Please cite this article as: Zenghui Miao, Xiaodong Ji*, Taichiro Okazaki, Noriyuki Takahashi. Pixel-level multicategory detection of visible seismic damage of reinforced concrete components. Computer-Aided Civil and Infrastructure Engineering, 2021, 1-18. https://doi.org/10.1111/mice.12667

Pixel-level multicategory detection of visible seismic damage of reinforced concrete components

Zenghui Miao[1] | Xiaodong Ji[1] | Taichiro Okazaki[2] | Noriyuki Takahashi[3]

[1]Key Laboratory of Civil Engineering Safety and Durability of China Education Ministry, Department of Civil Engineering, Tsinghua University, Beijing 100084, China

[2]Graduate School of Engineering, Hokkaido University, Sapporo, Hokkaido 060-8628, Japan

[3]Department of Architecture and Building Science, Tohoku University, Sendai, Miyagi 980-0845, Japan

Correspondence
Xiaodong Ji, Department of Civil Engineering, Tsinghua University, Beijing 100084, China.
E-mail: jixd@mail.tsinghua.edu.cn

Funding information
National Key R&D Program of China, Grant/Award Number: 2017YFC1500602; NSFC-JSPS International Joint Research Program, Grant/Award Number: 51811540032; Tsinghua University Initiative Scientific Research Program, Grant/Award Number: 20193080019

ABSTRACT

The detection of visible damage (i.e., cracking, concrete spalling and crushing, reinforcement exposure, buckling and fracture) plays a key role in post-earthquake safety assessment of reinforced concrete (RC) building structures. In this study, a novel approach based on computer-vision techniques was developed for pixel-level multicategory detection of visible seismic damage of RC components. A semantic segmentation database was constructed from test photos of RC structural components. Series of datasets were generated from the constructed database by applying image transformations and data-balancing techniques at the sample and pixel levels. A deep convolutional network (CNN) architecture was designed for pixel-level detection of visible damage. Two sets of parameters were optimized separately, one to detect cracks and the other to detect all other types of damage. A post-processing technique for crack detection was developed to refine crack boundaries, and thus improve the accuracy of crack characterization. Finally, the proposed vision-based approach was applied to test photos of a beam-to-wall joint specimen. The results demonstrate the accuracy of the vision-based approach to detect damage, and its high potential to estimate seismic damage states of RC components.

1 INTRODUCTION

Post-earthquake safety assessment of building structures has played a critical role in emergency treatment and post-hazard restoration of urban communities. For reinforced concrete (RC) buildings, post-earthquake safety assessment in the US (FEMA, 1998), Japan (JBDPA, 1997; MLIT, 2015) and China (CMC, 2016) require certified structural engineers or inspectors to visually inspect the damage state of individual buildings. This methodology is based on the relation between visible damage and performance degradation of structural components established from experimental data and observations of field performance. Although proven to be effective, this current procedure of safety assessment is time consuming and labor intensive. Furthermore, the evaluation results are reliant on the professional knowledge and experience of the engineers or inspectors (German et al., 2013), and thus might be influenced by human bias. With the advent of computer vision and machine learning, there is a high potential to develop a vision-based system for seismic damage

detection, which supports the engineers and inspectors to advance the current practice of post-earthquake safety assessment of building structures.

Vision-based damage detection has been extensively studied in the field of structural health monitoring over the past few decades. Vision-based crack detection approaches using conventional image processing techniques (IPTs) have found applications in crack detection on the surface of concrete and asphalt. The conventional IPTs mainly include the thresholding-based approach (Cheng et al., 2003; Fujita and Hamamoto, 2011; Nishikawa et al., 2012; Ying and Salari, 2010), the morphological approach (Iyer and Sinha, 2006; Nguyen et al., 2014) and the percolation-based approach (Yamaguchi et al., 2008). Two factors, i.e., the clear contrast of gray levels between the cracks and background and the narrow, line-like geometry of cracks, are vital for the success of the conventional IPTs-based crack detection. Therefore, the accuracy of these methods strongly relies on photo shooting conditions and the complexity of background clutter and occlusions. Moreover, the conventional IPTs-based methods are not suited to detect other types of damage, such as concrete spalling.

Over the past decade, machine learning techniques, including feature engineering, data clustering and classification, have been adopted to improve the flexibility and applicability of vision-based damage detection. Through feature engineering, a series of feature indexes, e.g., the geometric properties or the statistical indexes of color distributions, are defined and calculated for an area segmented from the input image. Then, by the application of machine learning algorithms for example the k-nearest neighbors (KNN) (Jahanshahi et al., 2013; Oliveira and Correia, 2008), support vector machine (SVM) (Chen et al., 2017a; Chen et al., 2012; Jahanshahi et al., 2013; Li et al., 2017; O'Byrne et al., 2014) and neural networks (NN) (Jahanshahi et al., 2013), the areas are clustered or classified based on the distribution of the feature indexes as one of the predefined damage categories. However, the success of these feature-based approaches relies on reasonable selection of pre-defined feature indexes, which requires considerable domain knowledge about the application scenarios. Therefore, these approaches are not suited for application to arbitrary damage categories in complex situations.

More recently, the convolutional networks (CNNs), which can be regarded as the combination of automatic feature extraction and classification, have been applied to visible damage detection in civil engineering. Some researchers proposed crack detection methods using CNN-based image classification (Cha et al., 2017; Ni et al., 2019), where the input image is firstly divided into a series of small image patches, which are then classified by the CNN as crack or non-crack. Other researchers developed the CNN-based object detection algorithms (Beckman et al., 2019; Li et al., 2018; Maeda et al., 2018), where multiple areas of visible damage are identified and localized in the form of rectangle regions with bounding boxes, and classified as one of the target damage categories, including crack, spalling, rebar exposure or steel corrosion. The CNN-based image classification and object detection are both region-level algorithms, i.e., visible damage is identified and localized from the input image with a series of image patches or bounding boxes. Such algorithms are incapable of segmenting the exact damage geometries from the input image, and thus, do not allow for quantitative assessment of damage for further analysis. For the purpose of quantitative analysis of visible damage, researchers developed pixel-level methods to clearly identify the arbitrary geometries of various damage categories through CNN-based semantic segmentation (Choi and Cha, 2020; Zou et al., 2019). These pixel-level damage detection methods were successfully applied to the maintenance inspection of pavement (Bang et al., 2019; Zhang et al., 2018) and bridge piers (Jang et al., 2020). Crack width and length can be calculated from the pixel-level detection results (Jang et al., 2020), which suggests high potential of these methods for further quantitative analysis.

Different from structural health monitoring of infrastructures that aims to detect damage at the early stages, damage detection applied to post-earthquake assessment of building structures requires the investigation of moderate and severe damage in order to accurately estimate the severity of damage. Moreover, wide diversity of shooting conditions and background clutter and occlusions should be considered to cope with the complex environment of practical applications. Past research in this area has focused on damage evaluation based on the outward appearances of damaged buildings. The CNN-based image classification was developed and applied for

collapse recognition (Gao and Mosalam, 2018; Xiong et al., 2020; Yeum et al., 2018) and damage degree evaluation (Ishii et al., 2018) of building structures. Pixel-level damage detection of building façades is achieved through CNN-based semantic segmentation (Chida and Takahashi, 2020; Hoskere et al., 2018). However, limited effort has been placed on damage detection at the component level. Recently, a pixel-level damage detection model for bridge piers (Liang, 2019) was shown to segment damage geometries from the background, but this study fell short of distinguishing the detailed damage categories.

To achieve pixel-level detection and quantitative analysis of multiple seismic damage categories for RC structural components, further development is required in the following two aspects: (1) In addition to cracks, pixel-level detection of concrete spalling and crushing, reinforcement exposure, buckling and fracture is needed because quantitative estimate of these seismic damage categories is essential to the assessment of seismic performance degradation of damaged RC components and post-earthquake safety of RC buildings; (2) Wide diversity in terms of failure modes of RC components, shooting conditions of input images as well as background clutter and occlusions should be considered for the effective detection of typical seismic damage.

The objectives of this study are to establish the techniques of computer vision (1) to localize, classify and segment typical seismic damage of RC components; and (2) to quantify and characterize the detected cracks and other damage. The major contribution of this study is threefold: (1) Data balancing techniques were proposed/developed to address the data imbalance issue of the constructed database, and stratified sampling was adopted to improve the recognition performance of complex background clutter and occlusions for the trained CNNs. (2) Two CNNs, i.e., the Crack-Net and 4Category-Net, were proposed and trained for pixel-level detection of cracks and other typical seismic damage categories (i.e., concrete spalling and crushing, reinforcement exposure, buckling and fracture), to enable quantitative assessment of seismic performance degradation of damaged RC components. (3) An effective yet simple post-processing technique was developed for Crack-Net outputs, which refined the boundaries of the detected cracks and thus enabled subsequent characterization of crack width to be accurate.

2 OVERVIEW OF CONVOLUTIONAL NETWORK OPTIMIZATION

The convolutional networks (CNNs) and the corresponding optimization, which form the basis of this study, are briefly described in this section. A convolutional network is a hierarchical combination of a few basic functions, which are called layers in CNN terminology, such as the convolutional layer, the ReLU layer and the pooling layer. The configuration of how these basic layers are organized and ensembled is referred to as the architecture of a CNN. Mathematically, a CNN can be interpreted as a complex function defined by its architecture, and can be formulated using symbolic notations as in Equation (1).

$$\boldsymbol{P} = \boldsymbol{f}(\boldsymbol{X}; \boldsymbol{\theta}) \tag{1}$$

In Equation (1), $\boldsymbol{f}(\cdot)$ is the function defined by the architecture, $\boldsymbol{X}$ and $\boldsymbol{P}$ are the (network) input and output, respectively, and $\boldsymbol{\theta}$ denotes the set of (network) parameters.

For a CNN used for semantic segmentation, the input $\boldsymbol{X}$ is an image with shape $H \times W \times c$, where H and W denote the height and width of the image in pixel-unit, respectively, and c denotes the number of color channels (e.g., $c = 3$ for an RGB image). The output of a CNN, $\boldsymbol{P}$, consists of $H \times W$ probability distributions, as in Equations (2) and (3), where $(p_k)_{i,j}$ denotes the probability that the pixel at the i-th row and j-th column belongs to category k, and K denotes the number of categories for a certain segmentation task. The (segmentation) prediction $\widehat{\boldsymbol{Y}}$ is further derived from the output $\boldsymbol{P}$ through Equations (4) and (5).

$$\boldsymbol{P} = \begin{bmatrix} \boldsymbol{p}_{0,0} & \boldsymbol{p}_{0,1} & \cdots & \boldsymbol{p}_{0,W-1} \\ \boldsymbol{p}_{1,0} & \boldsymbol{p}_{1,1} & \cdots & \boldsymbol{p}_{1,W-1} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{p}_{H-1,0} & \boldsymbol{p}_{H-1,1} & \cdots & \boldsymbol{p}_{H-1,W-1} \end{bmatrix} \tag{2}$$

$$\boldsymbol{p}_{i,j} = (p_0, p_1, \dots, p_{K-1})_{i,j}^T \tag{3}$$

$$\widehat{\boldsymbol{Y}} = \begin{bmatrix} \hat{y}_{0,0} & \hat{y}_{0,1} & \cdots & \hat{y}_{0,W-1} \\ \hat{y}_{1,0} & \hat{y}_{1,1} & \cdots & \hat{y}_{1,W-1} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{y}_{H-1,0} & \hat{y}_{H-1,1} & \cdots & \hat{y}_{H-1,W-1} \end{bmatrix} \tag{4}$$

$$\hat{y}_{i,j} = \underset{k=0,1,\dots,K-1}{\operatorname{argmax}} (p_k)_{i,j} \tag{5}$$

In semantic segmentation, a pair of an input image $\boldsymbol{X}$ and its corresponding pixel-level annotation (i.e., the ground truth) $\boldsymbol{Y}$, which is fed to a CNN for training or testing, is referred to as a sample. As formulated in Equation (6), the annotation $\boldsymbol{Y}$ of an input image $\boldsymbol{X}$ is a

matrix with shape $H \times W$, where $y_{i,j}$ denotes the actual category of the pixel at the i-th row and j-th column.

$$\boldsymbol{Y} = \begin{bmatrix} y_{0,0} & y_{0,1} & \cdots & y_{0,W-1} \\ y_{1,0} & y_{1,1} & \cdots & y_{1,W-1} \\ \vdots & \vdots & \ddots & \vdots \\ y_{H-1,0} & y_{H-1,1} & \cdots & y_{H-1,W-1} \end{bmatrix},$$

$$y_{i,j} \in \{0, 1, \dots, K-1\} \tag{6}$$

The training of a CNN is conducted to optimize its parameters $\boldsymbol{\theta}$ according to the available data (i.e., the dataset), such that the prediction $\widehat{\boldsymbol{Y}}$ of the CNN could be as identical as possible with the ground truth $\boldsymbol{Y}$ for a given input $\boldsymbol{X}$. The dataset, consisting of series of samples, is divided into two subsets, i.e., the training set and the test set. The training set is used to optimize the network parameters, while the test set is used to validate the prediction performance of the trained CNN. The loss function is introduced for the training of a CNN, which is defined as the mean value of prediction errors of all pixels in the training set, as in Equation (7). The prediction error of the CNN for a pixel is measured by an error function, such as the Negative Log-Likelihood (NLL) formulated in Equation (8), which is the commonly-used error function for CNNs.

$$L = \frac{1}{n}\sum_{s=0}^{n-1} l(\boldsymbol{p}_s, y_s) \tag{7}$$

$$l(\boldsymbol{p}_s, y_s) = -\ln\left(p_{y_s}\right)_s \tag{8}$$

In Equations (7) and (8), $\boldsymbol{p}_s$ and y_s denote the probability distribution and ground truth category of a pixel, respectively; and n denotes the number of pixels in the training set.

The training of a CNN is achieved by the gradient descent method, in which the parameters are iteratively optimized to minimize the loss function defined in Equation (7). However, because a large number of images may be included in a dataset for computer vision tasks, it is computationally impractical to calculate the mean value of errors and its corresponding gradient on the entire training set at each iteration. A compromise to overcome this difficulty is the stochastic gradient descent (SGD) method, in which a batch, i.e., a small number of samples randomly selected from the training set, is established at each iteration, and the loss function and its gradient are calculated for the batch. The loss function at iteration t for the training of a CNN using SGD can therefore be formulated as in Equation (9). The model parameters are then updated by the reverse of the gradient, multiplied by a small step size (i.e., the learning rate), α, as shown in Equation (10).

$$L_t = \frac{1}{n_\mathrm{b}}\sum_{s=0}^{n_\mathrm{b}-1} l(\boldsymbol{p}_s, y_s) \tag{9}$$

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \alpha\frac{\partial L_t}{\partial \boldsymbol{\theta}_t} = \boldsymbol{\theta}_t - \alpha\frac{1}{n_\mathrm{b}}\sum_{s=0}^{n_\mathrm{b}-1}\frac{\partial l(\boldsymbol{p}_s, y_s)}{\partial \boldsymbol{\theta}_t} \tag{10}$$

In Equations (9) and (10), n_b denotes the number of pixels in a batch.

The vanilla SGD can reduce the computing time significantly, but the gradient calculated by vanilla SGD is rather noisy and may cause slow convergence in the training. Several algorithms are proposed for a more effective training, such as Adam (Kingma and Ba, 2015) and RMSProp (Tieleman and Hinton, 2012) that was used for the training of CNNs in this study.

The trained CNN is used to produce the prediction of a new-coming input image, which is referred to as the inference of an image. One can understand from the CNN optimization that the prediction performance of a trained CNN is dependent on its architecture and the generality and representativity of the dataset.

3 DATABASE FOR VISIBLE DAMAGE OF RC COMPONENTS

A database which has sufficient number of labeled images and is representative of the application scenario is critical for any CNN-based system. A few databases have been established for the development of post-earthquake damage detection (Gao and Mosalam, 2018; Sajedi and Liang, 2020). In the database by Gao and Mosalam (2018), the samples were labeled for image classification, and were not annotated at the pixel level. In the database by Sajedi and Liang (2020), although samples were annotated at the pixel level, the damage geometries were identified without further distinction among different damage categories. To fulfill the long-term objective of quantitative assessment of typical seismic damage categories, in this study, images of damaged RC structural components were collected and annotated for multiple damage categories at the pixel level, and a semantic segmentation database was constructed for visible seismic damage categories of RC components. The images in the database are diverse in terms of visual appearances of seismic damage, shooting conditions (e.g., lighting conditions, scales and viewpoints), and

DS	Appearances of visible damage
I	Crack width less than 0.2mm
II	Crack width 0.2 ~ 1mm
III	Crack width 1 ~ 2mm; minor spalling of concrete
IV	Severe spalling of concrete; obvious exposure of reinforcement
V	Crushing of core concrete; buckling of reinforcement

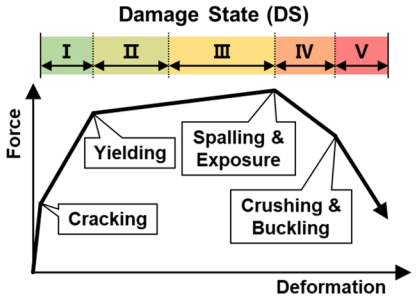


FIGURE 1. The relation between the visible damage and damage state of RC components (MLIT, 2015).

background clutter and occlusions. Database construction is divided into three key steps: (1) Definition of target categories of visible damage; (2) Image collection for the representation of selected damage categories; (3) Manual annotation of collected images. Special issues and considerations associated with these steps are discussed in this section.

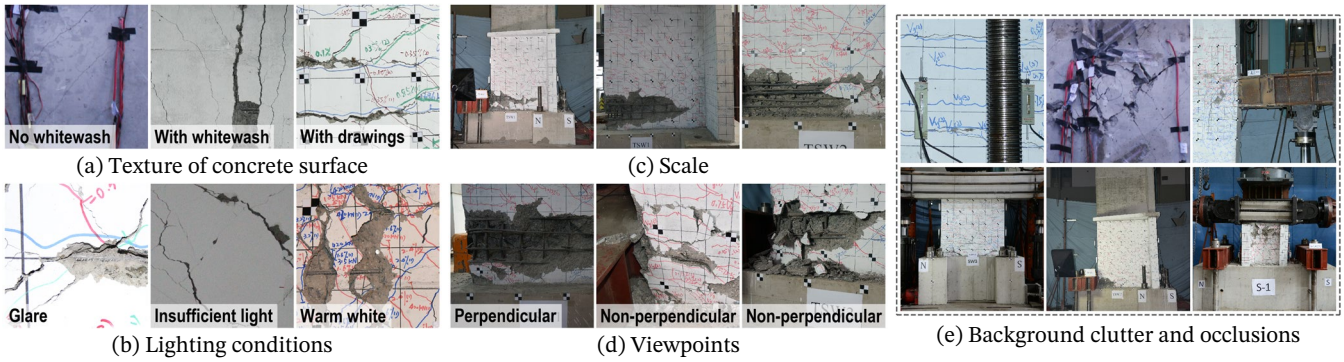
3.1 Definition of target damage categories

Seismic damage evaluation manuals define damage states and repair methods according to visible damage of RC components. For example, in the U.S., FEMA P-58 (FEMA, 2011) defines the damage states of slender RC walls as follows: Damage state DS1 (repair method: cosmetic repair) is associated with initial concrete cracking; DS2 (repair method: epoxy injection and patching) is associated with vertical cracks and spalling of cover concrete that does not reveal the longitudinal reinforcement; DS3 (repair method: replacement of concrete) is associated with spalling of cover concrete that exposes the longitudinal reinforcement; and DS4 (repair method: replacement of steel reinforcement and concrete) is associated with web concrete crushing, boundary element core crushing, rebar buckling or fracture. The Japanese standard for damage evaluation of seismic damaged buildings (MLIT, 2015) also specifies the damage states of RC components linked to their visible seismic damage, as illustrated in Figure 1.

Therefore, for effective estimation of the damage states, this paper categorizes visible damage as (a) concrete cracking, (b) cover concrete spalling, (c) exposure of reinforcement, (d) crushing of concrete, and (e) buckling and fracture of reinforcement.

3.2 Image collection and annotation

Seventy-six images with an average size of 3940 × 3940 pixels were selected to constitute the database. These images were selected from test photos of RC specimens including shear walls and joints, which were designed and tested by the authors. The test specimens ranged from mid-scale to full-scale, and were subjected to quasi-static cyclic loads following the loading protocol specified in Chinese specification for seismic test of buildings (CMC, 2015). The damage and failure modes of the specimens represent the seismic damage of RC components, and the images from experimental tests share similar visual characteristics of damage as those from post-earthquake field surveys. Figure 2 shows the diversity of the constructed database in terms of texture of concrete surface, lighting conditions, scales, viewpoints, and background clutter and occlusions. Diversity of the collected images was ensured through the following procedure. First, test specimens with a variety of section shapes (e.g., rectangular-shaped walls, T-shaped walls and I-shaped walls), shear-to-span ratios (in a range of 1.06 to 2.50), reinforcement configurations and failure



(a) Texture of concrete surface (c) Scale

(b) Lighting conditions (d) Viewpoints (e) Background clutter and occlusions

FIGURE 2. The diversity of the constructed database.

modes (e.g., flexural failure, shear failure, flexural-shear failure and sliding failure) were collected. Second, during the cyclic loading tests, cameras with different configurations were set up to shoot in diverse lighting conditions (see Figure 2(b)), scales (with a resolution ranging from 0.79 to 5.13 pixel/mm, see Figure 2(c)) and viewpoints (see Figure 2(d)). Third, image selection was conducted so that various background clutter and occlusions, such as wires, measurement devices and loading setups, would be included into the database, as demonstrated in Figure 2(e).

Manual annotation of the selected images was conducted by individuals who followed a comprehensive guideline. The guideline defined visual characteristics for each target category, and rules for pixel-level annotation to ensure the preciseness and consistency of manual annotation.

3.3 From database to datasets

In the context of this research, the term "database" denotes a set of images which occupy arbitrary sizes, and the term "dataset" denotes a set of samples with a standard, uniform size of 300 × 300 pixels.

Architectures of CNNs like FCN (Long et al., 2015), where each operation is reformulated as the convolution operation, allow images with arbitrary sizes to be fed to the networks. However, because the images in the database are large averaging 3940 × 3940 pixels, training CNNs directly through the database images would result in major drawbacks that significantly affect the training efficiency and recognition performance (i.e., detection accuracy). First, in terms of semantic segmentation, a fully convolutional architecture accepting large-size images as input is memory expensive. As an example, FCN-8s requires more than 4GB of memory for a sample with 1024 × 1024 pixels. The memory requirement of a sample limits the batch size to two samples for a NVIDIA GeForce GTX 1080 Ti GPU, which has a memory of 11GB and is a commonly-used GPU for general computing. Naturally, for an overly small batch size, the gradient of the loss function would be inaccurate, and training process will not effectively converge. Second, uniform-size samples facilitate the design of the convolutional network and simplify the programming of the proposed vision-based approach.

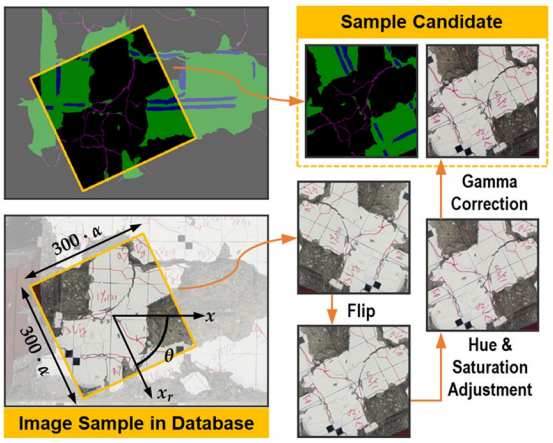


FIGURE 3. Flowchart for generation of sample candidates.

In this research, samples of 300 × 300 pixels were generated from images of the database to form a series of datasets, and were fed directly to the networks as training or test samples. As illustrated in Figure 3, sample candidates for the datasets are generated following a flowchart where image transformations are adopted to expand data diversity with respect to affine transformations and lighting conditions. Diversity of the datasets is enriched by randomly assigning the center point, rotation angle, scaling factor and flipping axis. Hue and Saturation adjustment and Gamma correction are further applied to simulate minor fluctuations of lighting conditions.

To guarantee the independence between training samples and test samples, images in the database are divided into two sub-databases, i.e., the training sub-database and test sub-database. Approximately 80% of images (i.e., 61 images) that are randomly selected from the database are included in the training sub-database, while the rest of images are left in the test sub-database. Afterwards, the training samples and test samples are generated from the training sub-database and test sub-database, respectively.

Separate datasets were generated for the crack category and the other four damage categories for the following two reasons: (1) Unlike the other four damage categories, its narrow line-shaped and delicate appearance forms its unique small-scale visual characteristics; (2) The crack category is relatively independent in spatial distribution, and limited contextual connections are shared with the other damage categories. In this study, datasets generated for the detection of crack and the other four damage categories are named with the prefix "Crack" and "4Category", respectively.

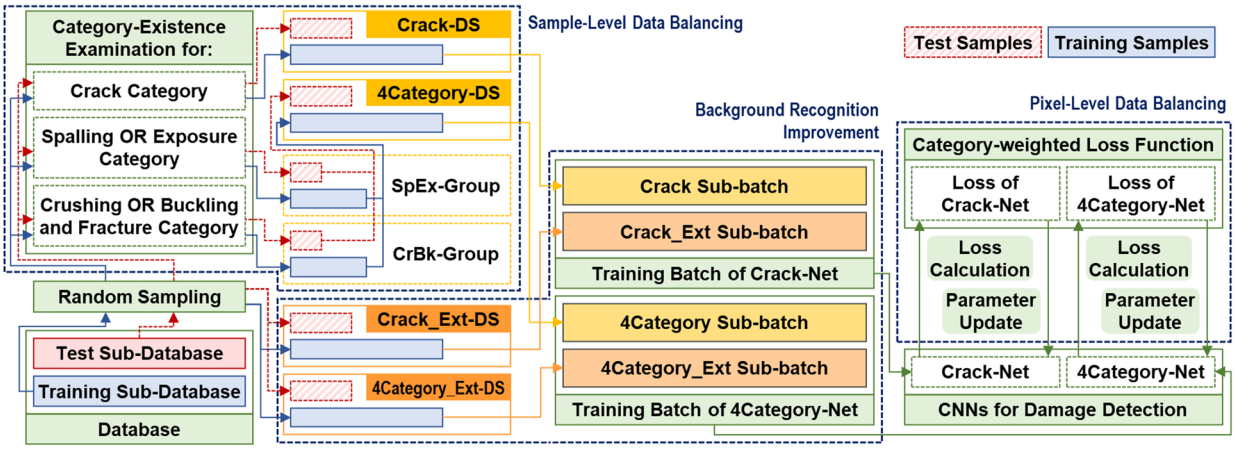


FIGURE 4. Flowchart of CNN-based damage detection and the corresponding data balancing techniques.

3.4 Data balancing

Significant difference in the number of samples and pixels of different target categories, i.e., data imbalance, is observed in the constructed database. Data imbalance is a common problem in CNNs and has major effect on the performance of a CNN-based system (Buda et al., 2018). At the sample level, the number of images which contain crushing, buckling and fracture categories is much less than the images containing other damage categories, since severe damage occurred only at the end of experimental testing. Moreover, the spatial distribution of each damage category leads to severe data imbalance at the pixel level. The pixels of background are dominant in the database, and for most image samples, the pixels of concrete damage (i.e., spalling and crushing) occupy much larger area than the pixels of reinforcement damage (i.e., exposure, buckling and fracture).

Data imbalance can extensively affect the training process (Buda et al., 2018). For instance, if one category is missing in a training batch, i.e., no damage area of the category is contained in any sample, the model parameters will be updated without taking account of the influence on this category, and consequently, the resulting CNN model will have no capability of detecting this category from the images.

In this study, techniques for data preparation and training of CNNs, including the sample-level and pixel-level data balancing and the background recognition improvement, were proposed/developed to mitigate data imbalance. The flowchart of the proposed CNN-based damage detection approach is demonstrated in Figure 4, where data balancing techniques functioned at different stages are highlighted. Combined use of these techniques allowed the CNNs to achieve balanced recognition performance for the target damage categories as well as improved recognition of background clutter and occlusions. Detailed explanation and theoretical formulation of these techniques are provided in this subsection.

3.4.1 Sample-level data balancing

The key of mitigating data imbalance at the sample level is to ensure that sufficient samples of each damage category are contained in the dataset, such that samples of each category will be included in any of the randomly selected batches. For this purpose, category-existence examination was conducted on sample candidates before including them in the datasets. For the crack category, a sample candidate generated from the database, following the procedure of Figure 3, was accepted as a sample of the dataset Crack-DS only if cracks were contained. To construct the dataset 4Category-DS, two groups of samples, denoted as SpEx-Group (Spalling / Exposure Group) and CrBk-Group (Crushing / Buckling Group), were firstly generated with category-existence examination (see Figure 4). Different sampling rates were assigned for the generation of the SpEx-Group and CrBk-Group, in order to ensure balance in size of the two groups. Dataset 4Category-DS was then constructed by combining the SpEx-Group and CrBk-Group. Statistical information of the resulted datasets is listed in Table 1.

To validate data balancing of 4Category-DS, the statistics of the number of samples of category k in a batch (i.e., $N_{b,k}$), including the mean value, standard deviation and the probability of no more than 8 samples (25% of the batch size) of category k in a batch, were calculated according to Equations (11) - (15).

TABLE 1. Statistical information of 4Category-DS and Crack-DS.

Dataset	*Category*	*Number of samples (Training / Test set)*	*Portion of pixels in training set*
4Category-DS	Background	3539 / 787	69.89%
	Spalling	3532 / 782	15.83%
	Exposure	2110 / 497	1.15%
	Crushing	1850 / 370	9.43%
	Buckling and Fracture	1793 / 365	3.70%
	Total	3545 / 787	-
Crack-DS	Background	2931 / 523	98.07%
	Crack		1.93%

TABLE 2. Statistics of the number of samples of damage categories in a batch.

Category	$EN_{\mathrm{b},k}$	$DN_{\mathrm{b},k}$	$P(N_{\mathrm{b},k} \le 8)$
Spalling	31.9	0.3	0
Exposure	19.0	2.8	7.3×10^{-5}
Crushing	16.7	2.8	1.6×10^{-3}
Buckling and Fracture	16.2	2.8	2.8×10^{-3}

$$P\left(N_{\mathrm{b},k}=i\middle|N,N_k,N_{\mathrm{b}}\right)=\frac{\binom{N-N_k}{N_{\mathrm{b}}-i}\cdot\binom{N_k}{i}}{\binom{N}{N_{\mathrm{b}}}} \tag{11}$$

$$EN_{\mathrm{b},k}=\sum_{i=0}^{N_{\mathrm{b}}}P\left(N_{\mathrm{b},k}=i\middle|N,N_k,N_{\mathrm{b}}\right)\times i \tag{12}$$

$$E\left(N_{\mathrm{b},k}\right)^2=\sum_{i=0}^{N_{\mathrm{b}}}P\left(N_{\mathrm{b},k}=i\middle|N,N_k,N_{\mathrm{b}}\right)\times i^2 \tag{13}$$

$$DN_{\mathrm{b},k}=\sqrt{E\left(N_{\mathrm{b},k}\right)^2-\left(EN_{\mathrm{b},k}\right)^2} \tag{14}$$

$$P\left(N_{\mathrm{b},k}\le 8\right)=\sum_{i=0}^{8}P\left(N_{\mathrm{b},k}=i\middle|N,N_k,N_{\mathrm{b}}\right) \tag{15}$$

In Equations (11) - (15), N is the number of samples in the training set of 4Category-DS, N_{b} denotes the number of samples in a batch, which is 32 in this study, N_k and $N_{\mathrm{b},k}$ are the number of samples of category k in the training set and in a batch, respectively. The computed results are listed in Table 2.

The mean value $N_{\mathrm{b},k}$ for each category ranges from 16.2 to 31.9 (i.e., approximately 50% to 100% of the batch size), and the probability that any batch includes more than 8 samples for each category is over 99.7%. Data balancing is achieved at the sample level through category-existence examination.

TABLE 3. Statistical information of the extended datasets.

Dataset	*Category*	*Number of samples (Training / Test set)*	*Portion of pixels in training set*
4Category_Ext-DS	Background	3528 / 819	94.30%
	Spalling	968 / 179	3.15%
	Exposure	282 / 62	0.24%
	Crushing	308 / 36	1.58%
	Buckling and Fracture	288 / 35	0.73%
	Total	3530 / 819	-
Crack_Ext-DS	Background	2782 / 487	99.35%
	Crack	1241 / 207	0.65%
	Total	2782 / 487	-

3.4.2 Background recognition improvement

By prioritizing category-existence examination, the aforementioned procedure for sample-level data balancing would result in lack of background clutter and occlusions in the datasets. There is a concern that models trained by these datasets may be incapable of distinguishing background clutter and occlusions from damage. Simply including more samples with background category is unfavorable, because it would harm sample-level data balancing. A method for batch construction using stratified sampling is proposed to solve this puzzle. In this method, samples are generated without category-existence examination to construct (background-) extended datasets, denoted as Crack_Ext-DS and 4Category_Ext-DS. The datasets Crack-DS and 4Category-DS, explained in Section 3.4.1, would be referred to as restricted datasets for better clarity. Statistical information of the datasets is listed in Table 3. A pair of restricted and extended datasets is used during the training process to assemble a batch at an iteration, where samples in the batch are selected using stratified sampling, i.e., N_{brst} samples come from the restricted dataset (i.e., the restricted sub-batch) and N_{bext} samples come from the extended dataset (i.e., the extended sub-batch).

In this research, $N_{\mathrm{brst}}=N_{\mathrm{bext}}=32$, and therefore a batch includes 64 samples. Through stratified sampling, samples of damage categories and background category are included in any batch simultaneously.

3.4.3 Pixel-level data balancing

Even if sample-level data balancing is achieved, data imbalance at the pixel level alone can hinder and stagnate the training process. A method was developed by Sajedi and Liang (2020), which functions in the inference phase to

minimize the negative impact of data imbalance. In this study, a different method is proposed that mitigates the pixel-level data imbalance in the training process of a CNN.

The problem of pixel-level data imbalance can be revealed by the loss function in Equation (9), which can be organized as an ensemble among categories, as expressed in Equation (16).

$$\begin{aligned} L_t &= \frac{1}{n_\mathrm{b}}\sum_{k=0}^{K-1}\sum_{s=0}^{n_{\mathrm{b},k}-1} l(\boldsymbol{p}_s,k) \\ &= \sum_{k=0}^{K-1}\frac{n_{\mathrm{b},k}}{n_\mathrm{b}}\left(\frac{1}{n_{\mathrm{b},k}}\sum_{s=0}^{n_{\mathrm{b},k}-1} l(\boldsymbol{p}_s,k)\right) \\ &= \sum_{k=0}^{K-1}\frac{n_{\mathrm{brst},k}+n_{\mathrm{bext},k}}{n_\mathrm{brst}+n_\mathrm{bext}}\left(\frac{1}{n_{\mathrm{b},k}}\sum_{s=0}^{n_{\mathrm{b},k}-1} l(\boldsymbol{p}_s,k)\right) \\ &= \sum_{k=0}^{K-1}\left(\rho_\mathrm{rst}\frac{n_{\mathrm{brst},k}}{n_\mathrm{brst}}+\rho_\mathrm{ext}\frac{n_{\mathrm{bext},k}}{n_\mathrm{bext}}\right) \\ &\qquad\cdot\left(\frac{1}{n_{\mathrm{b},k}}\sum_{s=0}^{n_{\mathrm{b},k}-1} l(\boldsymbol{p}_s,k)\right) \end{aligned} \tag{16}$$

$$\rho_\mathrm{rst} = \frac{n_\mathrm{brst}}{n_\mathrm{brst}+n_\mathrm{bext}} = \frac{N_\mathrm{brst}}{N_\mathrm{brst}+N_\mathrm{bext}} \tag{17}$$

$$\rho_\mathrm{ext} = \frac{n_\mathrm{bext}}{n_\mathrm{brst}+n_\mathrm{bext}} = \frac{N_\mathrm{bext}}{N_\mathrm{brst}+N_\mathrm{bext}} \tag{18}$$

In Equation (16), n_b, n_brst and n_bext denote the number of pixels in the current batch, the restricted sub-batch and the extended sub-batch, and $n_{\mathrm{b},k}$, $n_{\mathrm{brst},k}$ and $n_{\mathrm{bext},k}$ denote the corresponding numbers for the category-k pixels. ρ_rst and ρ_ext represent the portion of restricted and extended samples in the batch (see Equations (17) and (18)), which are both 0.5 in this study.

It should be noticed that the expression $n_{\mathrm{brst},k}/n_\mathrm{brst}$ represents the portion of the category-k pixels in the restricted sub-batch, and can be considered as an approximation of the category-k pixels in the entire training set of the restricted dataset, $n_{\mathrm{rst},k}/n_\mathrm{rst}$, where n_rst and $n_{\mathrm{rst},k}$ denote the number of the pixels and the category-k pixels in the training set of the restricted dataset, respectively.

The term $n_{\mathrm{b},k}/n_\mathrm{b}$ measures the contribution of category k to the loss function and the influence over the gradient updating. Gradients derived from the category with a small portion of pixels is likely to be neglected or even reversed, which would result in the indistinguishability of the model for the small-portion category.

The pixel-level data imbalance can be solved by reweighting each category in the loss function, following Eigen and Fergus (2015), as expressed in Equation (19). The weight factor for category k, α_k, is taken as Equation (20), which is the reciprocal of the mean portion of category-k pixels in a batch divided by the number of categories, K.

$$L_{t,\mathrm{w}} = \sum_{k=0}^{K-1}\alpha_k\cdot\frac{n_{\mathrm{b},k}}{n_\mathrm{b}}\left(\frac{1}{n_{\mathrm{b},k}}\sum_{s=0}^{n_{\mathrm{b},k}-1} l(\boldsymbol{p}_s,k)\right) \tag{19}$$

$$\alpha_k = \frac{1}{K}\cdot\left(\rho_\mathrm{rst}\frac{n_{\mathrm{rst},k}}{n_\mathrm{rst}}+\rho_\mathrm{ext}\frac{n_{\mathrm{ext},k}}{n_\mathrm{ext}}\right)^{-1} \tag{20}$$

The reweighted loss function can be viewed as the average of errors over categories instead of over pixels, as in Equation (21).

$$\begin{aligned} L_{t,\mathrm{w}} &= \sum_{k=0}^{K-1}\alpha_k\cdot\frac{n_{\mathrm{b},k}}{n_\mathrm{b}}\left(\frac{1}{n_{\mathrm{b},k}}\sum_{s=0}^{n_{\mathrm{b},k}-1} l(\boldsymbol{p}_s,k)\right) \\ &= \sum_{k=0}^{K-1}\alpha_k\left(\rho_\mathrm{rst}\frac{n_{\mathrm{brst},k}}{n_\mathrm{brst}}+\rho_\mathrm{ext}\frac{n_{\mathrm{bext},k}}{n_\mathrm{bext}}\right) \\ &\qquad\cdot\left(\frac{1}{n_{\mathrm{b},k}}\sum_{s=0}^{n_{\mathrm{b},k}-1} l(\boldsymbol{p}_s,k)\right) \\ &\approx \sum_{k=0}^{K-1}\alpha_k\left(\rho_\mathrm{rst}\frac{n_{\mathrm{rst},k}}{n_\mathrm{rst}}+\rho_\mathrm{ext}\frac{n_{\mathrm{ext},k}}{n_\mathrm{ext}}\right) \\ &\qquad\cdot\left(\frac{1}{n_{\mathrm{b},k}}\sum_{s=0}^{n_{\mathrm{b},k}-1} l(\boldsymbol{p}_s,k)\right) \\ &= \frac{1}{K}\sum_{k=0}^{K-1}\left(\frac{1}{n_{\mathrm{b},k}}\sum_{s=0}^{n_{\mathrm{b},k}-1} l(\boldsymbol{p}_s,k)\right) \end{aligned} \tag{21}$$

4 DAMAGE DETECTION MODEL

4.1 Convolutional network for semantic segmentation

In general, the architecture of CNNs used for semantic segmentation is composed of the encoder, decoder and a series of skip-layer data flows, such as skip connections used in U-Net (Ronneberger et al., 2015) and pooling indices used in SegNet (Badrinarayanan et al., 2017). In semantic segmentation, the encoder extracts visual features of multiple scales and coarsely predicts the categories of each region in the input image. Such prediction is then refined by the decoder to the size of the input. The information of multiscale local features is transferred through skip-layer data flows from the encoder to the decoder to assist the reconstruction of a finer prediction with respect to the local boundaries between the areas of various categories.

Major differences among various CNNs for segmentation lie in the design of the decoders and skip-layer data flows, where different techniques are utilized to refine the prediction generated by the encoder. The state-of-art architectures including FCN (Long et al., 2015), U-

Net (Ronneberger et al., 2015), DeepLab (Chen et al., 2017b) and SegNet (Badrinarayanan et al., 2017) were examined and compared during the pre-test. The 8-time ($8\times$) interpolation, applied in DeepLab to construct the final prediction, results in the incapability of DeepLab of capturing detailed boundaries of delicate objects (Chen et al., 2017b), such as the narrow, line-shaped areas of cracks. Different types of skip-layer data flows are utilized in FCN, SegNet and U-Net: In FCN, multiscale features in the encoder are compressed through convolutional layers before transferred to the decoder; in SegNet, pooling indices of pooling layers are transferred from the encoder to the decoder to assist prediction refinement; and in U-Net, multiscale features extracted in the encoder are directly transferred and concatenated with features in the decoder with modification or compression. Compared with FCN and SegNet, the type of skip-layer data flows used in U-Net involves more parameters to be trained, which makes U-Net less efficient. Nevertheless, U-Net is selected for this study because the intact feature transferring is beneficial for detecting delicate boundaries of cracks and rebars.

4.2 Damage-Net

In this study, a deep CNN architecture, named as Damage-Net, is proposed for visible damage detection of RC components. The architecture is based on U-Net, while adaptive improvements are carried out in terms of flexibility and training efficiency. First, the output prediction in U-Net is not of the same size as the input image, which makes the data pre- and post-processing cumbersome. In the proposed Damage-Net, layer configurations, such as padding size and stride size, are adjusted to ensure that the output size is equal to the input size. Second, the encoder of U-Net is not inherited from any known architecture of classification CNNs, therefore the parameters need to be trained from scratch. Training from scratch requires more training data, and may be confronted with issues like over-fitting and slow convergence. Damage-Net inherits its encoder from the convolutional layers of VGG-16 (Simonyan and Zisserman, 2014), a deep CNN that achieved excellent performance on the large-scale, general-purpose dataset ImageNet. The adaptation from VGG-16 enables Damage-Net to conduct transfer learning, which ensures the training on relatively small-scale datasets to gain improved convergence and high efficiency.

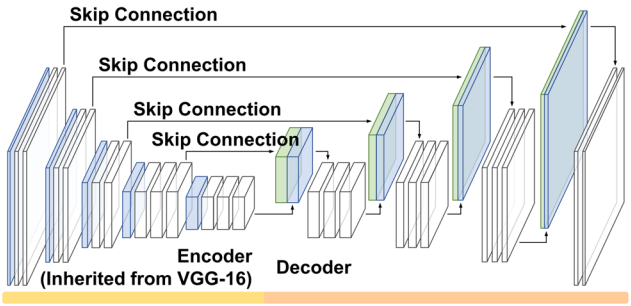


FIGURE 5. The architecture of the proposed Damage-Net.

The architecture of Damage-Net is illustrated in Figure 5. Four skip connections are introduced to effectively integrate multiscale information for delicate boundary construction. Damage-Net has 28.8 million parameters in total, and 14.7 million parameters are inherited from VGG-16.

4.3 Training of Damage-Net

Two CNN models, denoted as 4Category-Net and Crack-Net, are optimized separately by the training set of 4Category datasets and Crack datasets, respectively. The models are evaluated on aspects of resource consumption (i.e., computing time and memory usage) and recognition performance. The recognition performance of a model is evaluated on the test set using well-known metrics including per-category intersection-over-union (perIoU) and mean intersection-over-union (MIoU). The detailed formulation and explanation of these metrics can be found in Garcia-Garcia et al. (2017). In this study, the MIoU was considered as the major indicator for evaluation of the accuracy, since it is arguably the most used and accepted metric due to its representativeness and simplicity.

The proposed damage detection, consisting of dataset generation, data balancing and training and inference of CNNs (see Figure 4), was implemented in Python3 programming language, where modules of image transformations for dataset generation were coded based on scikit-image library (Van der Walt et al., 2014), and CNN-related modules were coded based on PyTorch library (Paszke et al., 2019).

4.3.1 Application of transfer learning

Transfer learning (Yosinski et al., 2014) is a technique for training CNNs on relatively small-scale datasets with limited computing resources. In brief, transfer learning is the application of certain parameters from a pretrained model into the target model, where part of the architecture

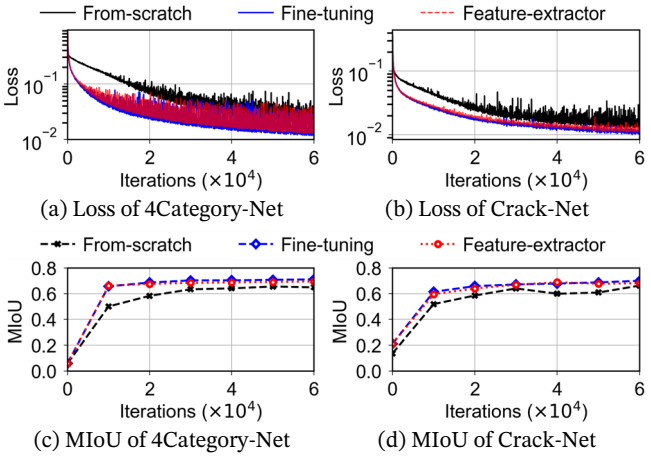
is shared between the two models. Most commonly, the pretrained model is the one trained on large-scale datasets (e.g., ImageNet), and thus tends to preserve broad generalization and possess excellent recognition of basic visual features. There are two strategies of transfer learning that can be deployed while training a CNN. The first one is referred to as the fine-tuning strategy, where parameters transferred from the pretrained model are updated iteratively using a small learning rate. The second one is called the feature-extractor strategy, where transferred parameters are fixed (i.e., frozen), and only the newly configured and randomly initialized parameters of the target model are optimized during the training.

Numerical tests were carried out to analyze the differences in performance, computing time and memory usage among the CNNs trained without transfer learning (i.e., from scratch) and with two different transfer learning strategies. The models were trained on the restricted datasets under the same configuration of base learning rate $\alpha = 1 \times 10^{-5}$, maximum iterations $n = 60000$, 32 batch size and RMSProp algorithm with hyper-parameter $\delta = 0.99$. To prevent overfitting, L2 regularization was added to the loss function, where the regularization strength was set to 0.002. The difference among these models was how to initialize and update the parameters. As for the from-scratch model, all the parameters were randomly initialized and iteratively optimized during training. For the two transfer-learning models, transferred parameters were initialized from the corresponding VGG-16 layers, while the rest were randomly initialized. During the training, transferred parameters in the fine-tuning model were updated with a smaller learning rate (i.e., 1% of the base learning rate), and the ones in the feature-extractor model were fixed.

Comparisons of resource consumption and recognition performance among the three models are shown in Table 4 and Figure 6. While the from-scratch model was less accurate, both transfer-learning models showed considerably similar detection accuracy. For computational efficiency, the feature-extractor strategy is more favorable, since this strategy resulted in a training time at least 25% shorter and a memory usage 25% less than training from scratch or the fine-tuning strategy.

TABLE 4. Computing resources required by models trained from scratch, with fine-tuning and feature-extractor strategy.

Network	*Strategy*	*Computing time per iteration / ms*	*Memory usage per sample / Mb*
4Category-Net	From scratch	1647	820
	Fine-tuning	1760	820
	Feature-extractor	1227	626
Crack-Net	From scratch	1600	813
	Fine-tuning	1707	813
	Feature-extractor	1220	619



(a) Loss of 4Category-Net (b) Loss of Crack-Net

(c) MIoU of 4Category-Net (d) MIoU of Crack-Net

FIGURE 6. Performance comparison of models trained from scratch, with fine-tuning and feature-extractor strategy.

4.3.2 Benefits of background-extended datasets

Two types of models were trained to demonstrate the effects of the extended datasets. The model 4Category_Rst-Net and Crack_Rst-Net (i.e., the Rst-Nets) were trained on the restricted dataset 4Category-DS and Crack-DS, respectively, while the model 4Category_Ext-Net and Crack_Ext-Net (i.e., the Ext-Nets) were trained simultaneously on the paired restricted and extended datasets, where the batch was constructed through stratified sampling. Curves of the loss function value and the MIoU on the test set of restricted datasets are plotted in Figure 7, which indicates comparable performance on damage recognition between the 4Category_Rst-Net (Crack_Rst-Net) and 4Category_Ext-Net (Crack_Ext-Net). In order to compare background recognition performance of Rst-Nets and Ext-Nets, several photos that were not contained in the database were used for damage detection by these models in Figure 8. A large photo was divided into a number of standard-size patches using overlap-tile strategy adopted from Ronneberger et al. (2015). The patches were analyzed by the trained models, and finally assembled as an entire output image. The fact

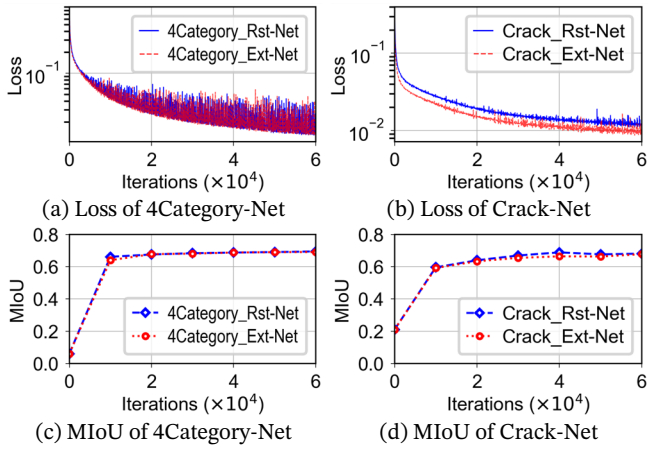


FIGURE 7. Performance comparison of models trained with only Rst-Dataset and with both Rst- and Ext-Dataset.

that the Ext-Nets surpassed Rst_Nets in distinguishing background from damage areas (see the areas highlighted with dashed-line boxes in Figure 8) suggests the success of the method proposed to improve background recognition.

4.4 Performance evaluation of Damage-Net

Several models were trained using Adam and RMSProp algorithms separately, with various training configuration (i.e., the learning rate, the hyper-parameters of Adam and RMSProp). Afterwards, the models that had the highest MIoU on the corresponding test sets were selected as the final 4Category-Net and Crack-Net used for subsequent analysis in this paper. For the final 4Category-Net and Crack-Net, metrics on the test sets of 4Category-DS and 4Category_Ext-DS, Crack-DS and Crack_Ext-DS are summarized in Table 5, respectively. Both models achieved

TABLE 5. Performance of the final models.

Network	*Category*	*perIoU /%*	*Category*	*perIoU /%*
4Category-Net	Background	97.97	Crushing	66.03
	Spalling	71.39	Buckling and Fracture	71.19
	Exposure	49.01		
	MIoU /%	71.12		
Crack-Net	Background	98.59	Crack	41.63
	MIoU /%	70.11		

satisfactory performance with MIoU over 70%. However, as can be noticed from Table 5, the recognition performance was lower for the damage categories than for background category. Further investigation is required in terms of architecture design and training strategies to improve the performance of the proposed 4Category-Net and Crack-Net, especially for the detection of exposed rebars and cracks. Several samples from the test sets of 4Category-DS and Crack-DS are demonstrated in Figure 9 to visualize the recognition performance of the 4Category-Net and Crack-Net.

4.5 Post-processing

Crack width is an important index to estimate the damage state of RC components. Crack-Net can detect cracks and track crack paths, but it has difficulties in delineating the boundaries of cracks. Crack-Net tends to extract a crack together with borders surrounding it, and thereby overestimates the crack width. A post-processing technique is proposed in this study to cope with this issue.

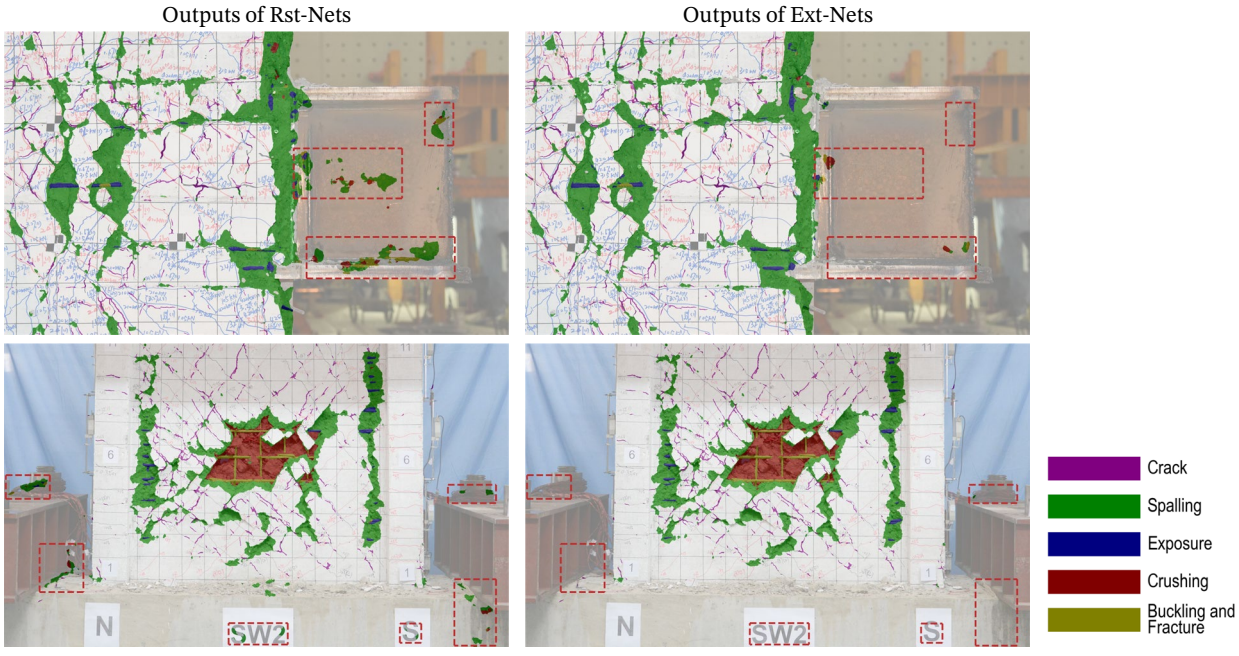


FIGURE 8. The outputs of photos predicted by Rst-Nets and Ext-Nets.

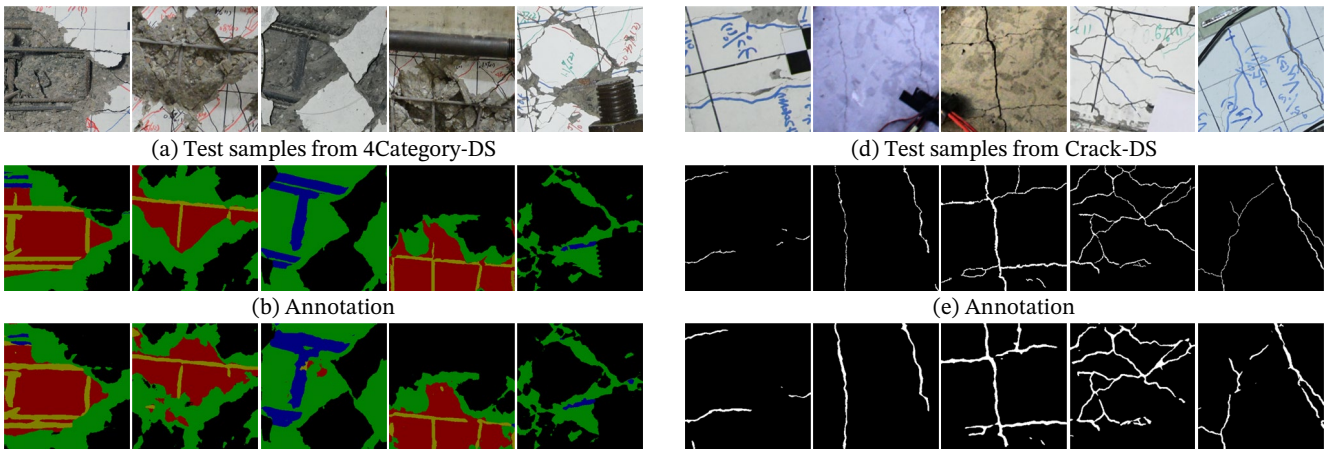


FIGURE 9. Visualization of test samples.

Figure 10 illustrates the flowchart of the proposed post-processing technique. The basic idea is to use the output of Crack-Net as a mask for crack detection, and the boundaries of cracks can further be determined through the contrast of gray levels between crack-pixels and background-pixels. The original image and the output of Crack-Net are overlapped to construct a masked image (see Image II in Figure 10), and then Histogram Equalization with Mask algorithm (Scikit-image, 2019) is conducted to improve the global contrast of the masked image. Afterwards, Otsu's thresholding (Otsu, 1979) is applied to the equalized masked image (see Image III in Figure 10) to refine the boundaries of the detected cracks. The proposed technique is named Threshold after Histogram Equalization (TaHE).

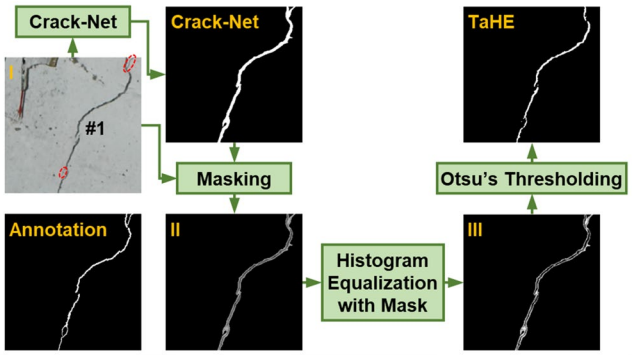


FIGURE 10. Flowchart of the proposed Threshold after Histogram Equalization (TaHE).

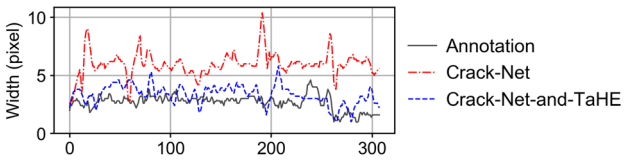
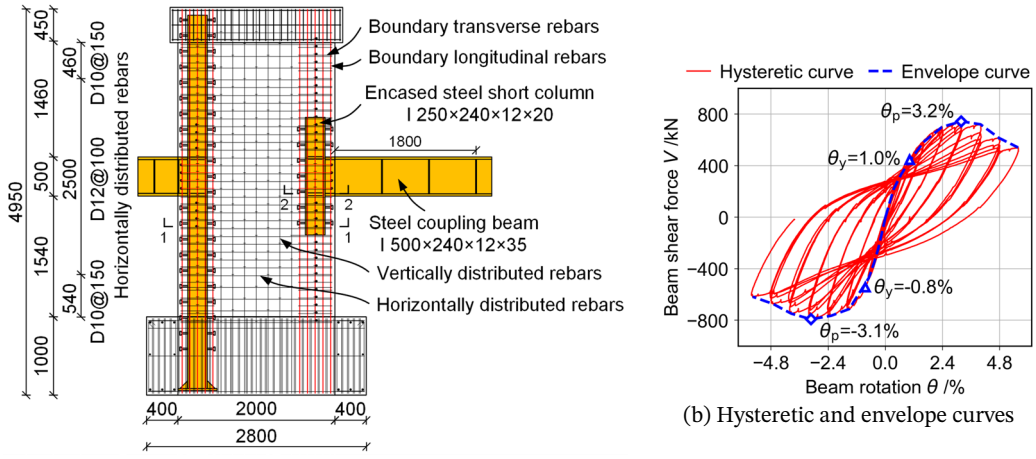


FIGURE 11. Comparison of crack width analyzed from Annotation, Crack-Net and Crack-Net-and-TaHE.

Afterwards, a filter-based algorithm (Ji et al., 2020) is used to quantify the geometric properties of the detected cracks. Figure 11 compares the width of cracks depicted in the images of (manual) annotation, (the output of) Crack-Net and (the result of) Crack-Net-and-TaHE in Figure 10. Through the proposed post-processing, the Crack-Net-and-TaHE result correlated well with the annotation (i.e., the ground truth), while the pure Crack-Net output led to significant overestimate of the crack width. The analysis results suggest the effectiveness of the proposed TaHE. Note that, subtle cracks, circled with red dashed lines in Image I in Figure 10, tend to be removed by TaHE, although are well detected by Crack-Net. Since wide cracks, compared with subtle cracks, have a dominating influence in the damage state estimation of RC components, omission of subtle cracks would not affect the post-earthquake safety assessment results.

5 APPLICATION ON STRUCTURAL SPECIMEN PHOTOS

The ability of the proposed vision-based models to detect damage and estimate damage states is demonstrated based on a series of test photos. The photos were taken from full-scale quasi-static tests conducted to study the cyclic behavior of a steel beam-to-RC wall joint (Leong, 2017). In the test specimen, a steel coupling beam was anchored to a short steel column embedded in the boundary element of a RC wall. Figure 12(a) shows the reinforcement details of the joint whose strength was designed according to Ji et al. (2019). The wall was rigidly clamped to the reaction floor, while cyclic shear load was applied to the steel cantilever beam to produce loading environment of a beam-to-wall



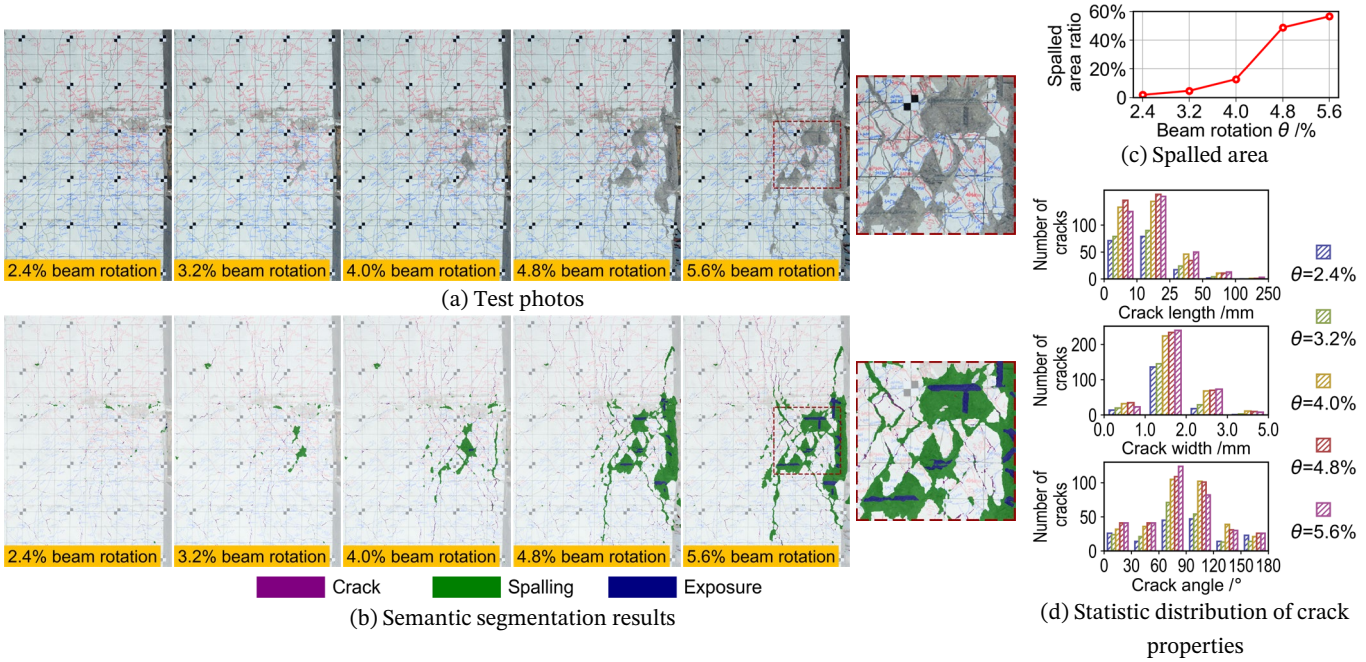
(a) Geometric dimensions and reinforcement details (unit: mm)

(b) Hysteretic and envelope curves

FIGURE 12. Reinforcement details and cyclic test results of beam-to-wall specimen.

joint. Figure 12(b) presents the hysteretic and envelope curves of the beam shear force-beam rotation relationship obtained from the test. The rotations at yield load and peak load (i.e., θ_y and θ_p) are identified in the figure. The specimen failed in panel shear failure mode of the joint, and the observed damage included concrete cracking and concrete spalling in the joint panel, and exposure of reinforcement.

A series of photos at various loading levels (see Figure 13(a)) were used as the inputs to Damage-Net for pixel-level damage detection. Note that in post-earthquake assessment, the visible damage is obtained after the earthquake shaking, rather than at the peak transient displacement. To mimic the assessment condition, each loading level was represented by a photo taken at the unloaded point after completing the cycles. Perspective transformation was utilized as pre-processing for lens distortion correction of the original photos, and the conversion factor between pixel-unit and engineering-unit was further derived from the corrected photos via markers mounted on the specimen. The photos were analyzed by 4Category-Net and Crack-Net separately, and the outputs were integrated to produce the ultimate semantic segmentation results visualized in Figure 13(b). Afterwards, the post-processing technique TaHE was applied to refine the crack boundaries. Cracks and spalling of concrete, which were prominent in this test specimen, were further quantified from the Damage-Net output. The spalled area was calculated by summing the pixels of the spalling category detected by 4Category-Net from each test photo. The development of spalled area at various loading levels is plotted in Figure 13(c), where the spalled area ratio



(a) Test photos

(b) Semantic segmentation results

(c) Spalled area

(d) Statistic distribution of crack properties

FIGURE 13. Application to a beam-to-wall joint test.

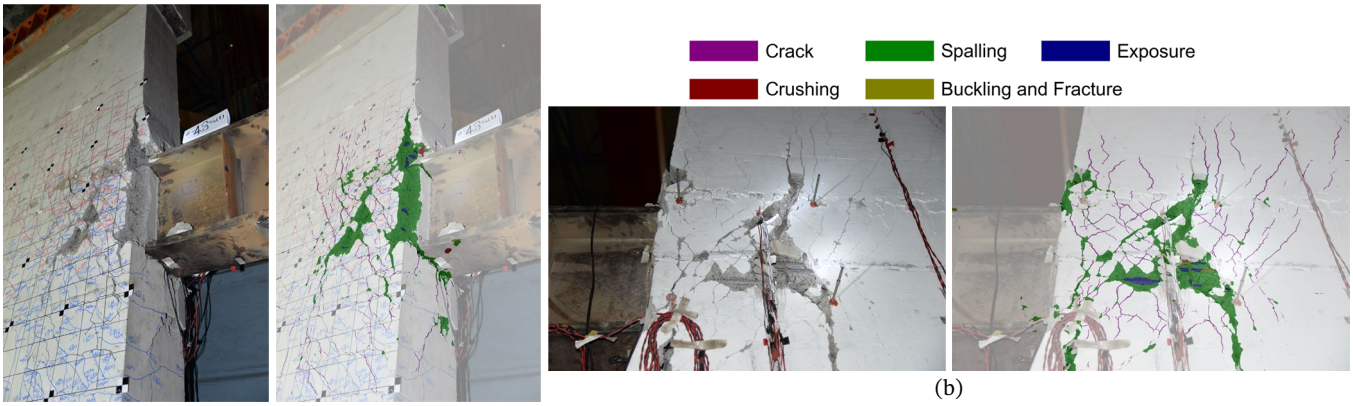


FIGURE 14. Damage detection using photos with unideal shooting conditions.

is calculated as the portion of spalled area in the joint panel. Labeling and quantification of cracks needs specific image processing techniques. In this study, the crack fields identified by Crack-Net and TaHE were further separated, labeled and quantified by the crack quantification algorithms developed by the authors (Ji et al., 2020). Using the algorithms, geometric properties (i.e., crack length, width and angle) of each crack were calculated. Statistical distributions of these geometric properties of cracks are shown in Figure 13(d).

Per Japanese provisions (MLIT, 2015), damage states of components or joints can be determined through visible damage (as illustrated in Figure 1). At 2.4% and 3.2% beam rotation, the majority of cracks were 1-2 mm in width and spalled area ratio was less than 10%, which corresponds to damage state DS III (see Figure 1). At 4.0% beam rotation, the spalled area ratio increased to 15% and horizontal reinforcement was exposed. At 4.8% and 5.6% beam rotation, spalled area ratio reached approximately 50%, and multiple rebars were exposed. However, neither crushing of concrete nor buckling and fracture of rebar was detected. Therefore, the specimen, at 4.0% to 5.6% beam rotation, corresponds to DS IV per Japanese provisions (see Figure 1). On the other hand, the damage states can be examined from the envelope curve in Figure 12(b) as well. The beam rotations of 2.4% and 3.2% fall into the yield-to-peak strength stage, belonging to DS III inferred from Figure 1. The beam rotations of 4.0%, 4.8% and 5.6% fall into the post-peak strength stage, belonging to DS IV. The damage states estimated from the vision-based damage detection based on Japanese provisions' criteria correlate well with those evaluated from the cyclic response data of the specimen. The success of the vision-based approach in this case study indicates a potential of its application in post-earthquake damage estimation of RC components.

In addition, the robustness of the proposed approach was validated using photos under various shooting conditions. Figure 14 shows two examples, where one was taken from the side with skewed angle and another was taken from the back of the beam-to-wall joint specimen. Obstructions, non-uniform lighting and random shadows can be observed in the photo of Figure 14(b). The detection results are comparable to those of photos with ideal shooting conditions (see Figure 13), which suggests that the proposed vision-based approach is capable of localizing, classifying and segmenting seismic damage with favorable accuracy even under unideal shooting conditions. Further study is warranted to extend the robustness of the approach to a wider range of unideal shooting conditions.

6 CONCLUSIONS

In this research, a novel vision-based approach is presented for semantic segmentation of seismic damage of RC structural components. A database that comprises pixel-level multicategory annotated images of RC test specimens was constructed for semantic segmentation of visible damage. Algorithms of computer vision were deployed to achieve pixel-level detection of multiple seismic damage categories of RC components, including cracking, concrete spalling and crushing, reinforcement exposure, buckling and fracture. The proposed approach was applied to a series of test photos of a beam-to-wall joint that was loaded to produce various damage categories, to validate its accuracy and effectiveness. The following conclusions are obtained from this study.

(1) Data balancing techniques at the sample and pixel levels were proposed/developed to generate a series of

datasets from the database. Examples were shown to demonstrate how these techniques mitigate data imbalance and facilitate the convolutional networks to achieve balanced recognition performance for each damage category.

(2) The ability of the CNNs to distinguish the complex background from the target damage categories was improved by generating separate datasets for background category and utilizing stratified sampling to construct the training batches.

(3) A CNN architecture for pixel-level damage detection, i.e., Damage-Net, was developed based on the state-of-art VGG-16 and U-Net. Through the application of transfer learning, improved recognition performance was achieved yet with savings of 25% computing time and 25% memory usage, compared with training the CNNs from scratch.

(4) Two models, i.e., Crack-Net and 4Category-Net, were optimized separately, the former to detect cracks and the latter to detect other four damage categories, i.e., concrete spalling and crushing, reinforcement exposure, buckling and fracture. Both models achieved satisfactory performance with a mean intersection-over-union (MIoU) over 70% (70.1% for Crack-Net and 71.1% for 4Category-Net).

(5) An effective yet simple post-processing technique, i.e., Threshold after Histogram Equalization (TaHE), was developed to refine the boundaries of cracks detected by Crack-Net, and thereby enable subsequent characterization of cracks to be accurate.

Room for improvement exists in the proposed vision-based damage detection: (1) Field-survey photos should be included into the database and delicate transfer learning techniques should be investigated, in order to enable damage detection for real post-earthquake field surveys; (2) Effective design and training strategies should be explored to improve the recognition performance of the proposed CNNs, especially for exposure of reinforcement category; (3) The number of parameters in Damage-Net is relatively large, therefore, more compact architectures should be further developed to decrease the computational costs of application.

This study contributes to the development of a vision-based safety assessment system for RC buildings. The system is aimed at providing quantitative damage detection results based on post-earthquake field-survey photos, and assisting engineers and inspectors to achieve an efficient and accurate safety assessment of damaged building structures. A major challenge is to construct a framework to correlate damage information at the component level with residual performance at the structure level. Two key issues need further investigation to construct such framework: (1) how to correlate the detected visible damage with degradation in mechanical properties (including stiffness, strength and deformation capacity) of the damaged components, and (2) how to correlate the degradation of individual structural components with the residual capacity of the structural system. Newly developed data-driven classification and regression algorithms (e.g., Ahmadlou and Adeli, 2010; Alam et al., 2020; Pereira et al., 2020; Rafiei and Adeli, 2017) shall be examined as possible tools for resolving these issues.

ACKNOWLEDGEMENT

The work presented in this paper was sponsored by the funds from the National Key R&D Program of China (Grant No. 2017YFC1500602), the NSFC-JSPS International Joint Research Program (Grant No. 51811540032) and Tsinghua University Initiative Scientific Research Program (Grant No. 20193080019). The financial support is sincerely appreciated.

REFERENCES

Ahmadlou M. & Adeli H. (2010). Enhanced probabilistic neural network with local decision circles: A robust classifier. *Integrated Computer-Aided Engineering*, *17*, 197-210.

Alam K. M. R., Siddique N. & Adeli H. (2020). A dynamic ensemble learning algorithm for neural networks. *Neural Computing and Applications*, *32*(12), 8675-8690.

Badrinarayanan V., Kendall A. & Cipolla R. (2017). SegNet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *39*(12), 2481-2495.

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

Beckman G. H., Polyzois D. & Cha Y.-J. (2019). Deep learning-based automatic volumetric damage quantification using depth camera. *Automation in Construction*, *99*, 114-124.

Buda M., Maki A. & Mazurowski M. A. (2018). A systematic study of the class imbalance problem in convolutional neural networks. *Neural Networks*, *106*, 249-259.

Cha Y. J., Choi W. & Büyüköztürk O. (2017). Deep learning-based crack damage detection using convolutional neural networks. *Computer-Aided Civil and Infrastructure Engineering*, *32*(5), 361-378.

Chen F. C., Jahanshahi M. R., Wu R. T. & Joffe C. (2017a). A texture-based video processing methodology using Bayesian data fusion for autonomous crack detection on metallic surfaces. *Computer-Aided Civil and Infrastructure Engineering*, *32*(4), 271-287.

Chen L. C., Papandreou G., Kokkinos I., Murphy K. & Yuille A. L. (2017b). DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *40*(4), 834-848.

Chen P. H., Shen H. K., Lei C. Y. & Chang L. M. (2012). Support-vector-machine-based method for automated steel bridge rust assessment. *Automation in Construction*, *23*, 9-19.

Cheng H. D., Shi X. J. & Glazier C. (2003). Real-time image thresholding based on sample space reduction and interpolation approach. *Journal of Computing in Civil Engineering*, *17*(4), 264-272.

Chida H. & Takahashi N. (2020). Study on image diagnosis of timber houses damaged by earthquake using deep learning. *Journal of Structural and Construction Engineering (Transactions of AIJ)*, *85*(770), 529-538. (in Japanese)

Choi W. & Cha Y. J. (2020). SDDNet: Real-time crack segmentation. *IEEE Transactions on Industrial Electronics*, *67*(9), 8016-8025.

CMC (2015): *Specification for Seismic Test of Buildings* (JGJ/T 101-2015). Beijing: China Ministry of Construction. (in Chinese)

CMC (2016): *Technical Guide for Post-Earthquake Emergency Assessment of Building Structures (2016 revision)* Beijing: China Ministry of Construction. (in Chinese)

Eigen D. & Fergus R. (2015). Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. *Proceedings of 2015 IEEE International Conference on Computer Vision*, Las Condes, Chile, 2650-2658.

FEMA (1998). *Evaluation of Earthquake Damaged Concrete and Masonry Wall Buildings: Basic Procedures Manual* (FEMA-306). Washington D.C.: Federal Emergency Management Agency.

FEMA (2011). *Fragility Functions for Slender Reinforced Concrete Walls* (FEMA P-58/BD-3.8.9). Washington, D.C.: Federal Emergency Management Agency.

Fujita Y. & Hamamoto Y. (2011). A robust automatic crack detection method from noisy concrete surfaces. *Machine Vision and Applications*, *22*(2), 245-254.

Gao Y. & Mosalam K. M. (2018). Deep transfer learning for image-based structural damage recognition. *Computer-Aided Civil and Infrastructure Engineering*, *33*(9), 748-768.

Garcia-Garcia A., Orts-Escolano S., Oprea S., Villena-Martinez V. & Garcia-Rodriguez J. (2017). A review on deep learning techniques applied to semantic segmentation. https://arxiv.org/abs/1704.06857.

German S., Jeon J. S., Zhu Z., Bearman C., Brilakis I., DesRoches R., et al. (2013). Machine vision-enhanced postearthquake inspection. *Journal of Computing in Civil Engineering*, *27*(6), 622-634.

Hoskere V., Narazaki Y., Hoang T. A. & Spencer B. F. (2018). Towards automated post-earthquake inspections with deep learning-based condition-aware models. *Proceedings of the 7th World Conference on Structural Control and Monitoring*, Qingdao, China.

Ishii Y., Matsuoka M., Maki N., Horie K. & Tanaka S. (2018). Recognition of damaged building using deep learning based on aerial and local photos taken after the 1995 Kobe earthquake. *Journal of Structural and Construction Engineering (Transactions of AIJ)*, *83*(751), 1391-1400. (in Japanese)

Iyer S. & Sinha S. K. (2006). Segmentation of pipe images for crack detection in buried sewers. *Computer-Aided Civil and Infrastructure Engineering*, *21*(6), 395-410.

Jahanshahi M. R., Masri S. F., Padgett C. W. & Sukhatme G. S. (2013). An innovative methodology for detection and quantification of cracks through incorporation of depth perception. *Machine vision and applications*, *24*(2), 227-241.

Jang K., An Y.-K., Kim B. & Cho S. (2020). Automated crack evaluation of a high-rise bridge pier using a ring-type climbing robot. *Computer-Aided Civil and Infrastructure Engineering*, https://doi.org/10.1111/mice.12550.

JBDPA (1997): *Emergency Risk Assessment Manual for Post-Earthquake Building Structures* Tokyo: Japan Building Disaster Prevention Association. (in Japanese)

Ji X., Cheng Y., Leong T. & Cui Y. (2019). Seismic behavior and strength capacity of steel coupling beam-to-SRC wall joints. *Engineering Structures*, *201*, 109820.

Ji X., Miao Z. & Kromanis R. (2020). Vision-based measurements of deformations and cracks for RC structure tests. *Engineering Structures*, *212*, 110508.

Kingma D. P. & Ba J. (2015). Adam: A method for stochastic optimization. *Proceedings of the 3rd International Conference for Learning Representations*, San Diego, USA.

Leong T. (2017). *Study on seismic behavior and design of connections between steel coupling beams and wall piers.* Tsinghua University, Beijing. (in Chinese)

Li G., Zhao X., Du K., Ru F. & Zhang Y. (2017). Recognition and evaluation of bridge cracks with modified active contour model and greedy search-based support vector machine. *Automation in Construction*, *78*, 51-61.

Li R., Yuan Y., Zhang W. & Yuan Y. (2018). Unified vision-based methodology for simultaneous concrete defect detection and geolocalization. *Computer-Aided Civil and Infrastructure Engineering*, *33*(7), 527-544.

Liang X. (2019). Image-based post-disaster inspection of reinforced concrete bridge systems using deep learning with Bayesian optimization. *Computer-Aided Civil and Infrastructure Engineering*, *34*(5), 415-430.

Long J., Shelhamer E. & Darrell T. (2015). Fully convolutional networks for semantic segmentation. *Proceedings of 2015 IEEE Conference on Computer Vision and Pattern Recognition*, Boston, USA, 3431-3440.

Maeda H., Sekimoto Y., Seto T., Kashiyama T. & Omata H. (2018). Road damage detection and classification using deep neural networks with smartphone images. *Computer-Aided Civil and Infrastructure Engineering*, *33*(12), 1127-1141.

MLIT (2015): *Technical Guide for Damage Estimation and Restoration of Post-Earthquake Building Structures (2015 revision)* Tokyo: Japan Building Disaster Prevention Association. (in Japanese)

Nguyen H. N., Kam T. Y. & Cheng P. Y. (2014). An automatic approach for accurate edge detection of concrete crack utilizing 2D geometric features of crack. *Journal of Signal Processing Systems*, *77*(3), 221-240.

Ni F., Zhang J. & Chen Z. (2019). Zernike-moment measurement of thin-crack width in images enabled by dual-scale deep learning. *Computer-Aided Civil and Infrastructure Engineering*, *34*(5), 367-384.

Nishikawa T., Yoshida J., Sugiyama T. & Fujino Y. (2012). Concrete crack detection by multiple sequential image filtering. *Computer-Aided Civil and Infrastructure Engineering*, *27*(1), 29-47.

O'Byrne M., Ghosh B., Schoefs F. & Pakrashi V. (2014). Regionally enhanced multiphase segmentation technique for damaged surfaces. *Computer-Aided Civil and Infrastructure Engineering*, *29*(9), 644-658.

Oliveira H. & Correia P. L. (2008). Supervised strategies for cracks detection in images of road pavement flexible surfaces. *Proceedings of the 16th European Signal Processing Conference*, Lausanne, Switzerland.

Otsu N. (1979). A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, *9*(1), 62-66.

Paszke A., Gross S., Massa F., Lerer A., Bradbury J., Chanan G., et al. (2019). Pytorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems 32*, Vancouver, Canada, 8026-8037.

Pereira D. R., Piteri M. A., Souza A. N., Papa J. P. & Adeli H. (2020). FEMa: a finite element machine for fast learning. *Neural Computing and Applications*, *32*(10), 6393-6404.

Rafiei M. H. & Adeli H. (2017). A new neural dynamic classification algorithm. *IEEE Transactions on Neural Networks and Learning Systems*, *28*(12), 3074-3083.

Ronneberger O., Fischer P. & Brox T. (2015). U-Net: Convolutional networks for biomedical image segmentation. *Proceedings of the 18th International Conference on Medical Image Computing and Computer Assisted Intervention*, Munich, Germany, 234-241.

Sajedi S. O. & Liang X. (2020). Uncertainty-assisted deep vision structural health monitoring. *Computer-Aided Civil and Infrastructure Engineering*, https://doi.org/10.1111/mice.12580.

Scikit-image (2019): *Histogram Equalization*. Retrieved from https://scikit-image.org/docs/dev/auto_examples/color_exposure/plot_equalize.html.

Simonyan K. & Zisserman A. (2014). Very deep convolutional networks for large-scale image recognition. *Proceedings of the 3rd International Conference on Learning Representations*, San Diego, USA.

Tieleman T. & Hinton G. (2012): Lecture 6.5-RMSProp: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*.

Van der Walt S., Schönberger J. L., Nunez-Iglesias J., Boulogne F., Warner J. D., Yager N., et al. (2014). Scikit-image: Image processing in Python. *PeerJ*, *2*, e453.

Xiong C., Li Q. & Lu X. (2020). Automated regional seismic damage assessment of buildings using an unmanned aerial vehicle and a convolutional neural network. *Automation in Construction*, *109*, 102994.

Yamaguchi T., Nakamura S., Saegusa R. & Hashimoto S. (2008). Image-based crack detection for real concrete surfaces. *IEEJ Transactions on Electrical and Electronic Engineering*, *3*(1), 128-135.

Yeum C. M., Dyke S. J. & Ramirez J. (2018). Visual data classification in post-event building reconnaissance. *Engineering Structures*, *155*, 16-24.

Ying L. & Salari E. (2010). Beamlet transform-based technique for pavement crack detection and classification. *Computer-Aided Civil and Infrastructure Engineering*, *25*(8), 572-580.

Yosinski J., Clune J., Bengio Y. & Lipson H. (2014). How transferable are features in deep neural networks? *Advances in Neural Information Processing Systems 27*, Montreal, Canada, 3320–3328.

Zhang A., Wang K. C. P., Fei Y., Liu Y., Tao S., Chen C., et al. (2018). Deep learning-based fully automated pavement crack detection on 3D asphalt surfaces with an improved CrackNet. *Journal of Computing in Civil Engineering*, *32*(5), 04018041.

Zou Q., Zhang Z., Li Q., Qi X., Wang Q. & Wang S. (2019). DeepCrack: Learning hierarchical convolutional features for crack detection. *IEEE Transactions on Image Processing*, *28*(3), 1498-1512.