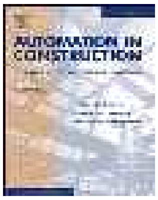
Automation in Construction 158 (2024) 105226

Contents lists available at ScienceDirect

Automation in Construction

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



Region of interest (ROI) extraction and crack detection for UAV-based bridge inspection using point cloud segmentation and 3D-to-2D projection

Jing-Lin Xiao, Jian-Sheng Fan, Yu-Fei Liu [*], Bao-Luo Li, Jian-Guo Nie

Key Lab. of Civil Engineering Safety and Durability of China Education Ministry, Dept. of Civil Engineering, Tsinghua University, Beijing 100084, China

ARTICLE INFO

Keywords:
Unmanned aircraft vehicles (UAVs)
Bridge inspection
Structure from motion (SfM)
Large-scale point clouds
Semantic segmentation
3D-to-2D projection
Crack identification
Deep learning

ABSTRACT

For digital-image-based bridge inspection tasks, images captured by camera-carrying unmanned aircraft vehicles (UAVs) usually contain both the region of interest (ROI) and the background. However, accurately detecting cracks in concrete surface images containing background information is challenging. To improve UAV-based bridge inspection, an image extraction and crack detection methodology is presented in this paper. First, a deep-learning-based semantic segmentation network RandLA-BridgeNet for large-scale bridge point clouds, which can facilitate 3D ROI extraction, is trained and tested. Second, an image ROI extraction method based on 3D-to-2D projection is presented to generate images containing only the ROI. Finally, a data-driven deep learning convolutional neural network (CNN) called the grid-based classification and box-based detection fusion model (GCBD) is utilized to identify cracks in the processed images. An experiment is conducted on highway bridge images to validate the presented methodology. The overall semantic segmentation and image ROI extraction accuracies are 97.0% and 98.9%, respectively. After ROI extraction, 47.9% of the grid cells, which represent background misrecognition, are filtered, greatly improving the crack identification accuracy.

1. Introduction

Concrete bridges that have been in service for decades are often afflicted by load overload, material aging, shrinkage, creep, and fatigue. To ensure the normal service of these structures, structural health monitoring and inspection have become a popular focus for academic research and engineering applications. Cracking is an important phenomenon for monitoring and inspection, and it is a major factor in the durability and safety of reinforced concrete bridges [1,2]. Cracking on concrete surfaces can lead to the concrete cover failure, and reinforcement bars are at risk of rusting. In addition, rapidly developing stress cracks are often a sign of structural failure and collapse.

Cracks can be measured and monitored by arranging contact sensors such as distributed fiber optic sensors [3]. However, when contact sensors are not applicable, the current primary means of crack identification is manually depicting the distribution pattern and measuring the width of the crack, which is time-consuming, laborious, and highly subjective, resulting in widespread errors and omissions. In a study by Graybeal et al. [4], different inspectors gave significantly different crack identification results for the same bridge. To improve the objectivity and efficiency of crack identification, digital image-based techniques have been rapidly developed in recent years. Liu et al. [5] proposed an adaptive digital image processing method for concrete surface crack identification, which includes three steps: image preprocessing, crack identification and extraction, and crack parameter calculation. With the booming development of artificial intelligence, deep learning methods have achieved great success in image recognition tasks. Thus, many researchers have used various kinds of deep neural networks, such as the dense convolutional network (DenseNet), region convolutional neural network (R-CNN), fully convolutional network (FCN), U-shaped network (U-Net), and segmentation network (SegNet), for crack identification [1,2,6–11]. More detailed reviews of the state-of-the-art crack identification methods can be found in [12]. These crack identification networks have been developed well under laboratory conditions. As indicated by Guo et al. [12], even dense interconnected microcracks in a strain-hardening cementitious composite (SHCC) can be satisfactorily characterized when images of the cracks are taken under fully controlled conditions.

The region of interest (ROI) is defined as the concrete component or surface where the crack of interest is located, while the remaining region

[*] Corresponding author.
E-mail addresses: xiaojl19@mails.tsinghua.edu.cn (J.-L. Xiao), fanjsh@tsinghua.edu.cn (J.-S. Fan), liuyufei@tsinghua.edu.cn (Y.-F. Liu), li-bl21@mails.tsinghua.edu.cn (B.-L. Li), niejg@mail.tsinghua.edu.cn (J.-G. Nie).

https://doi.org/10.1016/j.autcon.2023.105226
Received 17 May 2023; Received in revised form 25 November 2023; Accepted 27 November 2023
Available online 8 December 2023

is defined as the background. The aforementioned crack identification studies primarily used images containing only the ROI as the input of deep neural networks. When an image contains both an ROI and background, the deep-learning-based crack identification algorithm may misidentify some parts of the background as cracks, thus affecting the quality of crack recognition.

As reviewed by Poorghasem et al. [13] and Ranyal et al. [14], various robot-based automated systems, e.g., unmanned ground vehicles (UGVs) and unmanned aerial vehicles (UAVs) carrying vision systems such as cameras and optical lenses, were developed to facilitate image collection. Different robot-based automated systems and computer vision methods can be grouped for different engineering scenarios. Currently, robot-based image collection and digital-image-based crack identification are relatively mature methods for road pavement and tunnel surface scenarios [15,16]. In these two scenarios, pavement inspection vehicles and tunnel inspection vehicles, respectively, can be used to take digital images under fully controlled conditions. The obtained images usually cover only the surfaces of the inspected structure (ROI), with no extraneous backgrounds, and are of high quality. However, bridge surface distribution has a much higher complexity than road pavement and tunnel surfaces. Thus, contact inspection vehicles cannot be applied for this scenario. Instead, using camera-carrying UAVs is required to capture bridge surface images. UAVs are in a 6-way free state in flight, and the scope of the photographic scene is often difficult to control. Therefore, images taken by UAVs will inevitably contain extraneous background and are not suitable for direct use in crack identification.

In addition to crack identification, visualized crack localization is also a matter of concern for structural health monitoring and inspection. Liu et al. [17,18] fused two-dimensional (2D) digital image processing technology and three-dimensional (3D) reconstruction technology to achieve crack identification and localization. First, many 2D digital images taken in the field were used to complete the 3D reconstruction of the structure by Structure from Motion (SfM). Then, the images containing crack information were selected for crack identification. Finally, the projection method completed crack localization. Generally, images taken at a close distance with fine details improve crack identification, while images taken at a long distance containing scene geometric information are improve the success rate of 3D reconstruction. For images focusing on a local area of the concrete surface, state-of-the-art technology has been able to accurately complete crack identification. However, images containing only flat, smooth, feature-sparse structural surfaces are not sufficient for SfM to be successful. For successful 3D reconstruction, captured images with background information such as ground and vegetation are better, as the feature points in these images are more abundant. From this perspective, the background does not need to be excluded during UAV-based image capture.

Therefore, it is an objective requirement for UAV-based bridge inspection to accurately identify cracks from images with background. Extracting the ROI from the images is the best way to achieve this goal. Some recent researchers have extracted ROIs from 2D images based on semantic segmentation algorithms of deep learning. Taking bridge structures as an example, Narazaki et al. [19] constructed a semantic segmentation algorithm containing 45 convolutional layers for bridge component recognition after performing scene classification. A MATLAB GUI image semantic segmentation annotation tool was developed specifically to manually annotate thousands of images that were combined with an existing database to generate training data. Saovana et al. [20] trained a deep CNN (DCNN) for removing irrelevant features from bridge images. The training data were realistic scene images manually annotated using LabelMe, and the number of samples was expanded by rotating images and adjusting the brightness of images. Using 236 highway bridge images, Sajedi et al. [21] trained Fully Convolutional DenseNet (FC-DenseNet) to extract different kinds of bridge components from the images. These studies show that the main challenges for bridge component recognition in 2D images are as follows:

(1) Bridge images may contain complex background information, and the background can sometimes be so large that the bridge is not the dominant object in the image.
(2) The characteristics of the same type of bridge components may differ in images with different shooting distances and lighting conditions.
(3) Deep learning-based semantic segmentation algorithms rely heavily on the quality and quantity of training data. Unfortunately, there is currently a scarcity of open-source annotated data, and the available scenarios are not sufficiently comprehensive. Consequently, the reliability and portability of the trained network may suffer. Due to the diverse range of scenes encountered in practical engineering, annotating images for each scene class separately is time-consuming, and building a general dataset is extremely challenging.

Considering the opportunities and limitations mentioned above, this paper proposes a methodology for extracting image ROIs based on 3D point cloud segmentation and 3D-to-2D projection, aiming to improve crack identification from bridge images taken by UAVs and containing background information. Instead of directly extracting bridge components from the images as in previous studies [19–21], the proposed methodology offers enlightening insights about achieving this goal indirectly. The presented methodology integrates point cloud semantic segmentation and 3D-to-2D projection technologies into the UAV-based bridge crack detection task, contributing to advancements in the field. The proposed methodology has both practical purposes (as shown in Section 2) and great potential to improve crack detection results when handling images containing complex background information. The highway bridge is taken as an example in this study to facilitate discussion, but the proposed methodology is also applicable to other engineering scenarios.

2. Methodology framework

The framework of the proposed methodology is illustrated in Fig. 1. For the inspection task of concrete bridges discussed in this paper, the inspector takes numerous images manually or using a UAV. These images contain information about both the spatial composition of the scene and the cracks on the concrete surface. This information is used not only for performing SfM to reconstruct the 3D point cloud of the bridge but also for subsequently identifying cracks. First, RandLA-Net, a deep learning framework for semantic segmentation of large-scale point clouds, is adopted to construct a point cloud semantic segmentation network RandLA-BridgeNet for highway bridges. Bridge point clouds from an open-source dataset are annotated to train and test the network. A large-scale bridge point cloud can be input into RandLA-BridgeNet directly to complete semantic segmentation, and then the 3D ROI can be easily extracted from the point cloud. Second, for each image containing the bridge components to be inspected, the 3D-to-2D projection is performed based on the pinhole camera model. This step, calculating the projection of the 3D ROI in the 2D image (i.e., 2D ROI), is essentially the inverse process of SfM 3D reconstruction. Next, the edge detection algorithm is used to find the outer contour of the 2D ROI and generate a mask. The background pixels outside the outer contour of the 2D ROI are removed using the mask, and the 2D ROI is extracted, producing an image containing only the ROI. Finally, the ROI image is used for crack identification to effectively avoid background interference on the identification algorithm.

Notably, the proposed methodology framework is not limited to the adopted SfM-based 3D reconstruction technique. It can still be applied with minor adjustments when using other 3D reconstruction techniques and 2D digital images for bridge disease detection.

The remainder of this paper is organized as follows: Section 2 describes the point cloud semantic segmentation method used to extract 3D ROIs; Section 3 describes the 3D-to-2D projection and 2D ROI

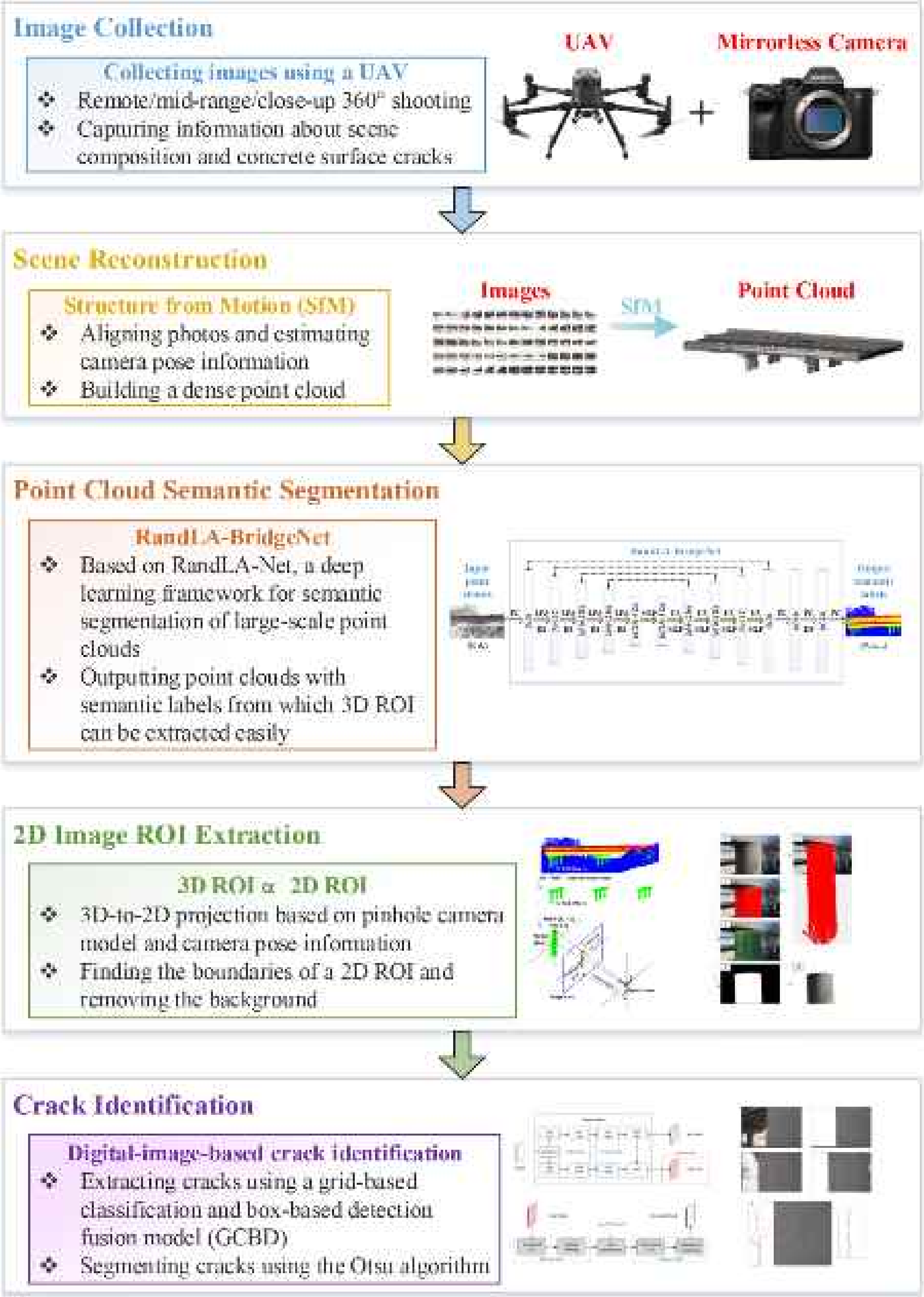


Image Collection
Collecting images using a UAV
Remote/mid-range/close-up 360° shooting
Capturing information about scene composition and concrete surface cracks
UAV
Mirrorless Camera
+
Scene Reconstruction
Structure from Motion (SfM)
Aligning photos and estimating camera pose information
Building a dense point cloud
Images
SfM
Point Cloud
Point Cloud Semantic Segmentation
RandLA-BridgeNet
Based on RandLA-Net, a deep learning framework for semantic segmentation of large-scale point clouds
Outputting point clouds with semantic labels from which 3D ROI can be extracted easily
2D Image ROI Extraction
3D ROI ⍺ 2D ROI
3D-to-2D projection based on pinhole camera model and camera pose information
Finding the boundaries of a 2D ROI and removing the background
Crack Identification
Digital-image-based crack identification
Extracting cracks using a grid-based classification and box-based detection fusion model (GCBD)
Segmenting cracks using the Otsu algorithm

Fig. 1. Methodology framework.

extraction method; Section 4 describes the digital-image-based crack identification method; Section 5 describes the experimental study on a real bridge for validating the proposed methodology; and Section 6 concludes this work.

3. Point cloud segmentation technique for extracting 3D ROI

3.1. Overview of point cloud segmentation

3D point cloud segmentation is a key research area in computer vision that aims to classify each point of the point cloud into one of several classes based on its spatial location, color feature, semantic information, etc. Classic segmentation methods include edge-based techniques, region growing, model fitting, and unsupervised clustering [22]. More recently, supervised deep learning methods have gained prominence [23], with voxel-based [24–26], multiview-based [27–29], and point cloud-based [30–35] methods emerging. Among these, point cloud-based methods have become common due to their ability to avoid partial information loss resulting from data preprocessing.

Some open-source deep learning semantic segmentation frameworks based on point clouds have been proposed, starting with the classical PointNet by Qi et al. [30]. PointNet directly uses 3D point clouds as input and has become the basis for many subsequently proposed methods. This framework, however, focuses too highly on global features, ignores local features, and does not consider the adverse effects of uneven point cloud density, which makes adapting it to complex scenes difficult. Thus, Qi et al. [31] proposed PointNet++, which overcomes the problems of feature extraction methods to a certain extent. However, it adopts the K-nearest neighbor search method, which may lead to the concentration of sampling points in one direction. Point cloud data are usually disordered and have density inhomogeneity. Therefore, Li et al. [32] proposed PointCNN to learn the local relationships of point clouds in space, which effectively reduces the time complexity and space complexity of segmentation. To address PointNet ignoring the correlation between neighboring points, Wang et al. [33] proposed a graph convolution-based DGCNN, which includes an EdgeConv operation that captures the distance information between each point and its neighboring points to learn edge features. While these frameworks are suitable for small-scale scenarios, they require block sampling when handling large-scale point clouds. More specifically, large-scale point clouds must be cut into 1 m × 1 m small blocks, and then each block must be sampled to obtain 4096 points as the network input [30–33]. To fully adapt to large-scale point clouds, Landrieu et al. [34] proposed the SPG framework based on the superpoint graph. SPG first divides the point cloud into geometrically simple but meaningful sets of superpoints, forms a superpoint graph, and then embeds each superpoint into a PointNet for semantic segmentation. However, dividing the superpoints is difficult to implement and prone to classification errors. Hu et al. [35] proposed RandLA-Net, a new framework that can directly handle large-scale point clouds. RandLA-Net achieved good segmentation results in the public large indoor and outdoor datasets S3DIS [36], Semantic3D [37] and SemanticKITTI [38].

In recent years, there have been several studies on point cloud segmentation of bridges. Due to the lack of a high-quality annotated bridge point cloud database, some researchers have suggested learning-independent segmentation methods [39–42]. Using the normal information of the points, Riveiro et al. [39] proposed a voxel-based method to recognize vertical and nonvertical components from the point cloud of a masonry arch bridge to divide the arch bridge into different parts. Yan et al. [40] proposed a heuristic algorithm for extracting structural components from the point clouds of steel bridges. Lu et al. [41] proposed a top-down point cloud segmentation algorithm for reinforced concrete bridges to complete bridge component recognition by stepwise classification. Truong-Hong et al. [42] used a cell- and voxel-based region growing method to extract surfaces individually from point clouds of reinforced concrete bridges. In conclusion, these segmentation methods use only one or a combination of classic point cloud segmentation techniques and depend heavily on domain knowledge such as geometric features specific to particular bridge types, e.g., common dimensions, axial orientation, and relative position relationships of components (piers, cap beams, girders, etc.). Therefore, these methods are only applicable to specific bridge types, and extending them to different scenarios may result in serious errors.

To address these issues, some scholars have applied deep learning-based methods to semantic segmentation of bridge point clouds. Kim et al. [43,44] utilized PointNet, PointCNN and DGCNN for bridge point cloud segmentation, and the three methods performed similarly overall. However, these methods require block sampling operations along the longitudinal direction of the bridge point cloud, and the size and overlap of the sampled blocks impact the segmentation results. Lee et al. [45] proposed hierarchical DGCNN (HGCNN) based on PointNet and DGCNN, which effectively improved the recognition of electric poles on bridges. Yang et al. [46] utilized the weighted SPG to directly process large-scale bridge point clouds, which performs better than PointNet and DGCNN and does not require block sampling. Referring to PointNet++, Jing et al. [47] developed BridgeNet for point cloud segmentation of masonry arch bridges and identified the bridge geometric parameters based on the segmented point clouds.

3.2. Proposed segmentation network RandLA-BridgeNet

3D point clouds of bridges obtained through SfM reconstruction often contain millions of points or more. While the classic PointNet framework and its variations are widely used, they rely on block sampling techniques to handle large-scale point clouds; these techniques can be sensitive to sampling parameters and may affect the segmentation results. To address this issue, the deep learning framework RandLA-Net [35] is adopted in this study to develop a robust point cloud semantic segmentation network called RandLA-BridgeNet, which directly takes the entire bridge point cloud as input. The network architecture is illustrated in Fig. 2. The network adopts an encoder-decoder architecture with residual connections. The input point cloud is progressively downsampled to extract the features of each point using a shared multilayer perceptron (MLP). Then, four encoding and decoding layers are utilized to learn the features of the points. Finally, three fully connected (FC) layers and a dropout layer are applied to predict the semantic labels of each point. Based on RandLA-Net [35], RandLA-BridgeNet follows most of the default parameter settings while adjusting the class definitions and the loss function to apply to the bridge point cloud dataset. Since the number of points of each class in the bridge dataset differs greatly, the weight of each class is calculated by dividing the number of points in each class by the total number of points in the dataset. Then, the value of 1/(weight + 0.02) is used as the weight for each class in the loss function.

To process a million-scale bridge point cloud directly with a deep neural network, it is necessary to gradually downsample while retaining as much geometric structure information as possible. Among the available sampling methods, farthest point sampling (FPS), inverse density importance sampling (IDIS), and generator-based sampling (GS) are computationally expensive, while continuous relaxation-based sampling (CRS) is demanding on GPU memory, and policy gradient-based sampling (PGS) has difficulty learning effective sampling strategies. Therefore, RandLA-BridgeNet adopts random sampling (RS), which is computationally efficient and has low memory overhead. However, RS results in a loss of useful information. To mitigate this issue, the network incorporates a local feature aggregation (LFA) module that complements RS. Fig. 2 illustrates the LFA module consisting of three submodules: Local Spatial Encoding (LocSE), Attentive Pooling, and Dilated Residual Block. The LocSE submodule encodes the 3D coordinate information and extracts neighborhood point features, enabling the network to better learn the geometric structure of the space from the relative position and distance information of points. Attentive pooling automatically learns

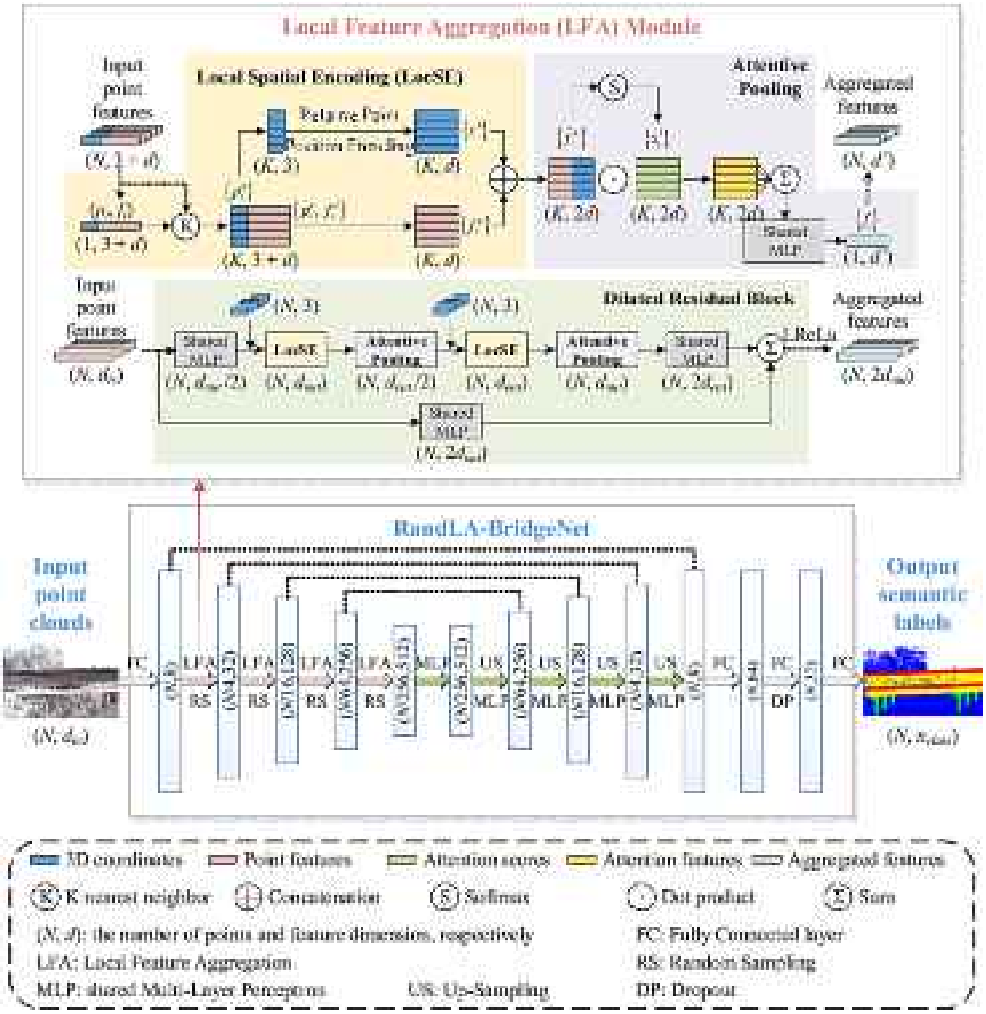


Fig. 2. Network architecture of RandLA-BridgeNet.

and aggregates useful information from neighboring point features. The dilated residual block connects two sets of LocSE and Attentive Pooling to cost-effectively increase the receptive field size of each point and facilitate feature propagation between neighboring points. The entire LFA module preserves the overall geometric details of the input point cloud even if the features of some points are randomly discarded by random downsampling.

3.3. Dataset

Fig. 3 and Table 1 present the dataset utilized to train and test RandLA-BridgeNet. The original data were provided by Lu et al. [41], who used a FARO Focus 3D X330 Terrestrial Laser Scanner to obtain the point clouds of ten reinforced concrete highway bridges. All the point clouds possess spatial location (XYZ) and color (RGB) information. To our knowledge, this is currently the most extensive open-source bridge point cloud database. The ten scanned reinforced concrete bridges represent the most typical type of highway bridges in China, North America and Europe; the bridges are composed of abutments, piers and continuous girders.

Before training the network, the original point clouds needed to be annotated. CloudCompare was employed to manually label the point clouds of the ten bridges, classifying the points into background, pier, superstructure and parapet and assigning the corresponding ground truth semantic labels. The final point cloud data used to train the network consisted of spatial location (XYZ), color (RGB), and semantic label information.

3.4. Network training and testing

RandLA-BridgeNet was implemented in Python using the TensorFlow platform. The high-performance CentOS 7 workstation used to train the network comprised an Intel Xeon Gold 6342 CPU, NVIDIA Ampere A 100 80 GB GPU and 100 GB RAM. The bridge point cloud dataset was split into three subsets: 60% of the data (6 bridges) was used for training, 20% (2 bridges) for validation, and 20% (2 bridges) for testing. More specifically, Bridges 1, 3, 4, 5, 7, and 8 were used for training; Bridges 2 and 10 were used for validation; Bridges 6 and 9 were used for testing.

To quantitatively evaluate the effectiveness of semantic segmentation, several evaluation metrics commonly used in classification problems were selected. For each class, the precision, recall, intersection over union (IoU) and F1 score were computed. The overall evaluation metrics comprised overall accuracy (OA), average recall (AR), mean intersection over union (mIoU), and average F1 score. The evaluation metrics are defined as follows:

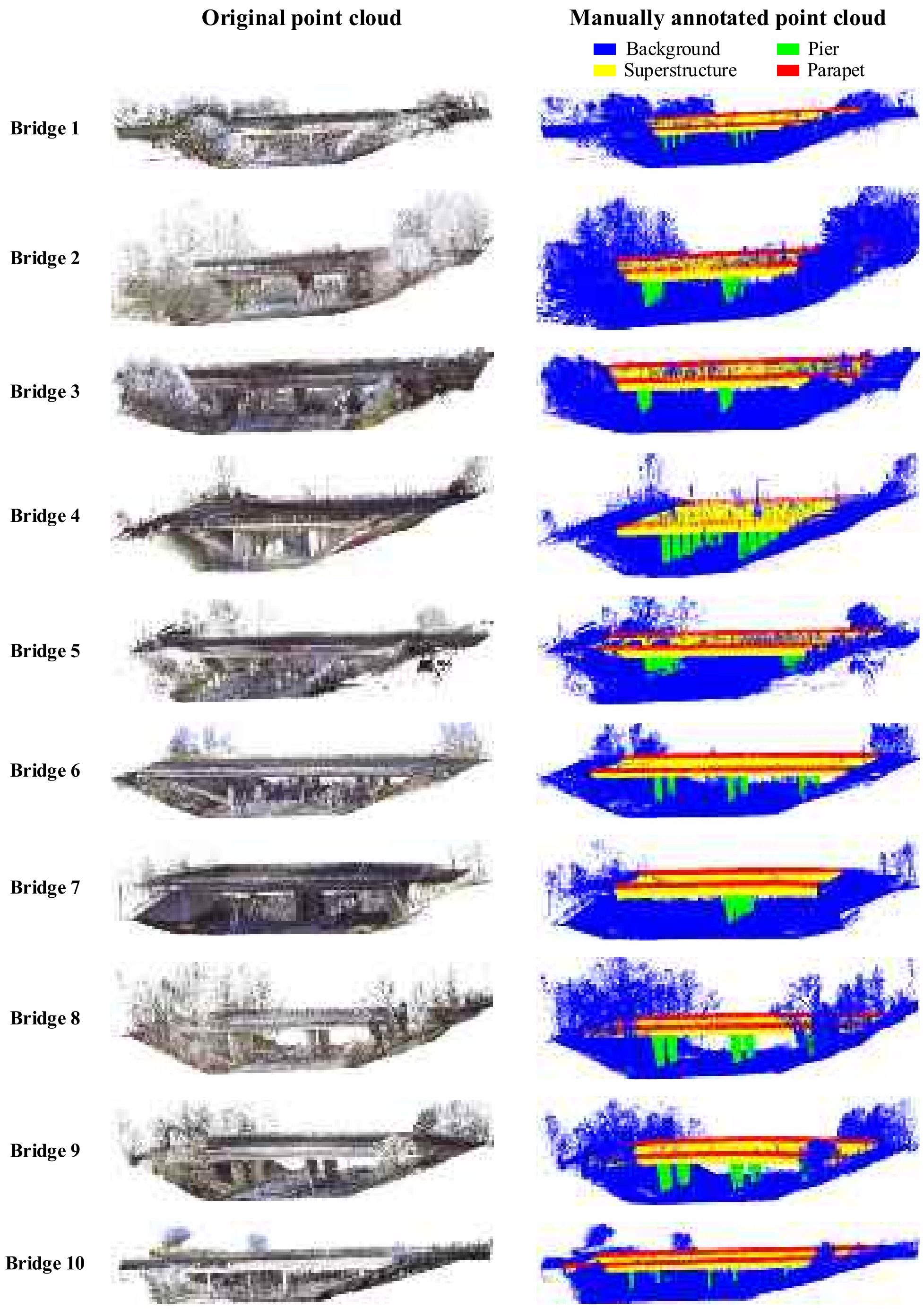


Fig. 3. Dataset used to train and test RandLA-BridgeNet.

Table 1
Metadata of the bridge point cloud dataset.

Bridge number	Number of points				
	Background	Pier	Superstructure	Parapet	Total
1	18,438,854	1,180,481	5,907,060	708,914	26,235,309
2	8,379,029	692,057	2,480,458	300,518	11,852,062
3	7,768,259	727,832	3,289,944	498,070	12,284,105
4	9,206,112	759,638	2,608,032	81,300	12,655,082
5	6,535,620	559,236	2,805,977	256,304	10,157,137
6	45,064,766	2,316,056	31,291,512	1,114,935	79,787,269
7	30,594,916	958,099	20,632,520	1,522,940	53,708,475
8	39,549,332	3,823,824	36,745,097	1,492,622	81,610,875
9	40,046,780	3,798,960	35,643,170	1,426,711	80,915,621
10	41,174,267	728,443	32,210,693	3,575,013	77,688,416

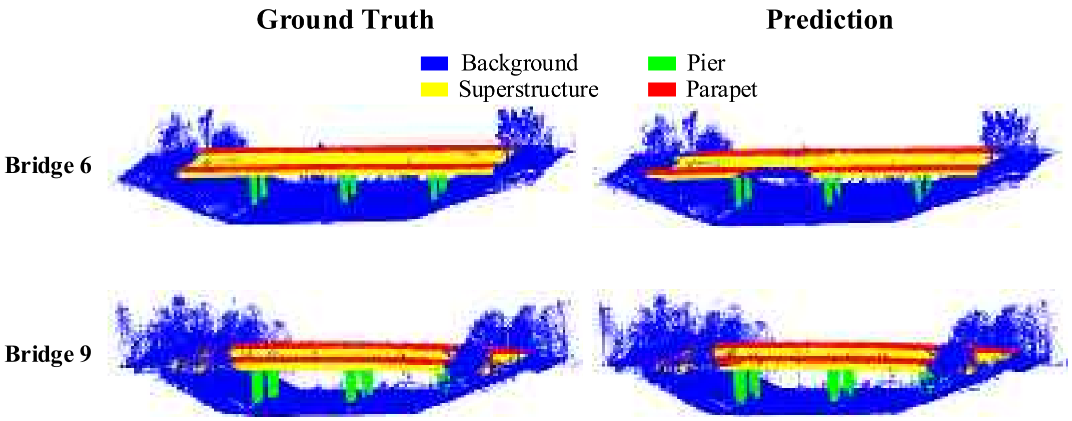


Fig. 4. Visualized comparison between prediction and ground truth of the test set.

Table 2
Quantitative evaluation of semantic segmentation of the test set.

Metrics for each class					Global metrics	
Metrics	Background	Pier	Superstructure	Parapet		
Precision	97.5%	97.5%	96.7%	94.7%	OA	97.1%
Recall	97.3%	90.6%	97.7%	90.7%	AR	94.1%
IoU	94.9%	88.5%	94.6%	86.3%	mIoU	91.1%
F1 score	97.4%	93.9%	97.2%	92.6%	Average F1 score	95.3%

$$\begin{cases} \text{Precision}_i = \dfrac{\text{TP}_i}{\text{TP}_i + \text{FP}_i} \\ \text{Recall}_i = \dfrac{\text{TP}_i}{\text{TP}_i + \text{FN}_i} \\ \text{IoU}_i = \dfrac{\text{TP}_i}{\text{TP}_i + \text{FN}_i + \text{FP}_i} \\ (\text{F1 score})_i = \dfrac{2 \times \text{Precision}_i \times \text{Recall}_i}{\text{Precision}_i + \text{Recall}_i} \\ \text{OA} = \dfrac{\sum_{i=1}^{n} \text{TP}_i}{N} \\ \text{AR} = \dfrac{\sum_{i=1}^{n} \text{Recall}_i}{n} \\ \text{mIoU}_i = \dfrac{\sum_{i=1}^{n} \text{IoU}_i}{n} \\ \text{Average F1 score} = \dfrac{\sum_{i=1}^{n} (\text{F1 score})_i}{n} \end{cases} \qquad (1)$$

where i represents the class number; TP stands for true positive, denoting the number of elements belonging to class i that are correctly classified into class i; FP stands for false positive, denoting the number of elements not belonging to class i that are falsely classified into class i; FN stands for false negative, denoting the number of elements belonging to class i that are falsely classified into the other classes instead of class i; and N and n are the total number of samples and classes, respectively.

The aforementioned evaluation metrics are applicable to classification tasks other than semantic segmentation, including image ROI extraction in Section 4, which can be regarded as a binary classification problem.

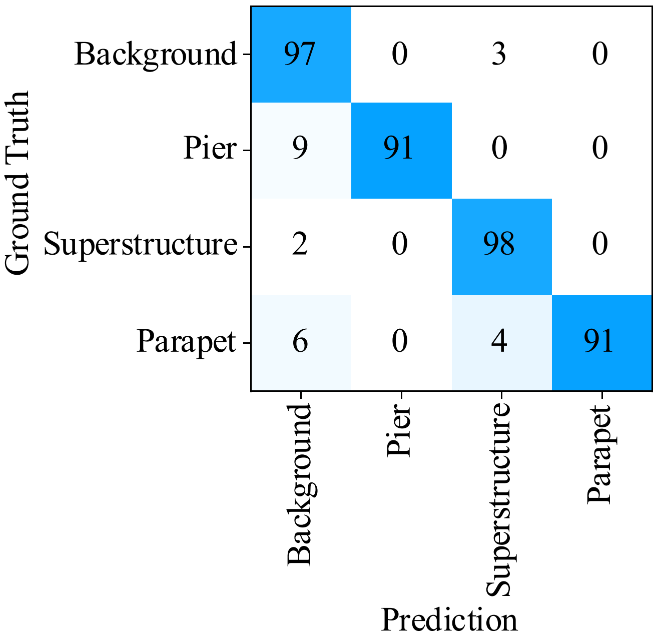


Fig. 5. Confusion matrix of RandLA-BridgeNet on the test set.

During the 100 training epochs, RandLA-BridgeNet achieved the best mIoU of 91.6% on the validation set at epoch 54, and the training process took a total of 17.2 h. The trained network was then applied to the test set. These results are visualized in Fig. 4, which shows a comparison between the prediction and ground truth. Overall, the network achieved good results in semantic segmentation. However, it still made errors at the boundaries of different parts due to the ambiguity of semantic information in those areas, meaning that one point may belong to more than one class simultaneously. For example, the points at the roots of the parapet can also be regarded as belonging to the superstructure class. In addition, the successive downsampling and upsampling operations of RandLA-BridgeNet may amplify these ambiguous areas.

Table 2 and Fig. 5 present the quantitative evaluations of the semantic segmentation of the test set. RandLA-BridgeNet performs remarkably overall, indicated by an OA of 97.1%. The network achieved

Table 3
Comparison of performance metrics between representative models.

Segmentation network	OA	mIoU	IoU for each class	
			Pier	Parapet
PointNet [46]	97.9%	90.0%	92.5%	85.8%
DGCNN [46]	97.3%	88.1%	96.7%	81.4%
SPG [46]	97.6%	92.4%	89.3%	90.3%
WSPG [46]	99.4%	96.5%	99.8%	90.0%
RandLA-BridgeNet (this study)	97.1%	91.1%	88.5%	86.3%

impressive results in all four classes, including background, pier, superstructure, and parapet, with F1 scores above 92% for each class. However, as indicated by the confusion matrix shown in Fig. 5, a small number of the pier points were misclassified as background due to the presence of a transition zone between the road and the root of each pier. There were also slight errors between the background and the superstructure due to vehicles, vegetation, and scanning artifacts on the bridge being included as background. Furthermore, the parapet stands directly on the bridge deck alongside these background elements, resulting in slight misclassification between the parapet, the superstructure and the background.

Table 3 compares some of the results in Table 2 with the results reported by Yang et al. [46]; both studies use the same dataset provided by Lu et al. [41]. However, Yang et al. [46] manually removed the extraneous background points from the original dataset and annotated the remaining points as pier, pier cap, girder, deck and parapet. Therefore, only the classes involved in both their paper and this study are listed in Table 3. Overall, RandLA-BridgeNet performs only slightly lower in most metrics than the state-of-the-art baselines, and it outperforms the PointNet and the DGCNN in the mIoU and the IoU for the parapet class. Since processing large-scale point clouds with extraneous background points directly is much more challenging than processing the cleaned point clouds of bridges, the performance of RandLA-BridgeNet is quite impressive.

4. Image ROI extraction based on 3D-to-2D projection

Upon deciding which bridge is to be inspected, plenty of on-site images should be captured. SfM can then be performed to reconstruct the 3D point cloud. The resultant point cloud is input into the trained RandLA-BridgeNet to obtain semantic labels, and the components requiring detection are then processed individually.

4.1. 3D-to-2D projection of ROI

Fig. 6 illustrates the 3D-to-2D projection principle of a bridge component, using pier 1 as an example. The current component of interest is pier 1, and the corresponding point cloud is defined as the 3D ROI. Points labeled as the pier are set to be visible in CloudCompare, while points with other labels are set to be invisible. Then, pier 1 can be easily extracted from the visible points.

The next step is finding the projection of the 3D ROI in image I containing pier 1. For any point P in the 3D ROI, its coordinates under the world coordinate system are (X, Y, Z). Then, the coordinates (x, y, z) of point P under the camera coordinate system xyz can be calculated by the following equation:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \mathbf{R}\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + \mathbf{T} \tag{2}$$

where **R** and **T** are the rotation matrix (3 × 3) and translation matrix (3 × 1) of image *I*, respectively, which represent the correspondence relationship between the camera coordinate system and the world coordinate system and are computed by the SfM algorithm.

According to the pinhole camera model, the following equation in the homogenous coordinate format holds:

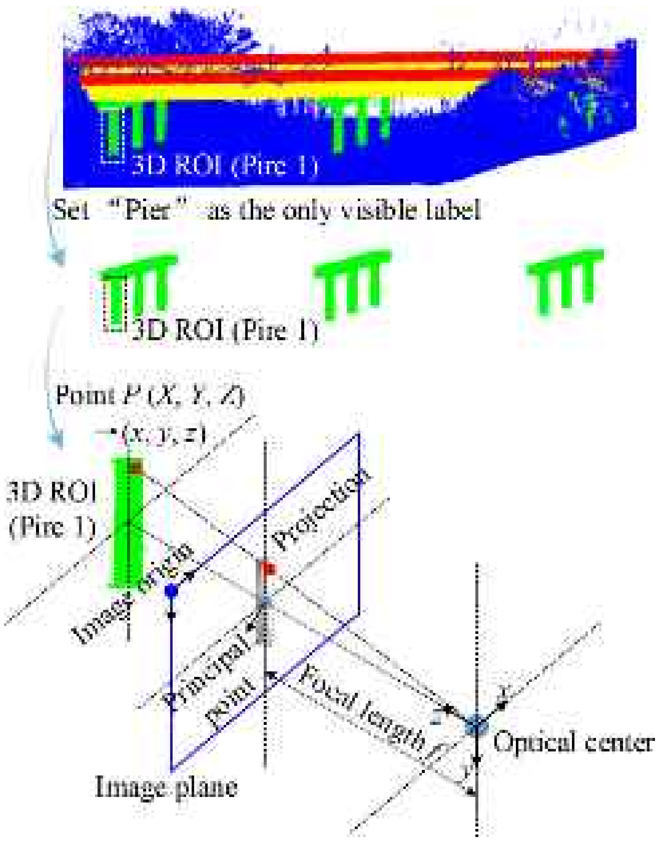


Fig. 6. Schematic of 3D-to-2D projection.

$$z\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & \frac{w}{2} \\ 0 & f & \frac{h}{2} \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \mathbf{K}\begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{3}$$

where f is the focal length of the camera; (u, v) are the coordinates of the projection point p in the image coordinate system where the origin is the upper left point; and **K** is the intrinsic parameter matrix (3 × 3) of the camera representing the correspondence relationship between the camera coordinate system and the image coordinate system, which is provided by the camera manufacturer and shown in the following equation:

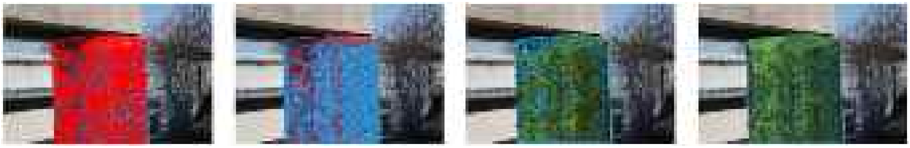
$$\mathbf{K} = \begin{bmatrix} f & 0 & \frac{w}{2} \\ 0 & f & \frac{h}{2} \\ 0 & 0 & 1 \end{bmatrix} \tag{4}$$

where w and h are the width and height of the image, respectively.

Then, the coordinates of the projection point p in the image coordinate system can be obtained as follows:

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \frac{x}{z}f + \frac{w}{2} \\ \frac{y}{z}f + \frac{h}{2} \end{bmatrix} \tag{5}$$

Note that all the scalars in Eqs. (2)–(5) should be unified in units, and converting all units into pixels is recommended.



(a) α=1 (b) α=10 (c) α=100 (d) α=1000

Fig. 7. Parametric analysis of the alpha shape algorithm.

4.2. Boundary detection of projection points

Using the method mentioned in the previous subsection, all points in the 3D ROI are projected into the 2D image. The resultant discrete projected points represent the 2D ROI. The alpha shape algorithm [48] is used in this study to compute a series of boundary line segments and generate a polygon that encloses the ROI.

The alpha shape algorithm can be implemented in MATLAB R2022a [49] using the "alphaShape" function, which relies on the value of the parameter α. Fig. 7 shows the boundary detection results using different α values, taking an image of a bridge pier as an example. The red points denote the projected points. The blue lines and green regions denote the detected boundary line segments and generated polygons, respectively. When α is set to 1 or 10, the generated polygons have many intersecting polylines, and the enclosing effect is weak. When α is set to 100, the generated polygons can enclose a portion of the projection points, but voids still exist inside. When α is set to 1000, the algorithm successfully captures the outer contours of the projection points and the generated polygons effectively enclose all the projection points. Thus, setting α to a larger value is recommended for common nonporous bridge components or surfaces. Since the images involved in this study do not exceed 10,000 pixels in either width or height, setting α to 1000 is appropriate. Other details and complete processing steps of this example are described in Section 4.3.

4.3. Batch processing algorithm for image ROI extraction

As shown in Fig. 8, the methods described above are integrated into an automatic MATLAB R2022a algorithm. The integrated algorithm can batch process all images containing the current component of interest by using the 3D ROI as input and outputting images that contain only the ROI. The algorithm has excellent operational efficiency, requiring an average processing time of only 8.3 s per 9504 × 6336 pixel image.

The right half of Fig. 8 illustrates the processing flow for a single image. Image 1 shows an original image, which contains the upper half of a pier (the current ROI) and its connection area with the pier cap. After performing the 3D-to-2D projection according to the pinhole camera model, image 2 is obtained. Since the 3D ROI includes the whole pier while the original image contains only the upper half of the pier, many projection points fall outside the scope of the image. After deleting these overrun points, image 3 is obtained. Then, the algorithm calls the "alphaShape" function to detect the boundaries of the projection points,

Algorithm: Batch processing for extracting the ROI from images

Input: 3D ROI matrix **P** ($N \times 3$); Original images $\mathbf{I_O}$; Camera parameters f_i, $\mathbf{R_i}$ (3×3), $\mathbf{T_i}$ (3×1) and $\mathbf{K_i}$ (3×3) of $\mathbf{I_{Oi}}$ in $\mathbf{I_O}$; Threshold value m for judging the correspondence between images and the current ROI

Output: Images $\mathbf{I_{ROI}}$ containing the ROI alone

```
Load $\mathbf{P}$ and $\mathbf{I_O}$

for $\mathbf{I_{Oi}}$ in $\mathbf{I_O}$:   (Image 1)
    Extend $\mathbf{T_i}$ to a $3 \times N$ matrix: $\mathbf{T_i}$ = repmat($\mathbf{T_i}$, 1, $N$)
    Calculate projection $\mathbf{P_C} = [\mathbf{K_i}(\mathbf{R_i}\mathbf{P}^T + \mathbf{T_i})]^T$
    2D ROI in image coordinate system $\mathbf{p_I} = [\mathbf{P_c}(:, 1) ./ \mathbf{P_c}(:, 3), \mathbf{P_c}(:, 2) ./ \mathbf{P_c}(:, 3)]$   (Image 2)

    for the j-th point $\mathbf{p_{Ij}}$ in $\mathbf{p_I}$:
      if $\mathbf{p_{Ij}}$ is outside of $\mathbf{I_{Oi}}$:
        Delete $\mathbf{p_{Ij}}$
      end
    end   (Image 3)
    if $\mathbf{p_{Ij}}$ contains less than $m$ (default value = 100)
      continue
    end

    Generate alphaShape on $\mathbf{p_I}$   (Image 4)
    Create a mask using the boundary of alphaShape (Image 5)
    Remove pixels of $\mathbf{I_{Oi}}$ that are located outside of the mask
    Save image $\mathbf{I_{ROIi}}$   (Image 6)
end
```

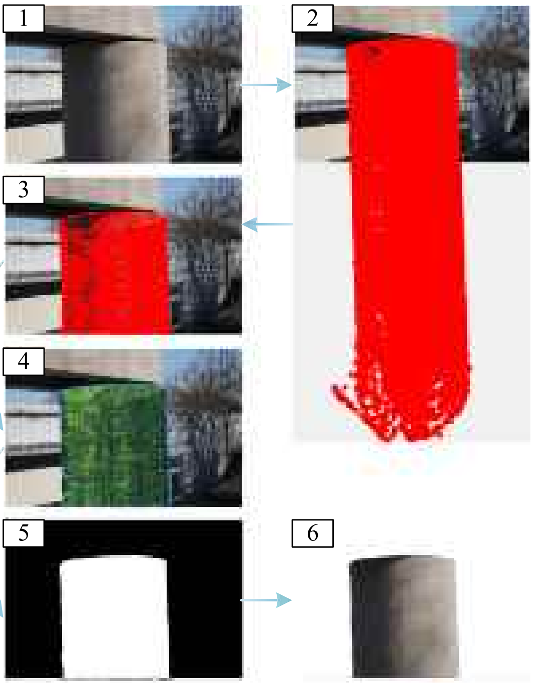


Fig. 8. Batch processing algorithm for image ROI extraction.

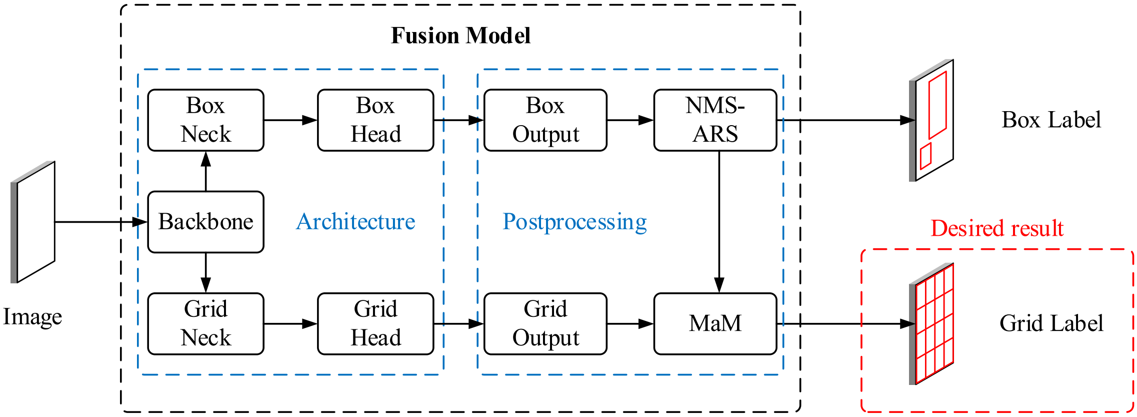


Fig. 9. Network architecture of the GCBD fusion model.

as shown in image 4. These boundary points are then used to generate a mask, as shown in image 5. Finally, the pixels outside the mask are removed to obtain image 6, which contains only the ROI. Notably, this algorithm not only separates the ROI from the background but also removes the background so that the resulting images can be directly used for crack identification.

5. Crack identification method

Digital image-based crack identification can be divided into two steps: crack extraction and crack segmentation. A data-driven deep learning convolutional neural network (CNN) called the grid-based classification and box-based detection fusion model (GCBD) is used for crack extraction [50]. The deep-learning-based method is more robust than traditional machine learning methods and can handle complex scenes in practical engineering. A typical threshold-based segmentation method [51] in digital image processing (DIP) is adapted for crack segmentation. The histogram distribution of cracks is bimodal, and the threshold-based method can obtain ideal crack segmentation results.

5.1. GCBD fusion model for crack extraction

The deep learning model used to extract cracks is the GCBD fusion model. The model has two output results: grid-based classification results and box-based detection results. In this paper, the grid-based classification results are the desired output, and the box-based detection results are not needed.

5.1.1. Grid-based classification branch

The network architecture has two branches, the grid-based classification branch and the box-based detection branch, as shown in Fig. 9. In this paper, we focus on the grid-based classification branch. The grid-based classification branch of the network resizes the image to 1440 × 960 and outputs a 45 by 30 grid mask. This branch integrates the features extracted by the backbone from three scales through the grid neck and finally outputs a grid mask through a layer of convolution on the 5-fold subsampled feature map.

Each grid cell in the grid mask has a confidence level. An appropriate threshold is set based on the scenario and the requirements for filtering out the grid cells below the threshold. The threshold can be chosen by testing model performance on a small dataset, which in this paper is size 5. The smaller the threshold is, the higher the recall. The larger the threshold is, the higher the precision. In identifying pier surface cracks, a threshold of 0.3 is set to find as many cracks as possible. A low threshold leads to some misidentifications, which are mostly non-pier surface disturbances that can be filtered through the background exclusion method proposed in the paper. In addition, when the threshold is set to approximately 0.3, the F1 score reaches the maximum value, and the recall and precision are perfectly balanced, as shown in Fig. 10. The remaining grid cells above the threshold are treated as areas containing cracks for further crack segmentation.

In addition, since only the grid-based classification branch is needed in this paper, the box neck and box head in the network can be removed through a pruning operation. This improves computing efficiency and does not affect grid output.

5.1.2. Generalization on OOD data

Concrete bridge images have different data distributions than asphalt pavement images. However, the weights used in the test are those trained on the asphalt pavement image dataset due to the lack of an available surface crack image dataset for concrete bridges. Thus, directly applying the fusion model on concrete piers requires strong out-of-distribution (OOD) generalization performance.

Because the fusion model adopts a shared backbone network, multitask learning and joint training, it is highly robust. The grid-based classification branch focuses on the local area, and the box-based detection branch focuses on the whole region. Fusing two tasks with different objectives drives the model to capture the common features of the cracks at both micro and macro scales. The experimental results show that the weight is still well generalized on concrete bridge surfaces, which will be demonstrated in Section 6.5.

5.2. Crack segmentation

5.2.1. Threshold-based method

Crack segmentation is performed in each grid cell using the Otsu algorithm [51]. The crack segmentation algorithm can be divided into three steps: preprocessing, segmentation and postprocessing, as shown in Fig. 11.

Preprocessing can be divided into two parts: image preprocessing and cracked region preprocessing. For cracked regions, grid cells with obvious misidentification are filtered out based on the confidence of each grid cell and the connectivity between all grid cells through connected component analysis (CCA). If a connected area is small and the average confidence is low, this area is considered an obvious misidentification area for filtering. Afterward, the median filter is used to smooth the image and filter out the salt and pepper noise.

The maximum interclass variance is calculated in each grid cell to obtain the local Otsu segmentation threshold. Because the image resolution is high and the coverage is wide, different areas in the image have inconsistent lighting. Therefore, using one threshold segmentation for the entire image will cause the local tiny crack to not be identified. This problem can be alleviated by using the local threshold method based on grid cells.

Preprocessing improves the reliability of the crack segmentation results on the macro scale, while postprocessing further refines the crack

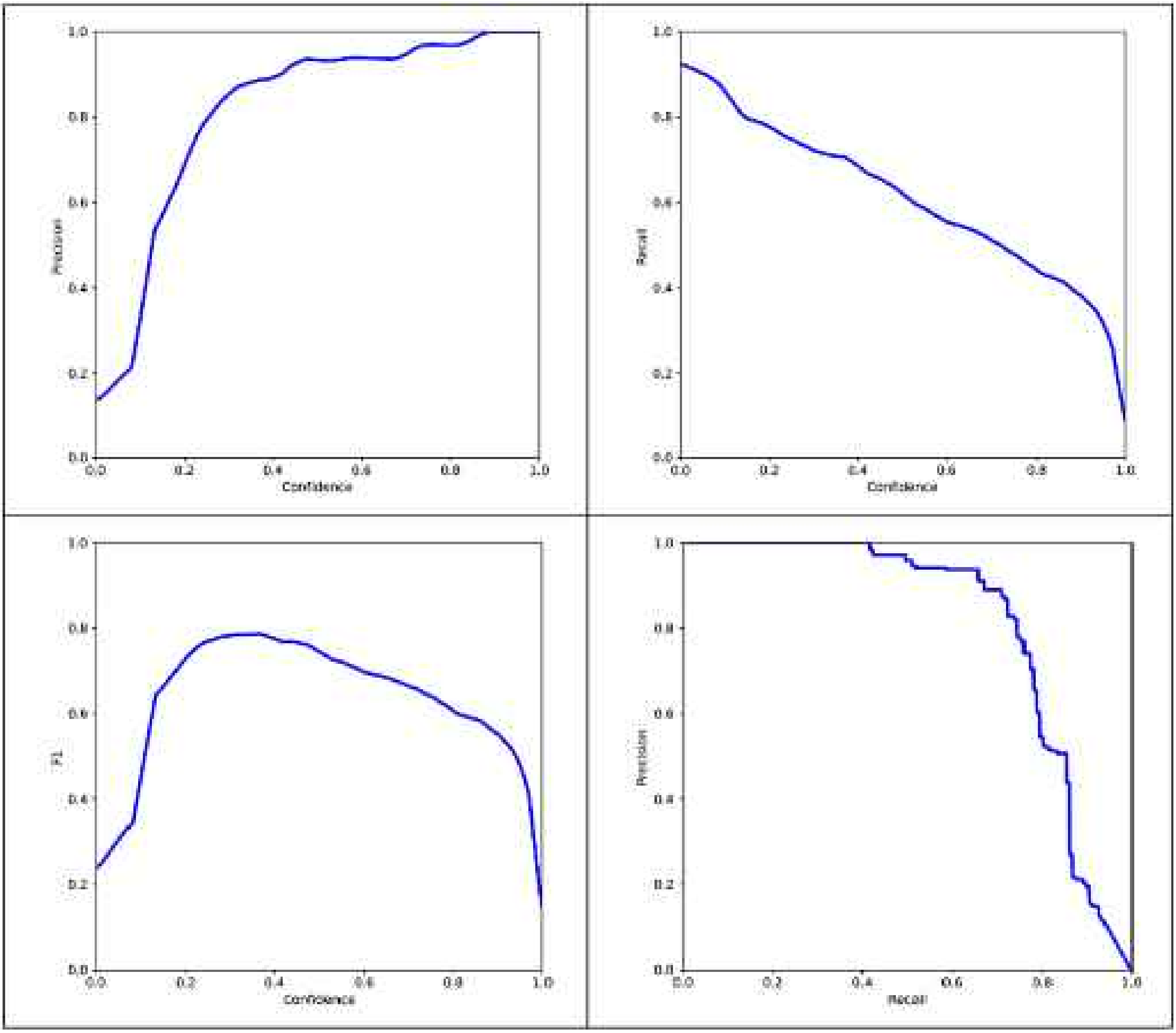


Fig. 10. Performance on the pier image dataset.

segmentation results on the micro scale. Calculating the connection relation of pixel points filters out noise such as holes. Then, the expansion corrosion closure operation is used to address edge nonclosure and internal cavities.

5.2.2. *Segmentation performance*

To test the performance of the segmentation algorithm, experiments are performed on the concrete crack dataset [53]. The edge of the crack is fuzzy, and there is a transition area between the crack pixel and the noncrack pixel, so the two adjacent pixels around the crack pixel can

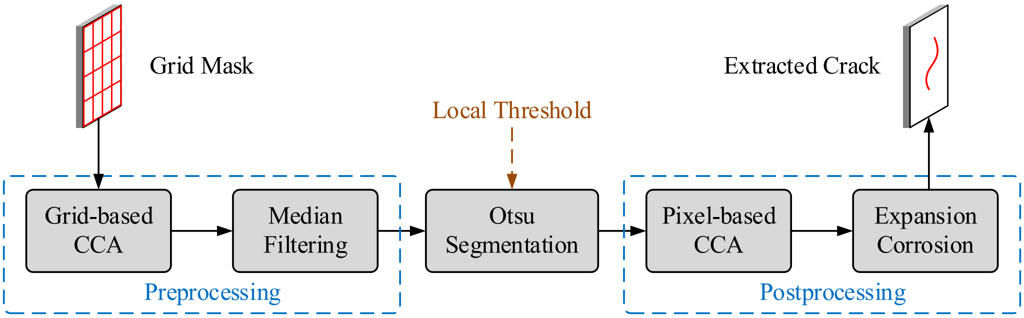
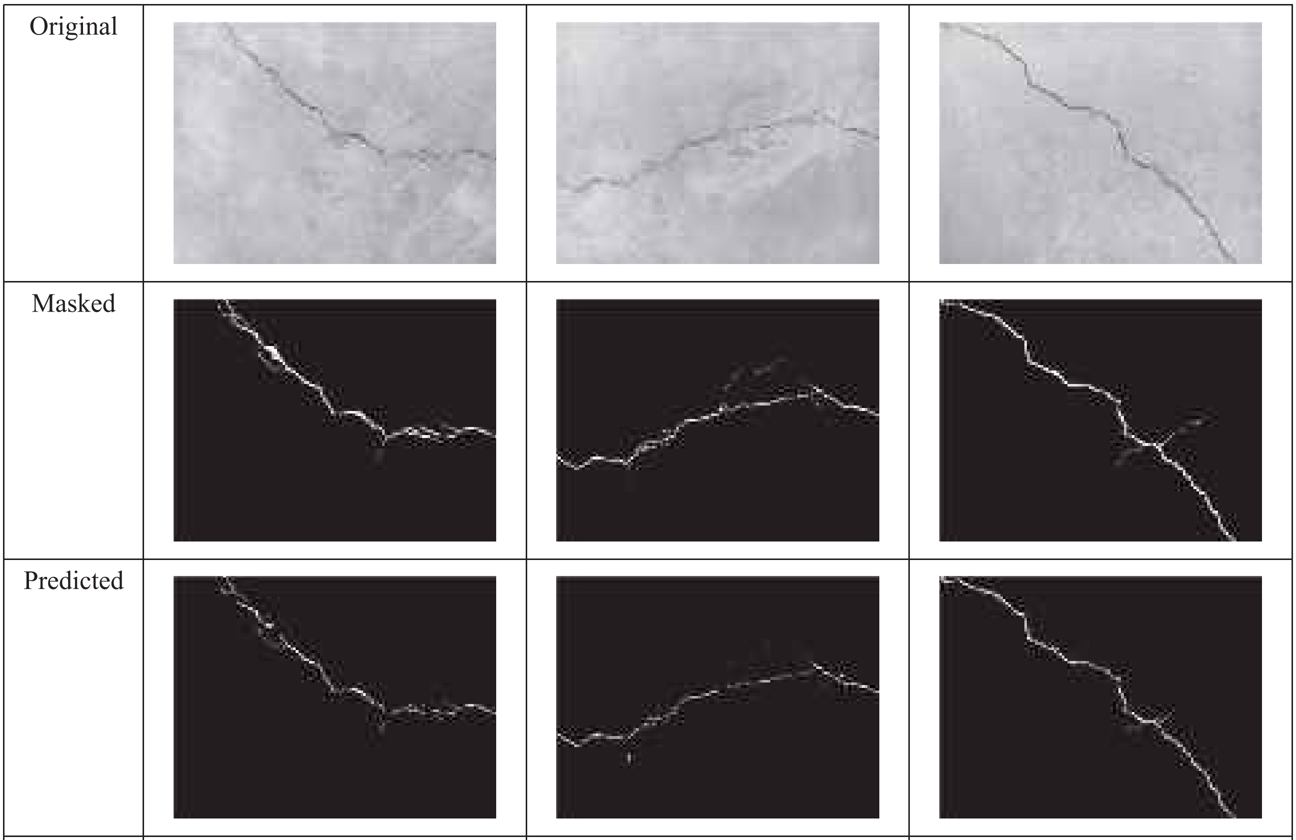


Fig. 11. Crack segmentation process.



Precision	88.7%	92.1%	95.2%
Recall	75.9%	72.8%	77.4%
F1 score	81.8%	81.2%	85.4%

Fig. 12. Crack segmentation results.

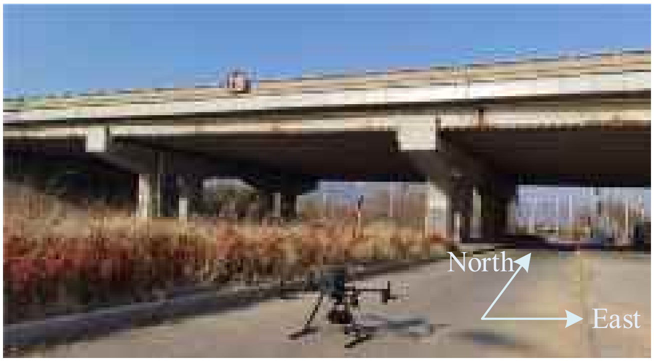


Fig. 13. Experimental scene: G7 bridge.

also be classified as true positives [54]. Precision, recall and F1 score as defined in Eq. (1) are used to evaluate the performance of the segmentation algorithm. Fig. 12 shows the results of crack segmentation in three typical images.

The experimental results show that the proposed segmentation algorithm can effectively extract cracks on the surface of concrete structures. The precision shows that the accuracy of the crack pixels identified by the algorithm can reach 92.0%. The recall shows that the proportion of cracked pixels identified by the algorithm to the actual cracked pixels is 75.4%. The F1 score reaches 82.8%, indicating that crack segmentation has excellent performance in general.

6. Experimental validation and discussion

To validate the proposed methodology, experimental studies were carried out in a real-world scenario. To find a suitable experimental scene, the authors conducted a field survey of 12 bridges along the G7 Beijing-Xinjiang Expressway and G6 Beijing-Tibet Expressway in China.

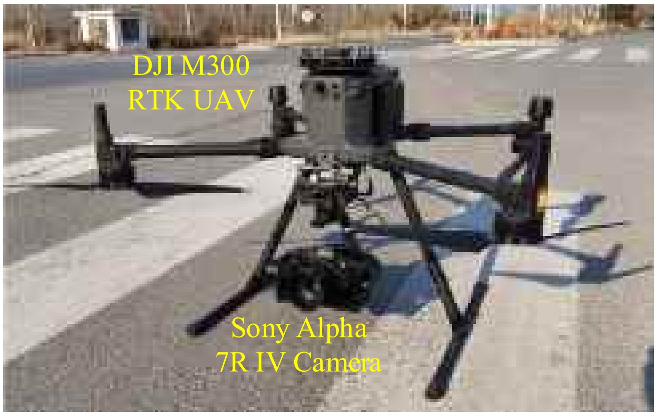
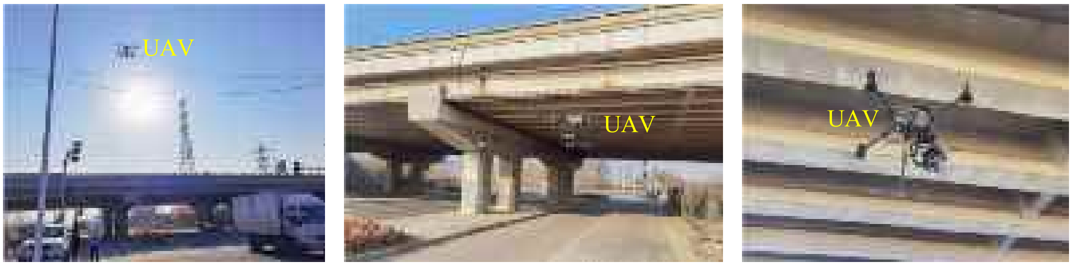


Fig. 14. DJI M300 RTK UAV equipped with a Sony Alpha 7R IV mirrorless camera and a self-developed gimbal system.



(a) Remote shooting (b) Mid-range shooting (c) Close-up shooting

Fig. 15. UAV executing the image collection mission.

The survey revealed that one bridge along the G7 Beijing-Xinjiang Expressway was moderately sized and had cracks on the surfaces of several concrete piers, making it a suitable candidate for this experiment. Therefore, this bridge (referred to as the G7 bridge hereafter) was selected as the experimental scene, as demonstrated in Fig. 13. The G7 bridge is a three-span continuous concrete bridge with six lanes in total, comprising 8 prismatic piers arranged in two rows.

The experiment was carried out by following the process shown in Fig. 1, and the results of each step are discussed in detail in the following subsections.

6.1. Image collection using UAV

This experiment utilized a DJI M300 RTK UAV equipped with a Sony Alpha 7R IV mirrorless camera and a self-developed gimbal system, as illustrated in Fig. 14. The Sony Alpha 7R IV was selected due to its sensor with up to 61 effective megapixels and lightweight compact body weighing only 665 g, which does not significantly increase the UAV's power consumption. Note that the camera selection should consider the detectable crack width standard required by the inspection criteria. Assuming that cracks wider than one pixel in the images can be detected, the following computed detectable crack width should surpass the required standard value:

$$\omega_{detectable} = \frac{FOV_w}{w} = \frac{S_w D}{fw} \quad \text{or} \quad \omega_{detectable} = \frac{FOV_h}{h} = \frac{S_h D}{fh} \tag{6}$$

where $\omega_{detectable}$ is the detectable crack width; FOV_w and FOV_h are the width and height of the field of view in millimeters, respectively; w and h are the width and height of the image in pixels, respectively; S_w and S_h are the width and height of the camera sensor in millimeters, respectively; D is the shooting distance; and f is the focal length of the camera in millimeters.

For the Sony Alpha 7R IV camera, S_w, S_h, w and h were 35.7 mm, 23.8 mm, 9504 pixels and 6336 pixels, respectively. The camera was equipped with a prime lens with a focal length f of 50 mm. Therefore, the detectable crack width $\omega_{detectable}$ was 0.3 mm when the shooting distance D was approximately 4 m. When the required detectable crack width is smaller or a camera with lower pixel resolution is used, a smaller shooting distance is recommended. Moreover, the superresolution processing technique can also be adopted to enhance the crack detection ability if necessary.

Since no product on the market integrates this camera with a UAV, our research team developed a gimbal system that supports the UAV to carry the camera. A corresponding software system was also developed to enable real-time control of the camera's three-axis rotation and shooting action via the UAV remoter.

Fig. 15 displays some photos of the UAV at work. The UAV was manually controlled to fly and photograph the G7 bridge from various angles and different distances. A total of 1577 images were obtained.

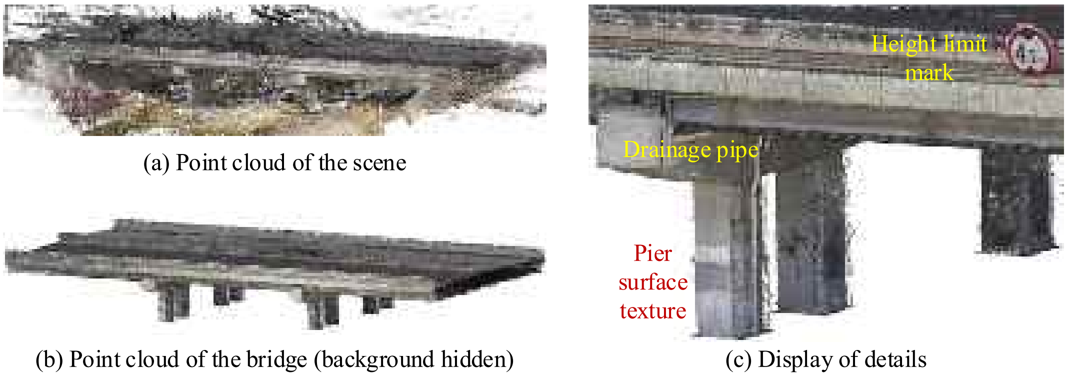
6.2. 3D scene reconstruction

The advanced commercial software Agisoft Metashape [52] was used to perform SfM-based 3D scene reconstruction on a high-performance Windows 10 workstation with an Intel Xeon W-2223 CPU and 192 GB RAM. All 1577 bridge images were imported into the software. The highest alignment accuracy was selected for photoalignment, which took approximately 2.5 h. The results showed that 1416 of the 1577 images were successfully aligned. Then, the estimated preselection mode in the reference preselection module of the alignment settings was selected. This mode uses estimated camera pose information to help process the photos that were not yet aligned. This step took approximately 1 min and resulted in the successful alignment of 1493 of the 1577 images. As discussed in Section 1, images that could not be aligned were mostly closely shot local images that lacked information about the remaining scene.

Next, a dense point cloud was built with the depth filtering mode set to aggressive to remove outlier points caused by noise or poor focus. To investigate the effect of the dense cloud quality setting, a parametric analysis was conducted. The results of this analysis are summarized in Table 4. For each degradation in the quality setting, the software downscales the preliminary image size by a factor of 4 (2 times on each side). When a relatively high quality is selected, depth map generation and dense cloud generation require a long processing time and a large amount of memory. Although high-performance workstations can be competent, these consumptions are not feasible for ordinary personal computers. In addition, when the 3D reconstruction yields too many points that make the file overly large, smoothly editing the exported point cloud in visualization software such as CloudCompare is difficult. Since the Sony Alpha 7R IV mirrorless camera captures high-resolution

Table 4
Parametric analysis of the dense cloud quality setting.

Quality setting	Depth map generation		Dense cloud generation		Number of points	File size/GB
	Processing time/h	Memory usage/GB	Processing time/h	Memory usage/GB		
Ultrahigh	47	21.22	32	135.18	3,690,575,911	57.53
High	15.8	21.8	9.5	70.05	953,498,233	14.02
Medium	3.9	5.87	3.5	35.48	333,855,364	4.84
Low	1.1	2.31	1	7.4	95,224,214	1.36
Lowest	0.5	1.96	0.5	5.59	31,993,679	0.46



(a) Point cloud of the scene

(b) Point cloud of the bridge (background hidden)

(c) Display of details

Fig. 16. Built point cloud of the G7 bridge (dense cloud quality: low).

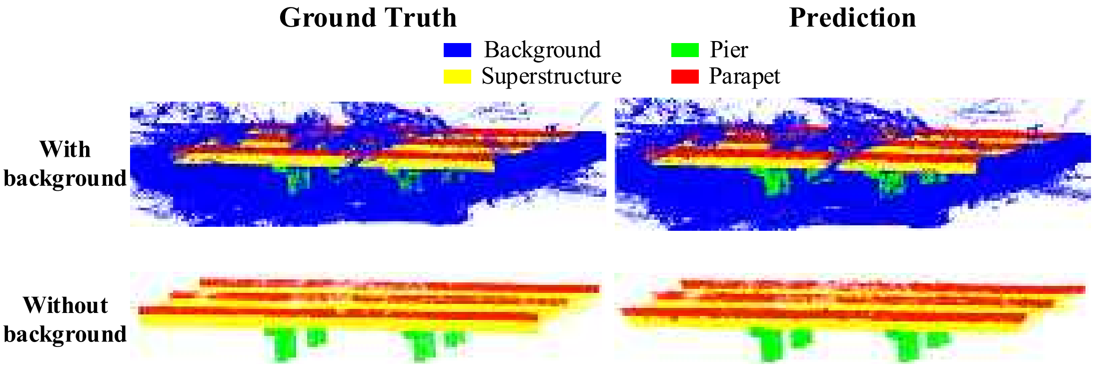


Fig. 17. Visualized comparison between prediction and ground truth of the G7 bridge.

Table 5
Quantitative evaluation of semantic segmentation of the G7 bridge.

Metrics for each class					Global metrics	
Metrics	Background	Pier	Superstructure	Parapet		
Precision	99.8%	83.8%	96.9%	82.4%	OA	97.0%
Recall	96.6%	99.9%	96.8%	99.8%	AR	98.3%
IoU	96.4%	83.8%	93.8%	82.3%	mIoU	89.1%
F1 score	98.2%	91.2%	96.8%	90.3%	Average F1 score	94.1%

images with 9504 × 6336 pixels, selecting a low quality can already generate 95,224,214 points with a point density similar to that of the dataset described in Section 3.3. Therefore, for this experiment and similar cases, a low dense cloud quality setting is recommended. The built point cloud is shown in Fig. 16. Details such as the drainage pipe, height limit mark and pier surface texture of the G7 bridge are quite

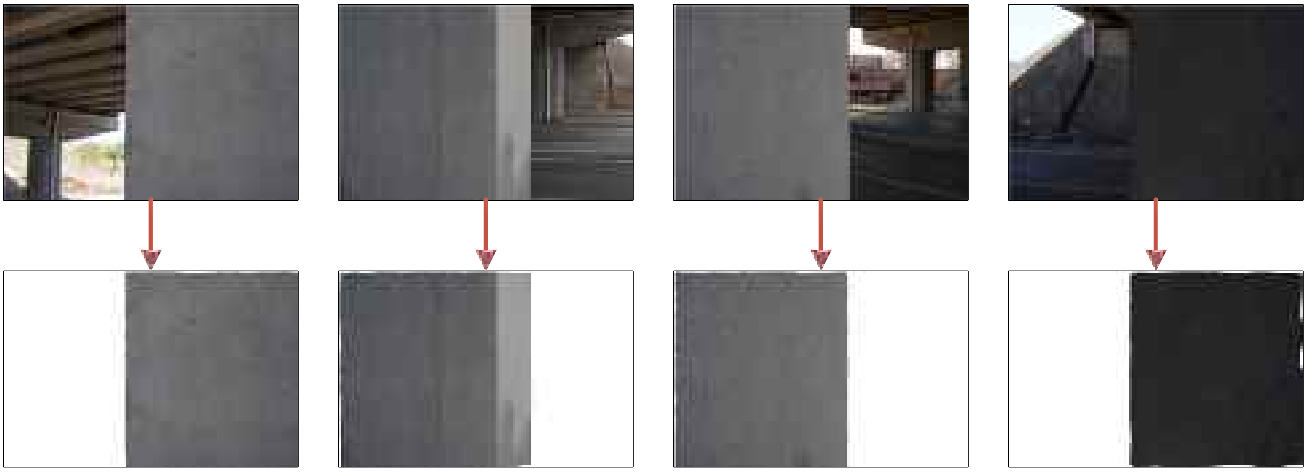


Fig. 18. ROI extraction results of some typical images.

Table 6
Quantitative evaluation of the image ROI extraction.

Metrics for each class			Global metrics	
Metrics	Background	ROI		
Precision	97.1%	99.9%	OA	98.9%
Recall	99.9%	98.2%	AR	99.1%
IoU	97.0%	98.2%	mIoU	97.6%
F1 score	98.5%	99.1%	Average F1 score	98.8%

clear, indicating that the scene reconstruction quality is good and the point density can meet the requirements. The subsequent processing and analysis are based on this point cloud.

6.3. *Point cloud semantic segmentation*

The built 3D point cloud of the G7 bridge was directly fed into the trained RandLA-BridgeNet to obtain a point cloud with predicted semantic labels. This cloud was obtained extremely quickly, with a processing time of only 125.4 s for the dense cloud containing approximately 95 million points from the approximately 40-m-long G7 bridge. CloudCompare was used to manually annotate the point cloud as the ground truth for further comparison. The comparison between the prediction and the ground truth of the G7 bridge is visualized in Fig. 17. The results show a small visual difference between the prediction and the ground truth, indicating that the semantic segmentation was generally successful.

The quantitative assessment results are presented in Table 5. The overall prediction performance is excellent, with the model achieving an OA of 97.0%, which is similar to that of the test set. The segmentation of the four classes, namely, background, pier, superstructure and parapet, is satisfactory, with F1 scores above 90% for each class.

6.4. *Image ROI extraction*

The identification targets of this experiment are the cracks on the concrete surfaces of piers. A total of 26 images containing crack information on pier surfaces were selected as the data source for crack identification. The 3D ROI corresponding to each image is the pier to which the cracked concrete surface in the image belongs; it can be easily extracted from the semantic-segmented point cloud of the G7 bridge, as described in Section 4.1. By batch processing the 26 images using the

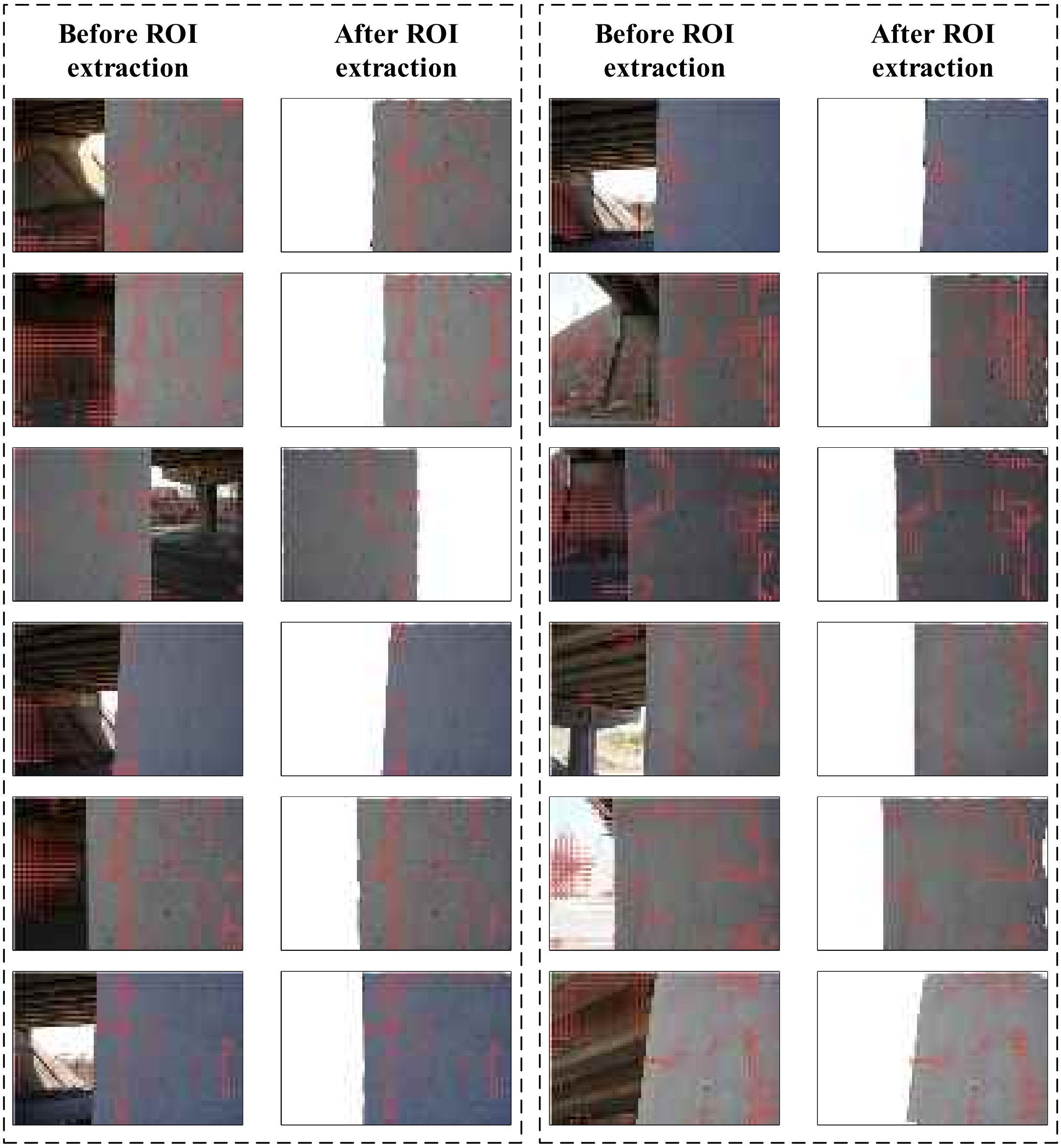


Fig. 19. Crack identification results of typical images before and after ROI extraction.

Table 7
Crack identification results.

Image number	Number of grid cells in ROI	Number of grid cells in background	Misidentification rate
1	144	149	50.9%
2	160	293	64.7%
3	56	44	44.0%
4	193	182	48.5%
5	117	30	20.4%
6	149	192	56.3%
7	146	10	6.4%
8	129	20	13.4%
9	57	65	53.3%
10	36	133	78.7%
11	141	216	60.5%
12	369	60	14.0%
13	142	60	29.7%
14	67	204	75.3%
15	45	246	84.5%
16	59	289	83.0%
17	210	65	23.6%
18	134	23	14.6%
19	113	19	14.4%
20	98	34	25.8%
21	99	33	25.0%
22	92	37	28.7%
23	173	168	49.3%
24	116	136	54.0%
25	163	140	46.2%
26	111	208	65.2%
All images	3319	3056	47.9%

algorithm described in Section 4.3, images containing only the 2D ROIs (cracked concrete surfaces of interest) were obtained.

Fig. 18 illustrates the ROI extraction results of some typical images. These images demonstrate the challenges mentioned by other researchers [19–21], showing that directly using a deep learning method for concrete component recognition in 2D images can be problematic. For the first three images, the deep learning method for 2D images recognizes all bridge piers as ROIs and keeps them, making directly obtaining the desired results impossible. The fourth image is taken in poor lighting conditions underneath the bridge, so the pier surface appears dark, while the abutment surface in the background appears brighter. Because of this, the deep learning method of 2D images is likely to misidentify the brighter abutment surface in the background as an ROI. The methodology proposed in this paper avoids this problem in principle, reasonably removing the background pixels and preserving the concrete surface of interest.

The aforementioned image ROI extraction can be regarded as a binary classification task to quantitatively evaluate its effectiveness. Specifically, this step involves classifying all pixels in each image as either background or ROI. The 26 images were manually annotated by removing the background to generate the ground truth. Table 6 presents the pixel-level evaluation metrics. There is only a slight difference between the evaluation metrics for the background and ROI. Moreover, all the evaluation metrics exceed 97%, indicating that the boundaries between the background and ROIs were accurately detected.

6.5. Crack identification

The images before and after ROI extraction were processed by the crack identification method. The crack identification results were obtained by extracting grid cells containing cracks from the GCDB fusion model, as shown in Fig. 19. When the background is excluded, the crack identification accuracy improves. Lines such as beams, branches, and the interface between the ROI and background can easily be misidentified as cracks when the background has not been excluded. Excluding the background can not only eliminate background line interference but also improve the accuracy of crack identification near the interface between the foreground and background.

The quantitative results of the crack identification are listed in Table 7, in which the misidentification rate denotes the ratio of the number of grid cells in the background to the total number of grid cells. Overall, the ROI extraction operation filters 47.9% of the grid cells; these filtered grid cells are misidentified background cells. Therefore, ROI extraction can effectively improve crack identification accuracy. Eliminating the interference of a complex background can make the network focus on the ROI, which is more consistent with the training data

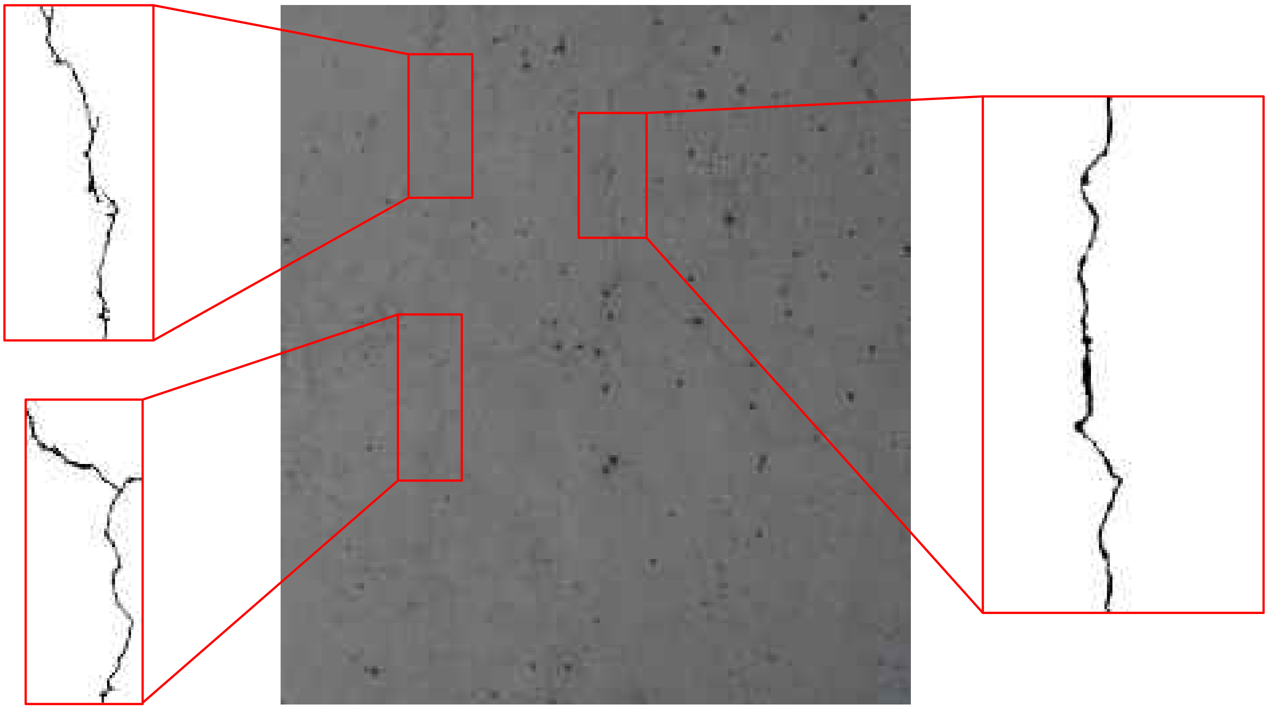


Fig. 20. Crack segmentation results of the concrete pier.

distribution and enhances the robustness of the network.

After crack extraction, the threshold segmentation method introduced in Section 5.2 is used to further segment cracks, as shown in Fig. 20. The proposed method can segment cracks effectively and can be used for subsequent crack assessment to assist with sophisticated maintenance decisions.

7. Conclusions

Accurately detecting cracks in concrete surface images with complex backgrounds is a challenging task. To improve the results of this task, an image ROI extraction methodology based on 3D point cloud semantic segmentation and 3D-to-2D projection is presented in this paper. First, a deep-learning-based semantic segmentation network RandLA-BridgeNet for large-scale bridge point clouds is constructed. A real-world bridge point cloud dataset is established for training and testing the network. Using the entire point cloud of the scene as input, RandLA-BridgeNet can perform semantic segmentation accurately and efficiently, achieving mIoUs of 91.6% and 91.1% on the validation set and the test set, respectively. Then, the 3D ROIs (concrete components of interest) are easily extracted from the segmented point cloud and projected into the corresponding images according to the pinhole camera model and camera pose information. Next, the alpha shape algorithm is used to detect the boundaries of the projected 2D ROI and remove the background, generating images that contain only the ROIs (concrete surfaces of interest). Finally, improved deep-learning-based crack identification can be performed using these processed images.

The methodology was validated by an experiment on an approximately 40-m-long bridge along the G7 Beijing-Xinjiang Expressway in China. For the point cloud reconstructed from 1577 UAV aerial images and containing approximately 95 million points, the inference of RandLA-BridgeNet took only 125.4 s. RandLA-BridgeNet achieved excellent semantic segmentation results, with F1 scores of 98.2%, 91.2%, 96.8% and 90.3% for the background, pier, superstructure and parapet, respectively. Image ROI extraction was performed on 26 images containing concrete surface cracks, with the overall extraction accuracy reaching 98.9%. A grid-based classification and box-based detection fusion model is used to identify cracks in the images. After ROI extraction, 47.9% of the grid cells, which represent background misrecognition, are filtered, greatly improving in the crack identification accuracy.

The presented methodology integrates point cloud semantic segmentation and 3D-to-2D projection technologies into the UAV-based bridge crack detection task, contributing to advancements in the field. As indicated by the field experimental validation presented in Section 6, the methodology framework shown in Fig. 1 has much potential for practical UAV-based bridge inspection applications and achieves impressive crack detection results when handling images containing complex background information.

However, some limitations still exist and call for future research efforts:

(1) Due to the relatively limited scenarios covered by the training data, the semantic segmentation network has limited applicability to various scenarios. A large open-source point cloud database that covers more bridge types needs to be established.

(2) The parameter setting method of the alpha shape algorithm needs to be further studied to extract the 2D ROI boundary accurately for bridge components or surfaces with holes.

(3) In the presented experiment, the manually controlled UAV flight was cumbersome and inefficient, requiring large battery consumption. Automatic path planning and control methods for UAV bridge inspection tasks need to be developed, and multiple UAVs may collaborate to further improve efficiency.

(4) The crack identification model used in this study was trained on the asphalt pavement image dataset due to the lack of available surface crack image datasets for concrete bridges. Although the identification results are generally satisfactory, transfer learning and fine tuning could feasibly further improve performance. The main feature extraction layer of asphalt pavement weight can be frozen, and a small amount of concrete surface data can be labeled to train the fusion model so that the model can better adapt to the data distribution of concrete bridge cracks. From another perspective, establishing a large concrete bridge crack image dataset for training a new crack identification model would also be meaningful.

CRediT authorship contribution statement

Jing-Lin Xiao: Data curation, Investigation, Methodology, Visualization, Writing – original draft. **Jian-Sheng Fan:** Conceptualization, Funding acquisition, Writing – review & editing, Supervision. **Yu-Fei Liu:** Conceptualization, Funding acquisition, Methodology, Supervision, Writing – review & editing. **Bao-Luo Li:** Investigation, Validation, Visualization. **Jian-Guo Nie:** Supervision, Writing – review & editing.

Declaration of Competing Interest

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

Data availability

Data will be made available on request.

Acknowledgments

The research is supported by the National Natural Science Foundation of China (No. 52192662 and 52121005). The authors express sincere appreciation for their contribution to this research.

References

[1] K. Chaiyasarn, A. Buatik, H. Mohamad, M. Zhou, S. Kongsilp, N. Poovarodom, Integrated pixel-level CNN-FCN crack detection via photogrammetric 3D texture mapping of concrete structures, Automation in Construction 140 (2022), 104388, https://doi.org/10.1016/j.autcon.2022.104388.

[2] S.Y. Kong, J.S. Fan, Y.F. Liu, X.C. Wei, X.W. Ma, Automated crack assessment and quantitative growth monitoring, Comput. Aided Civ. Inf. Eng. 36 (2021) 656–674, https://doi.org/10.1111/mice.12626.

[3] X. Tan, A. Abu-Obeidah, Y. Bao, H. Nassif, W. Nasreddine, Measurement and visualization of strains and cracks in CFRP post-tensioned fiber reinforced concrete beams using distributed fiber optic sensors, Automation in Construction 124 (2021), 103604, https://doi.org/10.1016/j.autcon.2021.103604.

[4] B.A. Graybeal, B.M. Phares, D.D. Rolander, M. Moore, G. Washer, Visual inspection of highway bridges, J. Nondestruct. Eval. 21 (3) (2002) 67–83, https://doi.org/10.1023/A:1022508121821.

[5] Y. Liu, S. Cho, B.F. Spencer, J. Fan, Automated assessment of cracks on concrete surfaces using adaptive digital image processing, Smart Struct. Syst. 14 (4) (2014) 719–741, https://doi.org/10.12989/sss.2014.14.4.719.

[6] R. Ali, J.H. Chuah, M.S.A. Talip, N. Mokhtar, M.A. Shoaib, Structural crack detection using deep convolutional neural networks, Automation in Construction 133 (2022), 103989, https://doi.org/10.1016/j.autcon.2021.103989.

[7] A. Zhang, K.C.P. Wang, B. Li, E. Yang, X. Dai, Y. Peng, Y. Fei, Y. Liu, J.Q. Li, C. Chen, Automated pixel-level pavement crack detection on 3D asphalt surfaces using a deep-learning network, Comput. Aided Civ. Inf. Eng. 32 (10) (2017) 805–819, https://doi.org/10.1111/mice.12297.

[8] S. Dorafshan, R.J. Thomas, M. Maguire, Comparison of deep convolutional neural networks and edge detectors for image-based crack detection in concrete, Construct. Build Mater. 186 (2018) 1031–1045, https://doi.org/10.1016/j.conbuildmat.2018.08.011.

[9] C.V. Dung, L.D. Anh, Autonomous concrete crack detection using deep fully convolutional neural network, Automation in Construction 99 (2019) 52–58, https://doi.org/10.1016/j.autcon.2018.11.028.

[10] S. Bang, S. Park, H. Kim, H. Kim, Encoder-decoder network for pixel-level road crack detection in black-box images, Comput. Aided Civ. Inf. Eng. 34 (8) (2019) 713–727, https://doi.org/10.1111/mice.12440.

[11] C. Xiang, W. Wang, L. Deng, P. Shi, X. Kong, Crack detection algorithm for concrete structures based on super-resolution reconstruction and segmentation network,

Autom. Constr. 140 (2022), 104346, https://doi.org/10.1016/j.autcon.2022.104346.

[12] P. Guo, X. Meng, W. Meng, Y. Bao, Monitoring and automatic characterization of cracks in strain-hardening cementitious composite (SHCC) through intelligent interpretation of photos, Compos. Part B Eng. 242 (2022), 110096, https://doi.org/10.1016/j.compositesb.2022.110096.

[13] S. Poorghasem, Y. Bao, Review of robot-based automated measurement of vibration for civil engineering structures, Measurement 207 (2023), 112382, https://doi.org/10.1016/j.measurement.2022.112382.

[14] E. Ranyal, A. Sadhu, K. Jain, Road condition monitoring using smart sensing and artificial intelligence: a review, Sensors 22 (8) (2022) 3044, https://doi.org/10.3390/s22083044.

[15] C. Chen, S. Chandra, Y. Han, H. Seo, Deep learning-based thermal image analysis for pavement defect detection and classification considering complex pavement conditions, Remote Sens. (Basel) 14 (1) (2022) 106, https://doi.org/10.3390/rs14010106.

[16] J. Guan, X. Yang, L. Ding, X. Cheng, V.C.S. Lee, C. Jin, Automated pixel-level pavement distress detection based on stereo vision and deep learning, Automation in Construction 129 (2021), 103788, https://doi.org/10.1016/j.autcon.2021.103788.

[17] Y.F. Liu, S. Cho, B.F. Spencer, J.S. Fan, Concrete crack assessment using digital image processing and 3D scene reconstruction, J. Comput. Civ. Eng. 30 (1) (2016) 04014124, https://doi.org/10.1061/(ASCE)CP.1943-5487.0000446.

[18] Y.F. Liu, X. Nie, J.S. Fan, X.G. Liu, Image-based crack assessment of bridge piers using unmanned aerial vehicles and three-dimensional scene reconstruction, Comput. Aided Civ. Inf. Eng. 35 (2020) 511–529, https://doi.org/10.1111/mice.12501.

[19] Y. Narazaki, V. Hoskere, T.A. Hoang, Y. Fujino, A. Sakurai, B.F. Spencer, Vision-based automated bridge component recognition with high-level scene consistency, Comput. Aided Civ. Inf. Eng. 35 (2020) 465–482, https://doi.org/10.1111/mice.12505.

[20] N. Saovana, N. Yabuki, T. Fukuda, Development of an unwanted-feature removal system for structure from motion of repetitive infrastructure piers using deep learning, Adv. Eng. Inform. 46 (2020), 101169, https://doi.org/10.1016/j.aei.2020.101169.

[21] S.O. Sajedi, X. Liang, Uncertainty-assisted deep vision structural health monitoring, Comput. Aided Civ. Inf. Eng. 36 (2021) 126–142, https://doi.org/10.1111/mice.12580.

[22] Y. Xie, J. Tian, X.X. Zhu, Linking points with labels in 3D: a review of point cloud semantic segmentation, IEEE Geoscience and Remote Sensing Magazine 8 (4) (2020) 38–59, https://doi.org/10.1109/MGRS.2019.2937630.

[23] Y. Guo, H. Wang, Q. Hu, H. Liu, M. Bennamoun, Deep learning for 3D point clouds: a survey, IEEE Trans. Pattern Anal. Mach. Intell. 43 (12) (2021) 4338–4364, https://doi.org/10.1109/TPAMI.2020.3005434.

[24] D. Maturana, S. Scherer, Voxnet: A 3D convolutional neural network for real-time object recognition, in: 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, 2015, pp. 922–928, https://doi.org/10.1109/IROS.2015.7353481.

[25] Y. Zhou, O. Tuzel, Voxelnet: End-to-End Learning for Point Cloud Based 3D Object Detection, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, 2018, pp. 4490–4499, https://doi.org/10.1109/CVPR.2018.00472.

[26] C.R. Qi, H. Su, M. Nießner, A. Dai, M. Yan, L.J. Guibas, Volumetric and Multi-View CNNs for Object Classification on 3D Data, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, 2016, pp. 5648–5656, https://doi.org/10.1109/CVPR.2016.609.

[27] H. Su, S. Maji, E. Kalogerakis, E. Learned-Miller, Multi-View Convolutional Neural Networks for 3D Shape Recognition, Proceedings of the IEEE International Conference on Computer Vision, Santiago, 2015, pp. 945–953, https://doi.org/10.1109/ICCV.2015.114.

[28] A. Boulch, J. Guerrv, B.L. Saux, N. Audebert, SnapNet: 3D point cloud semantic labeling with 2D deep segmentation networks, Computer & Graphics 71 (2018) 189–198, https://doi.org/10.1016/j.cag.2017.11.010.

[29] A. Milioto, I. Vizzo, J. Behley, C. Stachniss, RangeNet++: Fast and accurate LiDAR semantic segmentation, in: 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Venetian Macao, 2019, pp. 4213–4220, https://doi.org/10.1109/IROS40897.2019.8967762.

[30] C.R. Qi, H. Su, K. Mo, L.J. Guibas, PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Hawaii, 2017, pp. 652–660, https://doi.org/10.1109/CVPR.2017.16.

[31] C.R. Qi, L. Yi, H. Su, L.J. Guibas, PointNet++: deep hierarchical feature learning on point sets in a metric space, advances in neural information processing systems, Long Beach (2017) 5099–5108. https://dl.acm.org/doi/abs/10.5555/3295222.3295263.

[32] Y. Li, R. Bu, M. Sun, W. Wu, X. Di, B. Chen, PointCNN: convolution on x-transformed points, advances in neural information processing systems, Montreal (2018) 820–830. https://dl.acm.org/doi/10.5555/3326943.3327020.

[33] Y. Wang, Y. Sun, Z. Liu, S.E. Sarma, M.M. Bronstein, J.M. Solomon, Dynamic graph CNN for learning on point clouds, ACM Trans. Graph. 38 (5) (2019) 1–12, https://doi.org/10.1145/3326362.

[34] L. Landrieu, M. Simonovsky, Large-Scale Point Cloud Semantic Segmentation with Superpoint Graphs, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, 2018, pp. 4558–4567, https://doi.org/10.1109/CVPR.2018.00479.

[35] Q. Hu, B. Yang, L. Xie, S. Rosa, Y. Guo, Z. Wang, N. Trigoni, A. Markham, RandLA-Net: Efficient Semantic Segmentation of Large-Scale Point Clouds, Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, 2020, pp. 11108–11117, https://doi.org/10.1109/CVPR42600.2020.01112.

[36] I. Armeni, O. Sener, A.R. Zamir, H. Jiang, I. Brilakis, M. Fischer, S. Savarese, 3D Semantic Parsing of Large-Scale Indoor Spaces, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, 2016, pp. 1534–1543, https://doi.org/10.1109/CVPR.2016.170.

[37] T. Hackel, N. Savinov, L. Ladicky, J.D. Wegner, K. Schindler, M. Pollefeys, Semantic3D.net: A new large-scale point cloud classification benchmark, arXiv preprint arXiv (2017), https://doi.org/10.48550/arXiv.1704.03847, 1704.03847.

[38] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, J. Gall, SemanticKITTI: A Dataset for Semantic Scene Understanding of Lidar Sequences, Proceedings of the IEEE International Conference on Computer Vision, Seoul, 2019, pp. 9297–9307, https://doi.org/10.1109/ICCV.2019.00939.

[39] B. Riveiro, M.J. DeJong, B. Conde, Automated processing of large point clouds for structural health monitoring of masonry arch bridges, Automation in Construction 72 (2016) 258–268, https://doi.org/10.1016/j.autcon.2016.02.009.

[40] Y. Yan, J.F. Hajjar, Automated extraction of structural elements in steel girder bridges from laser point clouds, Automation in Construction 125 (2021), 103582, https://doi.org/10.1016/j.autcon.2021.103582.

[41] R. Lu, I. Brilakis, C.R. Middleton, Detection of structural components in point clouds of existing RC bridges, Comput. Aided Civ. Inf. Eng. 34 (2019) 191–212, https://doi.org/10.1111/mice.12407.

[42] L. Truong-Hong, R. Lindenbergh, Automatically extracting surfaces of reinforced concrete bridges from terrestrial laser scanning point clouds, Automation in Construction 135 (2022), 104127, https://doi.org/10.1016/j.autcon.2021.104127.

[43] H. Kim, J. Yoon, S.H. Sim, Automated bridge component recognition from point clouds using deep learning, Struct. Control Health Monit. 27 (2020), e2591, https://doi.org/10.1002/stc.2591.

[44] H. Kim, C. Kim, Deep-learning-based classification of point clouds for bridge inspection, Remote Sens. (Basel) 12 (22) (2020) 3757, https://doi.org/10.3390/rs12223757.

[45] J.S. Lee, J. Park, Y.M. Ryu, Semantic segmentation of bridge components based on hierarchical point cloud model, Automation in Construction 130 (2021), 103847, https://doi.org/10.1016/j.autcon.2021.103847.

[46] X. Yang, E.R. Castillo, Y. Zou, L. Wotherspoon, Y. Tan, Automated semantic segmentation of bridge components from large-scale point clouds using a weighted superpoint graph, Automation in Construction 142 (2022), 104519, https://doi.org/10.1016/j.autcon.2022.104519.

[47] Y. Jing, B. Sheil, S. Acikgoz, Segmentation of large-scale masonry arch bridge point clouds with a synthetic simulator and the BridgeNet neural network, Automation in Construction 142 (2022), 104459, https://doi.org/10.1016/j.autcon.2022.104459.

[48] H. Edelsbrunner, D. Kirkpatrick, R. Seidel, On the shape of a set of points in the plane, IEEE Trans. Inf. Theory 29 (1983) 551–559, https://doi.org/10.1109/TIT.1983.1056714.

[49] MATLAB R2022a, The MathWorks Inc., Natick, MA. https://ww2.mathworks.cn/help/matlab/, 2022 (accessed May 14, 2023).

[50] B.L. Li, Y. Qi, J.S. Fan, Y.F. Liu, C. Liu, A grid-based classification and box-based detection fusion model for asphalt pavement crack, Comput. Aided Civ. Inf. Eng. (2022), https://doi.org/10.1111/mice.12962.

[51] N. Otsu, A threshold selection method from gray-level histograms, IEEE Trans. Syst. Man Cybern. 9 (1979) 62–66, https://doi.org/10.1109/TSMC.1979.4310076.

[52] L.L.C. Agisoft, Agisoft Metashape user manual: professional edition, Version 2.0. https://www.agisoft.com/pdf/metashape-pro_2_0_en.pdf, 2023.

[53] C.F. Özgenel, Concrete crack images for classification, Mendeley Data V2 (2019), https://doi.org/10.17632/5y9wdsg2zt.2.

[54] J. Liu, X. Yang, S. Lau, X. Wang, S. Luo, V.C.S. Lee, L. Ding, Automated pavement crack detection and segmentation based on two-step convolutional neural network, Comput. Aided Civ. Inf. Eng. 35 (11) (2020) 1291–1305, https://doi.org/10.1111/mice.12622.