DOI: 10.1111/mice.12488

ORIGINAL ARTICLE

COMPUTER-AIDED CIVIL AND INFRASTRUCTURE ENGINEERING WILEY

Autonomous damage segmentation and measurement of glazed tiles in historic buildings via deep learning

Niannian Wang[1] | **Xuefeng Zhao**[1] | **Zheng Zou**[1] | **Peng Zhao**[2] | **Fei Qi**[2]

[1]School of Civil Engineering, Dalian University of Technology, Dalian, China

[2]Department of Architectural Heritage, The Palace Museum, Beijing, China

Correspondence
Xuefeng Zhao, School of Civil Engineering, Dalian University of Technology, Dalian 116024, LN, China.
Email: zhaoxf@dlut.edu.cn

Funding information
National Natural Science Foundation of China, Grant/Award Number: 51479031; The Palace Museum, Grant/Award Number: 2018-308

Abstract

Detecting and measuring the damage on historic glazed tiles plays an important role in the maintenance and protection of historic buildings. However, the current visual inspection method for identifying and assessing superficial damage on historic buildings is time and labor intensive. In this article, a novel two-level object detection, segmentation, and measurement strategy for large-scale structures based on a deep-learning technique is proposed. The data in this study are from the roof images of the Palace Museum in China. The first level of the model, which is based on the Faster region-based convolutional neural network (Faster R-CNN), automatically detects and crops two types of glazed tile photographs from 100 roof images (2,488 × 3,264 pixels). The average precision values (AP) for roll roofing and pan tiles are 0.910 and 0.890, respectively. The cropped images are used to form a dataset for training a Mask R-CNN model. The second level of the model, which is based on Mask R-CNN, automatically segments and measures the damage based on the cropped historic tile images; the AP for the damage segmentation is 0.975. Based on Mask R-CNN, the predicted pixel-level damage segmentation result is used to quantitatively measure the morphological features of the damage, such as the damage topology, area, and ratio. To verify the performance of the proposed method, a comparative study was conducted with Mask R-CNN and a fully convolutional network. This is the first attempt at employing a two-level strategy to automatically detect, segment, and measure large-scale superficial damage on historic buildings based on deep learning, and it achieved good results.

1 | INTRODUCTION

Periodic damage detection plays a crucial role in the maintenance and protection of existing buildings and infrastructure, especially for historic buildings (Kordatos, Exarchos, Stavrakos, Moropoulou, & Matikas, 2013; Ou & Li, 2010; Yan, Cheng, Wu, & Yam, 2007). Since the structural surface of historic buildings is degraded due to natural disasters and human factors, periodic inspection is necessary every year. The tiles of China's historic buildings are mainly yellow glazed tiles, which have glazed surfaces and provide excellent waterproofing. However, after hundreds of years, there are varying degrees of damage to the tiles. The damaged tiles not only affect the aesthetics of historic buildings but also reduce the waterproofing function of the roof (Botas, Veiga, & Velosa, 2017). However, the current inspections of infrastructure and historic buildings are carried out visually, which is time and labor intensive (Elmasry & Johnson, 2004; Gattulli & Chiaramonte, 2005; O'Byrne, Schoefs, Ghosh, & Pakrashi, 2013).

Machine vision techniques offer new and more effective solutions for visual inspection. Recently, many studies on machine vision techniques were conducted with the goal of replacing conventional visual inspection. For example, a robust automated image processing method based on multiple sequential image filtering was used to detect superficial cracks on concrete structures (Nishikawa, Yoshida, Sugiyama, & Fujino, 2012). Digital image correlation and feature tracking techniques were used to investigate the evolution of strain and deformation during uniaxial tensile tests and shear debonding tests in Fiber Reinforced Polymer (FRP)-masonry systems (Ghiassi, Xavier, Oliveira, & Lourenço, 2013). Automatic crack detection and classification methods that leveraged complementary metal-oxide semiconductor industrial cameras were used in a machine vision application for defect detection (Zhang, Zhang, Qi, & Liu, 2014). A vision-based automatic crack detection technique was proposed to inspect bridges (Prasanna et al., 2016; Yeum & Dyke, 2015; Zhu, German, & Brilakis, 2010). However, these methods have some limits, such as manual feature extraction and sensitivity to noise (stains, shadows, and nonuniform lighting conditions) (Koziarski & Cyganek, 2017; Ortega-Zamorano, Jerez, Gómez, & Franco, 2017). Hence, the detection results are unsatisfactory, and no optimization method exists.

Other machine learning methods have been studied to improve detection accuracy (Adeli & Yeh, 1989; Jiang & Adeli, 2007; Oh, Kim, Kim, Park, & Adeli, 2017). Kabir (2010) used a gray-level cooccurrence matrix and neural network classifier to detect alkali-aggregate reaction damage. Wu, Mokhtari, Nazef, Nam, and Yun (2014) used a neural network classifier to identify cracks and improve crack detection accuracy. Rafiei and Adeli (2017) developed the neural dynamics classification algorithm to monitor the global health of large structures. Although many machine learning methods have been used for crack or damage detection and have achieved good results, these methods have some limitations (Figueiredo, Park, Farrar, Worden, & Figueiras, 2011; Rivera et al., 2014). Since these methods always use image processing techniques such as edge detection, their robustness and adaptability are not significantly improved. Therefore, these methods do not have the ability to handle complex background images.

Rapidly developing deep-learning techniques are expected to solve the aforementioned problems (Molina-Cabello, Luque-Baena, López-Rubio, & Thurnhofer-Hemsi, 2018; Wang & Bai, 2018). Convolutional neural networks (CNNs) (Hinton & Salakhutdinov, 2006), without manual feature extraction, have achieved state-of-the-art performance in image classification (Russakovsky et al., 2015) and object detection (Everingham, Van Gool, Williams, Winn, & Zisserman, 2010). Compared with conventional methods, CNNs realize big data processing and parallel computing on graphics processing units (GPUs) (Lindholm, Nickolls, Oberman, & Montrym, 2008; Torres, Galicia, Troncoso, & Martínez-Álvarez, 2018). Recently, the development of deep learning led to technological progress, and many state-of-the-art deep-learning methods have been proposed, such as the region-based CNN (R-CNN) series of models (R-CNN: Girshick, Donahue, Darrell, & Malik, 2014; Fast R-CNN: Girshick, 2015; Faster R-CNN: Ren, He, Girshick, & Sun, 2015), DarkNet (Redmon, Divvala, Girshick, & Farhadi, 2016), single shot multi-box detector (SSD) (Liu et al., 2016), R-FCN (where FCN stands for full convolution network) (Dai, Li, He, & Sun, 2016), and Mask R-CNN (He, Gkioxari, Dollár, & Girshick, 2017). These deep-learning methods have good performance in object detection and object segmentation tasks. Moreover, they have gained great success in computer vision recognition, and many researchers have applied them to directly detect structural damage. Makantasis, Protopapadakis, Doulamis, Doulamis, and Loupos (2015) used a CNN with only two layers and a fully connected (FC) layer to detect tunnel cracks. Cha, Choi, and Büyüköztürk (2017) proposed a complete strategy for crack detection using an eight-layer deep CNN. Cha, Choi, Suh, Mahmoudkhani, and Büyüköztürk (2018) originally used the Faster R-CNN method to realize category 5 damage detection in real time in the field of structural damage detection. Y.Z. Lin, Nie, and Ma (2017) proposed a novel structural damage detection approach based on deep CNN to automatically extract damage features and identify damage locations. Zhang et al. (2017) developed a five-layer CNN for 3D asphalt pavement surface pixel wise classification. Wang, Zhao, Zhao, Li, and Zhao (2018) used a sliding window-based CNN method to identify and locate four categories of damage for historic masonry structures. Xue and Li (2018) proposed a new method for shield tunnel lining defect detection based on an FCN model. Beckman, Polyzois, and Cha (2019) originally proposed a method to detect and quantify volumetric damage based on deep learning and a depth camera, which was of great significance to superficial damage detection.

In summary, the aforementioned deep-learning methods are conceptually intuitive, flexible, and robust, and they achieve good performance in object detection for specific scenarios. However, these methods cannot automatically measure and assess the damage on small objects in large-scale structures, and it is necessary to manually crop large-scale images or collect small object samples. Taking superficial damage detection in glazed tiles as an example, the dataset for deep-learning methods consists of individual tile samples. The existing methods require manual cropping of large-scale roof images or the collection of small images to obtain individual tile samples, which is time and labor intensive.

Therefore, this paper proposes a two-level strategy that can address this limitation. The first level leverages the object detection technique to detect the classes and positions of tiles to automatically crop the tiles based on the position

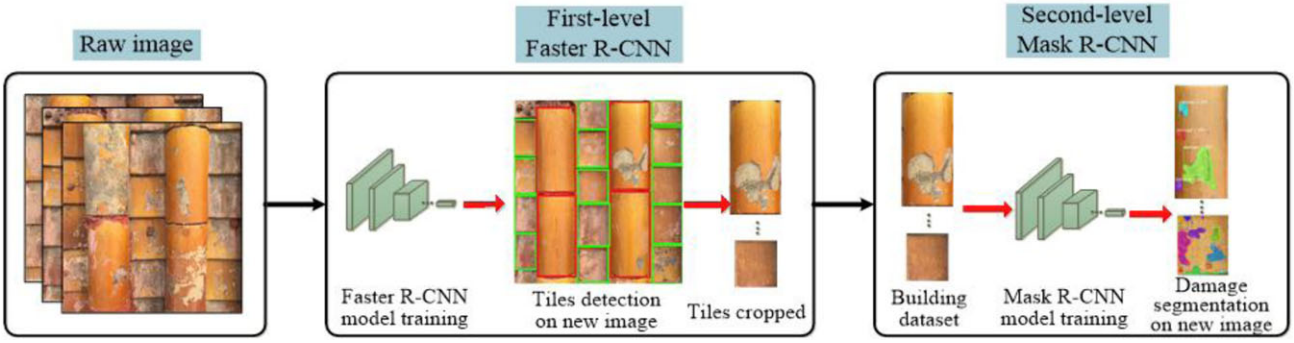

FIGURE 1 Flowchart for the two-level object detection strategy

information. The cropped tiles form a dataset for the second level. The second level originally uses the Mask R-CNN method to segment the pixel-level damage on historic tiles in the field of structural damage segmentation, which can provide a potential way to detect and measure the damage area and damage ratio in pixels. To verify the performance of the proposed method, a comparative study was conducted with Mask R-CNN and FCN. The results show that the proposed method has high identification accuracy, and the damage predictions of Mask R-CNN match well with the ground truth and the damage predictions of FCN. This is the first attempt at using a two-level strategy to automatically detect, segment, and measure large-scale superficial damage on historic buildings based on deep learning and contribute to the protection of historic buildings. The remainder of this paper is composed of Section 2—Methodology, Section 3—Implementation Details, Section 4—The Training Process, Section 5—Validating the Model, Section 6—Test Results and Discussion, and Section 7—Conclusions.

2 | METHODOLOGY

This paper proposes a novel two-level strategy based on a deep-learning technique to realize the automatic damage detection, segmentation, and measurement of individual glazed tiles on a large-scale historic building roof. The flowchart is shown in Figure 1. Overall, the strategy consists of two levels, and each level is divided into several steps. The first level includes the following steps: (a) preparing the dataset based on the collected raw images, (b) training the Faster R-CNN model, (c) detecting and locating the tiles in new images using the trained Faster R-CNN model and providing bounding boxes for the detected tiles (the details are explained in Section 2.1 and its subsections), and (d) cropping the tiles by utilizing the bounding boxes provided in step (c). The second level includes the following steps: (a) building the dataset based on the cropped tiles obtained from step (d) in the first level, (b) training the Mask R-CNN model (the details are explained in Section 3 and its subsections), and (c) segmenting and measuring the superficial damage on the new cropped tiles based on the trained Mask R-CNN model. The methods used in the two levels can be replaced by other methods according to the actual projects. In this study, the first level only needs the class and bounding box of the detected tiles and does not require the segmentation masks. Therefore, the Faster R-CNN method is chosen for the first level. Researchers can also use Mask R-CNN or other more appropriate algorithms according to the needs of actual projects.

2.1 | Faster R-CNN

Faster R-CNN is the most widely used and recognized algorithm in the field of target detection, and was jointly developed in 2015 by Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun (Ren et al., 2015). Based on Fast R-CNN, Faster R-CNN integrates the region proposal acquisition algorithm into the CNN and realizes end-to-end training and testing. The Faster R-CNN method has already been used to detect structural damage. Cha et al. (2018) originally used Faster R-CNN to detect category 5 damage in real time. In this paper, Faster R-CNN is used to automatically detect and crop individual roof tile images. The cropped tile images are used to form the database for Mask R-CNN, which is used to detect and segment the glazed tile damage.

2.1.1 | Data preparation

All the data in this study are from the glazed tiles of the Palace Museum in China. The historic buildings located at the Palace Museum were built in 1406 AD, dating back more than six centuries. The roof of the Palace Museum is a yellow glazed tile roof, as shown in Figure 2a. There are two main types of glazed roof tiles in Chinese historic buildings: roll roofing tiles and pan tiles. The color grade for Chinese historic buildings is very strict and distinct. The grade for yellow glazed tiles is the highest, followed by green, blue, purple, black, and white. Almost all of the glazed tiles on the historic buildings at the Palace Museum are yellow, which highlights that it had the highest status in terms of imperial power.

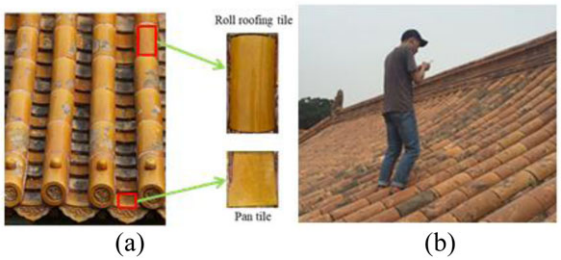

FIGURE 2 (a) The yellow glazed roof tiles and (b) on-site data collection

An iPhone 6 was used as the camera to collect the roof images. It provided the RGB image data with a 2,488 × 3,264 pixel resolution. The on-site data collection is shown in Figure 2b. The space position of the pan tile is lower than that of the roll roofing tile. Therefore, the camera should be parallel to the roof while collecting the images so that all the tiles can be photographed completely. The distance between the camera and the roof was from 1.3 m to 1.5 m.

After the data were collected, 400 images with a 2,488 × 3,264 pixel resolution were used as the database for Faster R-CNN. To annotate the labels (tile type: "roll roofing tile" and "pan tile") and the coordinates of the corresponding bounding boxes in the images, the free LabelImg annotation software was utilized. The training, validation, and testing datasets were manually produced to ensure that no image was in two datasets simultaneously. A total of 80 images were selected for the testing dataset, and the remaining 320 images were randomly selected to compose the training and validation datasets. A total of 100 new images were used as the new test images, which were not used in the training, validation, or testing datasets.

2.1.2 | Model training

Faster R-CNN contains two core components—the region proposal network (RPN) and Fast R-CNN. Both the RPN and Fast R-CNN networks use the same CNNs to extract features from raw images. Since ZF-Net has the fastest training and testing speed, as demonstrated in many studies (Li et al., 2016; Ren et al., 2015), it is used to train the RPN and Fast R-CNN (Zeiler & Fergus, 2014). The RPN extracts the region proposal and uses the "anchor" concept; anchors are a set of rectangular object proposals with different scales and aspect ratios. The anchor size and ratio parameters are very important for model training. In this paper, the optimal anchor sizes and ratios are found via trial and error. The anchor sizes and ratios are 128, 256, and 512 and 0.2, 0.85, and 1.7, respectively. Since the model in this study is structured for two categories ("roll roofing tile" and "pan tile"), the number of output neurons in the source code should be modified. The CNN and FC layers are initialized by a zero-mean Gaussian distribution with standard deviations of 0.01 and 0.001, respectively. The RPN and Fast R-CNN are trained at a basic learning rate of 0.001, an attenuation coefficient of 0.1, a momentum of 0.9, and weight decay of 0.0005 for 20,000 and 10,000 iterations, respectively.

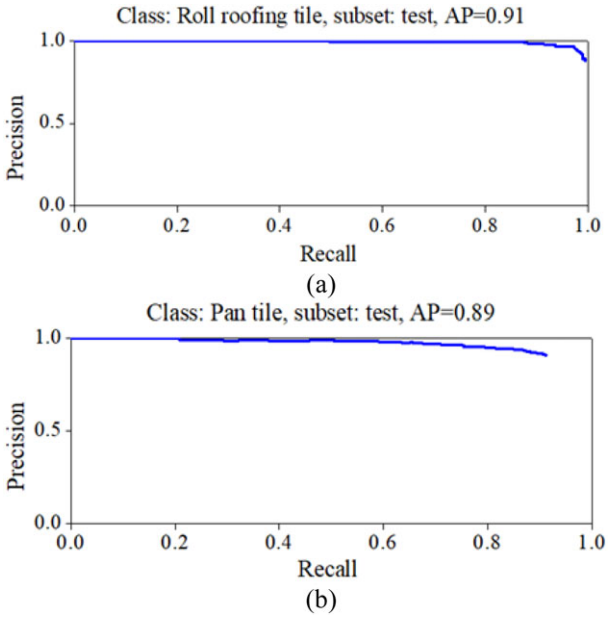

FIGURE 3 Precision/recall curve (a) for the roll roofing tile and (b) for the pan tile

The training platform is based on the Caffe framework and is implemented on a workstation (central processing unit [CPU]: Intel Xeon E5-2630 v4 @ 2.2 GHz, RAM: 32 GB, and GPUs: NVIDIA GeForce GTX 1080 Ti) using the GPU mode.

After the model is trained, average precision (AP) is used to evaluate the performance of the object detector (Girshick, 2015; Ren et al., 2015). The AP measures the area under the precision/recall curve (Everingham et al., 2010). The recorded AP values and precision/recall curves for the two types of tiles are shown in Figure 3. As shown in Figure 3, the AP values for the roll roofing and pan tiles are 0.91 and 0.89, respectively. The mean AP (mAP) is defined as the average of the calculated AP values for all the classes. The mAP for the two types of tiles is 0.90. The network requires 0.07 s in the GPU mode to evaluate each 2,488 × 3,264 pixel image.

2.1.3 | Tiles cropped

To verify the performance of the trained model, 100 new images were tested and cropped. Three of the test results are randomly displayed in Figure 4, which indicate that the output of the trained model for real-life tiles is very effective. The trained model not only outputs the type of tile but also the four parameters that represent the upper-left corner coordinates (T_x^l, T_y^l) and the lower-right corner coordinates (T_x^r, T_y^r) of the detected bounding boxes. Therefore, an individual tile can be automatically cropped from a roof image based on the position information. After cropping the tiles,

FIGURE 4 Tiles detection results based on Faster region-based convolutional neural networks

500 tile images were chosen from the cropped images to form a dataset for the second-level study of damage detection and segmentation.

2.2 | Mask R-CNN

Mask R-CNN is an end-to-end, pixel-to-pixel convolutional network for instance segmentation and is an extension of Faster R-CNN that adds a branch for estimating the segmentation masks on every region of interest (ROI) (He et al., 2017).

2.2.1 | The architecture of Mask R-CNN

Mask R-CNN is a broader version of Faster R-CNN; therefore, it has an architecture that is similar to that of Faster R-CNN. Compared with Faster R-CNN, a fully convolutional mask prediction branch is added in the network head to output the predicted mask, as shown in Figure 5. First, the raw images are put into the CNNs to extract the basic features. This paper uses the top-down combination of the ResNet101 model and feature pyramid network (FPN) (T.Y. Lin et al., 2017) in the Mask R-CNN backbone to extract the ROI features. Then, the region proposals are generated by the RPN, and each image has N proposals, which are mapped to the last convolution feature map of the CNN. Since ROI pooling does not have a pixel-to-pixel alignment, which greatly affects the precision of the prediction masks, Mask R-CNN uses ROI align instead of ROI pooling to increase the prediction precision. Then, each ROI generates a fixed size feature map using the ROI align layer. Finally, the fixed size feature maps are put into the FC layer. After several FC layers, the Mask R-CNN model outputs the damage category, location, pixel segmentation, damage area, and damage ratio in pixels.

Mask R-CNN performs class and position predictions in the same manner as that of Faster R-CNN. Segmentation for each ROI proposal is performed in the mask branch by an FCN that outputs $K \times m \times m$ binary masks. There are K categories and the ROIs have a size of $m \times m$. If the intersection-over-union (IoU) between the predicted ROI and the ground truth is more than 0.5 (Girshick, 2015), the predicted ROI is positive, and each positive ROI has K predicted masks. However, only the K_i mask can be output, and the K_i category is consistent with the predicted object category label according to the class branch. Finally, all the predicted masks for each damage category can be output.

2.2.2 | The loss function for Mask R-CNN

In the Mask R-CNN training process, a multitask loss for each sampled ROI is defined as $L = L_{cls} + L_{box} + L_{mask}$, in which L is the training total loss; L_{cls} is the classification loss, and L_{box} is the bounding-box loss. The definitions of L_{cls} and L_{box} are the same as those defined in Girshick et al. (2014). The variables for L_{cls} and L_{box} are defined as follows:

$$L_{cls}(p_i, p_i^*) = -\log[p_i^* p_i + (1 - p_i^*)(1 - p_i)] \tag{1}$$

$$L_{reg}(t_i, t_i^*) = \begin{cases} 0.5(t_i - t_i^*)^2 & (|x| < 1) \\ |x| - 0.5 & (|x| \geq 1) \end{cases} \tag{2}$$

where p_i is the probability of the ith anchor (proposed in Faster R-CNN) (Ren et al., 2015) being the "target." $p_i{}^*$ is the actual category evaluated by the IoU; $p_i{}^*$, the value for the "target," is 1; $p_i{}^*$, the value for "background," is 0. t_i and $t_i{}^*$ contain four components (two directions for translation and scaling), which are calculated as follows:

$$t_x = \frac{x - x_a}{w_a};\ t_y = \frac{y - y_a}{h_a};\ t_w = \log\frac{w}{w_a};\ t_h = \log\frac{h}{h_a} \tag{3}$$

$$t_x^* = \frac{x^* - x_a}{w_a};\ t_y^* = \frac{y^* - y_a}{h_a};\ t_w^* = \log\frac{w^*}{w_a};\ t_h = \log\frac{h^*}{h_a} \tag{4}$$

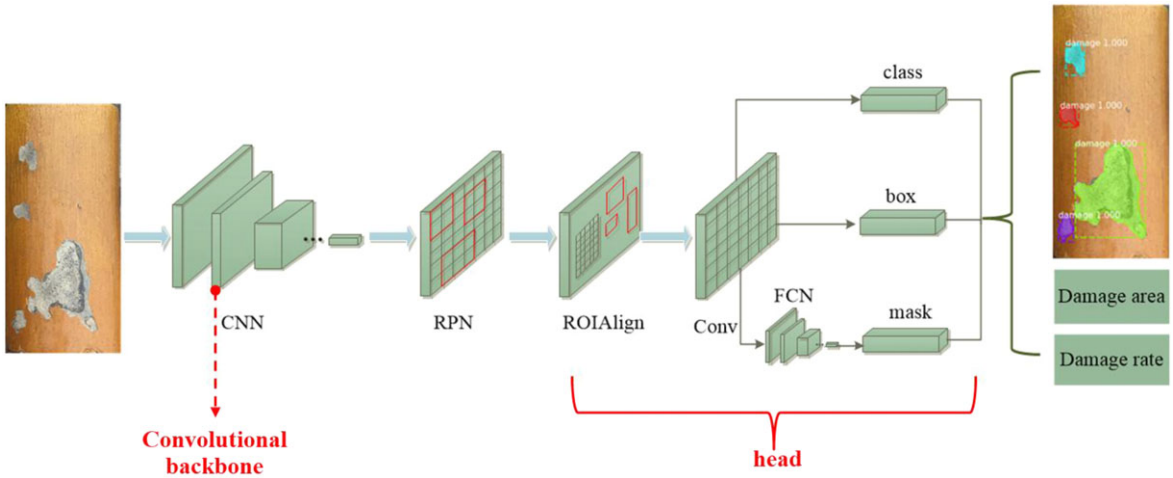

FIGURE 5 The architecture of Mask region-based convolutional neural networks (CNN)

where x, y, w, and h represent the x-coordinate, y-coordinate, x-direction length, and y-direction length of the center, respectively. The parameter containing the "*" symbol is the actual position of the nearest target. The parameter containing "a" is the anchor position. The parameter without any subscripts is the predicted position (i.e., the training position).

First, the sigmoid calculation is applied to each pixel. Then, L_{mask} is calculated by taking the average cross entropy of all the pixels on the ROI. Only the positive ROI, where the IoU between the ROI and the ground truth is more than 0.5 (Girshick, 2015), contributes the L_{mask}. The L_{mask} variables are defined as follows:

$$L_{mask} = -\frac{1}{N}[y_i \ln a_i + (1 - y_i)\ln(1 - a_i)] \tag{5}$$

$$y_i = 1/(1 + e^{-x_i}) \tag{6}$$

$$a_i = 1/(1 + e^{-b_i}) \tag{7}$$

where x_i is the prediction value of the ith pixel in the positive ROI, b_i is the true value of the ith pixel in the positive ROI, and N is the number of pixels in the positive ROI.

2.2.3 | ROIAlign

In the standard operation of Faster R-CNN, a small feature map is extracted from each ROI as ROI pooling (Girshick, 2015). There are two quantization operations in ROI pooling, and the schematic diagram is shown in Figure 6. An 800×800 image with a 665×665 bounding box is put into the CNN. After extracting the features via the CNN, the scaling stride of the feature map is 32. The size of the image and the bounding box are reduced to 1/32 of the raw image size. Eight hundred is exactly divisible by 32 into 25. However, 665 divided by 32 is a floating number equal to 20.78. Then, ROI pooling quantizes 20.78 to 20. Next, the feature map in the bounding box

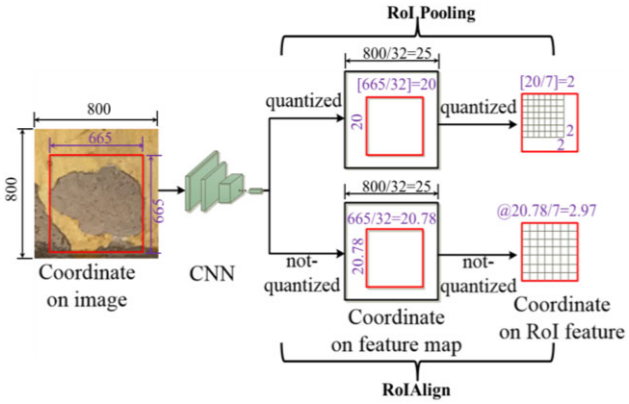

FIGURE 6 Schematic diagram of region of interest (ROI) pooling and ROI align

should be pooled to 7×7. Then, the bounding box is divided into 7×7 rectangular regions, obtaining a length of 2.86. ROI pooling quantifies 2.86 to 2. After two quantizations, the ROI and the extracted feature were obviously misaligned, as shown in Figure 6.

To solve this limitation, an ROI align layer is proposed for Mask R-CNN to replace the ROI pooling. The use of the ROI align layer avoids any quantization of the ROI boundaries or bins, as shown in Figure 6. Four general sampled locations in each ROI bin are selected, and a bilinear interpolation (Jaderberg, Simonyan, Zisserman, & Kavukcuoglu, 2015) is used to calculate the exact values for each location and average pooling is used to summarize the results.

3 | IMPLEMENTATION DETAILS

3.1 | Database

In the second-level dataset, the image size is not fixed, the minimum size is 384 pixels, and the maximum size is 1,408

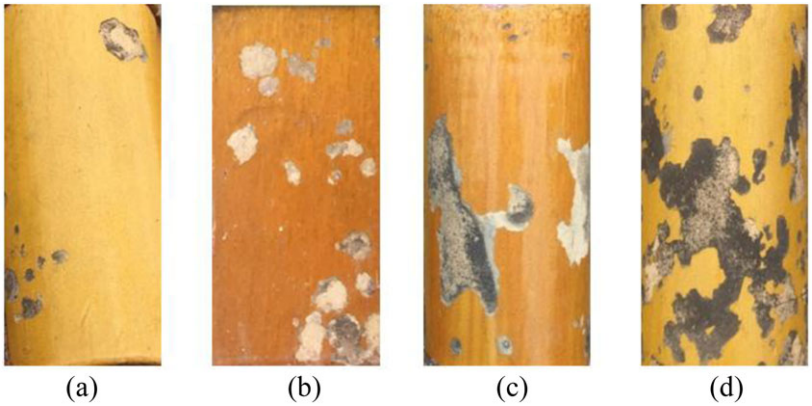

FIGURE 7 The training samples: (a) simple damage, (b) moderate damage, (c) serious damage, and (d) severe damage

TABLE 1 Description of the four kinds of damage to the historic glazed tile

	Damage category			
	Simple	**Moderate**	**Serious**	**Severe**
Damage feature	Damage area is small, the damage topology is simple, and the damage boundary is ellipsoidal	Damage area is larger than simple damage and there are many areas with simple damages on the tile surface	Damage area is large and there is less irregular mass damage. The damage topology is complex	Damage area is larger; there are many areas with irregular mass damage. The damage topology is more complex

pixels. The entire dataset is divided into the training and validation set (500 images) and the testing set (100 images). The damage characteristics of the glazed tiles vary, including simple, moderate, serious, and severe damage, as shown in Figure 7. A detailed description of the four kinds of damage is given in Table 1.

3.2 | Data preparation

The samples are put into a binary mask format to facilitate network training. Then, all the damage of the 500 samples is manually marked using the open-source Labelme software to generate the mask files. The objective only focuses on the detection and position of the spalling damage; thus, the researchers denote the background pixels as 0 and the damage pixels as 1. The process of generating the mask files is shown in Figure 8. First, the raw images are labeled, which means that the boundaries of all the damaged areas in the raw image are sketched out. Second, all the damaged areas are segmented, and each labeled damaged area generates a separate mask. Finally, the masked information is converted into an 8-bit mask file along with the corresponding mask coordinate and label name files. The labeled samples with the corresponding masks and labels are shown in Figure 8. During the annotation process, 5,524 spalling damage areas of the 500 images are labeled. Each image contains only one background label but may have more than one damage label (even dozens). Therefore, it is essential to decouple each damage mask from the background. In our research, each damaged area has separate mask information, which is intended for the damage segmentation process.

3.3 | Model initialization

The training process was carried out on a workstation with a high-performance GPU (NVIDIA Geforce GTX 1080 Ti) and a CPU (CPU: Intel Xeon E5-2630 v4 @2.2). The code was programmed using Python 3.6, and the virtual environment was established by TensorFlow 1.4 and Keras 2.1. The training process was performed on Linux systems. The hyper parameters of the model were set following the existing Fast/Faster R-CNN work (Girshick, 2015; T.Y. Lin et al., 2017; Ren et al., 2015). Although these strategies are used for object detection, it is robust for our instance segmentation model. To save training time and generalize the training process, the parameters in the convolutional layers of the Mask R-CNN backbone are initialized from a pretrained ResNet101 model (He, Zhang, Ren, & Sun, 2016), which is a state-of-the-art image classification network. According to the hardware and software conditions of the workstation, each minibatch has one image per GPU. Each image has N sampled ROIs, where N is 64 for the ResNet101 backbone and 512 for the FPN. The steps per epoch are 1,000, and there are 50 validation steps. The learning rate is 0.001, which is decreased by 10 at 10k steps. The momentum and the weight decay are assigned as 0.9 and 0.0001, respectively. The RPN anchors span five scales (48, 96, 192, 384, and 768) and three aspect ratios (0.5, 1, and 2). The ROI is considered positive

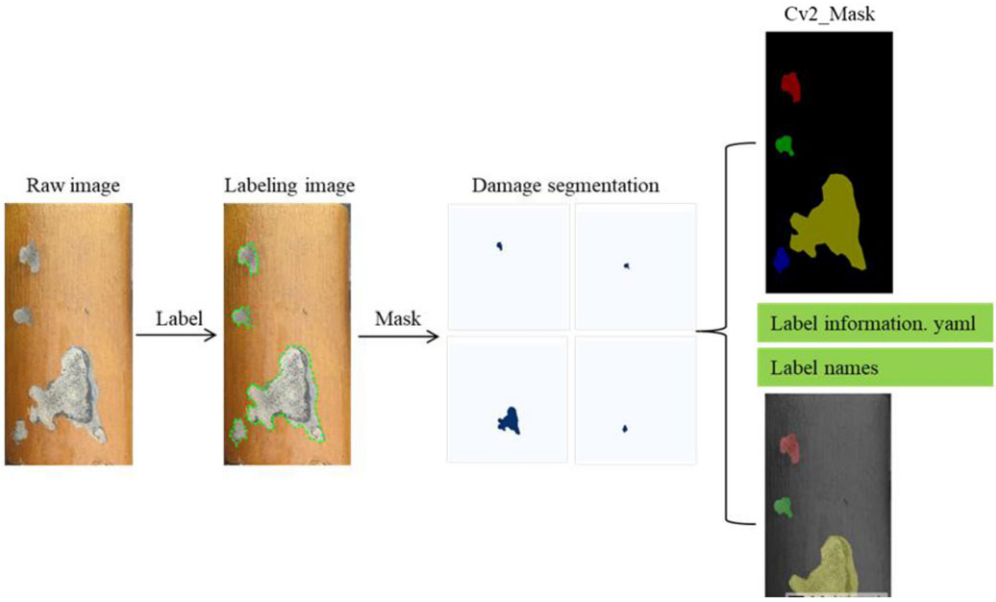

FIGURE 8 The process of generating the mask files

if the IoU between the ROI and the ground truth is more than 0.5 and negative otherwise.

4 | THE TRAINING PROCESS

In the training process, we first use the base layer to load the pretrained ResNet101 weights and then carry out the transfer learning based on the pretrained model. The detailed training process contains two steps: (a) Only the network *head* is trained to avoid destroying the extraction ability of the base layer. We freeze all the backbone layers and only train the randomly initialized layers. To train only the network head, the parameter of layers = "*heads*" should be added to the train method. The network head is trained in 10 epochs. (b) All the layers are fine-tuned after training the network head; to better adapt the new dataset, the layers = "*all*" parameter should be used to fine-tune all the layers. The fine-tuning process takes 30 epochs for training, and the total training time is 9 hr. The training and verification loss curves are shown in Figure 9, where $L = L_{cls} + L_{box} + L_{mask}$.

As shown in Figure 9, the darker curves represent the total loss, and other colorful curves represent L_{cls}, L_{box}, and L_{mask}, respectively. Figure 9a shows that the darker curves rapidly decrease from 1.0 and tend to stabilize at 20 epochs, and the ultimate total loss converges to approximately 0.2. As shown in Figure 9b, the darker curves rapidly decrease from 0.8, and the ultimate total loss converges to approximately 0.2. The decline process has large fluctuations and tends to stabilize at 20 epochs. The initial values of L_{box} in Figure 9a,b are small and eventually close to 0 after a slow decrease. The values of L_{cls} and L_{mask} in Figure 9a,b decrease slowly, and the ultimate mask loss converges to approximately 0.1. Here, all of the validating losses in Figure 9b perform close to the training losses in Figure 9a to ensure that the parameters of Mask R-CNN are not overfit during the training process.

5 | VALIDATING THE MODEL

During the training process, the validation dataset was used to evaluate the model performance. When the IoU threshold is 0.5, the AP of this verification is 0.975, which indicates that the performance of the model is very effective. To clearly understand the effect of the verification, we randomly display the visual results of six verified images, as shown in Figure 10. Figure 10 shows that the damage detection and segmentation results are very effective and can directly identify the damage boundaries of the glazed tiles. Next, the validation results will be discussed in detail.

In the damage detection task, a detected region is considered a true positive if it matches the damage type of the ground truth region and has an IoU of at least 50%. The extraneous detections are considered false positives. Figure 10 shows that almost all the damaged areas have been successfully detected, although the detection background is very complex (there are dozens of damages per image and the damage topology is complex).

To evaluate the damage segmentation accuracy, we apply the IoU as the evaluation metric, following the same method as in Garcia-Garcia, Orts-Escolano, Oprea, Villena-Martinez, and Garcia-Rodriguez (2017). In this study, the IoU is defined as the ratio of the overlap area to the union area between the predicted damage and ground truth damage. The IoU

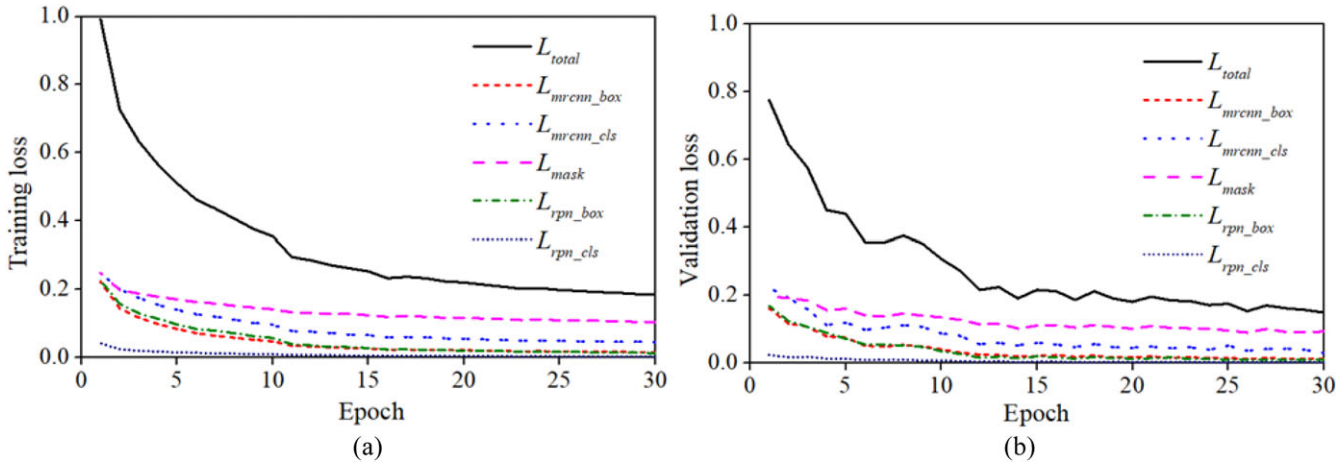

FIGURE 9 The losses in the training and validation process—(a) the training loss curve and (b) the validation loss curve

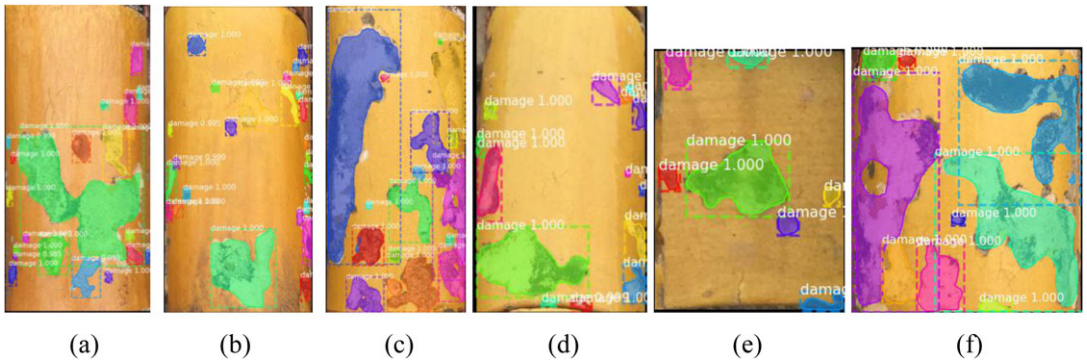

FIGURE 10 The validation results

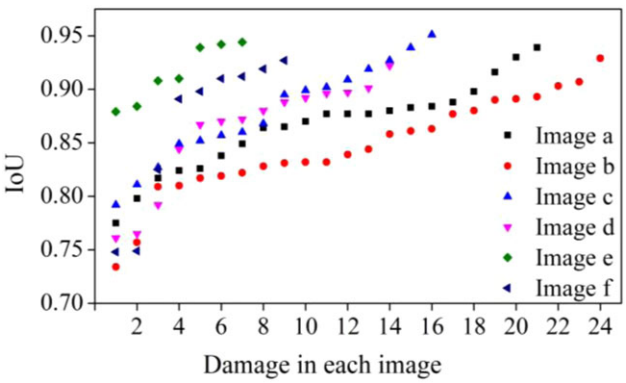

FIGURE 11 The match between the predicted damage and the ground truth of each image

for each damaged area in Figure 10 is plotted in Figure 11. Figure 11 shows the IoU values of all the predicted damage and ground truth in the image, and all IoU values are between 0.73 and 0.95. The mean IoU (mIoU) values of the images in Figure 10 are 0.865, 0.847, 0.879, 0.861, 0.915, and 0.864. The mIoU value of Figure 10e is the highest because it has less damage and the damage topology is simple. Figure 10c shows an example of complex damage (with a very complex damage topology); however, the mIoU value of Figure 10c is very high (0.879). Figure 10f shows that some small damaged areas at the boundary are not detected. These minor errors may be caused by the small training database. Therefore, the problem can be solved by expanding the database in a future study.

6 | TEST RESULTS AND DISCUSSION

6.1 | Testing new images

To evaluate the performance of the trained model, 100 new images, which were not used to train the model, were used for testing. Based on the trained model, we performed damage detection and segmentation on the 100 glazed tiles (including simple damage, moderate damage, serious damage, and severe damage). The recorded testing duration was approximately 0.1 s for each glazed tile. The total processing time from the first level to the second level for a roof image with 2,488 × 3,264 pixels is approximately 2.47 s. The test results of four kinds of typical damage are randomly

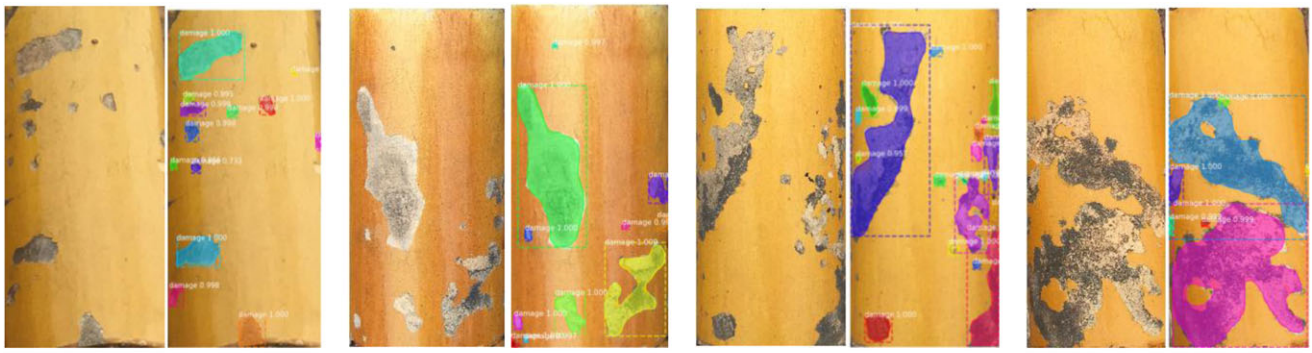

FIGURE 12 The test results of four kinds of typical damages: (a) simple damage, (b) moderate damage, (c) serious damage, and (d) severe damage

selected, as shown in Figure 12. The different colors of the damage segmentation areas in Figure 12 represent different individual damaged areas in one image. The glazed tile damage represented by the pixel-level mask can help to quickly calculate the damage area and damage ratio in pixels.

There are 11 simple damaged areas in Figure 12a, and the damage topology is regular. All the damage in Figure 12a was successfully detected and segmented, and the predicted damaged areas match very well with the ground truth damage. There are 10 damaged areas in Figure 12b, including two large damaged areas, but the overall degree of damage is not serious. Mask R-CNN still successfully identified all the damage, and the predicted damaged areas match very well with the ground truth damage. Figure 12c shows serious damage, and the damage topology is irregular. There are many small areas of damage and one large area of damage in Figure 12c, and the overall degree of damage is serious. It is satisfying that the damage detection and segmentation results of Figure 12c are very effective. Figure 12d shows severe damage, and the damage topology is more irregular. There are two major damaged areas in Figure 12d, and the damaged area is large. Surprisingly, Mask R-CNN successfully identified all the damage under challenging conditions. Despite a small undetected damaged area at the damage boundary, the predicted damage matches well with the ground truth.

In summary, Mask R-CNN achieves good segmentation results even under challenging conditions. Moreover, Mask R-CNN performs effectively where the damage topology ranges from regular to complex and the damage area ranges from small to large. Regardless of how complex the topology is and how much damage the image contains, Mask R-CNN is universal and effective.

6.2 | Test images with complex backgrounds

To detect the limits of the training model, new tiles with complex backgrounds were selected for the experiment. The tiles chosen in this test contained some ambient noise, such as dirt, stickers, weeds, and shadows. The test results are shown in Figure 13.

As shown in Figure 13, the test results of tiles with various complex backgrounds are very good. The tile in Figure 13a has an appendage and some dirt stains on the tile surface. All the damage in Figure 13a was successfully detected and segmented, indicating that the performance of the model is not affected by the dirt stains. In Figure 13b, the tile is covered with a sticker and a small number of weeds. All the damage in Figure 13b was also successfully detected and segmented, indicating that the performance of the model is not affected by the external stickers and a small number of weeds. Figure 13c–e is covered with different kinds of plants, and some of the damage is covered by the plant shadows. The tiles in Figure 13d and 13e are under dark lighting and intense lighting, respectively. It is satisfactory that the segmentation results of Figure 13c–e are very good, and almost all the damage on the three tiles was successfully segmented. The tile in Figure 13f is covered with some mortar, which is very similar to spalling damage. However, the model successfully detected all the damage without any disturbance from the mortar. There are some plants and branches on the tile in Figure 13g, and the damaged tile area is large. The test result in Figure 13g shows that boundary segmentation is not good enough for complex damage.

In general, despite the complex backgrounds, the performance of the model shows that it is effective, and almost all the damaged areas are successfully identified and segmented. It can be concluded that the damage segmentation method is insensitive to ambient noise.

6.3 | Measuring the damage

According to the above discussion, the predicted mask is accurate according to the ground truth damage. Based on the damage mask generated by Mask R-CNN, the damage

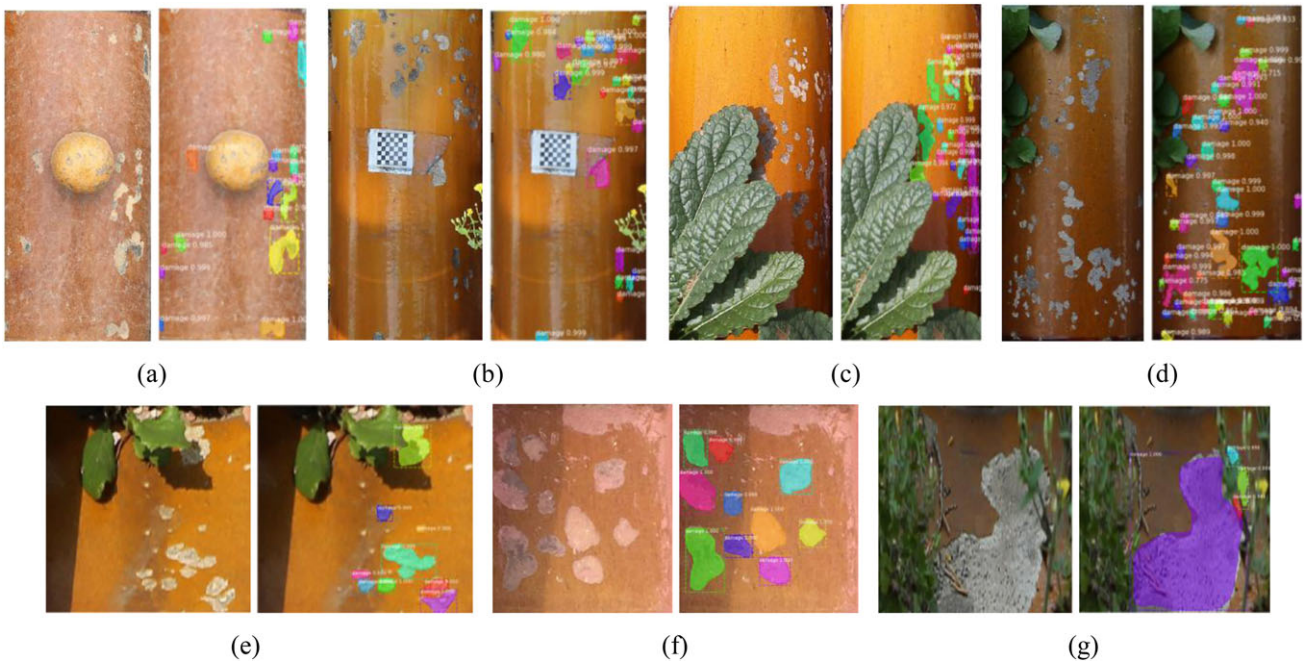

FIGURE 13 Test results with complex backgrounds

morphological features, such as the damage area and damage ratio, could be measured. The damage area in this study is calculated as the number of damage pixels (the pixel area). Since the output mask is binarized, where the damage region is 1 and the background is 0, the damage area can be measured by summing over the pixels of the damage mask. The damage ratio is defined as the number of pixels in the damaged area to the total number of pixels in the entire tile image (Skaloudova, Krivan, & Zemek, 2006). The damage ratio can provide a valuable reference for a glazed tile's working condition for experts. Meanwhile, if the physical dimensions of the tile are given, the damage area in pixels can be transformed into an actual physical area.

In this study, 100 tiles, which were not used for training and validation, were used to measure the damage. All the measurement results are shown in Figure 14, including the damage area and damage ratio. Figure 14a shows the comparative results of the ground truth and the predicted damage areas, which indicates that the predicted damage areas essentially agree with the ground truth areas. The damage area of the tile is mostly concentrated between 5,000 and 25,000 pixels, where the prediction results are very effective. Figure 14b shows the comparative results of the ground truth and the predicted damage ratio. The damage ratio of the tiles is between 5% and 55%. When the damage ratio is between 5% and 30%, the predicted result is very effective, and the prediction difference is between 0.2% and 2%. When the damage ratio is between 30% and 55%, the prediction difference is between 2% and 8%. The measurement results indicate that the proposed method is very effective for identifying small amounts of damage but has a slight error in identifying large amounts of damage.

The measurement results of four kinds of typical damage are randomly selected, as shown in Figure 15. The raw image area (RIA), the ground truth_damage area (GT_DA), the ground truth_damage ratio (GT_DR), the predicted_damage area (P_DA), the predicted_damage ratio (P_DR), and the predicted_damage area error (P_DAE) are presented in Figure 15. The measurement results shown in Figure 15 prove that the proposed method for measuring the damage on glazed tiles is very effective and accurate for simple and even complex damage. Figure 15a shows that there are three simple damaged areas on the tile surface, and the damage topology is regular. The three damaged areas are all successfully detected, and the predicted damage matches well with the ground truth. The GT_DR, P_DR, and P_DAE values in Figure 15a are 12.13%, 11.60%, and 4.03%, respectively, and the difference between the GT_DR and P_DR values is 0.70%, which indicates that the predicted damage matches well with the ground truth. Figure 15b shows moderate damage. The GT_DR, P_DR, and P_DAE values in Figure 15b are 15.71%, 15.20%, and 8.6%, respectively, and the difference between the GT_DR and P_DR values is 0.51%. For the serious damage in Figure 15c, the GT_DR, P_DR, and P_DAE values are 22.07%, 21.9%, and 0.93%, respectively, and the difference between the GT_DR and P_DR values is 0.08%, which shows an effective result. However, some of the damage boundaries in Figure 15d are not precisely segmented. The GT_DR, P_DR, and P_DAE values in Figure 15d are 42.85%, 41.20%, and 3.75%, respectively, and the difference between

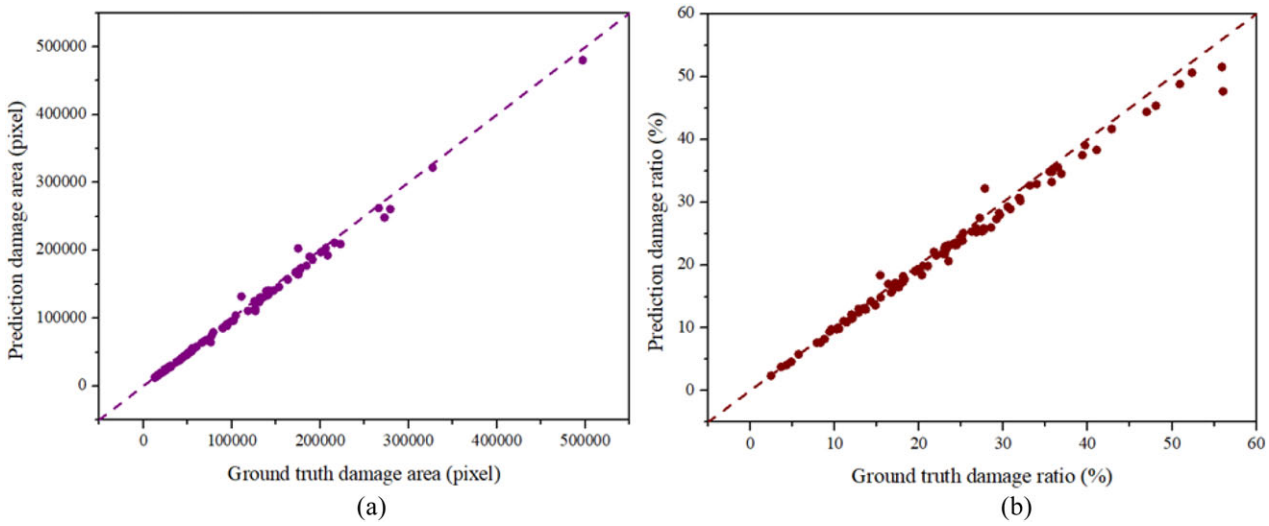

FIGURE 14 Measurement results of the ground truth damage and predicted damage—(a) the damage area result and (b) the damage ratio result

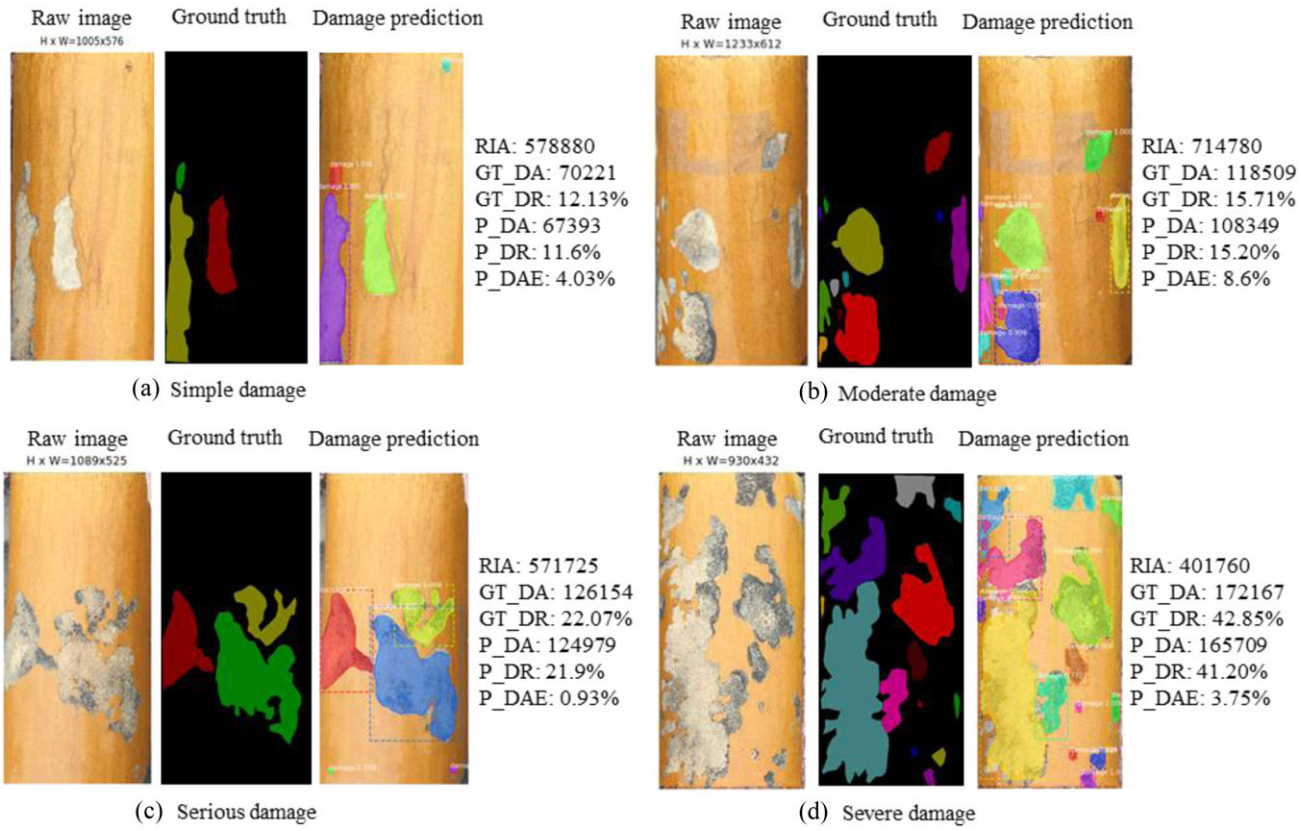

FIGURE 15 Measurement result of the four kinds of typical damage

the GT_DR and P_DR is 1.65%, which indicates that the detection results are not good enough for complex damage.

Overall, the predicted damage area is smaller than the ground truth, leading to a lower damage ratio. As mentioned in Section 5, some boundaries of spalling damage are not precisely detected. Therefore, the measured damage area is smaller. Despite minor errors, the test results demonstrate the effective performance of our proposed method for measuring the damage on historic glazed tiles. These minor errors may be caused by a small training database. In future research, the problem can be solved by expanding the database with more images of glazed tile damage under various conditions to improve the method's capacity and generalization.

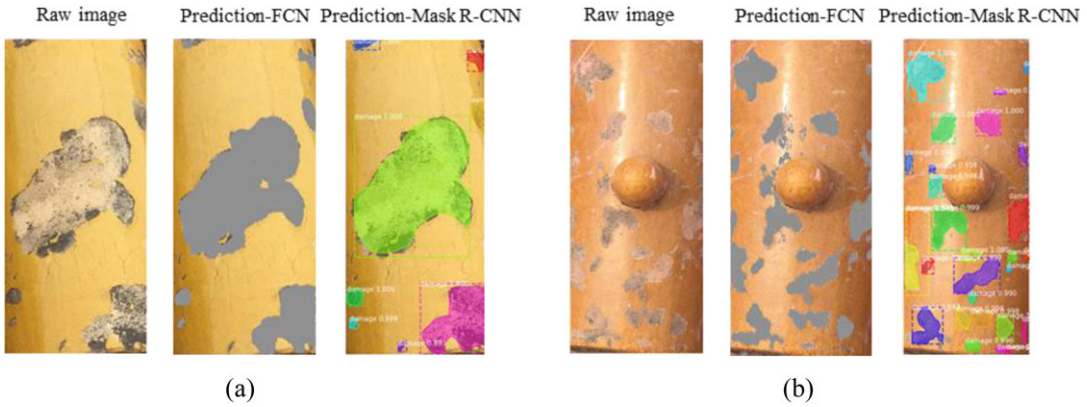

FIGURE 16 Comparison results of Mask region-based convolutional neural networks (R-CNN) and full convolution network (FCN)

6.4 | Comparative study

To compare the performance of the proposed approach with a state-of-the-art object segmentation method, the built database with glazed tile damages is used to train the FCN model (Yang et al., 2018). To adapt the input size of the FCN, all the images, and their labels in the training, validation and testing sets are resized to a 224×224 pixel resolution to generate a new database for training the FCN. The FCN is trained with a momentum of 0.9 and a weight decay of 0.0005 for 30,000 iterations. The batch size is set to two images to accelerate the convergence speed. The initial learning rate is set to 0.001.

Similar to the Mask R-CNN test, the trained FCN model is tested using the new images. The recorded test duration is 0.02 s for each image, which is shorter than that of Mask R-CNN (0.1 s) because of the smaller image size (224×224 pixel resolution). The images in Figure 16 include the raw image, the prediction result by FCN, and the prediction result by Mask R-CNN. As shown in Figure 16, the spalling damages are almost correctly identified by both the FCN and Mask R-CNN. However, for a large damaged area, Mask R-CNN cannot precisely detect the boundary, as shown in Figure 16a. The FCN precisely detects all the spalling damage. However, the FCN is so sensitive to spalling damage that false spalling damage is also detected. As shown in Figure 16b, the FCN incorrectly identifies the bright light area as spalling damage. However, Mask R-CNN is not sensitive to light, and it may accurately identify all the spalling damage in Figure 16b.

In conclusion, the two damage segmentation methods, the FCN and Mask R-CNN, have their own advantages. In this study, although Mask R-CNN has the drawback of inaccurate boundary detection for large areas of spalling damage, the stability of this method in cases with external noise is very good. In future research, potential ways to address the issue of inaccurate boundary detection can be explored, such as optimizing the damage detection algorithm and increasing the diversity of the dataset.

7 | CONCLUSIONS

This paper proposes a novel two-level object detection and segmentation strategy based on Faster R-CNN and Mask R-CNN. The first level detects and crops the tile images from raw roof photographs based on Faster R-CNN. The cropped tiles form a dataset for Mask R-CNN. The second level segments and measures the damage on individual tiles based on Mask R-CNN. Based on the two-level object detection and segmentation strategy, this paper realizes automatic identification and segmentation of the damage on morphological features. All the data in this study are from the glazed tiles of the Palace Museum in China. During model training, the validation dataset was used to verify the model performance. After the model was trained, 100 new images were used to evaluate the performance of the trained model. More importantly, based on the trained model, the spalling damage topology, damage area, and damage ratio can be quickly and effectively acquired according to the predicted damage mask. The conclusions are summarized as follows:

1. Based on Faster R-CNN, the first level of the model automatically detected and cropped the tile images from the roof photographs. The mAP of the trained model reached 0.90, and the cropped images formed a dataset for the second level.
2. Based on Mask R-CNN, the second level realized end-to-end pixel-to-pixel damage detection and segmentation (with a high validation AP of 0.975). The method can effectively realize damage detection and segmentation.
3. Based on the trained Mask R-CNN model, 100 new images were used to verify the performance of the model. Mask R-CNN achieved a high identification accuracy. Despite some minor undetected damage at the boundaries of large damaged areas, the predicted damage matched well with the ground truth. To verify the stability of the model, a validation experiment with complex backgrounds was

performed, and the results indicated that the method was insensitive to the ambient noise.

4. Based on the Mask R-CNN trained model, the predicted damage masks represented by pixels contribute to the spalling damage measurement. The morphological features of spalling damage can be extracted from the predicted damage mask, and no postprocessing or preprocessing is needed.
5. A comparative study between the FCN and Mask R-CNN was performed, which indicated some drawbacks of Mask R-CNN. When the damage area was large, some boundary damage was not precisely detected by Mask R-CNN.

Although Mask R-CNN shows effective performance in damage detection and segmentation, the detection of severe damage with very complex geometric features still has potential for improvement. In future research, it would be worthwhile to use a larger number of training samples and a state-of-the-art deep-learning method to improve the accuracy and precision of the damage detection and segmentation model for historic glazed tiles.

ACKNOWLEDGMENTS

The research was supported by the research project for information and disease AI identification of building components in the Palace Museum (2018-308) and National Natural Science Foundation of China (51479031).

REFERENCES

Adeli, H., & Yeh, C. (1989). Perceptron learning in engineering design. *Microcomputers in Civil Engineering*, *4*(4), 247–256.

Beckman, G. H., Polyzois, D., & Cha, Y. J. (2019). Deep learning-based automatic volumetric damage quantification using depth camera. *Automation in Construction*, *99*(3), 114–124.

Botas, S., Veiga, R., & Velosa, A. (2017). Air lime mortars for conservation of historic tiles: Bond strength of new mortars to old tiles. *Construction and Building Materials*, *145*, 426–434.

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.

Cha, Y. J., Choi, W., Suh, G., Mahmoudkhani, S., & Büyüköztürk, O. (2018). Autonomous structural visual inspection using region-based deep learning for detecting multiple damage types. *Computer-Aided Civil and Infrastructure Engineering*, *33*(9), 731–747. https://doi.org/10.1111/mice.12334

Dai, J., Li, Y., He, K., & Sun, J. (2016). R-FCN: Object detection via region-based fully convolutional networks. *30th Conference on Neural Information Processing Systems (NIPS 2016)*, Barcelona, Spain.

Elmasry, M. I., & Johnson, E. A. (2004). Health monitoring of structures under ambient vibrations using semiactive devices. *Proceedings of the 2004 American Control Conference*, Boston, 3526–3531. https://doi.org/10.23919/ACC.2004.1384458

Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., & Zisserman, A. (2010). The Pascal Visual Object Classes (VOC) challenge. *International Journal of Computer Vision*, *88*(2), 303–338.

Figueiredo, E., Park, G., Farrar, C. R., Worden, K., & Figueiras, J. (2011). Machine learning algorithms for damage detection under operational and environmental variability. *Structural Health Monitoring*, *10*(6), 559–572.

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, arXiv: 1704.06857, 1–23.

Gattulli, V., & Chiaramonte, L. (2005). Condition assessment by visual inspection for a bridge management system. *Computer-Aided Civil and Infrastructure Engineering*, *20*(2), 95–107.

Ghiassi, B., Xavier, J., Oliveira, D. V., & Lourenço, P. B. (2013). Application of digital image correlation in investigating the bond between FRP and masonry. *Composite Structures*, *106*(12), 340–349.

Girshick, R. (2015). Fast R-CNN. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Boston. https://doi.org/10.1109/ICCV.2015.169

Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Columbus. https://doi.org/10.1109/CVPR.2014.81

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

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas. https://doi.org/10.1109/CVPR.2016.90

Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *Science*, *313*(5786), 504–507.

Jaderberg, M., Simonyan, K., Zisserman, A., & Kavukcuoglu, K. (2015). Spatial transformer networks. *Proceedings of Advances in Neural Information Processing Systems*, Montreal.

Jiang, X., & Adeli, H. (2007). Pseudospectra, MUSIC, and dynamic wavelet neural network for damage detection of highrise buildings. *International Journal for Numerical Methods in Engineering*, *71*(5), 606–629.

Kabir, S. (2010). Imaging-based detection of AAR induced map-crack damage in concrete structure. *NDT & E International*, *43*(6), 461–469.

Kordatos, E. Z., Exarchos, D. A., Stavrakos, C., Moropoulou, A., & Matikas, T. E. (2013). Infrared thermographic inspection of murals and characterization of degradation in historic monuments. *Construction and Building Materials*, *48*(19), 1261–1265.

Koziarski, M., & Cyganek, B. (2017). Image recognition with deep neural networks in presence of noise—dealing with and taking advantage of distortions. *Integrated Computer-Aided Engineering*, *24*(4), 337–350.

Li, C., Kang, Q., Ge, G., Song, Q., Lu, H., & Cheng, J. (2016). DeepBE: Learning deep binary encoding for multilabel classification. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas. https://doi.org/10.1109/CVPRW.2016.98

Lin, T. Y., Dollár, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. (2017). Feature pyramid networks for object detection. *Proceedings of the Conference on Computer Vision and Pattern Recognition*, Venice. https://doi.org/10.1109/CVPR.2017.106

Lin, Y. Z., Nie, Z. H., & Ma, H. W. (2017). Structural damage detection with automatic feature-extraction through deep learning. *Computer-Aided Civil and Infrastructure Engineering*, *23*(12), 1025–1104.

Lindholm, E., Nickolls, J., Oberman, S., & Montrym, J. (2008). NVIDIA Tesla: A unified graphics and computing architecture. *IEEE Micro*, *28*(2), 39–55.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016). SSD: Single shot multibox detector. In B. Leibe, J. Matas, N. Sebe, & M. Welling (Eds.), *Proceedings of European Conference on Computer Vision* (pp. 21–37). Amsterdam: Springer. https://doi.org/10.1007/978-3-319-46448-0_2

Makantasis, K., Protopapadakis, E., Doulamis, A., Doulamis, N., & Loupos, C. (2015). Deep convolutional neural networks for efficient vision based tunnel inspection. *Proceedings of IEEE International Conference on Intelligent Computer Communication and Processing (ICCP)*, Cluj-Napoca. https://doi.org/10.1109/ICCP.2015.7312681

Molina-Cabello, M. A., Luque-Baena, R. M., López-Rubio, E., & Thurnhofer-Hemsi, K. (2018). Vehicle type detection by ensembles of convolutional neural networks operating on super-resolved images. *Integrated Computer-Aided Engineering*, *25*(4), 321–333.

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., Schoefs, F., Ghosh, B., & Pakrashi, V. (2013). Texture analysis based damage detection of ageing infrastructural elements. *Computer-Aided Civil and Infrastructure Engineering*, *28*(3), 162–177.

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

Ortega-Zamorano, F., Jerez, J. M., Gómez, I., & Franco, L. (2017). Layer multiplexing FPGA implementation for deep back-propagation learning. *Integrated Computer-Aided Engineering*, *24*(2), 171–185.

Ou, J., & Li, H. (2010). Structural health monitoring in mainland China: Review and future trends. *Structural Health Monitoring*, *9*(3), 219–231.

Prasanna, P., Dana, K. J., Gucunski, N., Basily, B. B., La, H. M., Lim, R. S., & Parvardeh, H. (2016). Automated crack detection on concrete bridges. *IEEE Transactions on Automation Science and Engineering*, *13*(2), 591–599.

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

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas. https://doi.org/10.1109/CVPR.2016.91

Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster RCNN: Towards real-time object detection with region proposal networks. *Proceedings of the Advances in Neural Information Processing Systems*, Montreal.

Rivera, N. V., Gómez-Sanchis, J., Chanona-Pérez, J., Carrasco, J. J., Millán-Giraldo, M., Lorente, D., … Blasco, J. (2014). Early detection of mechanical damage in mango using NIR hyperspectral images and machine learning. *Biosystems Engineering*, *122*, 91–98.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., … Fei-Fei, L. (2015). Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, *115*(3), 211–252.

Skaloudova, B., Krivan, V., & Zemek, R. (2006). Computer-assisted estimation of leaf damage caused by spider mites. *Computers and Electronics in Agriculture*, *53*(2), 81–91.

Torres, J. F., Galicia, A., Troncoso, A., & Martínez-Álvarez, F. (2018). A scalable approach based on deep learning for big data time series forecasting. *Integrated Computer-Aided Engineering*, *25*(4), 335–348.

Wang, N. N., Zhao, Q. A., Zhao, P., Li, S. Y., & Zhao, X. F. (2018). Damage classification for masonry historic structures using convolutional neural networks based on still images. *Computer-Aided Civil and Infrastructure Engineering*, *33*(12), 1073–1089.

Wang, P., & Bai, X. (2018). Regional parallel structure based CNN for thermal infrared face identification. *Integrated Computer-Aided Engineering*, *25*(3), 247–260.

Wu, L., Mokhtari, S., Nazef, A., Nam, B. H., & Yun, H. B. (2014). Improvement of crack detection accuracy using a novel crack defragmentation technique in image-based road assessment. *Journal of Computing in Civil Engineering*, *30*(1), 04014118.

Xue, Y., & Li, Y. (2018). A fast detection method via region-based fully convolutional neural networks for shield tunnel lining defects. *Computer-Aided Civil and Infrastructure Engineering*, *33*(8), 638–654.

Yan, Y. J., Cheng, L., Wu, Z. Y., & Yam, L. H. (2007). Development in vibration-based structural damage detection technique. *Mechanical Systems and Signal Processing*, *21*(5), 2198–2211.

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

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

Zeiler, M. D., & Fergus, R. (2014). Visualizing and understanding convolutional networks. In D. Fleet, T. Pajdla, B. Schiele, & T. Tuytelaars (Eds.), *Computer Vision—ECCV 2014: 13th European Conference*, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part I (pp. 818–833). Cham: Springer. https://doi.org/10.1007/978-3-319-10590-1_53

Zhang, A., Wang, K. C. P., Li, B., Yang, E., Dai, X., Peng, Y., & Chen, C. (2017). Automated pixel-level pavement crack detection on 3D asphalt surfaces using a deep-learning network. *Computer-Aided Civil and Infrastructure Engineering*, *32*(10), 805–819.

Zhang, W., Zhang, Z., Qi, D., & Liu, Y. (2014). Automatic crack detection and classification method for subway tunnel safety monitoring. *Sensors*, *14*(10), 19307–19328.

Zhu, Z., German, S., & Brilakis, I. (2010). Detection of large-scale concrete columns for automated bridge inspection. *Automation in Construction*, *19*(8), 1047–1055.

How to cite this article: Wang N, Zhao X, Zou Z, Zhao P, Qi F. Autonomous damage segmentation and measurement of glazed tiles in historic buildings via deep learning. *Comput Aided Civ Inf*. 2019;1–15. https://doi.org/10.1111/mice.12488