista, R. and Pfeil, M. (1999), "Fatigue cracks induced by traffic loading on steel bridges' slender orthotropic decks", The Ninth International Conference on Computational Methods and Experimental Measurements", *WIT Transact. Modell. Simul.*, **21**, 37-46. https://www.witpress.com/elibrary/wit-transactions-on-modelling-and-simulation/22/4967

Campbell, L.E., Connor, R.J., Whitehead, J.M. and Washer, G.A. (2021), "Human factors affecting visual inspection of fatigue cracking in steel bridges", *Struct. Infrastruct. Eng.*, **17**(11), 1447-1458.

https://doi.org/10.1080/15732479.2020.1813783

Cha, Y.-J., Choi, W. and Büyüköztürk, O. (2017), "Deep learning-based crack damage detection using convolutional neural networks", *Comput.-Aided Civ. 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. (2018), "Deeplab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs", *IEEE Trans. Pattern. Anal. Mach. Intell.*, **40**(4), 834-848. Doi: 10.1109/TPAMI.2017.2699184

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). https://doi.org/10.3390/s21124135

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

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

Gallwey, T.J. (1998a), "Evaluation and control of industrial inspection: Part I-Guidelines for the practitioner", *Int. J. Ind. Ergon.*, **22**(1-2), 37-49. https://doi.org/10.1016/S0169-8141(97)00066-8

Gallwey, T.J. (1998b), "Evaluation and control of industrial inspection: Part II–The scientific basis for the guide", *Int. J. Ind. Ergon.*, **22**(1-2), 51-65. https://doi.org/10.1016/S0169-8141(97)00067-X

Lee, J.K., Bae, D.S., Lee, S.P. and Lee, J.H. (2014), "Evaluation on defect in the weld of stainless steel materials using nondestructive technique", *Fusion Eng. Des.*, **89**(7-8), 1739-1745. https://doi.org/10.1016/j.fusengdes.2013.12.026

Mutlib, N.K., Baharom, S.B., El-Shafie, A. and Nuawi, M.Z. (2016), "Ultrasonic health monitoring in structural engineering: buildings and bridges", *Struct. Control Health Monit.*, **23**(3), 409-422. https://doi.org/10.1002/stc.1800

Ronneberger, O., Fischer, P. and Brox, T. (2015), "U-net: Convolutional networks for biomedical image segmentation", *Proceedings of International Conference on Medical Image Computing and Computer-Assisted Intervention*, Munich, Germany, October. https://doi.org/10.1007/978-3-319-24574-4_28

Ruder, S. (2016), "An overview of gradient descent optimization algorithms", *arXiv preprint*, arXiv:1609.04747. https://doi.org/10.48550/arXiv.1609.04747

See, J.E. (2012), "Visual inspection: a review of the literature", Report SAND2012-8590; Accessed on 2022.03.04. https://www.osti.gov/servlets/purl/1055636

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

Wang, X. and Wu, N. (2019), "Crack identification at welding joint with a new smart coating sensor and entropy", *Mech. Syst. Signal Process.*, **124**, 65-82. https://doi.org/10.1016/j.ymssp.2019.01.044

Wang, Y., Fu, Z., Ge, H., Ji, B. and Hayakawa, N. (2019), "Cracking reasons and features of fatigue details in the diaphragm of curved steel box girder", *Eng. Struct.*, **201**, 109767. https://doi.org/10.1016/j.engstruct.2019.109767

Xu, Y., Bao, Y., Chen, J., Zuo, W. and Li, H. (2018), "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

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 Civ. Infrastruct. 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 Civ. Infrastruct. 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), 567-585. https://doi.org/10.12989/sss.2019.24.5.567

Zhao, Z.Q., Zheng, P., Xu, S.T. and Wu, X. (2019), "Object detection with deep learning: a review", *IEEE*

Trans. *Neural Netw*. *Learn*. *Syst*., **30**(11), 3212-3232. Doi: 10.1109/TNNLS.2018.2876865

Zhu, X.P., Rizzo, P., Marzani, A. and Bruck, J. (2010), “Ultrasonic guided waves for nondestructive evaluation/structural health monitoring of trusses”, *Meas*. *Sci*. *Technol*., **21**(4). https://doi.org/10.1088/0957-0233/21/4/045701

TYx