DOI: 10.1111/mice.12971

RESEARCH ARTICLE

COMPUTER-AIDED CIVIL AND INFRASTRUCTURE ENGINEERING WILEY

Recording of bridge damage areas by 3D integration of multiple images and reduction of the variability in detected results

Tatsuro Yamane[1] | **Pang-jo Chun**[2] | **Ji Dang**[3] | **Riki Honda**[1]

[1]Department of International Studies, The University of Tokyo, Chiba, Japan

[2]Department of Civil Engineering, The University of Tokyo, Tokyo, Japan

[3]Department of Civil and Environmental Engineering, Saitama University, Saitama, Japan

Correspondence
Pang-jo Chun, Department of Civil Engineering, The University of Tokyo, Tokyo, Japan.
Email: chun@g.ecc.u-tokyo.ac.jp

Funding information
JST [Moonshot Research and Development], Grant/Award Number: JPMJMS2032; JSPS KAKENHI, Grant/Award Number: 21H01417 22J14364

Abstract

Machine learning models have been developed to perform damage detection using images to improve bridge inspection efficiency. However, in damage detection using images alone, the 3D coordinates of the damage cannot be recorded. Furthermore, the accuracy of the detection depends on the quality of the images. This paper proposes a method to integrate and record the damage detected from multiple images into a 3D model using deep learning to detect the damage from bridge images and structure from motion to identify the shooting position. The proposed method reduces the variability of the detection results between images and can assess the scale of damage or, conversely, where there is no damage and the extent of inspection omissions. The proposed method has been applied to a real bridge, and it has been shown that the actual damage locations can be recorded as a 3D model.

1 | INTRODUCTION

The aging of infrastructures, such as bridges and tunnels, has become a serious problem in various countries. For example, in the United States, 42% of the bridges were built more than 50 years ago, and 7.5% of the bridges in the country were reported to be in poor condition (ASCE, 2021). In Canada, 12% of bridges and tunnels are reported to be in poor or very poor condition (ACEC et al., 2019). Japan's Ministry of Land, Infrastructure, Transport, and Tourism (MLIT) has reported that 67% of the country's bridges will be more than 50 years old by 2033 (MLIT, 2018). Without proper maintenance, bridges and tunnels may collapse in worst-case scenarios (Horgan, 2019; Kawahara et al., 2014). Therefore, to operate aging infrastructure structures safely, it is necessary to conduct regular inspections to record the damage status and take appropriate measures, such as repair and reinforcement, depending on the degree of damage.

In the case of Japan, inspectors are required to approach the bridge and visually inspect it using periodic bridge inspection guidelines (MLIT, 2018). However, visual inspection is subject to variations in evaluation by different inspectors. Then, there is a risk that appropriate measures will not be taken, such as postponing the repair or reinforcement of bridges that should be prioritized. Additionally, inspections at high elevations require access to bridge members, for example, using bridge inspection vehicles, which results in a large budget and impacts vehicular traffic. Therefore, the development of efficient

structural health monitoring technology has become an important topic in recent years (Amezquita-Sanchez & Adeli, 2019; Amezquita-Sanchez et al., 2017). Various vibration-based and machine learning-based approaches have been proposed for efficient structural health monitoring and maintenance (Amezquita-Sanchez et al., 2018; Li et al., 2017; Perez-Ramirez et al., 2016). In recent years, there has been considerable research into damage detection using deep learning, among other machine learning techniques. These methods can be applied even where it is difficult to visually inspect the member and are expected to provide quantitative evaluation because they can remove the subjective judgment of humans.

Many studies on damage detection using images have focused on crack detection (Chun, Izumi, & Yamane, 2021; Prasanna et al., 2014; Sari et al., 2019; Yeum & Dyke, 2015). In recent years, many deep learning-based algorithms have been developed for crack detection via image classification (Y. Zhang & Yuen, 2021) and object detection (Deng et al., 2020), as well as pixel-level crack detection using images of road surfaces (Bang et al., 2019; J. Liu et al., 2020; Zhang et al., 2017) and concrete structures (Jang et al., 2021; Miao et al., 2021; Sajedi & Liang, 2021; Yamane & Chun, 2020; Yang et al., 2018). In particular, because deep learning can detect damage with high accuracy without extracting features in advance, its application has been expanded to detect various types of damage other than cracks. For example, research has been conducted to detect corrosion damage in images (Fondevik et al., 2020; Luo et al., 2021; Rahman et al., 2021; Xu et al., 2020) and to detect concrete delamination at the pixel level (Liang, 2019). Furthermore, several studies have been conducted to evaluate the degree of damage to structures based on vibrations (Chun et al., 2020; Lee & Kim, 2007; Mariniello et al., 2021; Rafiei & Adeli, 2017, 2018a; Soto & Adeli, 2019). In addition to directly detecting damage, various applications based on machine learning techniques have been considered, such as the assessment of the mechanical properties of damaged steel structures (Chun et al., 2019; Karina et al., 2017; Mazumder et al., 2021), estimation of strain response of the column (Oh et al., 2017), evaluation of seismic and post-earthquake soundness (Nagatani et al., 2021; Perez-Ramirez et al., 2019), description of damage situations from bridge images (Chun, Yamane, & Maemura, 2021), estimation of the mechanical and material performance of concrete structures (Nguyen et al., 2019; Okazaki et al., 2020; Rafiei et al., 2017; Ranjbar et al., 2022), and estimation of construction cost (Rafiei & Adeli, 2018b).

Although the performance of damage-detection algorithms based on machine learning has improved over the years, the detection accuracy still depends on the quality of the image to be analyzed. This is because, regardless of the sophistication of the detection algorithm, it is inevitable that some damage will be difficult to determine from a single image. For example, depending on the direction and distance of the camera when the image was captured, shadows and light reflections may make it difficult to determine the damage. Therefore, to accurately record damage, it is desirable to detect damage from images captured in various directions and distances and integrate the results.

Additionally, the results of current bridge inspections are often recorded in two-dimensional formats such as PDFs and tables. However, the method for recording information in the inspection report is not uniquely determined, even for the same damage. For example, whether to record extensive damage as a group of damages depends on the inspector's judgment. Additionally, it is inherently difficult to record the three-dimensional location of damage in a two-dimensional inspection report. Thus, it is difficult to compare the damage found in a previous inspection with the damage found in the reinspection. One solution to such problems is to standardize the data recording methods. For example, research is underway to introduce Construction Operation Building Information Exchange (COBie), a standard format for building information modeling (BIM)-based maintenance data transfer, to infrastructure maintenance (Shin et al., 2022). Although COBie makes it possible to link information, such as the degree of damage, to individual members, no format has been established to record the detailed location and size of the damage. Additionally, simply implementing COBie does not reduce labor because the information must be recorded manually. Recently, research has been conducted on bridge maintenance systems that can record the damage directly on a 3D bridge model; the detailed location and size of the damage can be recorded using these systems.

For example, Dang and Shim (2018) and Shim et al. (2017, 2019) have proposed a BIM-based framework for recording damage locations on bridges. Y. F. Liu et al. (2020) proposed a methodology for recording cracks detected from images captured by unmanned aerial vehicles (UAVs) in a 3D model of the pier. SeeBridge (Sacks et al., 2018), an infrastructure management system that records the condition of bridges using laser scans and photographs to build BIM models, has also been proposed. Using a 3D model to record damage in units of millimeters to centimeters rather than in units of members allows damage to be recorded more accurately and its progress to be determined more precisely than in conventional inspection reports. Additionally, the 3D model improves searchability by consolidating information and making it easier to understand the damage status of the bridge visually as a whole. In the future, it is expected that

the 3D model can be used to simulate the effects of damage on bridges. However, the current 3D model method of recording inspection results only records the damage found during inspection, so it is impossible to determine if there is unrecorded damage that may have been overlooked. In conventional inspection reports, photographs of damage are sometimes recorded, but undamaged areas are rarely photographed. This is because the amount of information to handle when photographs are taken comprehensively is huge, and it is difficult to determine which photograph was taken at which location. For undamaged areas, no more detailed information is recorded than the presence or absence of damage recorded at the member level; therefore, it is impossible to verify the correctness of the information later. This is because the purpose was to record only the damage information obtained by the inspection. This cannot be solved by simply replacing visual damage detection with image analysis because damage in areas that are not photographed may be overlooked. It is necessary to record the damage along with areas where there is no damage to solve this problem. Note that it is possible to make more sophisticated judgments based on the information about undamaged areas by recording areas where no damage exists. For example, if it is clear that damage is present on one side of an identically shaped member but not on the other, it can be assumed that the cause of the damage is not directly attributable to the member itself but to differences in peripheral factors such as the presence or absence of a drainage system.

This paper proposes a method to detect damage from multiple images using structure from motion (SfM) and deep learning and to integrate the results into a 3D model. The proposed method can record the damage more accurately than damage detection using a single image. This also makes it possible to determine areas that have not been properly photographed based on the 3D model, which enables us to objectively determine the scale of damage and degree of inspection omission.

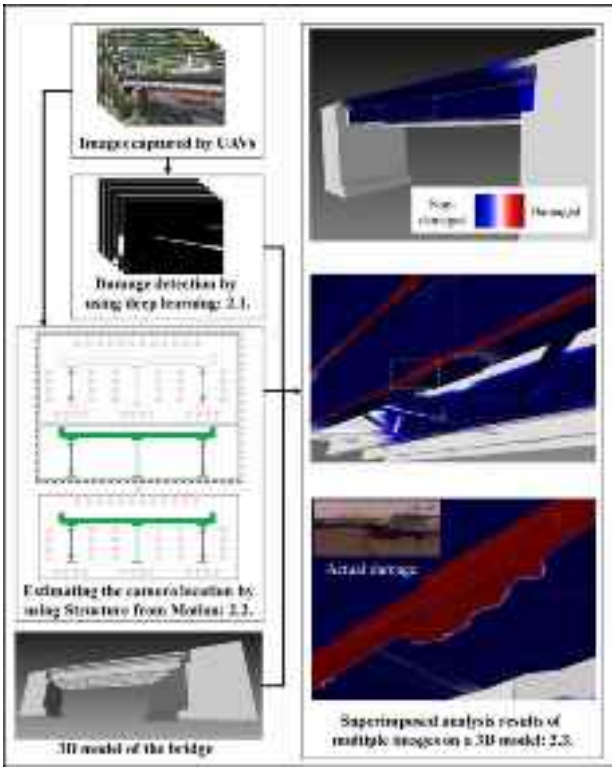


FIGURE 1 Diagram of the proposed damage detection and recording method. UAVs, unmanned aerial vehicles

2 | METHOD OF ESTIMATING THE 3D POSITION OF DAMAGES

2.1 | Overview of the proposed method

An overview of the proposed method is shown in Figure 1. First, members of the bridge were photographed using a UAV. Then, the damage was recorded in a 3D bridge model based on each method proposed in Sections 2.2–2.4.

In Section 2.2, a damage detection model using a Mask-R-CNN incorporating PointRend was proposed. In this study, a damage detection model was developed to detect corrosion as an example of damage. The application of a damage detection model incorporating PointRend improved the accuracy of damage detection.

Section 2.3 proposed a method for performing SfM for each member and integrating the coordinate system of the camera position using the least-squares method. This method identified the positional coordinates of images of a bridge taken from a short distance in the entire bridge system.

In Section 2.4, a method for reflecting the damage onto the 3D model was also proposed. This section showed that the 3D model could be converted into polygonal meshes, and the image analysis results could be projected onto each mesh by a projection transformation, thereby integrating the results of imaging and analysis from various angles.

2.2 | Damage detection by using deep learning

2.2.1 | Object segmentation

This study used an image segmentation algorithm based on deep learning to detect damage in images. Various image segmentation methods have been studied, including Graph Cuts (Blake et al., 2004) and Grow Cuts (Vezhnevets & Konouchine, 2005). With the recent development of

deep learning technology, the accuracy of image segmentation has significantly improved. Long et al. (2015) proposed a fully convolutional network (FCN) based on a convolutional neural network (CNN), which is an image analysis method using deep learning that enables image segmentation. CNNs are often used as classification models, wherein a fully connected layer is provided in the output layer. For image segmentation, a fully connected layer was not used in the output layer. When a fully connected layer is not used, the output is a feature map extracted from the input image via convolution, which shows the probability of the classification class for each pixel. Because the size of the feature map near the output layer of the CNN is much smaller than that of the input image, the feature map must be enlarged to superimpose it on the original image. However, because the information in the enlarged feature map is rougher than that in the original image, the FCN integrates the feature map output in the hidden layer into the output layer to produce a more detailed feature map. However, the FCN requires a large amount of memory to integrate the feature map output from the hidden layer into the output layer.

As a more accurate image segmentation method using deep learning, He et al. (2017) proposed Mask R-CNN based on Faster R-CNN (Ren et al., 2015). Faster R-CNN uses a CNN to detect object regions as a bounding box in the input image and classify the objects. Although object detection algorithms such as Faster R-CNN can generally detect damage, it is necessary to extract regions in pixels along the shape of the object to determine the damaged region accurately. Mask R-CNN extracts the region from the bounding box detected by Faster R-CNN, which enables highly accurate pixel-by-pixel image segmentation. In this study, we performed damage region extraction based on Mask R-CNN.

However, most CNN-based image segmentation methods do not generally have high accuracy in extracting regions at the boundaries between objects and backgrounds. This is because calculations are performed equally for all pixels when classifying each pixel in the image. Image segmentation is possible if the labels are determined around the boundary between the object and the background. However, if the calculations are performed equally for all pixels, more calculations are performed than necessary, even in low-frequency regions other than the boundary area, resulting in high computational costs. Subsequently, CNN-based image segmentation methods predict labels by dividing the image evenly on a low-resolution grid, thereby reducing the computational cost. To solve this problem, Kirillov et al. (2020) proposed PointRend, which can improve the accuracy of region extraction at the boundary between the object and background while reducing the computational cost.

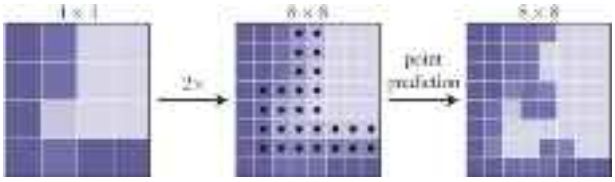


FIGURE 2 Example of upsampling by PointRend (Kirillov et al., 2020)

PointRend performs label prediction on a low-resolution grid, followed by upsampling using bilinear interpolation as shown in Figure 2. Then, for areas where the inference results are ambiguous (areas with black dots), the inference is repeated on a finer grid. Specifically, PointRend uses a simple multi-layer perceptron called Point Head, which iteratively performs class predictions only for each pixel with low confidence. PointRend is a generic module that can be flexibly integrated into existing region extraction models, such as the Mask R-CNN. Notably, in the case of the Mask R-CNN, PointRend can be incorporated into the model by replacing the mask head with a Point Head.

In this study, a damage-detection algorithm was developed using Mask R-CNN with integrated PointRend. Note that the entire framework proposed in this study is not dependent on the damage detection algorithm used and could be substituted if damage detection could be performed with improved accuracy. For example, although not used in this study, other algorithms, such as DeepLab v3+ (Chen et al., 2018), could be used to detect damage accurately.

2.2.2 | Segmentation model training

Various types of damage can occur to bridges. This study developed a deep learning model to detect corrosion-damaged regions as an example of damage. By extending the training dataset according to the detection target, the method shown in this study can be applied to other types of damage, such as exposed rebar and delamination of concrete. Figure 3 shows an example of the data used to train the model in this study. In this study, we prepared training data for 1966 images of corrosion damage as shown in Figure 3. These images were divided 80% (1572 images) into training data, 10% (197 images) into validation data, and 10% (197 images) into test data. Because the images were captured by various cameras, the number of pixels in each image was different, with the original images ranging from 315 to 1024 px in width and 238 to 768 px in height. Therefore, before training, the images were enlarged such that the width of all the images was 1024 px. Furthermore, the height of all images was set to

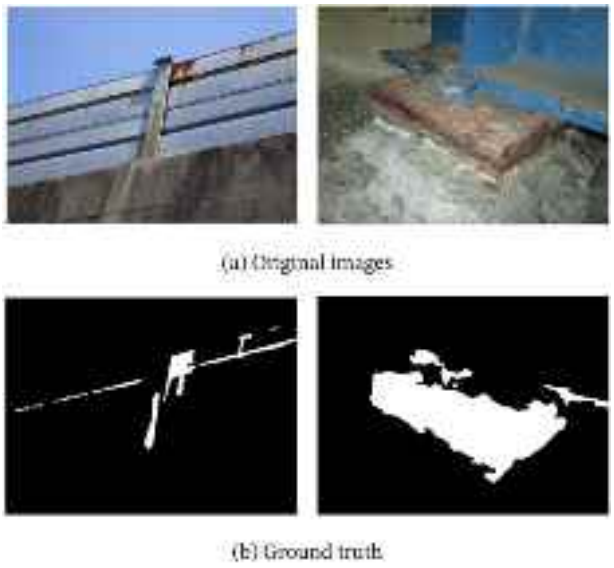


FIGURE 3 Training data of the corroded area

1024 px by applying appropriate padding to the top and bottom of the images. This study used an image segmentation algorithm based on deep learning to detect damage in images.

The Mask R-CNN model used in this study uses ResNet-50 as the backbone network, and the model was fine-tuned using the weights trained using the MS COCO dataset (Lin et al., 2014) as initial weights. The main hyperparameters of the Mask R-CNN were the same as those of Detectron2. During training, 4000 steps were performed with a batch size of two per step.

2.2.3 | Verification and evaluation

Using the trained model, corrosion damage was detected on 197 pieces of the test dataset, and an accuracy evaluation was performed. When detecting corrosion damage from the test dataset, the test data image was resized to 1024×1024 px for detection. Figure 4 shows an example of the detection results. The area shown in red in Figure 4 is where both the detection result and ground truth indicate corrosion. The blue area in Figure 4 is what is judged as corroded, even though there is no corrosion in the ground truth. The green area in Figure 4 is the area judged as not corroded, even though there is corrosion in the ground truth. As shown in Figure 4, most of the corrosion damage areas were detected accurately from the image using the developed model.

The accuracy of the corrosion damage detection model was evaluated based on the test data. The calculated results for each index were: accuracy = 0.94, precision = 0.80, sensitivity = 0.67, specificity = 0.97, and F-measure = 0.73. For reference, when a Mask R-CNN model without PointRend was used for training and detection under the same conditions, the F-measure was 0.62. This shows that integrating PointRend into the model improves the accuracy of corrosion detection.

2.3 | Estimating camera's location by using SfM

By using a damage detection model, such as the one developed in the previous section, damage can be automatically detected from the images. However, none of these images indicated which part of the bridge was photographed. Therefore, in this study, the camera coordinates were identified using SfM to calculate the location of the camera that captured each image, which was shot from around the bridge.

SfM is a method for estimating the position of the camera that captures each image, that is, the direction of rotation and translation of the camera, by determining the common tie-points of the objects based on the images captured from multiple viewpoints and using epipolar geometry (Zisserman & Hartley, 2003). Based on the camera position information identified by SfM, the 3D shape of structures, such as bridges, can be recovered by stereo matching (Khaloo et al., 2018; Morgenthal et al., 2019).

However, generally, SfM can be used to estimate the relative positioning of cameras but not the absolute distances or coordinates. Additionally, when using deep learning to detect damage, it is necessary to take images close to the bridge members because the pixel size of the camera mounted on the UAV may not be high. However, if the entire bridge is photographed from a short distance and SfM is performed simultaneously, it becomes difficult to perform SfM practically. This is because the number of images increases, increasing the computation time and requiring a large amount of computer memory. In this study, the positions of each camera taken from a short distance for each member were estimated, and the coordinates of each camera were integrated into a coordinate system for the entire bridge. Finally, the integrated positions of each camera were reflected in the coordinate space of the existing 3D model.

Figure 5 shows an overview of the integration method of the camera position and coordinate system of the bridge model shown in this study. First, the entire bridge was photographed from a distance of several tens of meters, and the global SfM of the entire bridge was performed. In parallel, the images were taken from a distance close to the bridge, and the local SfM of each member was performed. Some of the photos used for the SfM of the entire bridge were used when performing local SfM. In this manner, the coordinate information of the common camera can be

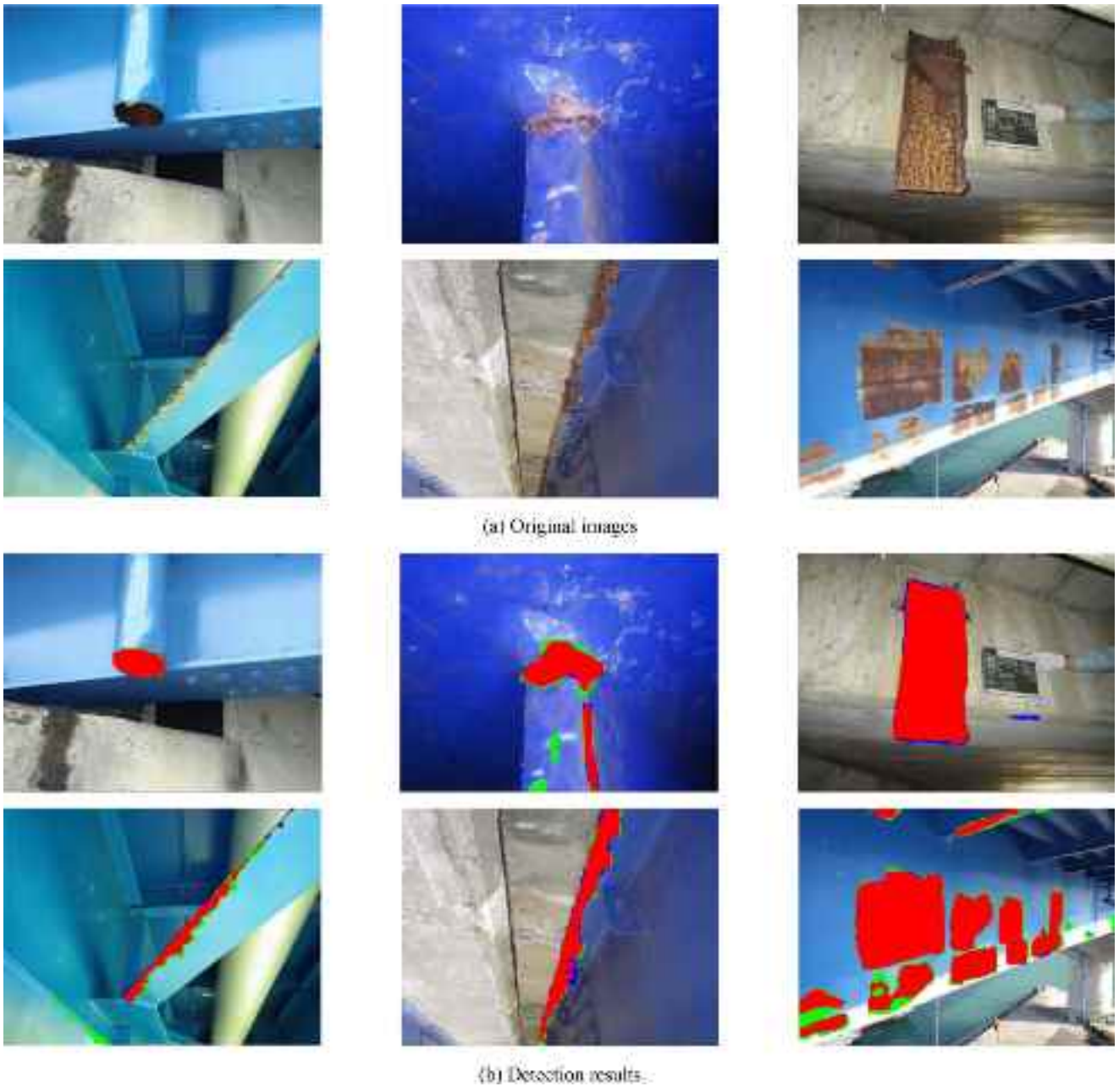


FIGURE 4 Example of detection results

treated as a reference point, and the camera positions can be integrated. Then, the camera coordinates integrated in this manner are converted collectively to the coordinates of the 3D model of the bridge with the real scale that has been constructed in advance. For this purpose, several reference point coordinates were obtained from the surface of the bridge model constructed using the SfM and the 3D model using the real scale. Subsequently, based on the coordinate information of these reference points, the position information of each camera is reflected in the bridge 3D model. Therefore, by performing SfM separately for the entire bridge and individual components, it is possible to take photographs suitable for damage detection by deep learning while reducing computational cost and obtaining the position coordinates of these cameras in the entire bridge system.

Note that if the space under the girders is narrow, the difference in distance between photos taken from long and close ranges must be reduced. In such cases, the number of photos taken from a long range will increase slightly, but the calculation cost can still be reduced because the group of photos from a long range does not include the photos between the girders.

In this study, the least-squares method was used to integrate the camera positions. First, the integration of the position coordinates of the global and local

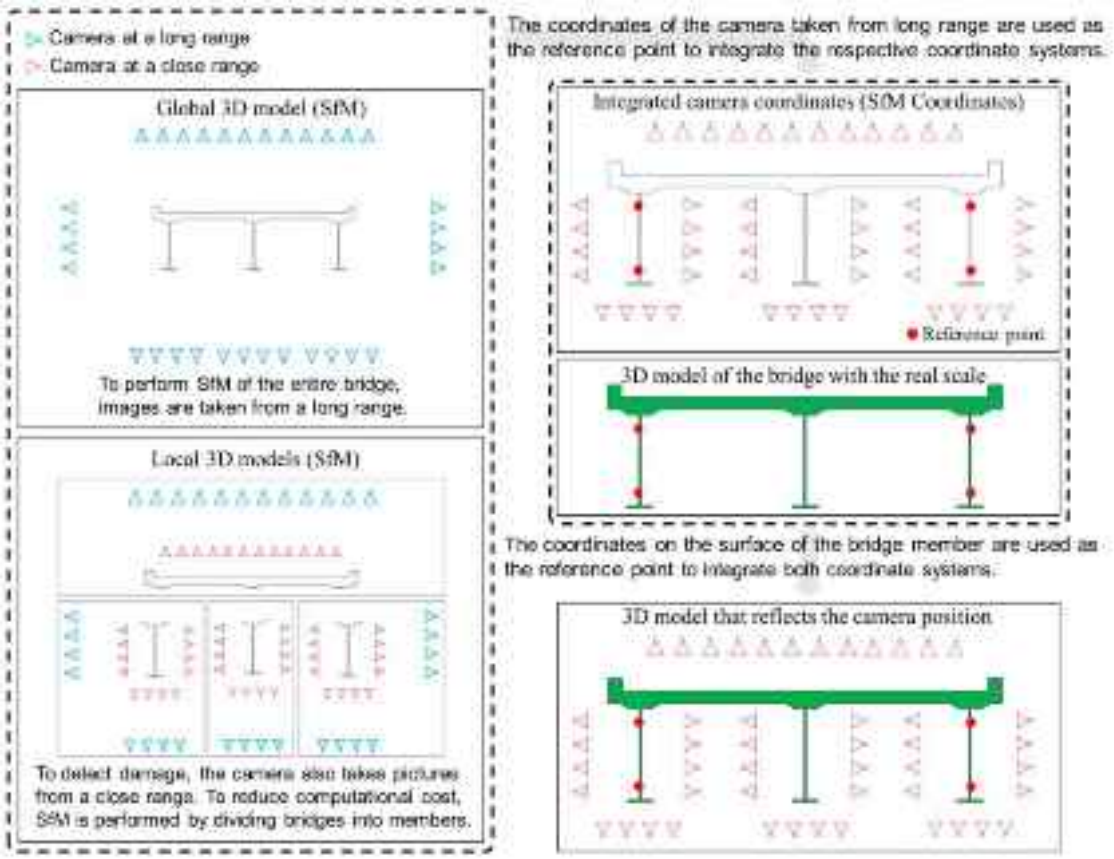


FIGURE 5 Overview of camera position integration. SfM, structure from motion

3D models is explained. The center-of-gravity coordinates G of the reference points (common camera coordinates) in both models were calculated. Then, the coordinate system of the local 3D model was shifted in parallel such that the calculated center-of-gravity coordinates G coincided. Next, the local 3D model was rotated, with the center-of-gravity coordinate G as the origin. The rotation matrices around the X-, Y-, and Z-axes for the local 3D model are expressed by Equations (1)–(3), respectively. Then, as shown in Equation (4), the parameters $\hat{\phi}$, $\hat{\theta}$, and $\hat{\psi}$ for rotating the local model were estimated by setting ϕ, θ, and ψ to minimize the residual sum of squares of the coordinate positions of the camera common to both models. The normalized vectors of the camera coordinates common to the local and global 3D models were $\tilde{\mathbf{u}}$ and $\tilde{\mathbf{v}}$, respectively, and the number of reference points (the number of cameras common to both models) was n.

$$R_x(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix} \tag{1}$$

$$R_y(\theta) = \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} \tag{2}$$

$$R_z(\psi) = \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3}$$

$$\left(\hat{\phi},\hat{\theta},\hat{\psi}\right) = \underset{\phi,\theta,\psi}{arg\,min} \sum_{i=1}^{n} \left(\tilde{\mathbf{v}}_i - R_z(\psi) R_y(\theta)\ R_x(\phi)\,\tilde{\mathbf{u}}_i\right)^2 \tag{4}$$

After aligning the direction of rotation, as shown in Equation (5), the scale of the local 3D model was adjusted to match the scale of the two models.

$$\mathbf{w}' = \mathbf{G} + (\mathbf{w} - \mathbf{G}) \sum_{i=1}^{n} \frac{(\mathbf{v}_i - \mathbf{G})}{(\mathbf{u}_i - \mathbf{G})} \tag{5}$$

First, the distance from the center-of-gravity coordinate G to each reference point and the corresponding average values were calculated. The scale of each camera in the local 3D model was adjusted with the center-of-gravity coordinate G as the origin such that these values were the same for both models. $\mathbf{w}$ was the vector of all the camera coordinates of the local 3D model, and $\mathbf{u}$ and $\mathbf{v}$ were the vectors of the camera coordinates common to the local and global 3D models, respectively. $\mathbf{G}$ is the position vector of the center of gravity coordinate G.

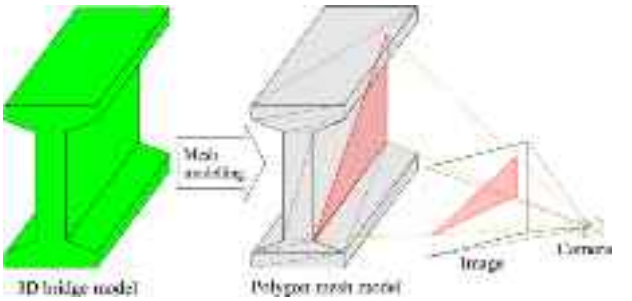


FIGURE 6 Projection from camera coordinates to the 3D bridge model

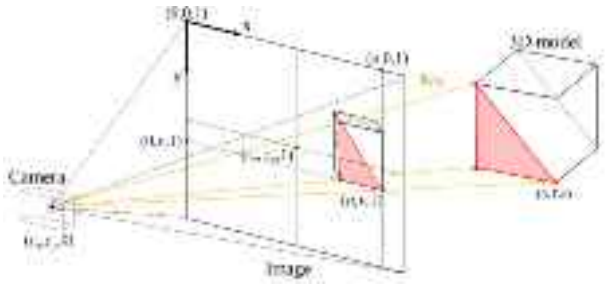


FIGURE 7 Relationship between camera, image, and 3D model

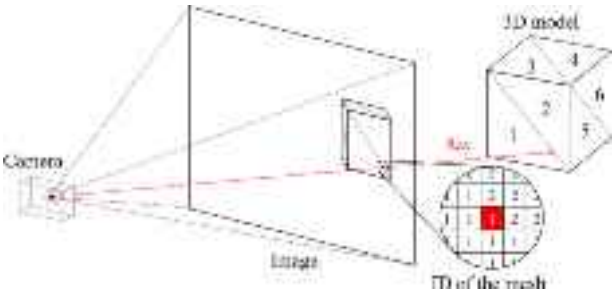


FIGURE 8 ID of the mesh stored in each pixel

Therefore, the camera positions in the local 3D model can be integrated into the coordinate system of the entire bridge. Then, when the integrated camera position is reflected in the 3D model of the bridge, the coordinate transformation is performed in the same way, using the reference point obtained from the surface of the member.

Note that in this study, the selection of common cameras was done manually but could be automated using simultaneous localization and mapping technology on UAVs. Future research is expected on these issues. However, in practice, the selection of common images can be conducted manually. This is because, when photographing entire bridges with UAVs, the image capture flow is usually divided into sections by member, lattice, and capture distance at the stage of planning the capture, and the images are captured in that order. If photography is not conducted in this way, it is impossible to know during photography which areas have been photographed. Therefore, after filming, the near, far, and common images are sorted almost completely and can be used as they are.

2.4 | Integration of analysis results of multiple images on a 3D model

The methods described in the previous sections provided damage detection results from each image and information regarding where the camera that captured each image is located in the 3D model space. Based on this information, the image analysis results were superimposed on a 3D model of the bridge. In this study, each surface of the 3D model was divided into a triangular mesh in advance to reflect the damage to any shape of a member with a single algorithm as shown in Figure 6. Then, the damage-detected results are reflected in each mesh that constitutes the 3D model. Note that in this study, 3D models are assumed to be pre-built models (e.g., based on BIM-based 3D models). This is because pre-built models are more reliable in terms of geometry than the 3D models generated by SfM. In this study, SfM is only used to estimate the camera coordinates.

The relationship between the mesh, image, and camera constituting the 3D model is shown in Figure 7. When the coordinate position of the camera is represented by $(c_x, c_y, 0)$, the upper-left coordinate of the image is (0,0,1), and the optical axis center coordinate of the image is $(c_x, c_y, 1)$. As shown in Figure 8, when a point (X, Y, Z) on the 3D model is located at $(u, v, 1)$ in the image, these relationships are expressed by Equation (6). The distortion caused by the lens was assumed to be corrected.

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} [\mathbf{R}|\mathbf{t}] \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (6)$$

where f_x and f_y are the focal lengths of the cameras, and $[\mathbf{R}|\mathbf{t}]$ is the external parameter matrix representing the translation and rotation of the camera, respectively. The external parameter matrix for each camera was obtained using the SfM. Equation (6) can be used to determine the location of each coordinate position in the 3D model in the image.

In this study, the results of the image analysis were reflected in each mesh constituting the 3D model. Based

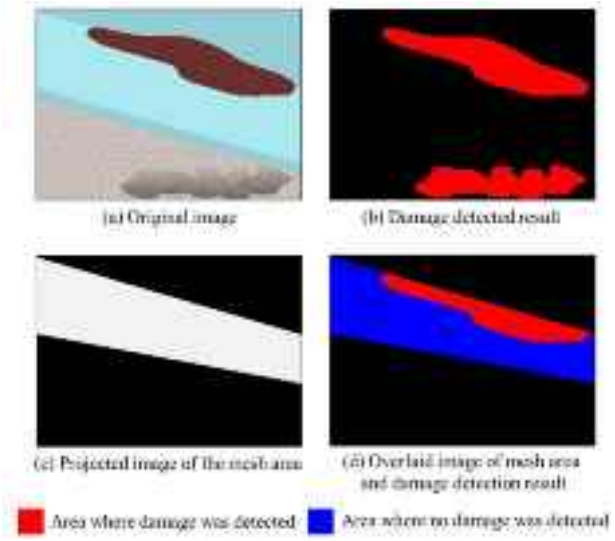


FIGURE 9 Superimposing model mesh and image analysis results

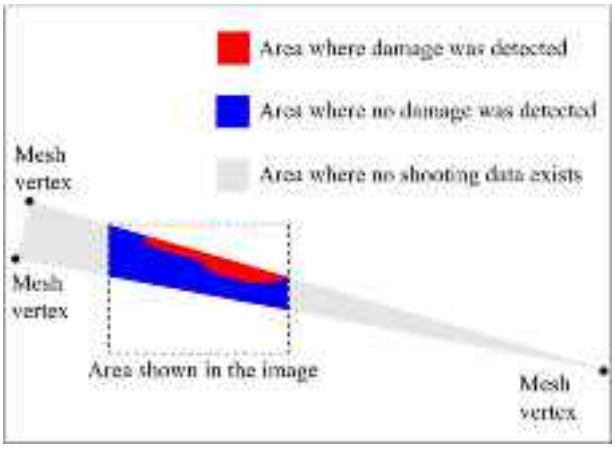


FIGURE 10 Example of mesh vertex positions on an image

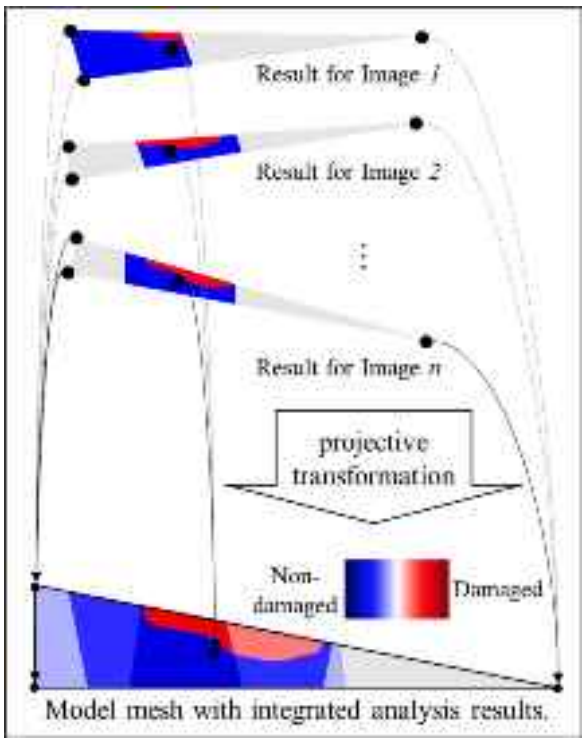


FIGURE 11 Projection of each analysis result onto the model mesh

on the camera coordinates, a vector was calculated from each pixel in the image to the model, and the mesh of the model where the vector intersects was calculated as shown in Figure 8. Each mesh of the model was assigned a unique ID in advance, and the ID of the intersecting mesh was stored for each pixel of the image. As each vector may intersect the mesh of the model multiple times, the distance from the camera coordinates to the intersection point with the mesh was calculated, and the ID of the mesh with the shortest distance was stored in each pixel. Thus, only the meshes captured in the image were stored in each pixel, and the ID of the mesh of the 3D model was associated with each pixel of every image.

By storing the ID of the mesh in each pixel, it was possible to map the mesh of interest to the image analysis results as shown in Figure 9. Note that the area shown in light blue in Figure 9a is the structural member, and the area shown in brown is the damaged area. First, suppose that image analysis is performed based on the captured image, as shown in Figure 9a, and the resultant image for damage detection is shown in Figure 9b. In parallel, an image representing only an arbitrary mesh of interest is created as shown in Figure 9c. By superimposing Figure 9b,c, the image analysis results for the mesh in focus were obtained as shown in Figure 9d. Note that, as shown in the lower part of Figure 9a,b, if false positives occur in areas other than the structure, these false positives will not be a problem because they will be removed in principle when the mesh is superimposed.

The mesh vertices projected onto the image did not necessarily fall within the image range as shown in Figure 10. In other words, information regarding the presence or absence of damage could not be obtained for areas outside the image, and only the portion of the mesh that appeared in the image was updated. Similarly, if there were other meshes between the mesh of interest and the camera, no information regarding the presence or absence of damage in that region could be obtained.

Thus, the image analysis results could be mapped to each mesh of the model. Because damage detection and mesh mapping are based on multiple images, it is necessary to integrate the results of multiple image analyses. Therefore, as shown in Figure 11, the mesh regions

obtained from each image were integrated by projective transformation into the mesh of the model. The transformation equation for a point p on the 2D image plane to a mesh vertex P' is expressed in Equation (7).

$$P' = \begin{bmatrix} X' \\ Y' \\ W' \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = Hp \tag{7}$$

H is the homography matrix, a necessary parameter for the projective transformation, which is determined based on four points on the image plane. Therefore, in addition to the vertices of the triangular mesh, the coordinates of the geometric center of the mesh were projected onto the image, and the projection transformation was performed by calculating the homography matrix based on these four points. By performing these operations on each image, the results of multiple image analyses could be integrated into the 3D model, thereby reducing the variability of the detection results for each image. Additionally, areas not captured in the images were recorded as having no information obtained, making it possible to record the damage and determine from the 3D model whether the inspection was conducted comprehensively.

Although there are various possible methods for calculating the integration of the results of each image analysis, in this case study, the analysis results were superimposed according to Equation (8).

$$I(i,j) = min\left\{max\left\{0,\ \frac{1}{2} + \frac{1}{k}\sum_{i=1}^{n}\left(d_n(i,j) - \frac{1}{2}\right)\right\},1\right\} \tag{8}$$

I is the pixel value on the surface of the 3D model obtained by superimposing the results of each analysis and is a value between 0 and 1. The closer I is to 1, the more likely it is to be a damaged area, and the closer it is to 0, the more likely it is to be a non-damaged area. d is the value of each pixel obtained from the analysis using the deep learning model. A pixel with $d = 0$ indicates no damage was detected, whereas a pixel with $d = 1$ indicates that damage was detected. n is the number of analyzed images, k is a constant, and a large value of k makes it difficult for the pixel values to converge to 0 or 1.

This formula resulted in a pixel value near 0.5 in areas where the number of shots was small, such as one or two shots. Then, areas where no pictures are taken and areas where the number of pictures taken is small can be treated as areas where the judgment result is not fixed. Thus, even if an erroneous analysis result is obtained accidentally at a location where the number of photographed images is small, the effect can be reduced.

FIGURE 12 Steel girder bridge used in this study

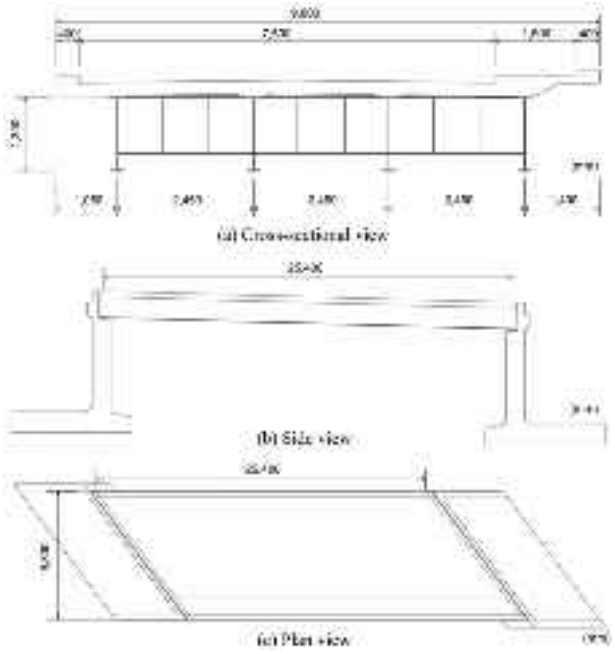


FIGURE 13 Drawing of the bridge

3 | APPLICATION TO THE REAL BRIDGE

This section presents a case study in which the proposed method is applied to an actual bridge.

3.1 | Field test investigations

The bridge used in this field test was a steel girder bridge, as shown in Figure 12, managed by the Japanese MLIT. As shown in Figure 13, the span length and total width of this bridge were 25.4 and 9.8 m, respectively. The bridge was

FIGURE 14 Photographing bridges using an unmanned aerial vehicle

FIGURE 15 Inspection using bridge inspection vehicles

constructed in 1970 in a mountainous area of Yamanashi Prefecture, Japan. The bridge has several damaged parts, particularly the main girders, which are heavily corroded. In this study, corrosion damage was detected using the deep learning model trained in Section 2.2, and the results were reflected in the 3D model built using 3D CAD.

In this study, Skydio 2, a small UAV developed by Skydio, was used to capture photographs of this bridge as shown in Figure 14. The focal length of the camera mounted on the UAV was 4 mm, the 35 mm focal length was 21 mm, and the pixel count was approximately 12.3 megapixels (4056 × 3040 px). This bridge was inspected using inspection vehicles as shown in Figure 15. If the UAV-based inspection method shown in this study is realized, it will greatly contribute to improving the efficiency of maintenance.

In this study, 2316 images of the bridge were captured from various angles, and the SfM process was performed using Agisoft Metashape. Note that as long as the camera position can be accurately estimated, the framework proposed in this study is independent of the type of UAV and SfM software used. The camera coordinates estimated using SfM are shown in Figure 16.

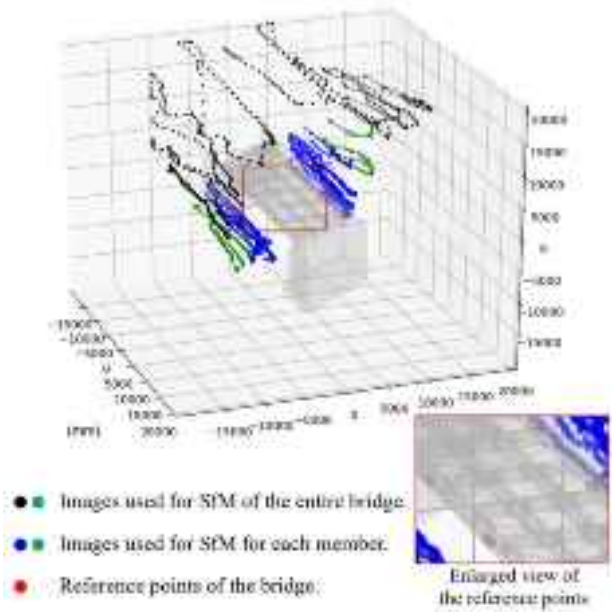


FIGURE 16 Coordinate position of each camera estimated using structure from motion (SfM)

The global SfM of the entire bridge was determined using 870 images obtained from the positions indicated by the black and green markers in Figure 16. The local SfM of the upstream and downstream exterior girders was performed using 1691 images (taken from the locations indicated by the blue and green markers in Figure 16). For the SfM of the upstream exterior girder, 954 images were used; for the SfM of the downstream exterior girder, 737 images were used.

Because the SfM of the entire bridge and that of each exterior girder were performed separately, the camera coordinates were estimated using completely different coordinate systems. Therefore, as explained in Section 2.3, a coordinate transformation was performed using the least-squares method such that the camera coordinates common to both (points indicated by green markers in Figure 16) would match.

Reference points (indicated by red markers in Figure 16) were set on each exterior girder of the 3D model to match the camera coordinates estimated in this manner with the coordinate positions of the 3D model of the bridge. Then, a coordinate transformation was performed using the least-squares method such that the reference points on the model constructed using SfM matched those on the 3D model.

Then, corrosion damage was detected from the captured images using the deep learning model trained

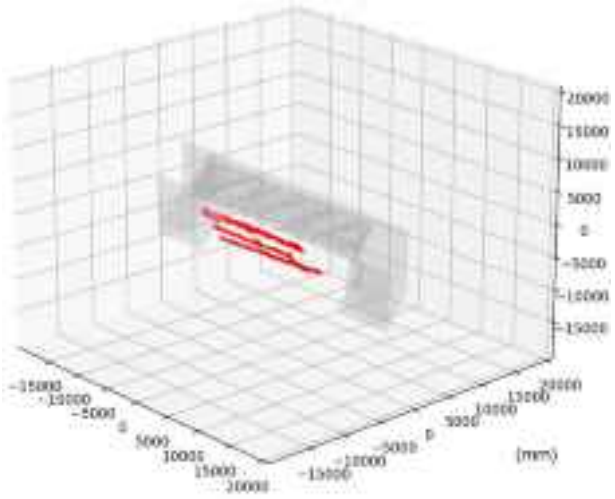


FIGURE 17 Location coordinates of cameras that captured the images used for damage detection

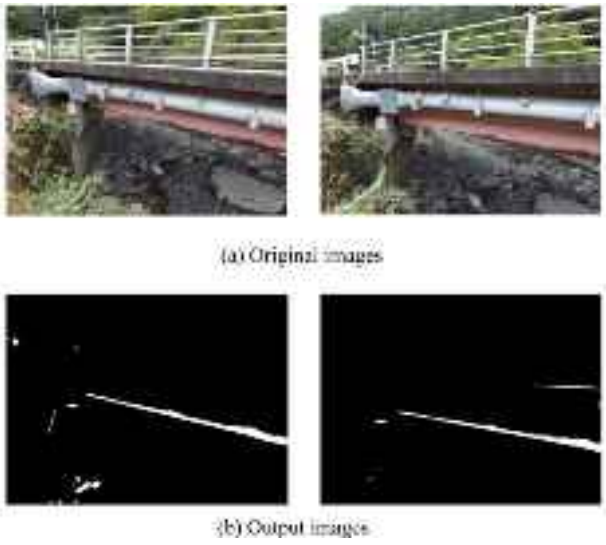


FIGURE 19 Example of output images from the deep learning model

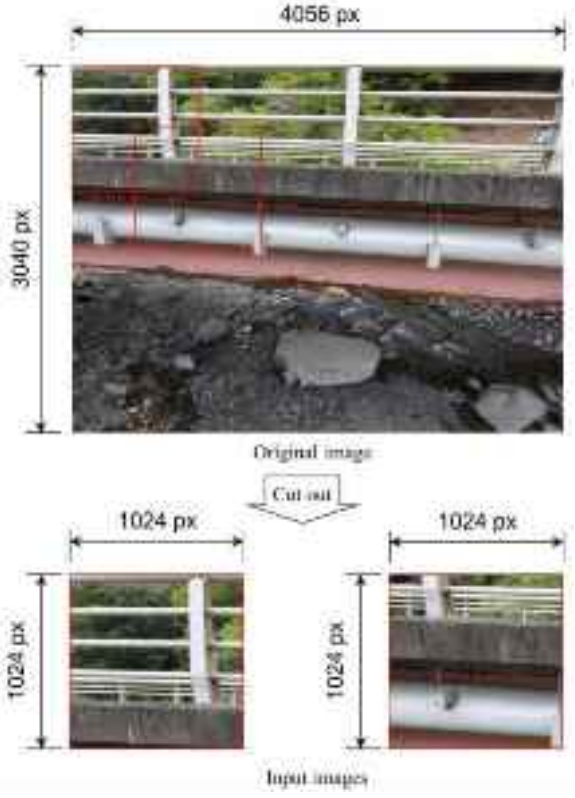


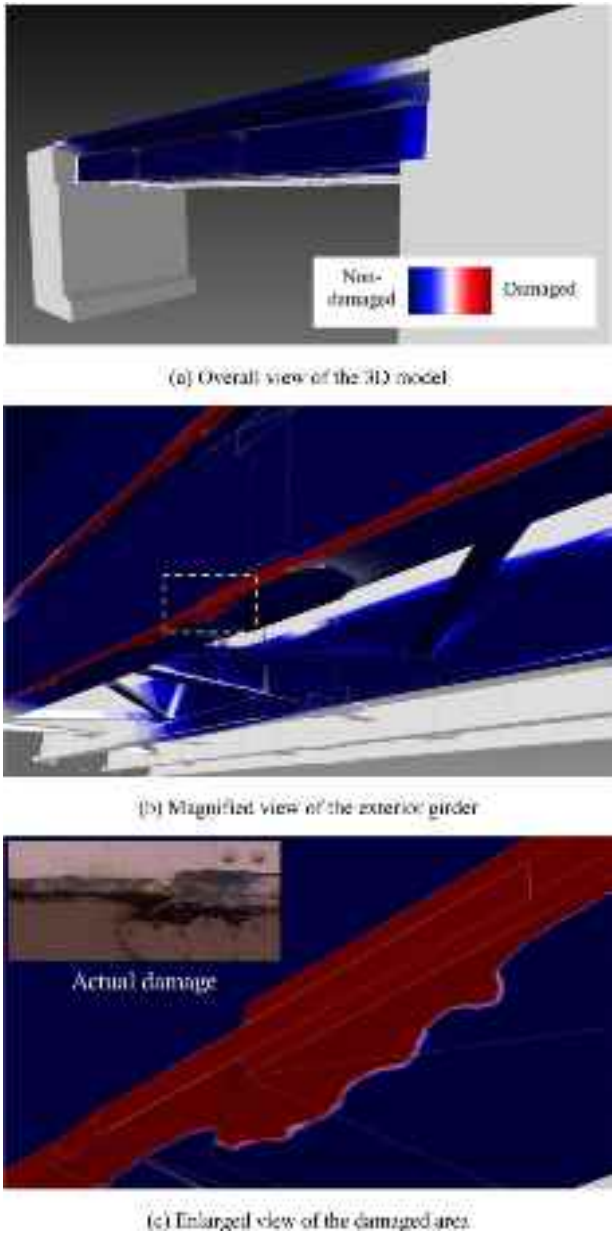
FIGURE 18 Example of small image cut-outs used for analysis

in Section 2.2. For simplicity, this case study used 233 images taken from the downstream exterior girder, where significant corrosion had occurred, to detect corrosion damage. The coordinates of the images used in the analysis are shown in Figure 17.

The image was cropped to a small area (1024 × 1024 px), as shown in the red box in Figure 18, to detect corrosion damage and was used as the input image for the deep learning model. This is because the size of the original image is larger than that of the input layer of the deep learning model. If the original image is resized and input into the model, information on the detailed damage is lost during the resizing. Therefore, detection accuracy can be ensured by cropping instead of resizing. The small cropped areas are overlapped by half to make the detection accuracy less dependent on the position of the cropped image.

Figure 19 shows an example of the analysis results obtained using the deep learning model. As shown in Figure 19, the model could correctly detect most areas where corrosion occurred.

Finally, the image analysis results are reflected in the 3D model based on the method described in Section 2.4. In this case study, k in Equation (8) was set to 10. The resulting 3D model is shown in Figure 20. According to the color map shown in Figure 20, the pixel values on the 3D model's surface were converted for ease of viewing. The blue region on the 3D model's surface in Figure 20 is where the probability of damage was judged to be low in most images. The red area on the 3D model's surface is where the probability of damage was judged to be high in most images. However, the uncolored areas on the 3D model's surface are where the analysis result has not been determined, either because no images were analyzed or because the analysis results differed. For example, areas hidden by other members, such as the backside of the lower lateral bracing, the top of the inner girder, or the upstream exterior girders that were not used in the deep learning analysis, were not colored because sufficient analysis results were not obtained as shown in Figure 20b. Figure 20c shows the area enclosed by the yellow frame in Figure 20b. As shown in Figure 20c,



(a) Overall view of the 3D model

(b) Magnified view of the exterior girder

(c) Enlarged view of the damaged area

FIGURE 20 3D model that integrates and reflects multiple damage detection results

there is an area of ambiguous color on the border between the damaged and undamaged areas.

This is due to slight deviations in the projected positions on the model caused by damage-detection errors in each image and estimation errors in the camera coordinates. However, as shown in Figure 20c, this region was small in this case study and negligible for practical purposes. As described above, the method proposed in this study allowed us to integrate and reflect on the analysis results of the captured images of the bridge in the 3D model.

For reference, the computation time taken to reflect the analysis results in the 3D model in this case study was 342 min. A Core i7-1065G7 CPU was used for the calculations. Note that this computation time varies greatly depending on the number of images and the target to be reflected, and parallel computation using GPUs may enable even faster speeds.

3.2 | Results and discussion

As shown in Section 3.1, by constructing a 3D model reflecting the analysis results, the location and shape of the damage can be determined. Moreover, this section discusses the results of the proposed method in comparison with the methodologies proposed in other studies thus far.

Table 1 shows a comparison of the results of this study with those reported by others. This study developed a machine learning model for detecting corrosion. Most of the studies are aimed at detecting cracks, but very few studies have targeted corrosion. Shi et al. (2021) detected corrosion using VGG-Unet. Although an exact comparison of this study with the current study is difficult due to the use of different datasets, the corrosion detection model developed in this study could detect corrosion with high accuracy. It should be noted that in the proposed method, the integration of the results of several analyses is more important than the accuracy of each analysis. Most currently proposed damage detection models have high accuracy for ordinary images. The major factor in the low average accuracy is the lack of accurate detection of rare, poor-quality images. The proposed method can minimize the effect of these rare false positives by integrating the analysis results from multiple images.

In addition, a method has been proposed to record the damage in a 3D model of the bridge. However, existing proposed methods only record the analysis results of the inspected areas. In such a case, it is impossible to determine whether the areas where damage is not recorded have been inspected or whether they have been determined to be truly undamaged. The ability to distinguish between undamaged and non-inspected areas is an important novelty of the proposed method as shown in Figure 20. It was possible to make sophisticated decisions based on the information that no damage existed. For example, if the damage is present on one side of an identically shaped member but not on the other, it can be assumed that the cause of the damage is not directly attributable to the member itself but to the differences in the peripheral factors such as a drainage system.

In addition, existing methods mainly perform SfM at the member level, for example, SfM for piers only. When damage is detected using machine learning, the images used must be taken from a close range, which is difficult to apply to an entire bridge for SfM. This study overcomes this prob-

TABLE 1 Comparison of results with other proposals reviewed

		Achievements of each method		
	Detectable damage	**Reflection of damage on a 3D model**	**Distinction between undamaged areas and non-inspected areas**	**Detecting invisible damage (e.g., concrete floating and delamination)**
Proposed method	Corrosion (accuracy = 0.94)	✓	✓	
Shi et al. (2021)	Corrosion (accuracy = 0.82)			
Dang and Shim (2018)	Crack	✓		
Shim et al. (2017)	Crack	✓		
Y. F. Liu et al. (2020)	Crack	✓		
Sirca and Adeli (2018)	Floating and delamination			✓
Chun and Hayashi (2021)	Floating and delamination			✓

lem by integrating the SfM results at the member level into the entire bridge system.

Note that, as visible images were used in this study, damage such as concrete delamination, which cannot be detected from the visible images, could not be reflected in the 3D model. However, some studies have successfully detected damage, such as concrete delamination, using infrared images (Chun & Hayashi, 2021; Sirca & Adeli, 2018). The findings of these studies can be combined with the present proposal to expand the scope of the application of the proposed method.

The proposed method is more advanced as it can replace conventional visual bridge inspections and contribute toward improving the efficiency of bridge inspections. Note that in current bridge inspections, at least in the United Kingdom and Japan, bridge inspectors have to report details such as "the location, severity, extent, and type of all defects in the structure." The applicability of the damage detection model proposed in this study is limited to corrosion; it is therefore impossible to record all the damages using this model. However, other types of damage can be detected by building similar detection models. In addition, the proposed method cannot detect the severity of the damage; this can be addressed by building a suitable detection model in the future.

4 | CONCLUSION

This paper proposes a method to detect damage from images of bridges taken from various directions using SfM and deep learning to integrate the results of image analysis into a 3D model. The proposed method can reduce the variation in detection results among images and evaluate the scale of damage and extent of inspection omissions.

The proposed method is based on UAV imaging and damage detection using deep learning, which contributes to improving the efficiency of bridge inspections.

One problem with damage detection using deep learning is that it is inevitable that areas of an image will be difficult to determine from a single image alone. This study proposed a methodology to detect damage from images taken from various angles and distances and integrate these results and record them in a 3D model. The proposed method was applied to an actual bridge, and it was shown that the actual damaged areas could be recorded in the 3D model.

Additionally, because the conventional inspection method only records noticeable damage, it is impossible to determine whether damage exists or whether the inspection is inadequate for areas where no record of damage exists in the inspection report. The proposed method comprehensively recorded even areas where no damage was detected, allowing for the determination of the adequacy of inspections.

Several individual problems were also addressed. In this study, a deep learning model for damage detection was developed using corrosion as an example of the damage to be handled. To detect damage accurately, the damage-detection algorithm in this study was developed using Mask R-CNN with integrated PointRend. The developed damage detection model had accuracy = 0.94, precision = 0.80, sensitivity = 0.67, specificity = 0.97, and F-measure = 0.73. When the Mask R-CNN model without PointRend was used, F-measure = 0.62. This shows

that integrating PointRend into the model improved the accuracy of corrosion detection.

Furthermore, the resolution of cameras mounted on UAVs is not necessarily high; therefore, to accurately detect damage, it is necessary to capture images from close to the bridge members. Therefore, this study proposed a method for performing SfM for each member and integrating the coordinate system of the camera position using the least-squares method. This method was shown to identify the positional coordinates of images of a bridge taken from a short distance in the entire bridge system.

A format for reflecting the damage on the 3D model was also proposed. It was shown that to reflect damage in the 3D model, the 3D model could be converted into polygonal meshes, and the results of image analysis could be projected onto each mesh by a projection transformation, thereby integrating the results of imaging and analysis from various angles.

If the results obtained from the proposed method are linked as attributes of a BIM-based model, they can be used to evaluate the overall health of the bridge and estimate the cost of repair. Based on the results of this study, advances in bridge maintenance using 3D models are expected.

ACKNOWLEDGMENTS

The Kofu River and National Highway Office, the Kanto Regional Development Bureau, and the Ministry of Land, Infrastructure, Transport, and Tourism of Japan allowed us to take measurements of the bridges under their management. We would like to express our sincere gratitude to these organizations for making this research possible. This research was funded by the Japan Science and Technology Agency (Moonshot Research and Development) under grant number (JPMJMS2032) and the JSPS Grant-in-Aid for Scientific Research(B) (Grant Number 21H01417).

REFERENCES

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

Amezquita-Sanchez, J. P., & Adeli, H. (2019). Nonlinear measurements for feature extraction in structural health monitoring. *Scientia Iranica-A*, *26*(6), 3051–3059.

Amezquita-Sanchez, J. P., Park, H. S., & Adeli, H. (2017). A novel methodology for modal parameters identification of large smart structures using MUSIC, empirical wavelet transform, and Hilbert transform. *Engineering Structures*, *147*, 148–159. https://doi.org/10.1016/j.engstruct.2017.05.054

Amezquita-Sanchez, J. P., Valtierra-Rodriguez, M., & Adeli, H. (2018). Wireless smart sensors for monitoring the health condition of civil infrastructure. *Scientia Iranica-A*, *26*(6), 2913–2925.

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

Blake, A., Rother, C., Brown, M., Perez, P., & Torr, P. (2004). Interactive image segmentation using an adaptive GMMRF model. *European Conference on Computer Vision*, Prague, Czech Republic pp. 428–441. https://doi.org/10.1007/978-3-540-24670-1_33

Chen, L. C., Zhu, Y., Papandreou, G., Schroff, F., & Adam, H. (2018). Encoder-decoder with atrous separable convolution for semantic image segmentation. *Proceedings of the European Conference on Computer Vision (ECCV)*, Munich, Germany pp. 801–818. https://doi.org/10.1007/978-3-030-01234-2_49

Chun, P. J., & Hayashi, S. (2021). Development of a concrete floating and delamination detection system using infrared thermography. *IEEE/ASME Transactions on Mechatronics*, *26*(6), 2835–2844. https://doi.org/10.1109/TMECH.2021.3106867

Chun, P. J., Izumi, S., & Yamane, T. (2021). Automatic detection method of cracks from concrete surface imagery using two-step light gradient boosting machine. *Computer-Aided Civil and Infrastructure Engineering*, *36*(1), 61–72.

Chun, P. J., Yamane, T., Izumi, S., & Kameda, T. (2019). Evaluation of tensile performance of steel members by analysis of corroded steel surface using deep learning. *Metals*, *9*(12), 1259. https://doi.org/10.3390/met9121259

Chun, P. J., Yamane, T., Izumi, S., & Kuramoto, N. (2020). Development of a machine learning-based damage identification method using multi-point simultaneous acceleration measurement results. *Sensors*, *20*(10), 2780. https://doi.org/10.3390/s20102780

Chun, P. J., Yamane, T., & Maemura, Y. (2021). A deep learning based image captioning method to automatically generate comprehensive explanations of bridge damage. *Computer-Aided Civil and Infrastructure Engineering*, *37*(11), 1387–1401. https://doi.org/10.1111/mice.12793

Dang, N. S., & Shim, C. S. (2018). BIM authoring for an image-based bridge maintenance system of existing cable-supported bridges. *IOP Conference Series: Earth and Environmental Science*, *143*(1), 012032. https://doi.org/10.1088/1755-1315/143/1/012032

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

Fondevik, S. K., Stahl, A., Transeth, A. A., & Knudsen, O. Ø. (2020). Image segmentation of corrosion damages in industrial inspections. *2020 IEEE 32nd International Conference on Tools with Artificial Intelligence (ICTAI)*, Baltimore, MD pp. 787–792. https://doi.org/10.1109/ICTAI50040.2020.00125

He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask R-CNN. *Proceedings of the IEEE International Conference on Computer Vision*, Venice, Italy pp. 2961–2969. https://doi.org/10.1109/ICCV.2017.322

Horgan, R. (2019). *Fatal Taiwan bridge collapse is latest example of maintenance failings*. https://www.newcivilengineer.com/latest/fatal-taiwan-bridge-collapse-is-latest-example-of-maintenance-failings-07-10-2019/

Jang, K., An, Y. K., Kim, B., & Cho, S. (2021). Automated crack evaluation of a high-rise bridge pier using a ring-type climbing robot. *Computer-Aided Civil and Infrastructure Engineering*, *36*(1), 14–29. https://doi.org/10.1111/mice.12550

Karina, C. N. N., Chun, P. J., & Okubo, K. (2017). Tensile strength prediction of corroded steel plates by using a machine learning approach. *Steel and Composite Structures*, *24*, 635–641.

Kawahara, S., Shirato, M., Kajifusa, N., & Kutsukake, T. (2014). Investigation of the tunnel ceiling collapse in the central expressway in Japan. *Transportation Research Board 93rd Annual Meeting*, Washington, DC.

Khaloo, A., Lattanzi, D., Cunningham, K., Dell'Andrea, R., & Riley, M. (2018). Unmanned aerial vehicle inspection of the Placer River Trail Bridge through image-based 3D modelling. *Structure and Infrastructure Engineering*, *14*(1), 124–136. https://doi.org/10.1080/15732479.2017.1330891

Kirillov, A., Wu, Y., He, K., & Girshick, R. (2020). PointRend: Image segmentation as rendering. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Seattle, WA pp. 9799–9808. https://doi.org/10.1109/CVPR42600.2020.00982

Lee, J., & Kim, S. (2007). Structural damage detection in the frequency domain using neural networks. *Journal of Intelligent Material Systems and Structures*, *18*(8), 785–792. https://doi.org/10.1177/1045389x06073640

Li, Z., Park, H. S., & Adeli, H. (2017). New method for modal identification of super high-rise building structures using discretized synchrosqueezed wavelet and Hilbert transforms. *The Structural Design of Tall and Special Buildings*, *26*(3), e1312. https://doi.org/10.1002/tal.1312

Liang, X. (2019). Image-based post-disaster inspection of reinforced concrete bridge systems using deep learning with Bayesian optimization. *Computer-Aided Civil and Infrastructure Engineering*, *34*(5), 415–430. https://doi.org/10.1111/mice.12425

Lin, T. Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., & Zitnick, C. L. (2014). Microsoft COCO: Common Objects in Context. *In European Conference on Computer Vision*, Zurich, Switzerland pp. 740–755.

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

Liu, Y. F., Nie, X., Fan, J. S., & Liu, X. G. (2020). Image-based crack assessment of bridge piers using unmanned aerial vehicles and three-dimensional scene reconstruction. *Computer-Aided Civil and Infrastructure Engineering*, *35*(5), 511–529. https://doi.org/10.1111/mice.12501

Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Boston, MA pp. 3431–3440. https://doi.org/10.1109/CVPR.2015.7298965

Luo, C., Yu, L., Yan, J., Li, Z., Ren, P., Bai, X., Yang, E., & Liu, Y. (2021). Autonomous detection of damage to multiple steel surfaces from 360° panoramas using deep neural networks. *Computer-Aided Civil and Infrastructure Engineering*, *36*(12), 1585–1599. https://doi.org/10.1111/mice.12686

Mariniello, G., Pastore, T., Menna, C., Festa, P., & Asprone, D. (2021). Structural damage detection and localization using decision tree ensemble and vibration data. *Computer-Aided Civil and Infrastructure Engineering*, *36*(9), 1129–1149. https://doi.org/10.1111/mice.12633

Mazumder, R. K., Salman, A. M., & Li, Y. (2021). Failure risk analysis of pipelines using data-driven machine learning algorithms. *Structural Safety*, *89*, 102047. https://doi.org/10.1016/j.strusafe.2020.102047

Miao, Z., Ji, X., Okazaki, T., & Takahashi, N. (2021). Pixel-level multicategory detection of visible seismic damage of reinforced concrete components. *Computer-Aided Civil and Infrastructure Engineering*, *36*(5), 620–637. https://doi.org/10.1111/mice.12667

Ministry of Land, Infrastructure, Transport and Tourism (MLIT). (2018). *Roads in Japan 2018*. https://www.mlit.go.jp/road/road_e/index_e.html

Morgenthal, G., Hallermann, N., Kersten, J., Taraben, J., Debus, P., Helmrich, M., & Rodehorst, V. (2019). Framework for automated UAS-based structural condition assessment of bridges. *Automation in Construction*, *97*, 77–95. https://doi.org/10.1016/j.autcon.2018.10.006

Nagatani, K., Abe, M., Osuka, K., Chun, P. J., Okatani, T., Nishio, M., Chikushi, S., Matsubara, T., Ikemoto, Y., & Asama, H. (2021). Innovative technologies for infrastructure construction and maintenance through collaborative robots based on an open design approach. *Advanced Robotics*, *35*(11), 715–722. https://doi.org/10.1080/01691864.2021.1929471

Nguyen, T., Kashani, A., Ngo, T., & Bordas, S. (2019). Deep neural network with high-order neuron for the prediction of foamed concrete strength. *Computer-Aided Civil and Infrastructure Engineering*, *34*(4), 316–332. https://doi.org/10.1111/mice.12422

Oh, B. K., Kim, K. J., Kim, Y., Park, H. S., & Adeli, H. (2017). Evolutionary learning based sustainable strain sensing model for structural health monitoring of high-rise buildings. *Applied Soft Computing*, *58*, 576–585. https://doi.org/10.1016/j.asoc.2017.05.029

Okazaki, Y., Okazaki, S., Asamoto, S., & Chun, P. J. (2020). Applicability of machine learning to a crack model in concrete bridges. *Computer-Aided Civil and Infrastructure Engineering*, *35*(8), 775–792. https://doi.org/10.1111/mice.12532

Perez-Ramirez, C. A., Amezquita-Sanchez, J. P., Adeli, H., Valtierra-Rodriguez, M., Camarena-Martinez, D., & Romero-Troncoso, R. J. (2016). New methodology for modal parameters identification of smart civil structures using ambient vibrations and synchrosqueezed wavelet transform. *Engineering Applications of Artificial Intelligence*, *48*, 1–12. https://doi.org/10.1016/j.engappai.2015.10.005

Perez-Ramirez, C. A., Amezquita-Sanchez, J. P., Valtierra-Rodriguez, M., Adeli, H., Dominguez-Gonzalez, A., & Romero-Troncoso, R. J. (2019). Recurrent neural network model with Bayesian training and mutual information for response prediction of large buildings. *Engineering Structures*, *178*, 603–615. https://doi.org/10.1016/j.engstruct.2018.10.065

Prasanna, P., Dana, K. J., Gucunski, N., Basily, B. B., La, H. M., Lim, R. S., & Parvardeh, H. (2014). Automated crack detection on concrete bridges. *IEEE Transactions on Automation Science and Engineering*, *13*(2), 591–599. https://doi.org/10.1109/TASE.2014.2354314

Rafiei, M. H., & Adeli, H. (2017). A novel machine learning-based algorithm to detect damage in high-rise building structures. *The Structural Design of Tall and Special Buildings*, *26*(18), e1400. https://doi.org/10.1002/tal.1400

Rafiei, M. H., & Adeli, H. (2018a). A novel unsupervised deep learning model for global and local health condition assessment of structures. *Engineering Structures*, *156*, 598–607.

Rafiei, M. H., & Adeli, H. (2018b). Novel machine-learning model for estimating construction costs considering economic variables and indexes. *Journal of Construction Engineering and Management*, *144*(12), 04018106. https://doi.org/10.1061/(ASCE)CO.1943-7862.0001570

Rafiei, M. H., Khushefati, W. H., Demirboga, R., & Adeli, H. (2017). Supervised deep restricted Boltzmann machine for estimation of concrete. *ACI Materials Journal*, *114*(2), 237–244. https://doi.org/10.14359/51689560

Rahman, A., Wu, Z. Y., & Kalfarisi, R. (2021). Semantic deep learning integrated with RGB feature-based rule optimization for facility surface corrosion detection and evaluation. *Journal of Computing in Civil Engineering*, *35*(6), 04021018. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000982

Ranjbar, I., Toufigh, V., & Boroushaki, M. (2022). A combination of deep learning and genetic algorithm for predicting the compressive strength of high-performance concrete. *Structural Concrete*, *23*(4), 2405–2418. https://doi.org/10.1002/suco.202100199

Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster R-CNN: Towards real-time object detection with region proposal networks. *Proceedings of 28th Conference on Advances in Neural Information Processing Systems*, *1*, 91–99.

Sacks, R., Kedar, A., Borrmann, A., Ma, L., Brilakis, I., Hüthwohl, P., Daum, S., Kattel, U., Yosef, R., Liebich, T., Barutcu, B. E., & Muhic, S. (2018). SeeBridge as next generation bridge inspection: Overview, information delivery manual and model view definition. *Automation in Construction*, *90*, 134–145. https://doi.org/10.1016/j.autcon.2018.02.033

Sajedi, S. O., & Liang, X. (2021). Uncertainty-assisted deep vision structural health monitoring. *Computer-Aided Civil and Infrastructure Engineering*, *36*(2), 126–142. https://doi.org/10.1111/mice.12580

Sari, Y., Prakoso, P. B., & Baskara, A. R. (2019). Road crack detection using support vector machine (SVM) and OTSU algorithm. *2019 6th International Conference on Electric Vehicular Technology (ICEVT)*, Bali, Indonesia pp. 349–354. https://doi.org/10.1109/ICEVT48285.2019.8993969

Shi, J., Dang, J., Cui, M., Zuo, R., Shimizu, K., Tsunoda, A., & Suzuki, Y. (2021). Improvement of damage segmentation based on pixel-level data balance using VGG-Unet. *Applied Sciences*, *11*(2), 518. https://doi.org/10.3390/app11020518

Shim, C. S., Dang, N. S., Lon, S., & Jeon, C. H. (2019). Development of a bridge maintenance system for prestressed concrete bridges using 3D digital twin model. *Structure and Infrastructure Engineering*, *15*(10), 1319–1332. https://doi.org/10.1080/15732479.2019.1620789

Shim, C. S., Kang, H., Dang, N. S., & Lee, D. (2017). Development of BIM-based bridge maintenance system for cable-stayed bridges. *Smart Structures and Systems*, *20*(6), 697–708.

Shin, S., Moon, H., & Shin, J. (2022). BIM-based maintenance data processing mechanism through COBie standard development for port facility. *Applied Sciences*, *12*(3), 1304. https://doi.org/10.3390/app12031304

Sirca, Jr, G. F., & Adeli, H. (2018). Infrared thermography for detecting defects in concrete structures. *Journal of Civil Engineering and Management*, *24*(7), 508–515. https://doi.org/10.3846/jcem.2018.6186

Soto, M. G., & Adeli, H. (2019). Semi-active vibration control of smart isolated highway bridge structures using replicator dynamics. *Engineering Structures*, *186*, 536–552. https://doi.org/10.1016/j.engstruct.2019.02.031

The Association of Consulting Engineering Companies Canada (ACEC), The Canadian Construction Association (CCA), the Canadian Parks and Recreation Association (CPRA), the Canadian Public Works Association (CPWA), the Canadian Society for Civil Engineering (CSCE), the Canadian Urban Transit Association (CUTA), the Canadian Network of Asset Managers (CNAM), the Federation of Canadian Municipalities (FCM). (2019). *The Canadian infrastructure report card 2019*. http://canadianinfrastructure.ca/en/index.html

Vezhnevets, V., & Konouchine, V. (2005). GrowCut: Interactive multi-label ND image segmentation by cellular automata. *Proceedings of 15th International Conference on Computer Graphics and Applications*, *1*(4), 150–156.

Xu, J., Gui, C., & Han, Q. (2020). Recognition of rust grade and rust ratio of steel structures based on ensembled convolutional neural network. *Computer-Aided Civil and Infrastructure Engineering*, *35*(10), 1160–1174. https://doi.org/10.1111/mice.12563

Yamane, T., & Chun, P. J. (2020). Crack detection from a concrete surface image based on semantic segmentation using deep learning. *Journal of Advanced Concrete Technology*, *18*(9), 493–504. https://doi.org/10.3151/jact.18.493

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

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

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

Zhang, Y., & Yuen, K. V. (2021). Crack detection using fusion features-based broad learning system and image processing. *Computer-Aided Civil and Infrastructure Engineering*, *36*(12), 1568–1584. https://doi.org/10.1111/mice.12753

Zisserman, A., & Hartley, R. (2003). *Multiple view geometry in computer vision*. Cambridge University Press.

How to cite this article: Yamane, T., Chun, P.-J., Dang, J., & Honda, R. (2023). Recording of bridge damage areas by 3D integration of multiple images and reduction of the variability in detected results. *Computer-Aided Civil and Infrastructure Engineering*, *38,* 2391–2407. https://doi.org/10.1111/mice.12971