Mechanical Systems and Signal Processing 160 (2021) 107850

Contents lists available at ScienceDirect

Mechanical Systems and Signal Processing

journal homepage: www.elsevier.com/locate/ymssp

Synthetic environments for vision-based structural condition assessment of Japanese high-speed railway viaducts

Yasutaka Narazaki [a,*], Vedhus Hoskere [b], Koji Yoshida [c], Billie F. Spencer [d], Yozo Fujino [e]

[a] Zhejiang University/University of Illinois at Urbana-Champaign Institute, Zhejiang University, China
[b] Department of Civil and Environmental Engineering, the University of Houston, USA
[c] Central Japan Railway Company, Japan
[d] Department of Civil and Environmental Engineering, University of Illinois at Urbana-Champaign, USA
[e] Institute of Advanced Sciences, Yokohama National University, Japan

ARTICLE INFO

Article history:
Received 19 August 2020
Received in revised form 10 January 2021
Accepted 10 March 2021

Keywords:
Automated structural inspection
Reinforced concrete
Railway viaduct
Synthetic environment
Semantic segmentation
Monocular depth estimation

ABSTRACT

Civil infrastructure condition assessment using visual recognition methods has shown significant potential for automating various aspects of the problem, including identification and localization of critical structural components, as well as detection and quantification of structural damage. The application of those methods typically requires large amounts of training data that consists of images and corresponding ground truth annotations. However, obtaining such datasets is challenging, because the images are annotated manually in most existing approaches. With the limited availability of datasets, development of effective visual recognition systems that can extract all required information is not straightforward. This research leverages synthetic environments to develop a unified system for automated vision-based structural condition assessment that can identify and localize critical structural components, and then detect and quantify damage of those components. The synthetic environments can produce images and associated ground truth annotations for semantic segmentation of structural components and damage, as well as monocular depth estimation for structural component localization. To illustrate the approach, automated vision-based structural condition assessment of reinforced concrete railway viaducts for a Japanese high-speed railway line (the Tokaido Shinkansen) is explored. The effectiveness of the synthetic environments and the generated dataset (the Tokaido dataset) is demonstrated by training fully convolutional network-based semantic segmentation and monocular depth estimation algorithms, and then testing the networks using both synthetic and real-world images. Finally, all trained algorithms are combined to realize an automated system for structural condition assessment.

1. Introduction

Visual recognition approaches for civil infrastructure condition assessment have been investigated actively, demonstrating significant potential for improving the assessment process in terms of efficiency, reliability, and the degree of autonomy [19,57]. In particular, algorithms based on convolutional neural network (CNN) architectures have shown to be effective to implement the steps of the civil infrastructure condition assessment using image data, including the detection and localiza-

* Corresponding author at: Room C323, Engineering Building, 718 East Haizhou Road, Haining, Zhejiang 314400, China.
E-mail address: narazaki@intl.zju.edu.cn (Y. Narazaki).

https://doi.org/10.1016/j.ymssp.2021.107850
0888-3270/

tion of critical structural components and damage to those components. In the existing literature, those algorithms are applied by preparing image data and the corresponding ground truth annotations (training data), and then optimizing the parameters of the CNN-based detection, segmentation, and regression models using gradient-based iterative algorithms (training). After training, the performance of the model is evaluated using a dataset that has not been seen by the model during the training process (testing). Depending on the target problems and the specific contexts in which the algorithms are applied, existing work applies different approaches to prepare training and testing datasets to enable their investigations.

A summary of the existing literature about civil structural component detection and localization using image data is provided in Table 1, with an emphasis on dataset preparation approaches. Narazaki et al. [42] investigate the application of semantic segmentation algorithms (estimating pixel-wise labels), such as fully convolutional networks (FCN) [36] and SegNet [2], to perform the vision-based bridge component recognition task automatically. In Liang [35], an object detection algorithm (Fast R-CNN [17]) is applied to draw bounding boxes around reinforced concrete bridge columns in the input image. In these studies, the training data is generated manually, limiting the amount of data. Furthermore, accurately annotating structural components for all types and scales is not always straightforward, introducing additional error to the recognition results. To alleviate the need of large amount of training data, Gao et al. [16] investigate the transfer learning approach, where an existing network (VGGNet) trained on the ImageNet image classification dataset is fine-tuned using 2,000 additional images of civil structures, and then the structural components are localized using class activation map (CAM) [67]. While the training data generation step is simplified significantly in this research (single label per image is specified, instead of pixel-wise labels or object bounding boxes), the benefit comes at the expense of the limited accuracy for the structural component localization. Therefore, while the existing literature has demonstrated significant potential for those approaches, the performance of the visual recognition algorithms needs to be improved by increasing the size and quality of the training data.

Structural damage detection and quantification is another research area for which visual recognition approaches have been proven effective. Earlier research efforts focus on images of the structural surface viewed from close or controlled distances; in other words, the decision is binary, i.e., damaged surface or intact surface. For example, Cha et al. [10] apply convolutional neural networks to sub-images to detect and localize concrete cracks. Xu et al. [65] investigate steel crack recognition problem by sub-image classification approach using the CNN improved to leverage multi-scale features (Fusion CNN). A publicly available dataset of annotated images of concrete cracks has been developed to support the investigation of visual recognition approaches [37]. Following the promising results of such efforts, the problem has been generalized to more complex scenarios, where the input image is not necessarily the close-up view of the target structural surface. Yeum [66] performed one of the pioneering work in this category, where an object detection algorithm (R-CNN [18]) is applied to find concrete spalling in images of complex scenes of structures. Cha et al. [11] use the Faster R-CNN object detection algorithm [52] to recognize steel corrosion (two severity levels), steel delamination, bolt corrosion, and concrete cracks. Pan et al. [48] investigate a four-class image classification problem for the damage states of reinforced concrete columns, as well as the exposed rebar recognition using object detection algorithm (YOLOv2 [51]). Application of semantic segmentation algorithms has also been investigated to obtain the maps of structural damage at the resolution of the input image. Liang [35] applies the SegNet algorithm to identify concrete damage in the images of bridges. To obtain high accuracy even for the images of complex scenes, Hoskere et al. [22] combine two semantic segmentation tasks, segmentations based on material types and damage types, and apply a multi-task FCN with the homoscedastic loss function. Hoskere et al. [21] developed a software called InstaDam, where methods to annotate structural damage accurately and efficiently are implemented. The selected existing literature about damage detection and quantification is summarized in Table 2, with additional notes on datasets and annotations.

The size of the dataset in the majority of the existing literature is less than 1,000 unique images, except for image-wise annotations that provide limited information about the location, size, and shape of the damage. Despite the potential of performance improvement, increasing the size of training datasets is not straightforward, because images are annotated manually in the existing approaches. To increase the apparent size of the datasets, the images and the corresponding annotations are augmented by different methods, such as cropping, rotation, and intensity scaling. However, the benefit of such data augmentation approaches is marginal, and increasing the number of unique images is the desirable way to improve the training and testing processes.

Depth map estimation is a critical step to recognizing the geometric structure of the surrounding environment using a monocular camera [13,14,34]. The problem is frequently discussed in the context of vision-based navigation of robotic platforms [6,60], and in particular, autonomous driving [45][63]. While the findings from the literature in the related areas are expected to be applicable to the autonomous navigation of robotic platforms (e.g. unmanned aerial vehicles, or UAVs) to col-

Table 1
Selected existing literature about civil structural component detection and localization using image data.

Literature	Component types	Algorithms used	Notes on dataset and annotation
[42]	RC bridge columns, beams & slabs	FCN, SegNet	1,563 images (up to 320 × 320), pixel-wise annotation.
[35]	RC bridge columns	Fast R-CNN	236 images (430 × 400), bounding boxes.
[16]	Wall, columns/beams	CNN + CAM	2,000 images, image-wise annotation.

Table 2
Selected existing literature about damage detection and quantification using image data (CCR: Concrete cracks, CS: Concrete spalling, SC: Steel corrosion, SCR: Steel cracks, SD: Steel delamination, ACR: Asphalt. cracks).

Literature	Damage types	Algorithms used	Notes on dataset and annotation
[22]	CCR, CS, SC, SCR, ACR	Multi-task FCN	1,695 images (600×500), cropped from 339 high-resolution images, pixel-wise annotation.
[48]	CCR, CS	CNN, YOLOv2	2,260 images for component-level damage state classification (224×224 or 227×227), image-wise annotation. 780 images for exposed rebar detection (224×224 or 227×227), bounding boxes.
[35]	CCR, CS	SegNet	436 images (430×400), pixel-wise annotation.
[11]	CCR, SC, SD	Faster R-CNN	2,366 images (500×375), cropped from 297 high-resolution images ($6,000 \times 4,000$), bonding boxes.
[37]	CCR	–	Dataset containing 56,092 images (256×256), cropped from 230 high-resolution images, image-wise annotation.
[65]	SCR	Fusion CNN	67,200 images (64×64) cropped from 350 high-resolution images ($3,264 \times 4,928$), image-wise annotation.
[10]	CCR	CNN	40 K images (256×256) cropped from 332 high-resolution images ($4,928 \times 3,264$ or $5,888 \times 3,584$), image-wise annotation.
[66]	CS	R-CNN	1,086 images, bonding boxes.

lect image data for the civil structure condition assessment, lack of appropriate dataset has become a bottleneck to investigate such topics. One of the preliminary efforts toward monocular depth estimation has been presented by Narazaki et al. [44]. However, the research has been limited to the dataset generation in a single synthetic environment that contains reinforced concrete bridges created manually.

In all the three visual recognition tasks discussed herein, a large amount of data is needed to enable the application of visual recognition methods in a realistic scenario of civil structure condition assessment, and to improve the performance of those methods. For example, structural component recognition is difficult when the components are far or not the main objects in images. Even when all structural components are identified accurately, the performance of damage recognition systems developed individually is uncertain, because such systems may not be trained sufficiently for the damage of those components. Without rich training data from the desired application scenario, visual recognition systems that automate multiple aspects of structural condition assessment are unlikely to perform satisfactorily.

An approach that inspires this research is the use of synthetic environment to investigate the recognition of general scene objects for autonomous driving purposes [54]. The study has shown that the semantic segmentation accuracy improves significantly by using 13,400 synthetic images with 200–300 real-world images. The accuracy improvement reaches more than 10% in some test cases. To bring the benefit of synthetic environment into the visual recognition tasks for civil infrastructure condition assessment, the authors of this study have investigated approaches to model civil infrastructure in various structural and damage conditions. In particular, the authors [23,24] have proposed a modeling approach, termed physics-based graphics models (PBGMs), in which damage hot spots identified by the finite element analysis are used to realize photo-realistic synthetic damage scenarios. The synthetic environment and the PBGMs have been first applied to the semantic segmentation of steel surface damage (e.g. corrosion) of navigation infrastructure on the U.S. inland waterways, and later extended to displacement and strain measurement of miter gates [42] and an experimental bridge structure [41]. These studies have shown significant potential of facilitating civil infrastructure condition assessment by augmenting training data for visual recognition algorithms, as well as providing means of the quantitative evaluation of algorithm performance.

This research leverages synthetic environments to develop a unified system for automated vision-based structural condition assessment that can identify and localize critical structural components, and then detect and quantify damage of those components. The synthetic environments can produce images and associated ground truth annotations for semantic segmentation of structural components and damage, as well as monocular depth estimation for structural component localization. To illustrate the approach, this research explores automated vision-based structural condition assessment of reinforced concrete (RC) railway viaducts for a Japanese high-speed railway line (the Tokaido Shinkansen) operated and maintained by the Central Japan Railway Company. Such viaducts are designed by following a standard design procedure, which had been developed as simple and systematic steps to determine the geometric details using a few parameters, including slab height and track curvature [32]. When the Tokaido Shinkansen was constructed, such viaducts were mass-produced, comprising the vast majority of the 116 km of railway bridges of the 515 km-length Tokaido Shinkansen line [47]. Besides the generic geometry, this type of structures bring in an additional benefit for the investigation of damage recognition problems: major structural damage to the RC viaducts after earthquakes are cracks, spalling, and exposed rebar on the column surfaces [28,39]. The synthetic environments developed in this research consist of 2,000 viaducts with random geometry realized by the standard design procedure, as well as random damage scenarios (concrete cracks, spalling, and exposed rebar) of the viaduct columns. The synthetic environments are used to produce a dataset of 8,648 images for structural component recognition and depth estimation, as well as 7,990 images for damage recognition. The image resolution is $1,920 \times 1,080$, and each image is associated with ground truth pixel-wise information of structural component types, damage types, the depth values (Fig. 1). The produced dataset, termed Tokaido dataset, is used effectively to train visual recognition algorithms for the structural condition assessment tasks. The performance of the trained algorithms is then tested using both synthetic and real-world images.

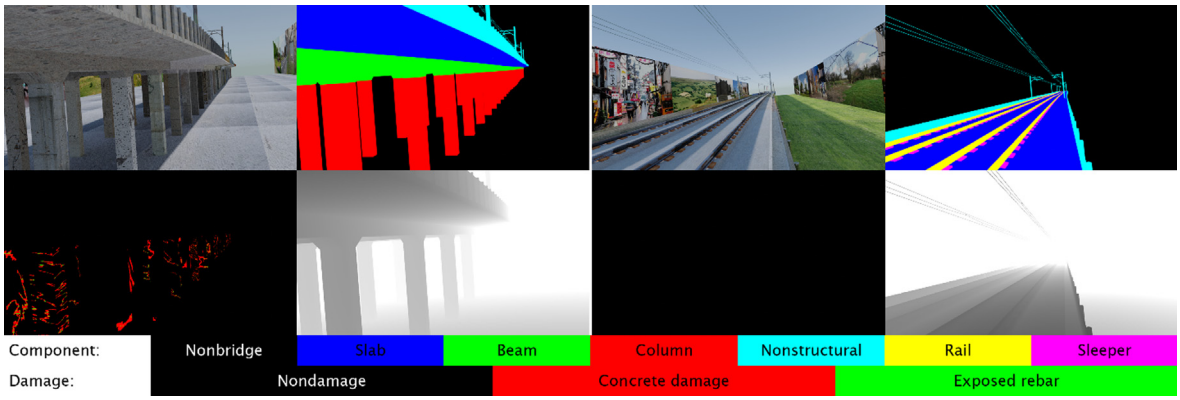


Fig. 1. Example images (top left), structural component label maps (top right), damage maps (bottom left), and depth maps (bottom right) from the synthetic Tokaido dataset.

Finally, the trained visual recognition algorithms are integrated to realize an automated system for structural condition assessment, which can extract type, location, and shape of structural components in images, as well as any damage to those structural components.

The next section discusses the steps to develop the synthetic environments for civil infrastructure condition assessment. Then, in Section 3, the synthetic environment of the Japanese high-speed railway viaducts is developed, and the detail of the produced dataset (the Tokaido dataset) is described. In Section 4, the semantic segmentation and depth estimation algorithms are applied to demonstrate the effectiveness of the Tokaido dataset. Section 5 discusses the results of data analysis, and Section 6 presents concluding remarks.

2. Synthetic environments for civil infrastructure condition assessment

2.1. Overview

This section describes the steps of developing synthetic environments that can produce training data for visual recognition tasks related to civil infrastructure condition assessment. The overview of the steps is depicted in Fig. 2. First, meshes are created to represent the geometry of the target structure (Step 1 in Fig. 2). Then, appropriate textures for each part of the meshes are determined (Step 2 in Fig. 2). A simple method for this step is to specify images that contain information about mesh surface colors. Alternatively, multiple images can be specified to control the detail of the mesh surface, such as color, roughness, metallic properties, and shading effect caused by uneven surface normal fields. In step 3, the selected images are assigned to the model by defining mappings between the images and the mesh surface. After the model of the target structure is created, synthetic cameras are placed to render images (Step 4 in Fig. 2). At the same time, one can obtain ground truth information for the visual recognition tasks, such as ground truth maps of structural component labels, ground truth

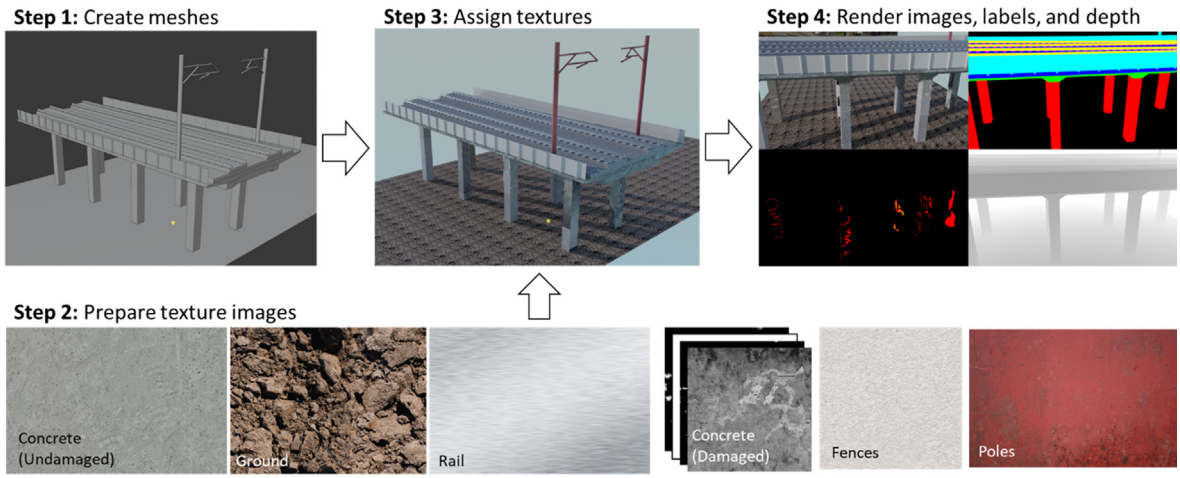


Fig. 2. Overview of the computer graphics approach for civil infrastructure condition assessment.

maps of damage labels, and ground truth depth maps. The remaining part of this section discusses key concept and implementation procedure of each step presented in Fig. 2.

Throughout this research, Blender 3D computer graphics modeling software is used to implement the modeling steps. Besides, the Blender-Python API [3,4] is used to automate repeated Blender operations, so that many random viaducts and damage scenarios can be produced efficiently.

2.2. Mesh creation

The first step of developing the synthetic environments is to define meshes that specify the geometry of structures. In Blender, mesh can be created either manually by combining basic operations, such as adding primitive shapes and scaling, moving, and extruding those shapes, or programmatically by directly specifying mesh vertex coordinates and faces. This research defines the meshes programmatically using Python to enable the creation of random railway viaduct models.

2.3. Texture image preparation

As discussed in Section 2.1, the simple method for defining a texture is to specify an image that represents the color of the mesh surface. To define realistic textures that are appropriate for the random creation of railway viaduct models, this research develops a repository of images of concrete (534 images), steel (403 images), painted surface (318 images), wood (439 images), and ground (113 images), downloaded from the Internet using the Bulk Bing Image Downloader (BBID) [8] and from https://www.textures.com [62]. To create the image repository of surrounding scenes, 371 images of urban and rural scenes are downloaded from the internet similarly. When a texture in each category is required by the model, an image is sampled randomly from the corresponding category of the repository.

A more advanced approach for texturing is physically-based rendering [9]. In this approach, various properties of the mesh surface, as well as RGB color, can be controlled by specifying multiple texture images. In particular, this research controls the following four properties to generate realistic textures of damage to the bridge columns: color, roughness, metallic properties, and surface normal field. The effect of controlling each of those properties is shown in Fig. 3 for sample texture images available at textures.com. By using the RGB image and the metallic map, the proportion of reflection and refraction when the light hit the object surface can be specified: large metallic value (white color) assigns highly reflective surface (metal), while the light passes through the surface (refraction), scattered, and re-emitted outward when the small value is specified (dark color). The roughness map controls a statistical model of the micro-structure of the target surface termed microfacets [49]. The small value (dark color) specifies smooth surface, while the light is scattered more when the roughness value is high (white color). Finally, geometric structure of the target surface is modeled by defining a surface normal map, based on which the shading on the target surface is adjusted to add visual effects that represents the geometry. In Fig. 3, the visual effect of the protrusions is attained by controlling the shading on the surface this way, rather than deforming the mesh, an approach that is significantly more expensive computationally.

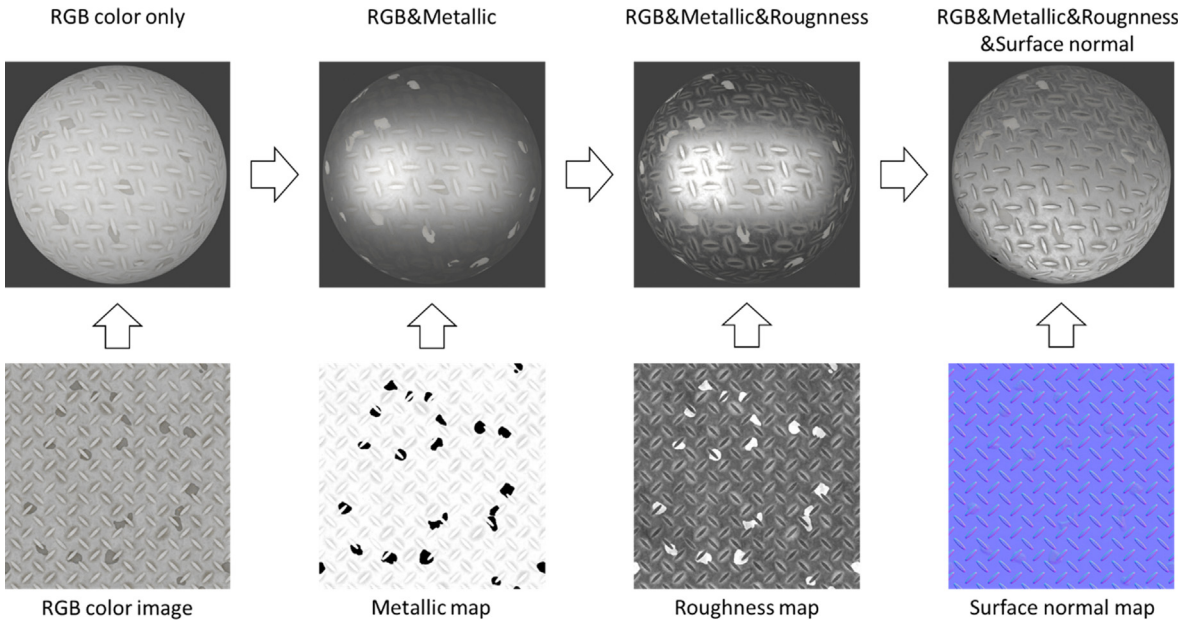


Fig. 3. Steps of physically-based rendering.

2.4. Texture assignment

The defined textures can be assigned to the surface of the mesh using the Blender Principled BSDF node (Fig. 4) by either specifying a default value or connecting prepared images (Image Texture node) to the appropriate entry of the node. Besides examples shown in Fig. 3, an example of applying the Principled BSDF node to generate concrete crack model is presented in Fig. 5, where steps discussed in a Blender tutorial Tutorials [5] are implemented. The method generates noise texture, and successively applies thresholding and scaling to create a displacement map that can be converted to the surface normal map using the Blender Bump node. At the same time, the method defines different diffuse colors between the damaged and undamaged regions to enhance the reality. The shapes and the appearances of the cracks can be controlled by adjusting parameters of noise generation, thresholding, scaling, and the textures of damaged and undamaged regions.

2.5. Rendering images, labels, and depth

Once the synthetic models of the structures are created, synthetic cameras are placed to render photo-realistic images. At the same time, ground truth label maps for structural components and damage can be rendered by assigning additional textures that contains such information (e.g. pure red color can be assigned to the mesh surface belonging to bridge columns). To eliminate unwanted shading effects in the label maps, Blender emission shader is used, instead of the principled BSDF. The corresponding ground truth depth maps can be obtained using Blender compositing functionality.

3. Development of synthetic environment of Japanese high-speed railway viaducts

This section describes the process of developing synthetic environments of Japanese high-speed railway viaducts, and then presents the detail of the dataset (the Tokaido dataset) produced using the synthetic environments. The emphasis is put on how this research enables the creation of random viaduct models that conform to the standard design, as well as how this research implements random structural damage that mimics the actual damage of the target structure.

3.1. Mesh generation based on the standard design

Typical RC viaducts with the standard design are shown in Fig. 6. Each viaduct is 24 m, comprising of three central spans (6 m each) and cantilever-type end spans (3 m each). The detailed design of a viaduct is determined based on three param-

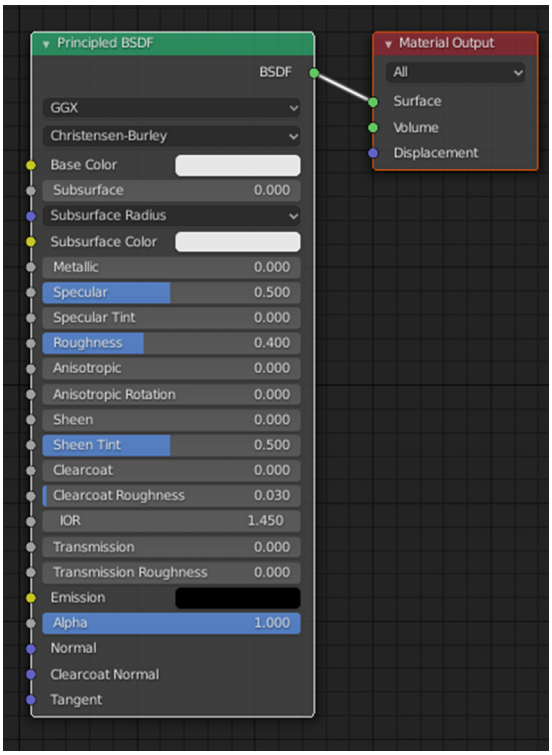


Fig. 4. Blender principled BSDF node.

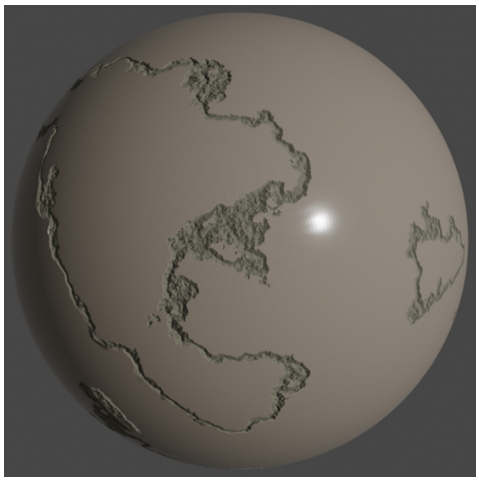


Fig. 5. Concrete crack model created by steps presented in [5].

Fig. 6. Google street view image of typical RC viaducts with the standard design.

eters: loading conditions, viaduct height, and geotechnical conditions. The loading conditions of viaducts are classified based on their trajectories: straight trajectories when the radius of curvature is larger than 5,000 m, and curved trajectories when the radius of curvature is smaller than 5,000 m. Viaducts with curved trajectories are stiffer in the transverse directions and have 20% more rebars than those with straight trajectories. The dimensions of cross-sections of viaduct columns and intermediate beams are presented in Table 3 for different loading conditions and viaduct heights. Geotechnical conditions affect the design of footing underground, which is not modeled in this research. The readers are directed to the original Japanese literature for other dimensions and the further detail of the design concept [32].

This research first generates random viaduct trajectories, from which viaduct meshes are generated following the standard design procedure. The steps to generate a trajectory are described below:

(i) Determine the number of viaducts N and the initial viaduct height. In this research, $N = 10$, and the initial height is sampled uniformly from the range [6.5 m, 12 m].
(ii) Generate a random trajectory using the inverse Fast Fourier Transform (FFT) of rectangular spectra with uniform random phase (the cutoff frequency is $2/N$). The trajectory is defined using three noise sequences that represent ground height increment, viaduct height increment, and the curvature of the viaduct centerline (inverse of the radius of curvature). The amplitudes of the FFT spectra are set to 1.0 for the ground height increment and 0.2 for the viaduct height increment. The noise sequence for the viaduct curvature is scaled, so that the minimum radius of curvature along the trajectory is a value sampled from a uniform distribution in the range [500,5000].
(iii) Apply constraints to the trajectory. Based on the typical values mentioned in the existing literature [27,32], this research defines the constraints as shown in Table 4.

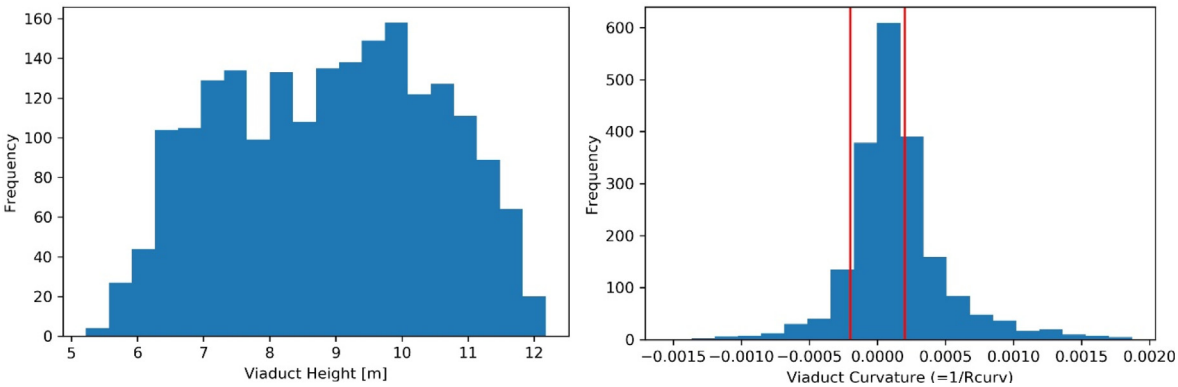
Following the steps (i)-(iii), 200 trajectories of 10 viaducts are generated (2,000 viaducts in total). The histograms of viaduct height and curvature are shown in Fig. 7. With the sampling scheme implemented in this research, the height and curvature of the generated viaducts span the range of those parameters of actual structures realized by the standard design.

Table 3
Dimensions of structural component cross sections [cm × cm] [32] Straight: radius of curvature is larger than 5,000 m. Curved: radius of curvature is smaller than 5,000 m,

Viaduct height H[m]	Columns (Straight)	Intermediate beams (Straight)	Columns (Curved)	Intermediate beams (Curved)
$5.5 < H \leq 7$	60×60	None	60×70	None
$7 < H \leq 8.5$	70×70	None	70×80	None
$8.5 < H \leq 10$	80×80	None	80×90	None
$10 < H \leq 12$	70×70	80×60	70×85	80×60
$12 < H \leq 14$	80×80	80×60	80×95	80×60

Table 4
Constraints used in this research.

Viaduct Height$H$[m]	$5.0 \leq H \leq 14$
Viaduct slope$S$	$\|S\| < 0.02$



(a) Viaduct height (b) Viaduct curvature

Fig. 7. Histograms of viaduct height and curvature. Red lines separate straight and curved trajectories.

Information about the design of the superstructure of this type of viaducts, such as tracks, poles, and wires, is relatively sparse in the literature. To supplement the information, this research refers to literature about other railway lines in Japan as well [15,27,40,61]. The modeling accuracy needs to be improved to enable further investigation of inspection and monitoring tasks that require precise information of those components, which is part of the future work.

3.2. Random texture generation and assignment

Once the meshes of the viaducts are generated, surface textures are defined using the images in the repository discussed in Section 2.3. The texturing process for each component of the synthetic environment is as follows:

Slabs, beams, and undamaged columns. One concrete texture image is selected randomly for each slab, beam, or undamaged column, and the selected image is assigned using the Principled BSDF node, where the "Metallic" and "Roughness" parameters are set to 0.0 and 1.0, respectively. Moreover, the scale of the texture is sampled uniformly from the range [0.1,1.0].

Sleepers. For each viaduct, sleeper material is sampled from concrete, steel, and wood materials with equal probabilities. Then, one image is selected randomly from the images of the selected material. For concrete and wood materials, the Metallic and Roughness parameters of the principled BSDF block are set to 0.0 and 1.0, respectively. For steel materials, the Metallic parameter is set to 1.0, and the Roughness parameter is sampled uniformly from the range [0, 1]. The scale of the texture is sampled uniformly from the range [0.1,1.0].

Rails and wires. One image is sampled randomly from the set of steel images. The other parameter setting follows the same steps as that of steel materials for sleepers.

Fences and poles. One image is sampled randomly from the set of painted surface images. Both the Metallic and the Roughness parameters are sampled uniformly from the range [0, 1]. The scale of the texture is sampled uniformly from the range [0.1,1.0].

Ground. One image is sampled randomly from the set of ground images. The Metallic and the Roughness parameters are set to 0.0 and 1.0, respectively, and the texture scale is sampled randomly from the range [0, 0.1].

Surrounding scene. The modeling of the surrounding environment of the viaducts is simplified in this research, considering that the focus of the research is on the viaducts: the surrounding environment is modeled using panels of outdoor scene images placed at the distance of 30 m from the viaduct centerline. An image of the surrounding scene is selected randomly from the repository for each panel. To avoid shading caused by the panels, emission shader is used with low intensity values (0.5).

3.3. Random damage texture generation and assignment

This research extends the method discussed in the aforementioned tutorial [5] to generate random damage to the RC viaduct columns. The idea of texture generation for concrete damage is shown in Fig. 8. The cracks shown in Fig. 5 can be characterized by the following three properties: (i) random geometry realized by thresholding and scaling noise textures, (ii) displacement texture that induces shading effects that are characteristic to concrete damage, and (iii) texture discontinuity that represents the texture difference between the exterior surface (typically subject to dirt, dust, paint etc.) and the fresh concrete surface exposed by the removal of the exterior surface concrete. This research *defines* the synthetic concrete damage as the surface regions with all the three properties, and randomly sample parameters of those properties to realize concrete damage textures. Note that the defined concrete damage is not always realistic, as shown in Fig. 8. For example, the defined damage can occur in every part of the structural surface with equal probability, regardless of the shear and the bending moment distributions. Similarly, the shapes of concrete damage are generated purely randomly, without referring to the structural mechanics theory. Ideally, structural analysis/mechanics should be incorporated to make the set of defined damage as close as possible to the set of realistic damage. On the other hand, for the simplicity of implementation, this research designed the dataset such that the defined damage covers the realistic damage, and therefore reliable detection of the defined damage leads to the reliable detection of the realistic damage.

The implementation procedure for concrete damage texture generation is shown in Fig. 9. First, noise texture of the resolution $2,048 \times 2,048$ is generated using Perlin simplex noise noise (PyPI [46]), where the number of octaves is 8, and the frequency parameters are sampled uniformly between 20% and 200% of the texture dimensions. After taking the absolute values, the noise is clipped at the predetermined threshold value to get the synthetic crack profile. Mathematically, these operations can be written as

$$\boldsymbol{P}_{\mathbf{c}}^{-} = \min\left(t, \mathrm{abs}\left(\boldsymbol{N}\left(frq_x, frq_y, oct\right)\right)\right) \tag{1}$$

where $\boldsymbol{N}\left(frq_x, frq_y, oct\right)$ is a $2,048 \times 2,048$ noise texture parameterized by frequency parameters in image X- and Y-directions, frq_x, frq_y, and the number of octaves, oct, and t is the threshold value. $\boldsymbol{P}_{\mathrm{c}}^{-}$ is a map with resolution $2,048 \times 2,048$ that represents the geometric profile of the generated concrete damage. The profile consists of the intact surface specified as the value t, and the damaged regions where the geometry is expressed by the offset from the value t. The map $\boldsymbol{P}_{\mathrm{c}}^{-}$ can be converted to the displacement map, $\boldsymbol{P}_{\mathrm{c}}$, that are applied to mesh surface to represent concrete damage geometry (bottom of the second column of Fig. 9) by

$$\boldsymbol{P}_{\mathbf{c}} = d \cdot \mathrm{normalize}\left(t - \boldsymbol{P}_{\mathbf{c}}^{-}\right) \tag{2}$$

where d is the maximum depth of the concrete damage, and $\mathrm{normalize}(\cdot)$ indicates the normalization operation to make the maximum value of the map one.

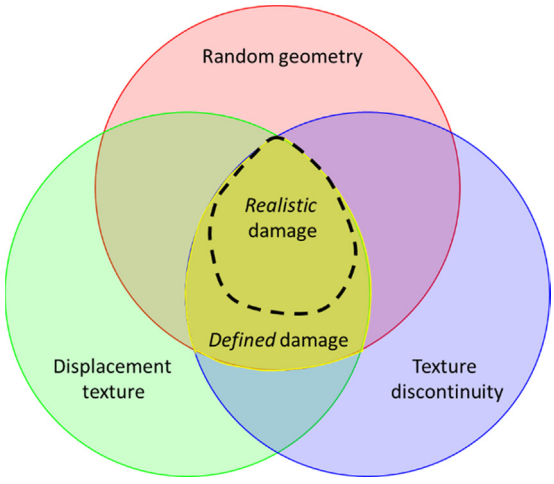


Fig. 8. Concrete damage texture generation concept.

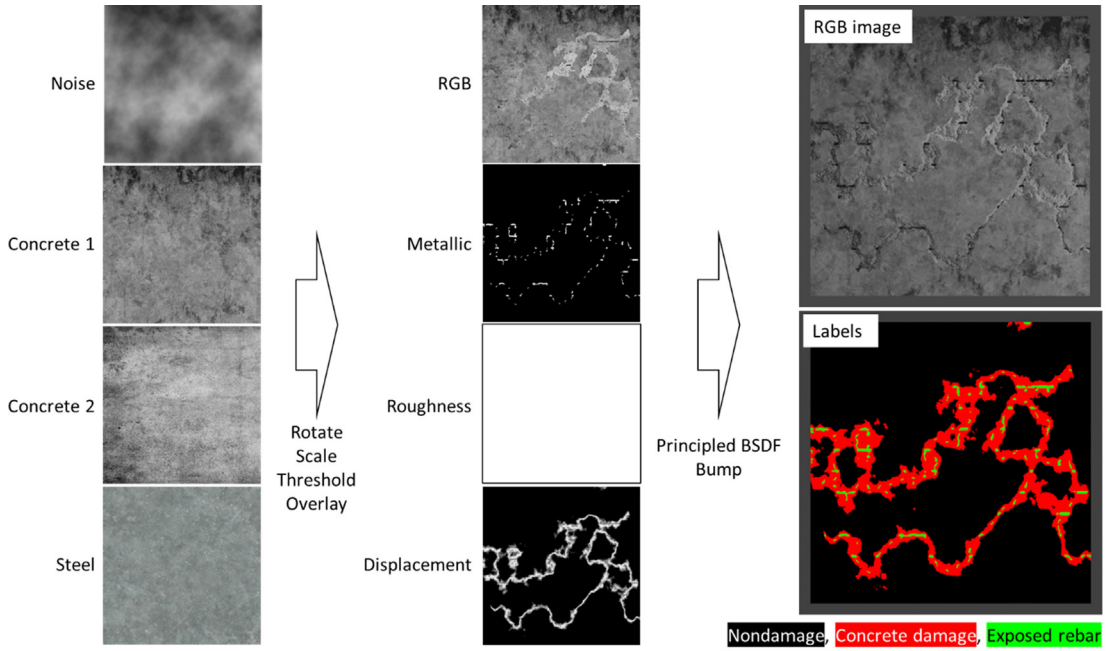


Fig. 9. Random damage texture generation procedure.

When the depth of the concrete damage is larger than the thickness of the cover concrete, rebar is exposed, affecting the structural assessment significantly. This research models the rebar following a concept similar to the one explained in Fig. 8: the rebar is modeled such that the range of its design parameters cover the corresponding range of the actual rebar. With this concept, the geometry of the rebar is modeled using cylinders with sinusoidal surface ripples (ripple amplitude is 10% of the cylinder radius). The cylinder radii of the axial rebar and stirrups are 1/200 and 1/300 of the texture image dimension, respectively. To create rebar shape profiles, the thickness of the cover concrete for axial rebar, c_a, is first sampled uniformly from the range [2.5 cm, 7.5 cm], referring to the typical values for the actual design of concrete columns [7,28]. Then, the cover thickness for the stirrups, c_s is determined by

$$c_s = c_a - 2 \text{ [cm]} \tag{3}$$

The intervals of the axial rebar and stirrups are sampled uniformly from the three and ten times the corresponding diameters. The shape profiles of the rebar are created based on these parameters and, when the concrete damage depth exceeds the depth of the rebar surface, the rebar shape profiles are used in place of the damaged concrete profile. The bottom row of the second column of Fig. 9 shows the displacement profile after incorporating the rebar profile.

With the procedure discussed in this section, masks of the non-damage, concrete damage, and exposed rebar can be obtained. The RGB image and the metallic map used for the Blender principled BSDF block can be generated using the masks as follows:

(RGB-1) Two images are selected randomly from the set of concrete images, corresponding to the concrete texture in the damaged and undamaged regions.
(RGB-2) A single image is selected randomly from the set of steel images, corresponding to the exposed rebar.
(RGB-3) The selected images are overlaid using the masks of nondamaged, concrete damage, and exposed rebar.
(Metallic) Set metallic to 1.0 for the regions of the exposed rebar, and 0.0 for other regions.

The first and the second images of the second column of Fig. 9 show the RGB image and the metallic map generated for this example. The third image shows the roughness map, which is 1.0 everywhere. Once the RGB image, metallic map, roughness map, and displacement map are obtained, the damaged concrete texture can be defined and assigned following the steps discussed in Sections 2.3-2.5. An image rendered using a plate with the generated damaged concrete texture and the corresponding label map are shown in the right column of Fig. 9.

The threshold t and the maximum depth parameter d control the severity of the damage. The area of concrete damage increases as t increases, and the depth of the concrete damage increases as d increases, exposing more rebar. This research defines three damage severity categories (slight, medium, severe), and samples the values of those parameters using distributions defined for each category. Slight damage is designed such that the area of damage is between 0% and 1% of the total area (sampled uniformly). Once the ratio of the damaged area is determined, the corresponding value of t can be determined

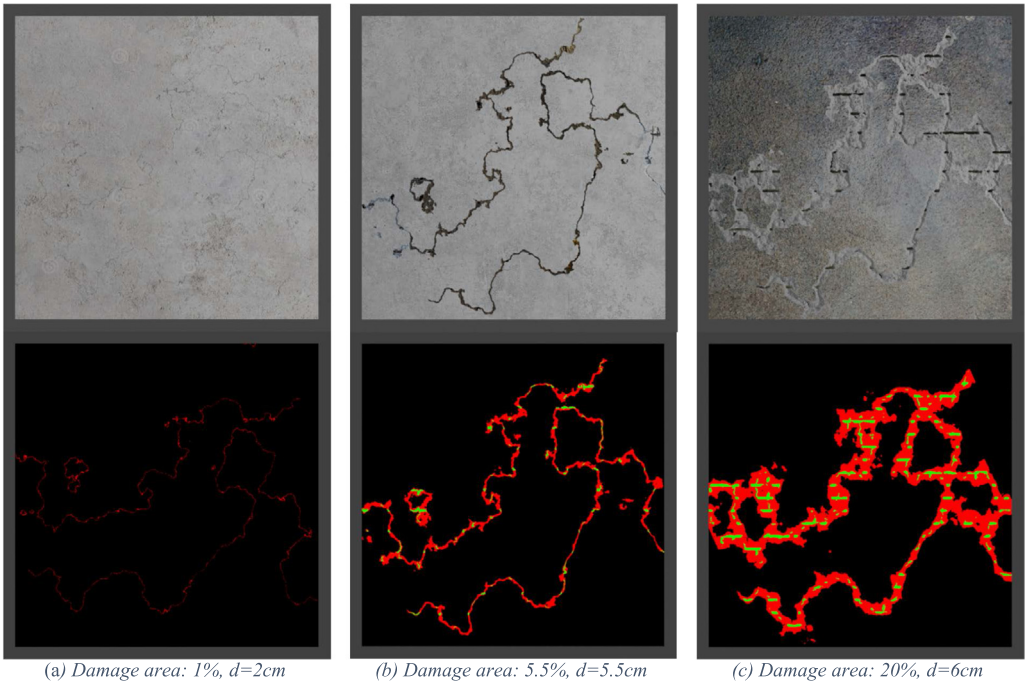
uniquely. The depth parameter d for the slight damage is sampled uniformly from the range [1 cm, 2 cm]. The range of the ratio of the damaged area for the medium damage is [90%, 99%], and the maximum depth is sampled from the range [1 cm, 10 cm]. For severe damage, the ratio of the damaged area is sampled from the range [70%, 90%], and the maximum depth is sampled from [2 cm, 10 cm]. Example damage textures generated using the same noise texture are shown in Fig. 10 for each of the slight, medium, and severe categories.

Following the steps described in this section, 1,000 damage textures are generated for each of the slight, medium, and severe categories. The textures are assigned to the viaduct columns by the following rule: (i) a column is not damaged with probability 0.1 (texturing following the steps discussed in the Section 3.2), (ii) a column is damaged with probability 0.9, (iii) for a damaged column, the damage category is chosen from the three categories with equal probability, (iv) once the damage category is determined, a damage texture is selected from the 1,000 textures generated by the steps discussed in this section, and (v) the selected texture is assigned to the column using the Blender Principled BSDF node and the Bump node, with texture scale sampled uniformly from the range [0.2, 1.0].

3.4. Synthetic data generation

A synthetic dataset for vision-based structural condition assessment of Japanese high-speed railway bridges, termed Tokaido dataset in this research, is created using 200 synthetic environments, each of which contains 10 viaducts generated randomly by the steps discussed in this section. For each environment, 100 synthetic cameras are defined by sampling focal length uniformly from the range [15 mm, 55 mm], and setting the sensor size and camera resolutions to 36 mm and $1,920 \times 1,080$, respectively. The location of each camera is sampled uniformly from the space defined by the distance from the viaduct centerline, D, and height range $[h_0, h_1]$: $D = 10\text{m}, h_0 = 1\text{m}, h_1 = 10\text{m}$ for 50 cameras, and $D = 5\text{m}, h_0 = 1\text{m}, h_1 = 6.5\text{m}$ for the remaining 50 cameras. With this parameter setting, the first 50 cameras take images of the global structures as well as close-up images of the structural components ("regular images"), while the remaining 50 cameras take more close-up images of viaduct columns ("close-up images"). Camera rotations (in XYZ Euler angles used by default by the Blender software) are also sampled uniformly, such that the rotation about X and Y axes are in the ranges $[\pi/4, 3\pi/4]$ and $[-\pi/12, \pi/12]$, respectively. The rotation about Z axis is sampled from the range $[-\pi/2, \pi/2]$, measured from the direction that points to the viaduct. Once the RGB image is rendered, associated ground truth structural component label map, damage label map, and depth map are collected.

This process generates a dataset of 10,000 regular images and the same number of close-up images, as well as the associated ground truth maps. This raw dataset needs to be post-processed to remove unrealistic images: a camera may be placed too close to an object, or even inside the mesh of an object. This research applies the following criteria to filter out such unwanted images: (i) An image is removed when the average pixel intensity is less than 30, (ii) an image is removed if the image contains a single structural component label only, and (iii) an image is removed when the minimum depth from



(a) *Damage area: 1%, d=2cm* (b) *Damage area: 5.5%, d=5.5cm* (c) *Damage area: 20%, d=6cm*

Fig. 10. Example images and labels of damage for different severity levels. (a) Slight, (b) Medium, and (c) Severe.

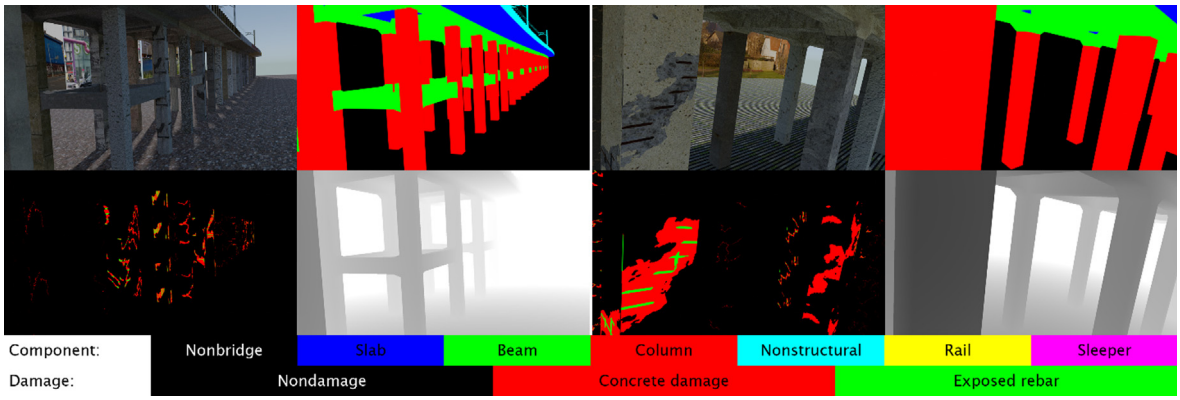


Fig. 11. Additional example images (top left), structural component label maps (top right), damage label maps (bottom left), and depth maps (bottom right) from the Tokado dataset.

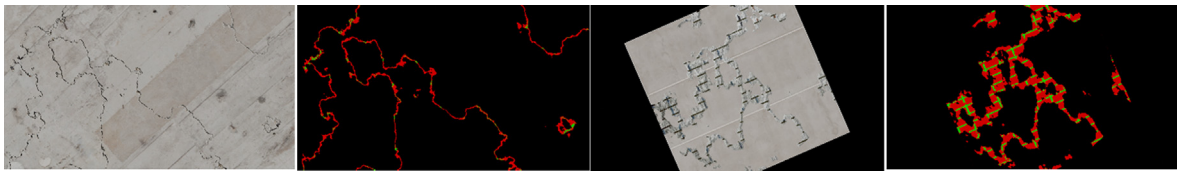


Fig. 12. Example pure texture images and associated labels.

the camera is not larger than 50 cm. After the image removal, the dataset contains 8,648 regular images, 7,288 close-up images, and the associated ground truth maps. Examples of the RGB images, structural component label maps, damage label maps, and depth maps are shown in Fig. 1 and Fig. 11.

In addition to the regular and close-up images, this research generates 3,000 images of damaged RC plates to further augment training data for damage recognition. The damage textures are obtained by the steps discussed in Section 3.3, and the textures are applied to the plates without additional scaling. The plates are then rotated randomly, and photographed using synthetic cameras placed at random distances. The 3,000 images thus generated are termed "pure texture" images in this research. Examples of the pure texture images are shown in Fig. 12. The composition of the Tokaido dataset is summarized in Table 5.

4. Data analysis

This section demonstrates the effectiveness of the Tokaido dataset for training visual recognition algorithms that perform three tasks of the vision-based structural assessment of railway viaducts: recognition of structural components, recognition of damage, and localization of the target structural components by depth estimation. The first two tasks are implemented using a semantic segmentation algorithm, fully convolutional network (FCN) [36], that estimates label maps, instead of image-wise labels. The FCN is selected in this research due to the simplicity of implementation and the successful recent applications to civil engineering problems [22,42]. The depth map estimation for the localization of the structural component is implemented using the FCN with reverse Huber loss function [34,45]. Tensorflow 2 [1] is used to implement the machine learning models discussed in this section.

4.1. Fully convolutional network architecture

A network architecture with 58 convolutional layers, termed FCN58, is used in this research. The FCN58 architecture is an extension of the FCN45 architecture used in [43] to recognize bridge components in image data automatically. The FCN45 architecture is a relatively compact network, compared to general visual recognition networks that deal with significantly larger number of classes (e.g. VGG16 [56] and ResNet-50 [20] architectures trained to solve the 1,000-class ImageNet classification problem [55]. The FCN45 network also distributes the parameters relatively evenly along the network depth to perform recognition of objects of different sizes. On the other hand, the FCN45 architecture is updated in this research to (i) accommodate the larger input image size ($1,920 \times 1,080$, compared to up to 320×320 images processed by the FCN45 networks), (ii) reduce the effect of overfitting by using depth-wise separable convolution [12] in the layers with large number of

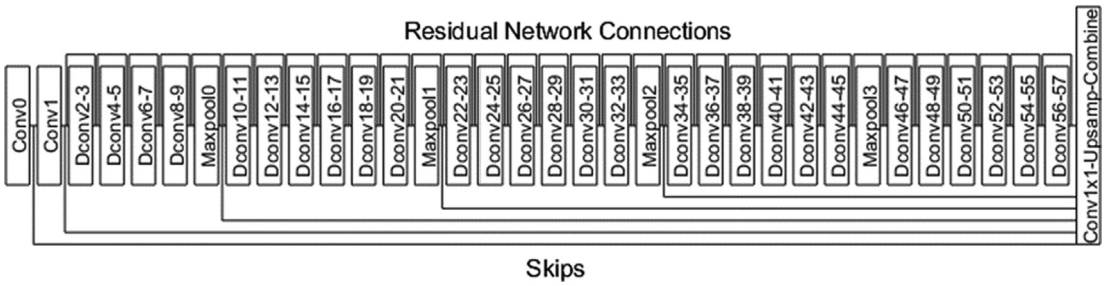


Fig. 13. Illustration of FCN58 architecture. Conv: Ordinary convolutional layer, Dconv: Depthwise separable convolutional layer.

Table 5
Composition of the Tokaido dataset.

Number of images	Number of environments	Number of viaducts	Structural component labels	Damage labels	Depth range
8,648 (regular) 7,288 (close-up) 3,000 (pure texture)	200	2,000	Nonbridge, Columns, Beams, Slabs, Rails, Sleepers, Other nonstructural components	No Damage Concrete damage Exposed rebar	0.5 m ~ 30 m

Table 6
Detail of the FCN58 architecture (Parameter names follow the Tensorflow convension).

Layer name	Filters	Kernel size	Strides
Conv0	16	7×7	3×3
Conv1	64	7×7	2×2
Dconv2-9	64	3×3	1×1
Maxpool0	–	2×2	2×2
Dconv10-21	128	3×3	1×1
Maxpool1	–	2×2	2×2
Dconv22-33	128	3×3	1×1
Maxpool2	–	2×2	2×2
Dconv34-45	128	3×3	1×1
Maxpool3	–	2×2	2×2
Dconv46-57	128	3×3	1×1

filters. The depth-wise separable convolution first applies a 2D spatial convolution to each channel separately (depth-wise convolution), and then applies a 3D convolution with kernel size 1×1 that mixes the information contained in different channels (point-wise convolution), reducing the number of parameters and computational cost significantly. Detailed discussions about the effectiveness of the depth-wise separable convolution for object detection and semantic segmentation tasks can be found in [25,50].

The FCN58 architecture used in this research is illustrated in Fig. 13, and the detail of each layer is provided in Table 6. The first two layers apply ordinary convolutions, reducing the feature dimensions from $1,920 \times 1.080$ to 640×360 and 320×180, respectively. Then, 56 depth-wise separable convolutional layers are applied. Each convolutional layer consists of a convolution operation, followed by batch normalization [29] and ReLu activation function. To train the deep networks effectively, residual network connections are implemented, as specified in Fig. 13. The features at multiple scales (layer outputs from Conv0, Conv1, and Maxpool0-2) are then extracted and integrated using bilinear up-sampling and skip connections as described in [36]. With this architecture, the networks output label maps or depth maps of the resolution 640×360.

4.2. Dataset considerations

The Tokaido dataset is split into training, validation, and testing sets as specified in Table 7. To ensure the uniqueness of the testing images from those used during training process, data from 25 environments out of 200 environments of the Tokaido dataset are used as the testing set. The data from the remaining 175 environments is mixed, shuffled, and further

Table 7
Dataset specifications for the training, validation, and testing of FCNs.

Tasks	Training		Validation	Testing	
	Real-world*	Synthetic	Synthetic	Real-world	Synthetic
Component recognition	51	7,275	300	50	1,073
Damage recognition	61 + 101 (262)	6,781	300	60 (165)	909
Depth estimation	–	7,275	300	–	1,073

*Numbers in parentheses indicate the numbers of cropped images.

divided into training set and small validation set. The label maps and the depth maps are resized to 640 × 360 to accommodate the FCN58 networks.

The FCNs for the structural component recognition and depth map estimation need to be trained for both global views of the structures and close-up views of structural components. For that purpose, regular images are used for those tasks, and are split into training set (7,275 images), validation set (300 images), and testing set (1,073 images).

The FCN for damage recognition is trained more effectively for close-up images of structural surfaces. The preliminary analysis revealed that using all regular and close-up images and the associated damage labels leads to large errors. Concrete damage and exposed rebar often exhibit thin and subtle patterns, which are hardly discernable when the camera is not placed nearby. On the other hand, ground truth damage labels are always available, even when the image resolutions are not enough to characterize the damage fully. The labels of far damage therefore confuse the network during training process, causing error to the recognition results. To address the problem, this research uses the ground truth depth map to identify the image regions of more than 1.5 pixels per centimeter (pixel/cm), and discards the damage labels outside of those regions. After applying this pre-processing to both regular and close-up images and removing images that do not contain damage in the specified regions, the number of images reduces to 4,990 (4,381 images for training and validation, 609 images for testing). To supplement the reduced amount of data, this research uses pure texture images (2,700 images for training and validation, 300 images for testing) in combination with the regular and close-up images. Similar to the structural component recognition and depth map estimation, the validation set is created by selecting 300 images randomly.

This research combines small amount of real-world data with the synthetic Tokaido dataset for the structural component recognition and damage recognition tasks. The image data for the structural component recognition is downloaded from the Google Street View by manually selecting viewpoints that result in images of the Tokaido shinkansen viaducts with the standard design. The collected images are labelled manually using Labelbox [33], a simple and user-friendly online tool for the image annotation for semantic segmentation tasks. Structural components that are not modeled in the Tokaido dataset are labelled as either "Other structural" (e.g. braces installed to improve the seismic resistance) or "Other nonstructural" (e.g. plumbing), and are not focused in the subsequent part of this paper. The real-world data for structural component recognition consists of 51 images for training and 50 images for testing, with the associated labels, as shown in Table 7.

Real-world images for damage recognition are collected from two sources: (i) images of general RC damage downloaded from the internet, and (ii) images of damaged RC viaducts collected by Professor Yoshikazu Takahashi at the Kyoto University after the 2011 Tohoku Earthquake [58]. The viaducts in the set (ii) have the structures similar to the ones with the standard design. The collected images are labelled manually using the Labelbox. In addition to the data thus created, the training set includes 101 real-world images from the structural component recognition task with "No Damage" label assigned to everywhere ("+101" in Table 7). When the image size is not consistent with those of the Tokaido dataset, the image is rescaled, and multiple sub-images are cropped to cover the entire region of the original image (numbers in parentheses in Table 7). An appropriate real-world image dataset for depth estimation for bridge component localization is not available currently, and therefore only synthetic data is used to test the depth estimation network as a preliminary investigation. Note that the numbers of real-world images are significantly smaller than the numbers of images used in the existing literature, and therefore not enough to obtain reliable results.

4.3. Network training schemes

When the networks are trained, training images, both real-world and synthetic, are mixed, shuffled, and flipped (left–right) randomly. Then, mini batches of 8 data samples are created and fed into the network to update the network weights using Adam optimizer [31]. Median frequency balancing technique [13] is used to reduce the effect of data imbalances for the structural component recognition and damage recognition tasks. For all tasks, weight decay of 0.0001 is applied to reduce the overfitting effect.

The number of training iterations are specified in terms of epochs, that is, how many times the networks see the entire training data. Each epoch contains 915, 880, and 909 iterations for structural component recognition, damage recognition, and depth map estimation tasks, respectively. The networks are first trained with the learning rates of 0.001 for 200 epochs, and then for 50 epochs with the learning rate of 0.0001 and 10 epochs with the learning rate of 0.00001. The accuracy and loss function on the validation set are evaluated after every epoch to monitor the training progress.

4.4. Structural component recognition results

The FCN58 network is trained for the structural component recognition task using the data described in Section 4.2 and the training scheme discussed in 4.3. The performance of the trained network on the testing set is evaluated by taking argmax of the softmax probability re-weighted by the inverse of the median frequency balancing weights [38]. Three performance criteria are evaluated and presented in Table 8: precision (TP/(TP + FP)), recall (TP/(TP + FN)), and intersection

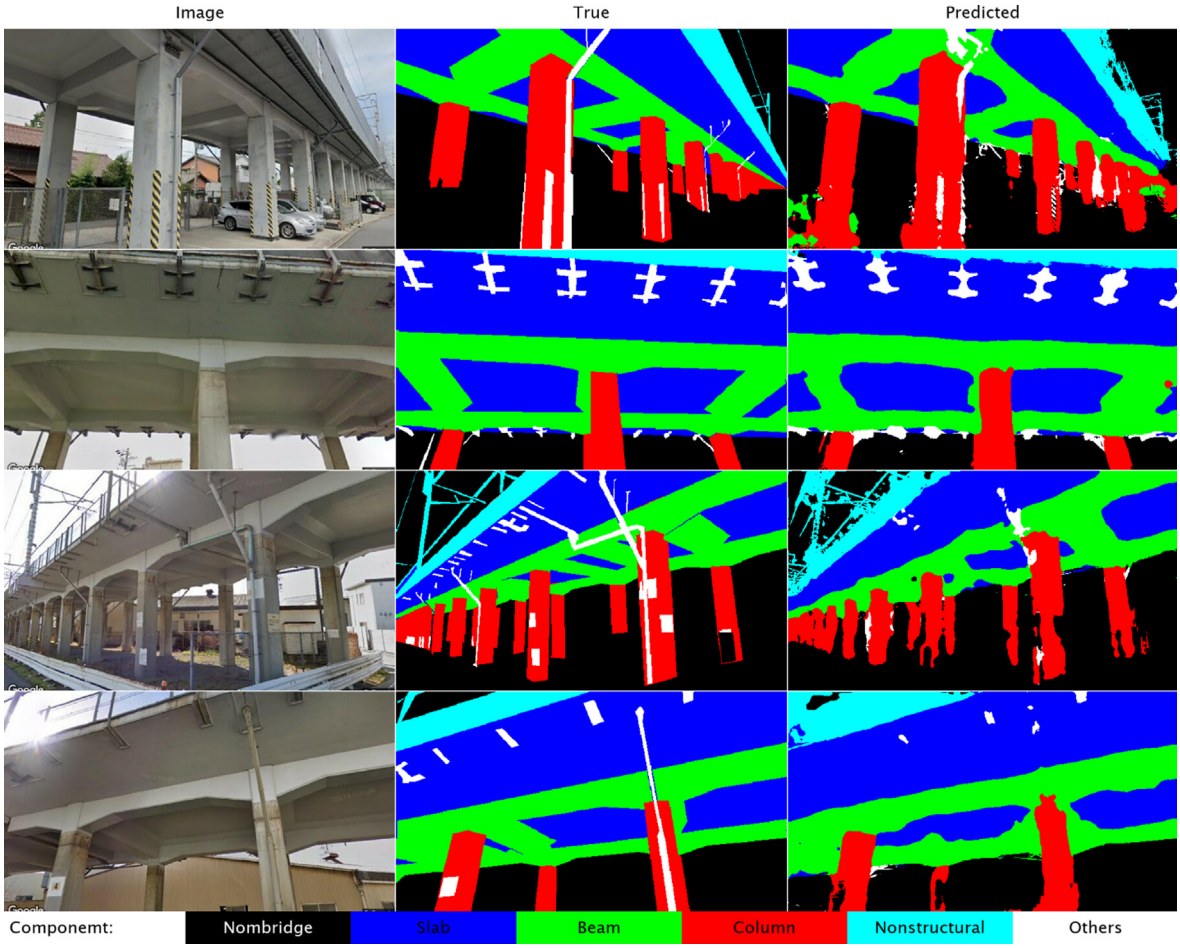


Fig. 14. Example structural component recognition results (real-world images).

Table 8
Network performance on testing set [%] (Structural component recognition).

	Synthetic images			Real-world images		
	Precision	Recall	IoU	Precision	Recall	IoU
No Bridge	98.8	99.4	98.2	89.6	93.3	84.2
Slab	96.1	94.6	91.1	85.2	86.7	75.3
Beam	94.0	93.9	88.6	84.1	82.5	71.3
Column	96.6	97.6	94.4	84.9	90.5	78.0
Nonstructural	97.5	91.7	89.5	88.0	66.3	60.8
Rail	95.2	91.6	87.6	NA	NA	NA
Sleeper	84.2	75.3	66.0	NA	NA	NA
Others	NA	NA	NA	55.4	30.2	24.3
Mean*	94.6	92.0	87.9	86.4	83.9	73.9

* "Others" class is not included.

over union (IoU, TP/(TP + FP + FN)), where TP, FP, and FN are the numbers of true positive, false positive, and false negative pixels. The performance for the "Others" class for the synthetic data and the performance for the "Rail" and "Sleeper" classes for the real-world data are not provided because the dataset does not contain such pixels. The performance for the "Others" class for the real-world data is provided, but is not focused in the subsequent discussions because those objects are not modeled in the synthetic environment (Section 4.2). The low performance for this class indicates that real-world images in the training set alone are not enough to train networks that can produce accurate results.

The performance of the network on the synthetic data is high, including more than 90% of IoU for slabs and columns, as well as nearly 90% of IoU for beams, nonstructural components, and rails. The results indicate that the training using 175 synthetic environments can be generalized well to new synthetic environments with different viaduct trajectories, dimensions, and textures. The performance for the sleepers tends to be lower than that of other components, which could be explained by the fact that the dataset contains more images of substructures than those of rails and sleepers (camera is placed in the height range of [1 m, 10 m]).

The results of real-world images show that the training process can take advantage of the rich synthetic data to improve the network performance significantly. Example structural component recognition results are presented in Fig. 14, with the corresponding image-wise performance criteria provided in Table 9. The trained network can capture the global structures, as well as fine detail of the components (e.g. wires and poles in the first and third images). On the other hand, the ground truth labels created manually follow the rules of the KITTI dataset when the structural components can be seen through objects, such as fences and transparent objects (e.g. the leftmost column in the first image): labels of the closest objects are assigned even when structural components are visible behind them. Those situations, as well as components categorized into "Others" class, can be modeled in the synthetic environment to further improve the accuracy.

4.5. Damage recognition results

The FCN58 network is trained for the damage recognition task using the data described in Section 4.2 and the training scheme discussed in 4.3. The performance of the trained network on the testing set is shown in Table 10. During the evaluation, argmax of the softmax probability is computed without re-weighting, because the approach produces better results of detecting damage that has rare probability of occurrence ("Concrete Damage" and "Exposed Rebar" classes constitute 3.9% and 0.4% of the total number of pixels, respectively). The recall values for the synthetic testing images show that the Tokaido dataset is effective for training networks that can detect concrete damage and exposed rebar, even when the input images are complex, containing multiple structural components in general scenes. On the other hand, precision and IoU values are relatively low, indicating that accurate pixel-level localization of the structural damage is challenging: concrete damage and exposed rebar are often thin and subtle in images, as discussed during dataset considerations.

The network performance for the real-world images, example damage recognition results, and the corresponding image-wise performance criteria are shown in Table 10, Fig. 15, and Table 11, respectively. The first three images in Fig. 15 demonstrate the effectiveness of the Tokaido dataset and the trained network to detect and localize concrete damage and exposed rebar in various sizes, shapes, and contexts. The fourth image shows both potential and challenge. While the network returns reasonable results partially, the overall accuracy of damage recognition for this image is not as high as other images. Two types of errors can be observed herein: (i) deformed rebar and missing core concrete are not labeled accurately (right col-

Table 9
Image-wise performance criteria of example structural component recognition results.

Image	Slab			Beam			Column			Nonstructural		
	PR	REC	IoU	PR	REC	IoU	PR	REC	IoU	PR	REC	IoU
1	90.3	80.3	74.0	83.1	88.1	74.7	71.3	91.3	66.7	60.5	73.7	49.7
2	95.9	92.6	89.1	92.2	94.5	87.6	89.0	96.4	86.1	94.9	87.3	83.3
3	84.8	93.0	79.7	86.6	89.6	78.7	84.0	80.8	70.0	64.2	90.1	60.0
4	93.7	92.5	87.0	86.2	93.3	81.2	78.2	85.2	68.8	77.0	94.5	73.8

PR: Precision, REC: Recall, IoU: Intersection over Union.

Table 10
Network performance on testing set [%] (Damage recognition).

	Synthetic			Real-world		
	Precision	Recall	IoU	Precision	Recall	IoU
No Damage	99.6	97.9	97.5	94.2	95.6	90.3
Concrete Damage	59.5	84.1	53.4	65.1	63.0	47.1
Exposed Rebar	56.2	83.6	50.6	49.4	35.3	25.9
Mean	71.8	88.6	67.2	69.5	64.6	54.4

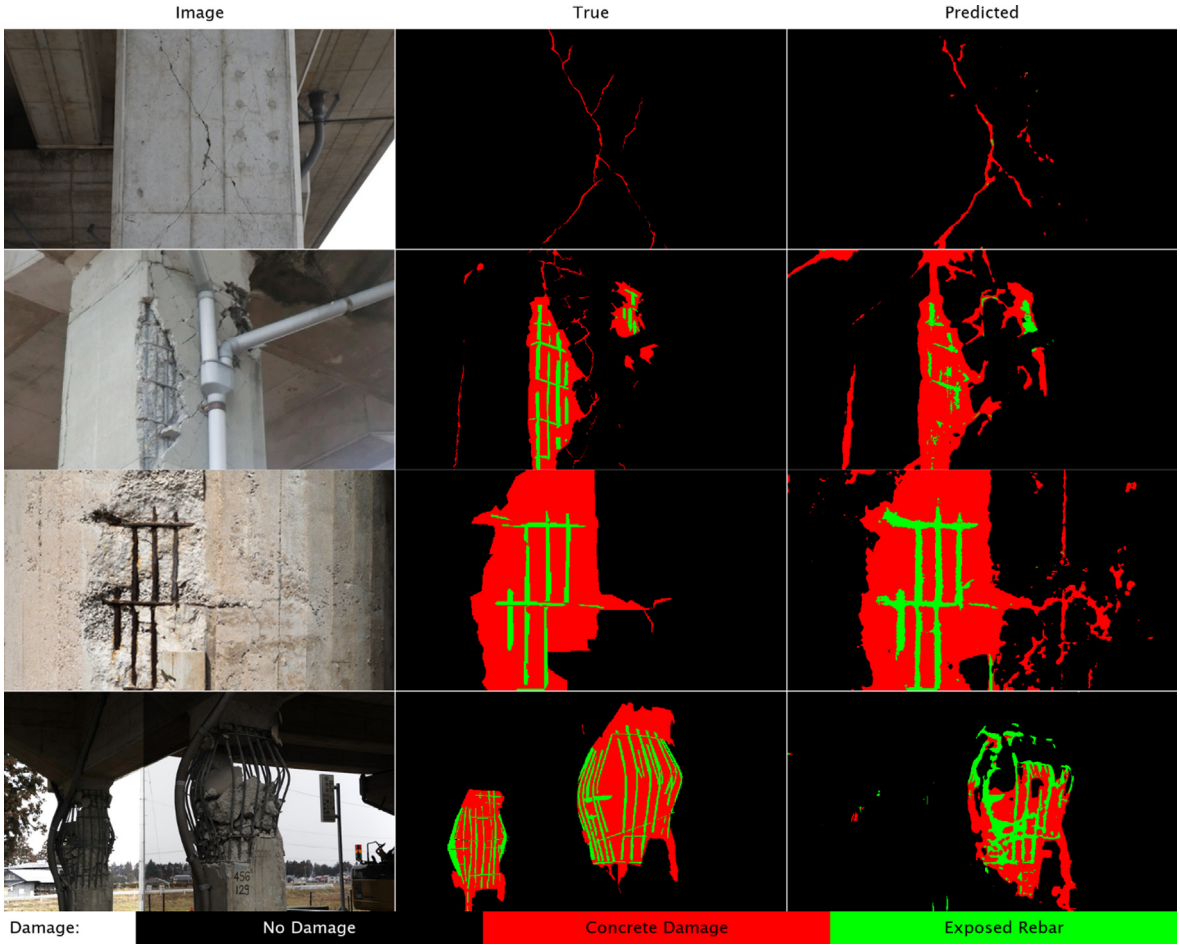


Fig. 15. Example damage recognition results (real-world images fom [58]).

Table 11
Image-wise performance criteria of example damage recognition results.

Image	Concrete Damage			Exposed Rebar		
	PR	REC	IoU	PR	REC	IoU
1	31.9	65.6	27.3	NA	NA	NA
2	53.0	85.1	48.5	52.2	29.2	23.1
3	73.8	90.6	68.5	55.6	97.2	54.7
4	62.3	29.5	25.1	43.1	35.1	24.0

PR: Precision, REC: Recall, IoU: Intersection over Union.

umn), and (ii) far damage is not recognized at all (left column). The first type of error is caused because synthetic dataset does not contain such images, and the number of real-world images that contain such patterns is not enough to perform reliable recognition (only a few images of such damage are included in the 61 real-world images used for training). The second type of error is caused by the pre-processing of the synthetic dataset: far damage labels were removed before using the data to train FCNs. This type of error leads to the reduced recall values in Table 10.

The first type of error can be reduced by either increasing real-world training images or modeling additional phenomena, such as deformed rebar and core concrete removal, in the synthetic environment. Algorithms that implement attention mechanisms can also be investigated to improve the accuracy for smaller or thinner damage, as well as near and large damage (e.g. multi-scale attention masks can be aggregated by hierarchical approach, as proposed in [59]). The second type of error can be controlled by combining the damage recognition results with the depth estimation results, which is discussed in the next sections.

4.6. Depth map estimation results

The FCN58 network for depth map estimation is trained using the reverse Huber loss function and training scheme described in Section 4.3. This research evaluates the trained network using three criteria used in existing literature about depth map prediction (e.g. [14,34,45,64]): (i) mean absolute error (MAE), (ii) root mean squared error (RMSE), and (iii) absolute relative distance (ARD). Using the depth estimation error at each pixel, e_i, and the true depth at the corresponding pixel, $d_{\text{true}i}$, those criteria can be computed by (i) MAE $:= \text{mean}_i(|e_i|)$, (ii) RMSE $:= \sqrt{\text{mean}_i(e_i^2)}$, and (iii) ARD $:= \text{mean}_i(|e_i/d_{\text{true}i}|)$. The performance evaluation criteria computed using the testing set is provided in Table 12, showing that the algorithm can localize structural components of interest. Example depth map estimation results are shown in Fig. 16 with the corre-

Table 12
Performance evaluation criteria of the trained depth map estimation network.

MAE [m]	RMSE [m]	ARD [%]
1.20	1.85	10.02

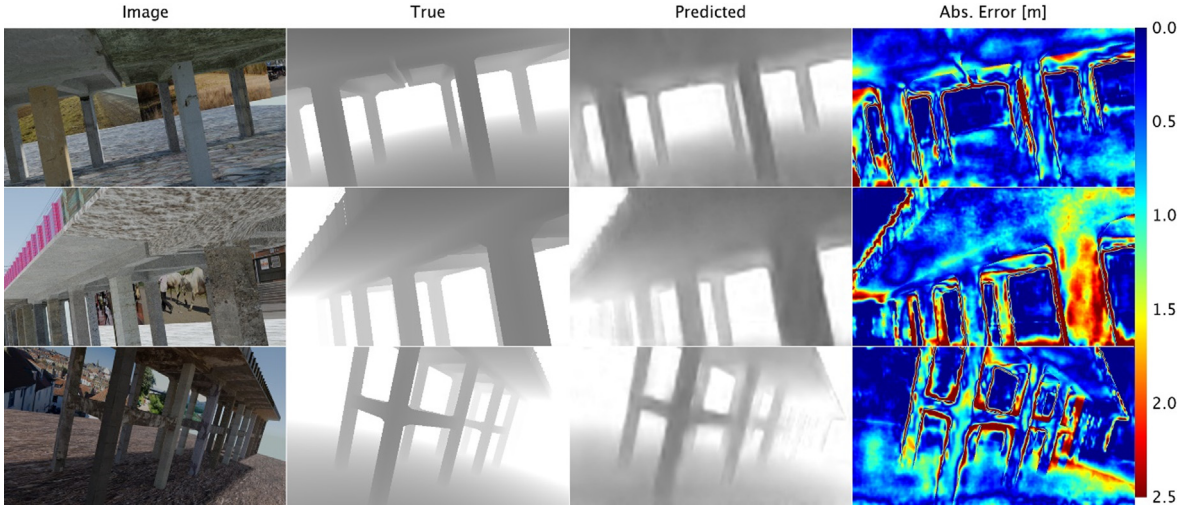


Fig. 16. Example monocular depth map estimation results.

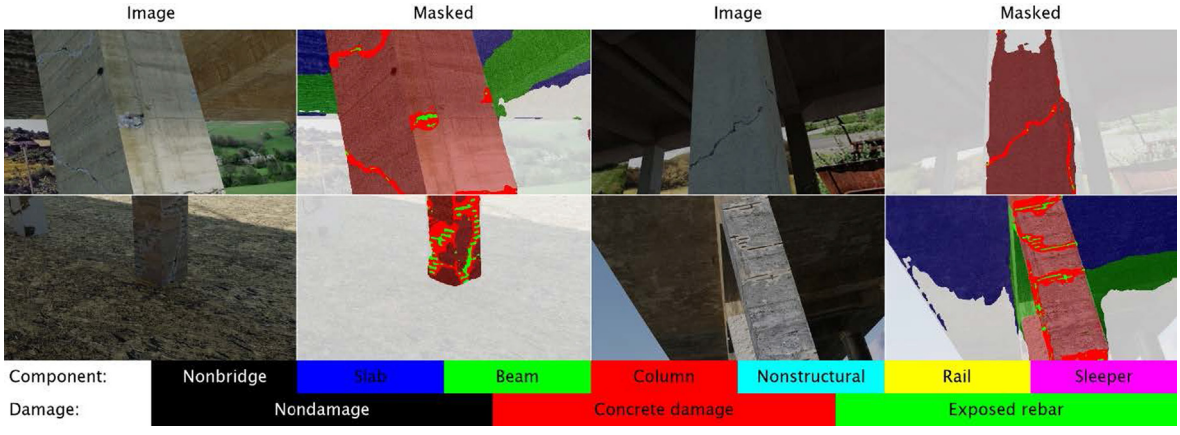


Fig. 17. Example structural condition assessment results. Far or non-bridge regions are grayed out.

sponding maps of absolute estimation error. The example results indicate that the localization tends to be successful for structural components near the camera, while the accuracy decreases for far or partially occluded components, as well as image regions near object boundaries. These observations are consistent with results reported by the existing literature about general depth estimation problems (e.g. [34].

4.7. Structural condition assessment of high-speed railway viaducts

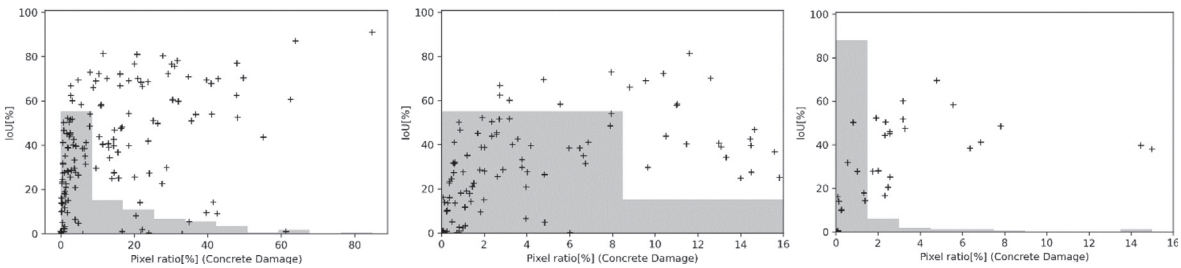
The results of structural component recognition, damage recognition, and depth map estimation can be combined to perform automated structural condition assessment of the high-speed railway viaducts. First, damage labels are obtained by applying the damage recognition network. Because this network is trained using close-up images, image regions that are expected to produce accurate results (regions of reliable damage recognition, or RRDRs) are then identified using the predicted depth map. The damage recognition results in the RRDRs are further processed by combining the structural component recognition results: the damage is categorized based on the structural component types, and at the same time predicted damage in the non-bridge region is removed. This process is demonstrated in Fig. 17 for synthetic testing images, where the RRDRs are defined as image regions where the resolution is 1.5 pixel/cm or more. The example results show the effectiveness of the proposed approach to extract information relevant to the structural conditions of railway viaducts.

Performance improvement for damage recognition is also expected by specifying RRDRs derived using the predicted depth map. The performance criteria presented in Table 10 are evaluated again in Table 13, using labels in the RRDRs only. Here, the RRDRs are defined to have the predicted depth between 0.5 m and 20 m (camera focal lengths are not available, and therefore the threshold cannot be applied in the form of pixel/cm). Note that predicted depth of real-world images is not accurate, because the network is not trained for the structures contained in the images. The predicted depth map indicates whether the network "thinks" that each part of image is close to the camera or not. With this post-processing, the recall value for concrete damage improves significantly, approaching the levels presented in Table 10 for synthetic images.

To further explain the results presented in Table 13, image-wise IoUs are plotted against pixel ratio (number of damage pixels divided by total number of pixels in an image) in Fig. 18. By considering RRDRs only, damage recognition results with relatively low accuracy for each range of the pixel ratio can be removed automatically. The apparent decrease of some of the performance criteria after post-processing observed in Table 13 is caused by larger proportion of images with small pixel ratios, where accurate recognition is challenging inherently. This is partly because of imperfect depth estimation, which, for example, recognizes that the damage in the second image in Fig. 15 is in a far surface, and therefore not inside the RRDR. Large damage recognized with high accuracy in Fig. 18(a) can be recovered by improving the depth estimation network, which is part of the future work. With the improved depth estimation network, the post-processing using the RRDRs can be an effective way to control the second type of error discussed in Section 4.5.

Table 13
Network performance on testing set [%] (Damage recognition, after post-processing).

	Real-world		
	Precision	Recall	IoU
No Damage	99.8	95.9	95.4
Concrete Damage	11.3	74.6	10.9
Exposed Rebar	13.8	20.2	8.9



(a) Before post-processing (b) Before post-processing (zoomed) (c) After post-processing

Fig. 18. Image-wise IoU for concrete damage before and after post-processing using regions of reliable damage recognition. Histogram shows proportion of testing images with the given range of concrete damage pixel ratio.

5. Discussions

The results obtained in Section 4 show the effectiveness of the developed synthetic environments to train reliable deep FCN-based algorithms for structural component recognition, damage recognition, and monocular depth map estimation. The potential of the proposed approach can be further clarified by referring to the previous research about bridge component recognition using semantic segmentation[42], which reports less than 50% IoU for columns (accurate localization was challenging because of their slender shapes), and less than 70% IoU for the beams and slabs. Three factors contributed to the significantly higher recognition performance obtained in this research: (i) large number of training images, (ii) highly accurate annotation, and (iii) clearer scope of the target structure. Obtaining datasets that satisfy (i) and (ii) is straightforward by using the synthetic environment. The factor (iii) needs additional considerations when the approach is applied to problems that are not modeled explicitly in this research. The networks trained using the Tokaido dataset alone are likely to be confused by component, damage, and depth patterns that do not exist in the dataset, e.g. components of bridges not designed by the standard design, deformed rebar, and debris. Depending on the application scenarios, the dataset should be combined with additional data that is highly relevant to the problem at hand.

Another advantage of the proposed approach is that images for different structural condition assessment tasks can be sampled consistently from synthetic environments that are relevant to the desired application scenario. Compared with existing approaches that develop different visual recognition systems independently, the systems developed for this research can be integrated in an efficient and straightforward manner (Section 4.7). The unified system for the structural condition assessment can parse complex images using the masks of structural components and damage, as well as the regions of reliable damage recognition at high accuracy.

With the synthetic environment and the Tokaido dataset, various state-of-the-art algorithms for general semantic segmentation and depth recognition problems, such as SegNet [2], U-net [53], and its successors [26,68], need to be investigated in the future to improve the structural condition assessment accuracy. Another area of extending this work is the investigation of advanced methods for correcting bias of the synthetic datasets, such as transfer learning and sym2real approaches [30].

Future extensions of the proposed approach include the application to long-term structural health monitoring (SHM) scenarios. Compared to the post-earthquake structural assessment where the detection and quantification of severe damage are important, finding damage at early stages is more important for the long term SHM scenarios. The results presented in this research suggest that accurate recognition and quantification of subtle damage (thin cracks) are not straightforward even with the large-scale synthetic dataset. To address the challenge, the data collection scheme in the synthetic environment needs to be reformulated: (i) images should be collected from closer distances, (ii) higher image resolutions should be used, and (iii) severe damage should be excluded because such damage causes data imbalances.

Finally, the availability of the large-scale synthetic dataset does not affect the paramount importance of real-world data. When the prototype systems are applied in the field environment, the recognition performance needs to be improved by fine-tuning the networks using large amount of real-world data. The importance of the synthetic environment and the Tokaido dataset resides in the ease of getting baseline performance that enables such fine-tuning, as well as the investigation of larger frameworks that are based on the semantic segmentation and depth prediction capabilities discussed in this research.

6. Conclusions

This research investigated a unified system for the automated vision-based structural condition assessment that can identify and localize critical structural components, and then detect and localize structural damage to those components. To address the challenge of data scarcity, this research proposed to leverage synthetic environments that can generate target structures and damage scenarios randomly. The approach was illustrated for the structural condition assessment of reinforced concrete railway viaducts for a Japanese high-speed railway line (the Tokaido Shinkansen). This research developed 200 different synthetic environments that contain 2,000 railway viaducts with the standard design. The synthetic environments are used to produce a dataset of 8,648 images for structural component recognition and depth estimation, as well as 7,990 images for damage recognition (the Tokaido dataset). The image resolution is $1,920 \times 1,080$, and each image is associated with ground truth pixel-wise information of structural component types, damage types, the depth values. The developed dataset was validated for three tasks: structural component recognition, damage recognition, and depth map estimation. For each task, the effectiveness of the developed dataset was demonstrated by training fully convolutional networks with 58 convolutional layers using both the Tokaido dataset and small number of real-world images. For the structural component recognition, the trained network achieved 87.9% of IoU for synthetic testing images, as well as 73.9% of IoU for the real-world testing images. The trained damage recognition network can detect concrete damage (Recall: 84.1% for synthetic images, 63.0% for real-world images without post-processing, and 74.6% for real-world images with post-processing), as well as exposed rebar (Recall: 83.6% for synthetic images, 35.3% for real-world images). The results of depth map estimation are also promising, with 1.20 m of mean absolute error, 1.85 m of root mean squared error, and 10.02% of the absolute relative distance in the range [0.5 m, 30 m]. Finally, the results of structural component recognition, damage recognition, and depth map estimation are combined to demonstrate automated structural condition assessment of high-speed railway via-

ducts. The proposed approach will facilitate the investigations of automated vision-based structural condition assessment, including those using the Tokaido dataset for the structural engineering community to be fully benefitted from the recent advance of the computer vision and data science field.

Future work includes (i) further improvement of recognition accuracy by investigating and comparing state-of-the-art semantic segmentation and depth prediction algorithms proposed for general visual recognition problems, (ii) incorporating structural analysis/mechanics to generate realistic synthetic damage textures, (iii) improving the accuracy and robustness of the networks for structural assessment by increasing the amount of real-world data, and (iv) investigating different data collection schemes to apply the approach for long-term structural health monitoring scenarios. These extensions will further facilitate the applications of visual recognition algorithms in autonomous structural inspection contexts, eventually leading to resilient society based on effective infrastructure management.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

The authors would like to acknowledge the financial support by the U.S. Army Corps of Engineers (Contract/Purchase Order No. W912HZ-17-2-0024). This research was also supported in part by the National Natural Science Foundation of China Grant No. 51978182. The authors would like to acknowledge Professor Yoshikazu Takahashi at the Kyoto University for providing images of damaged RC railway viaducts that were needed for this research.

References

[1] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., ... Zheng, X. (2016). TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems. Retrieved from http://arxiv.org/abs/1603.04467
[2] V. Badrinarayanan, A. Kendall, R. Cipolla, SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation, IEEE Trans. Pattern Anal. Mach. Intell. 39 (12) (2017) 2481–2495, https://doi.org/10.1109/TPAMI.2016.2644615.
[3] Blender. (n.d.). Retrieved from https://www.blender.org/
[4] Blender API Documentation. (n.d.). Retrieved September 3, 2019, from https://docs.blender.org/api/2.79/
[5] Blender Video Tutorials. (n.d.). Retrieved May 16, 2020, from http://www.littlewebhut.com/blender/
[6] Bloesch, M., Czarnowski, J., Clark, R., Leutenegger, S., & Davison, A. J. (2018). CodeSLAM-Learning a Compact, Optimisable Representation for Dense Visual SLAM.
[7] Building Code Requirements for Structural Concrete (ACI 318-14) Commentary on Building Code Requirements for Structural Concrete (ACI 318R-14) An ACI Standard and Report from IHS. (2014).
[8] Bulk Bing Image downloader. (n.d.). Retrieved May 16, 2020, from https://github.com/ostrolucky/Bulk-Bing-Image-downloader
[9] B. Burley, W. Disney, Physically-Based Shading at Disney, Retrieved from (2012), www.merl.com/brdf.
[10] Y.-J. Cha, W. Choi, O. Büyüköztürk, Deep Learning-Based Crack Damage Detection Using Convolutional Neural Networks: Deep learning-based crack damage detection using CNNs, Comput.-Aided Civ. Infrastruct. Eng. 32 (5) (2017) 361–378, https://doi.org/10.1111/mice:12263.
[11] Y.J. Cha, W. Choi, G. Suh, S. Mahmoudkhani, O. Büyüköztürk, Autonomous Structural Visual Inspection Using Region-Based Deep Learning for Detecting Multiple Damage Types, Comput.-Aided Civ. Infrastruct. Eng. 33 (9) (2018) 731–747, https://doi.org/10.1111/mice.12334.
[12] Chollet, F. (2017). Xception: Deep Learning with Depthwise Separable Convolutions.
[13] Eigen, D., & Fergus, R. (2016). Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. Proceedings of the IEEE International Conference on Computer Vision, 11-18-Dece, 2650–2658. https://doi.org/10.1109/ICCV.2015.304
[14] Fu, H., Gong, M., Wang, C., Batmanghelich, K., & Tao, D. (2018). Deep Ordinal Regression Network for Monocular Depth Estimation.
[15] S. Fujita, III. 電車線路設備, The Journal of the Institute of Electrical Engineers of Japan 102 (2) (1982) 113–114.
[16] Y. Gao, K.M. Mosalam, Deep Transfer Learning for Image-Based Structural Damage Recognition, Comput.-Aided Civ. Infrastruct. Eng. 00 (2018) 1–21.
[17] R. Girshick, Fast R-CNN, ArXiv Preprint (2015), ArXiv:1504.08083.
[18] R. Girshick, J. Donahue, T. Darrell, J. Malik, Region-Based Convolutional Networks for Accurate Object Detection and Segmentation, IEEE Trans. Pattern Anal. Mach. Intell. 38 (1) (2016) 142–158, https://doi.org/10.1109/TPAMI.2015.2437384.
[19] Y. Ham, K.K. Han, J.J. Lin, M. Golparvar-Fard, Visual monitoring of civil infrastructure systems via camera-equipped Unmanned Aerial Vehicles (UAVs): a review of related works, Visualization in Engineering 4 (1) (2016) 1, https://doi.org/10.1186/s40327-015-0029-z.
[20] K. He, X. Zhang, S. Ren, J. Sun, Deep Residual Learning for Image Recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.
[21] Hoskere, V., Amer, F., Friedel, D., Narazaki, Y., Yang, W., Tang, Y., ... Smith, M. D. (2020). InstaDam: A Semi-Automated Software Platform for Rapid Pixel-wise Annotation of Structural Damage in Images. Submitted to Applied Sciences.
[22] V. Hoskere, Y. Narazaki, T.A. Hoang, B.F. Spencer, MaDnet: multi-task semantic segmentation of multiple types of structural materials and damage in images of civil infrastructure, J Civil Struct Health Monit 10 (5) (2020) 757–773, https://doi.org/10.1007/s13349-020-00409-0.
[23] V. Hoskere, Y. Narazaki, B.F. Spencer, Learning to Detect Important Visual Changes for Structural Inspections using Physics-based Graphics Models, 9th International Conference on Structural Health Monitoring of Intelligent Infrastructure. St. Louis, USA, 2019.
[24] V. Hoskere, Y. Narazaki, B.F. Spencer, M.D. Smith, January 1). *Deep learning-based damage detection of miter gates using synthetic imagery from computer graphics*, DEStech Publications Inc., 2019, pp. 3073–3080.
[25] Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., ... Adam, H. (2017). MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. Retrieved from http://arxiv.org/abs/1704.04861
[26] Huang, H., Lin, L., Tong, R., Hu, H., Zhang, Q., Iwamoto, Y., ... Wu, J. (2020). UNet 3+: A Full-Scale Connected UNet for Medical Image Segmentation. 1055–1059. Retrieved from http://arxiv.org/abs/2004.08790
[27] Ikuma, M., & Naito, M. (2011a). Structure Planning for Shinkansens. Concrete Journal, 49(1), 27–31. https://doi.org/10.3151/coj.49.1_27
[28] Inaguma, H., & Seki, M. (2004). 鉄道高架橋柱のポリエステル繊維巻き耐震補強に関する実験的研究 (Experimental study on earthquake strengthening using polyester sheets of RC railway viaduct columns, in Japanese). 構造工学論文集 A (JSCE Journal of Structural Engineering A), 50A(2), 515–526.
[29] S. Ioffe, C. Szegedy, Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift, The 32nd International Conference on Machine Learning, 2015.

[30] N. Jaipuria, X. Zhang, R. Bhasin, M. Arafa, P. Chakravarty, S. Shrivastava, V.N. Murali, Deflating Dataset Bias Using Synthetic Data Augmentation, Retrieved from (2020). http://arxiv.org/abs/2004.13866.
[31] D.P. Kingma, J. Ba, A method for stochastic optimization, Proc. International Conference for Learning Representations (2015) 1–15.
[32] Kono, M., & Matsumoto, Y. (1965). DESIGN OF THE STANDARD RIGID FRAME RAILWAY BRIDGE IN NEW TOKAIDO LINE. Transactions of the Japan Society of Civil Engineers, 1965(115), 13–25. https://doi.org/10.2208/jscej1949.1965.115_13
[33] Labelbox: The leading training data platform. (n.d.). Retrieved May 19, 2020, from https://labelbox.com/
[34] Laina, I., Rupprecht, C., Belagiannis, V., Tombari, F., & Navab, N. (2016). Deeper depth prediction with fully convolutional residual networks. Proceedings - 2016 4th International Conference on 3D Vision, 3DV 2016, 239–248. https://doi.org/10.1109/3DV.2016.32
[35] X. Liang, Image-based post-disaster inspection of reinforced concrete bridge systems using deep learning with Bayesian optimization, Computer-Aided Civil and Infrastructure Engineering 34 (5) (2019) 415–430, https://doi.org/10.1111/mice:12425.
[36] E. Shelhamer, J. Long, T. Darrell, Fully Convolutional Networks for Semantic Segmentation, IEEE Trans. Pattern Anal. Mach. Intell. 39 (4) (2017) 640–651, https://doi.org/10.1109/TPAMI.2016.2572683.
[37] Maguire, M., Dorafshan, S., & Thomas, R. J. (2018). SDNET2018: A concrete crack image dataset for machine learning applications. https://doi.org/https://doi.org/10.15142/T3TD19
[38] D.D. Margineantu, When Does Imbalanced Data Require Cost-Sensitive Learning? more than, Retrieved from (2000), www.aaai.org.
[39] K. Masashi, K. Shinoda, K. Mizuno, S. Nozawa, T. Ishibashi, Study on damage caysed to Shinkansen RC viaducts by the 2011 off the pacific coast of Tohoku earthquake, Journal of Japan Society of Civil Engineers A1 70 (4) (2014) 688–700.
[40] Matsushige, S. (1964). 国鉄東海道新幹線工事について(in Japanese). Manufacturing and Technology, 16(9), 1–7.
[41] Y. Narazaki, F. Gomez, V. Hoskere, M.D. Smith, B.F. Spencer, Efficient development of algorithms for vision-based dense 3D displacement measurement using physics-based graphics models. *Submitted to the Journal of*, Structural Health Monitoring. (2020).
[42] Y. Narazaki, V. Hoskere, B.A. Eick, M.D. Smith, B.F. Spencer, Vision-based dense displacement and strain estimation of miter gates with the performance evaluation using physics-based graphics models, Smart Structures and Systems 24 (6) (2019) 709–721. https://doi.org/10.12989/SSS.2019.24.6.709.
[43] Y. Narazaki, V. Hoskere, T.A. Hoang, Y. Fujino, A. Sakurai, B.F. Spencer Jr., Vision-based automated bridge component recognition with high-level scene consistency, Computer-Aided Civil and Infrastructure Engineering 35 (5) (2020) 465–482, https://doi.org/10.1111/mice:12505.
[44] Y. Narazaki, V. Hoskere, T.A. Hoang, B.F. Spencer, Automated Bridge Component Recognition using Video Data, The 7th World Conference on Structural Control and Monitoring (7WCSCM). Qingdao, China, 2018.
[45] D. Neven, B. De Brabandere, S. Georgoulis, M. Proesmans, L. Van Gool, Fast Scene Understanding for Autonomous Driving, Retrieved from, Deep Learning for Vehicle Perception (DLVP), 2017, http://arxiv.org/abs/1708.02550.
[46] noise · PyPI. (n.d.). Retrieved May 18, 2020, from https://pypi.org/project/noise/
[47] Ohba, M. (2013). The Design History of the Railway Viaduct from the Design of Tokaido Shinkansen to the Recent Design. Concrete Journal, 51(1), 112–115. https://doi.org/10.3151/coj.51.112
[48] X. Pan, T.Y. Yang, Postdisaster image-based damage detection and repair cost estimation of reinforced concrete buildings using dual convolutional neural networks, Computer-Aided Civil and Infrastructure Engineering 35 (5) (2020) 495–510, https://doi.org/10.1111/mice:12549.
[49] Pharr, M., & Humphreys, G. (2010). Physically Based Rendering: From Theory To Implementation. In Physically Based Rendering: From Theory To Implementation. https://doi.org/10.1016/C2009-0-30446-8.
[50] Poudel, R. P. K., Liwicki, S., & Cipolla, R. (2019). Fast-SCNN: Fast Semantic Segmentation Network. Retrieved from http://arxiv.org/abs/1902.04502
[51] J. Redmon, A. Farhadi, YOLO9000: Better, Faster, Stronger, 2017, Retrieved from http://pjreddie.com/yolo9000/.
[52] S. Ren, K. He, R. Gershick, J. Sun, R. Girshick, J. Sun, Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks, Advances in Neural Information Processing Systems 39 (6) (2015) 91–99, https://doi.org/10.1109/TPAMI.2016.2577031.
[53] O. Ronneberger, P. Fischer, T. Brox, U-net: Convolutional networks for biomedical image segmentation, Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 9351 (2015) 234–241, https://doi.org/10.1007/978-3-319-24574-4_28.
[54] G. Ros, L. Sellart, J. Materzynska, D. Vazquez, A.M. Lopez, The SYNTHIA Dataset: A Large Collection of Synthetic Images for Semantic Segmentation of Urban Scenes, IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2016 (2016) 3234–3243, https://doi.org/10.1109/CVPR.2016.352.
[55] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A.C. Berg, L.i. Fei-Fei, ImageNet Large Scale Visual Recognition Challenge, Int J Comput Vis 115 (3) (2015) 211–252, https://doi.org/10.1007/s11263-015-0816-y.
[56] K. Simonyan, A. Zisserman, Very Deep Convolutional Networks for Large-Scale Image Recognition, International Conference on Learning Representations (ICRL) 1–14 (2015), https://doi.org/10.1016/j.infsof.2008.09.005.
[57] B.F. Spencer Jr., V. Hoskere, Y. Narazaki, Advances in Computer Vision-Based Civil Infrastructure Inspection and Monitoring, Engineering 5 (2) (2019) 199–222, https://doi.org/10.1016/j.eng.2018.11.030.
[58] Takahashi, Y. (2011). 東北地方太平洋沖地震 構造物被害報告 (in Japanese). Retrieved from http://committees.jsce.or.jp/2011quake/system/files/Takahashi_vol1.pdf
[59] A. Tao, K. Sapra, B. Catanzaro, Hierarchical Multi-Scale Attention for Semantic Segmentation, Retrieved from (2020). http://arxiv.org/abs/2005.10821.
[60] K. Tateno, F. Tombari, I. Laina, N. Navab, CNN-SLAM: Real-Time Dense Monocular SLAM With Learned Depth Prediction CVPR (2017) 6243–6252.
[61] Tateyama, M. (2009). 鉄道における土工技術と性能規定化の動向 (in Japanese). 建設の施工企画, 709, 33–39.
[62] Textures for 3D, graphic design and Photoshop! (n.d.). Retrieved June 5, 2020, from https://www.textures.com/
[63] The KITTI Vision Benchmark Suite. (n.d.). Retrieved April 20, 2020, from http://www.cvlibs.net/datasets/kitti/
[64] Uhrig, J., Cordts, M., Franke, U., & Brox, T. (2016). Pixel-level Encoding and Depth Layering for Instance-level Semantic Labeling. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 9796 LNCS, 14–25. Retrieved from http://arxiv.org/abs/1604.05096
[65] Y. Xu, Y. Bao, J. Chen, W. Zuo, H. Li, Surface fatigue crack identification in steel box girder of bridges by a deep fusion convolutional neural network based on consumer-grade camera images, Structural Health Monitoring 18 (3) (2019) 653–674, https://doi.org/10.1177/1475921718764873.
[66] C.M. Yeum, Computer vision-based structural assessment exploiting large volumes of images, Theses and Dissertations Available from ProQuest, 2016.
[67] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, A. Torralba, Learning Deep Features for Discriminative Localization, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Las Vegas, NV, USA, 2016.
[68] Z. Zhou, M.M.R. Siddiquee, N. Tajbakhsh, J. Liang, UNet++: A Nested U-Net Architecture for Medical Image Segmentation, Retrieved from (2018). http://arxiv.org/abs/1807.10165.