Front. Struct. Civ. Eng. 2022, 16(11): 1378–1396
https://doi.org/10.1007/s11709-022-0855-8

RESEARCH ARTICLE

Detecting large-scale underwater cracks based on remote operated vehicle and graph convolutional neural network

Wenxuan CAO[a], Junjie LI[a,b*]

[a] *Faculty of Infrastructure Engineering, Dalian University of Technology, Dalian 116024, China*
[b] *College of Water Conservancy and Hydropower Engineering, Hohai University, Nanjing 210098, China*
[*] *Corresponding author. E-mail: lijunjie@dlut.edu.cn*

ABSTRACT It is of great significance to quickly detect underwater cracks as they can seriously threaten the safety of underwater structures. Research to date has mainly focused on the detection of above-water-level cracks and hasn't considered the large scale cracks. In this paper, a large-scale underwater crack examination method is proposed based on image stitching and segmentation. In addition, a purpose of this paper is to design a new convolution method to segment underwater images. An improved As-Projective-As-Possible (APAP) algorithm was designed to extract and stitch keyframes from videos. The graph convolutional neural network (GCN) was used to segment the stitched image. The GCN's *m-IOU* is 24.02% higher than Fully convolutional networks (FCN), proving that GCN has great potential of application in image segmentation and underwater image processing. The result shows that the improved APAP algorithm and GCN can adapt to complex underwater environments and perform well in different study areas.

KEYWORDS underwater cracks, remote operated vehicle, image stitching, image segmentation, graph convolutional neural network

1 Introduction

In recent decades, with the development of Ocean Engineering, Hydraulic Engineering, Bridge and Tunnel Engineering, etc., a series of new underwater structure models has been formed [1,2]. However, due to the effects of the external environment (such as wind and wave, corrosion, hydraulic flushing, temperature stress, etc.) or human factors (such as design errors or improper selection of materials), underwater structures may have various degrees of damage, which may lead to cracks during long-term service [3]. At present, it is still difficult to simulate the crack propagation and formation mechanism [4]. Many scholars have proposed algorithms to simulate crack propagation, such as NHPD [5] and GCEM [6]. Cracks seriously damage the reliability and longevity of the structure; they can exist not only at the structure's surface but can also extend into the interior [7,8]. Zhang and Zhuang [9,10] proposed a self-propagating strong discontinuity embedded approach with the statically optimal symmetric (SDA-SOS) formulation to study the propagation law of cracks. Their computational examples showed that cracks seriously degrade structure durability. Rezaiee-Pajand and Tavakoli [11] thought cracks are the external manifestations of the accumulation of fatigue in the structure. Therefore, it is essential to monitor structures in time to prevent crack expansion.

Due to the complexity of the underwater environment, only a few technologies have been applied to underwater crack detection, such as electrical exploration, elastic wave testing, radar, etc. [12,13]. For example, Li et al. [1] proposed a high sensitivity rotating alternating current field to measure underwater cracks; Luo et al. [14] detected concrete cracks using a tapered polymer fiber sensor; Shi et al. [15] used sonar images to detect and classify below-water-level cracks in dams. However, these methods have shortcomings in common: shallow measurement depth, inability to fully examine deep-water

Article history: Received Feb 23, 2022; Accepted Apr 3, 2022

structures, large positioning errors, low efficiency, and weak adaptability. Moreover, these methods are costly, and neither convenient nor reliable [15,16]. Visual estimation is more efficient and inexpensive for obtaining crack information such as location and shape [17]. With the rapid development of machine vision technology, some crack detection methods based on image processing have been proposed [18]. In the last century, Belytschko et al. [19] began developing and designing the first camera-based road damage detection vehicle GERPHO. Ukai [20,21] proposed an image acquisition system using a multi-eye line array camera to monitor tunnel cracks in 2000 and 2007. Since the start of the 21st century, research on crack detection has been further deepened. Lu et al. [22] proposed a road crack detection algorithm based on adaptive threshold segmentation; Talab et al. [23] used the Sobel filter and Otsu algorithm to detect concrete cracks, which had an accuracy of 85% on their data sets, but the algorithm was sensitive to changes in shooting angle and light; Xiao and Li [24] combined the adaptive Canny operator with seepage theory and proposed a crack detection algorithm. However, these methods were limited to traditional image processing technology. The algorithms were susceptible to environmental factors, had large errors, and had low generalization ability [25].

Deep learning (DL) has an advantage of being little affected by noise, being able to migrate to different environments, and high accuracy. The rapid development of DL provides different ideas for people to solve problems. For example, Nguyen-Thanh et al. [26] solved potential energy problems in parametric deep energy methods based on physical information neural networks (PINN). Nguyen-Thanh et al. [27] also presented a deep energy method for finite deformation hyperelasticitiy using deep neural networks (DNNs), which could avoid entirely a discretization like FEM. Guo et al. [28] proposed a deep collocation method (DCM) for thin plate bending problems. Zhuang et al. [29] present a deep autoencoder based energy method (DAEM) for bending, vibration and buckling analysis of Kirchhoff plates; Guo et al. [30] present a stochastic deep collocation method (DCM) based on a neural architecture search (NAS) and transfer learning for heterogeneous porous media. DL has been widely used in many fields such as solving partial differential equations in Computational Mechanics [31,32], dam subsidence prediction [33,34], urban traffic monitoring, etc. Many scholars have tried to use DL to detect cracks. Cha et al. [35] used a five-layer convolutional neural network (CNN) to detect concrete cracks and processed gridded images based on sliding window technology. Kim and Cho [36] used CNN with Alexnet as the backbone for accurate classification of five observable entities such as cracks, plants and concrete. In 2015, Long et al. [37] proposed fully convolutional networks (FCN) by replacing the full connection layer of CNN with convolution layer. FCN realizes semantic segmentation in the real sense. Image semantics segmentation based on DL has also been widely used in crack detection. Dung and Anh [38] realized the automatic detection of concrete cracks by deep FCN. Their results show that cracks are reasonably detected, and crack density is also accurately evaluated. Bang et al. [17] introduced an attention model into image semantics segmentation and obtained good results in detecting road cracks. Zhang et al. [39] proposed a neural network with multiple convolution layers and combine context information to detect cracks in structures. The method adopted an end-to-end training approach, and could realize pixel level processing of images of any size. Zhang et al. [40] proposed a faster, simpler single-stage detector based on YoLoV3 for detecting multiple concrete bridge damages. Liu et al. [41] combined target detection with semantic segmentation, and designed a two-step network. Zhang and Yuen [42] designed a novel crack detection system based on a broad learning system, and their system can be accelerated without GPU during training, which reduces the requirement of the computer configuration.

However, cracks are usually continuous, long-distance, and large-scale. It is difficult to get a complete crack in the field of view of a single image, either above or below water level [43,44]. Assessment of complete cracks is significant for analyzing damage degree and true working form of underwater structures. Therefore, it is necessary to determine the complete shape of a crack by stitching of multiple images. There are mainly two steps of image stitching technology: registration and fusion. In the 1980s, Burt and Adelson [45] proposed an image fusion method based on the Laplace Pyramid. The image pyramid and scale transformation lay an important foundation for subsequent research. In 2004, Lowe [46] proposed an image registration method, SIFT, which performs image registration based on the eigenvectors of the image's feature point. After Lowe, Bay et al. [47] proposed SURF, which uses integrated images to achieve faster image registration. Rublee et al. [48] proposed the ORB algorithm, which has a strong advantage in registration speed. In recent years, image stitching technology has begun to be applied to structural health monitoring (SHM). Zhu et al. [49] stitched different positions' concrete column images based on traditional feature-based image stitching technique. Won et al. [50] automatically generated panoramic bridge images using deep matching. Based on the SIFT algorithm, Wang et al. [43] obtained the complete shapes of cracks in a dam. Wu et al. [44] stitched large-scale panoramic cracks using Oriented FAST and Rotated BRIEF feature matching algorithm.

However, due to the complex underwater optical

environment, underwater images have low contrast and much noise. Moreover, due to the multi-interface refraction of light when using a fisheye camera, the collected images are often distorted [51,52], which means that the underwater image is essentially in Non-Euclidean Space compared to the above water image. The current image stitching and semantics segmentation algorithms are based on image data's translation invariant, scale invariant and rotation invariant. In other words, these methods are aimed at data in Euclidean space, and they may not accurately detect cracks from underwater images.

The APAP algorithm proposed by Zaragoza et al. [53] is an image fusion algorithm. The APAP algorithm first grids the images, then spatially warps each grid cell using its corresponding local homograph matrix, and finally superimposes them on the canvas to complete the image fusion. Because the APAP algorithm uses a local-global fusion strategy, it can eliminate errors caused by image distortion when image stitching. At present, APAP is mainly used to stitch large-scale remote sensing images [54], and it has not been studied for SHM.

Graph convolution neural network (GCN) is a new DL model proposed for data in non-Euclidean space. It was first proposed in 2017 and achieved the detection of high-dimensional data features by constructing nodes and edges. Landrieu and Simonovsky [55] proposed a large-scale points-cloud segmentation algorithm based on superpoint graphs. In 2018, the OCNet was proposed by Yuan and Wang [56], which could apply a non-local operator to segmentation. To date, GCN has had little broad application and research, including in image segmentation and underwater image processing, but it has great potential.

Based on the above discussion, our paper's primary work and purpose are to study the large-scale underwater cracks detection method using image stitching and semantic segmentation. Remote Operated Vehicles (ROVs) were used to collect images of cracks in different underwater structures. A dataset of underwater cracks in three areas was created for training, validation, and other GCN processes. Since the data collected by the ROV is transmitted in the form of videos, an improved APAP algorithm was designed, which can automatically extract keyframes from the video and then stitch the images. Then, an image segmentation algorithm based on GCN was designed, which takes the pre-trained Resnet101 as the backbone. In addition, our research also compared the effect of GCN and traditional FCN in segmenting underwater crack images.

2 Dataset and methods

In the work reported in this paper, an underwater cracks detection method was designed, composed of the proposed image stitching algorithm and GCN algorithm. Our study can be divided into three parts: data acquisition, image stitching and image segmentation. The detailed process of our study is shown in Fig. 1. Firstly, the ROV was used to get underwater videos. Next, image stitching was done directly on videos. The stitched image of different study areas was obtained through the improved APAP algorithm. Then, a dataset of underwater cracks was created by framing videos. These images were stochastically selected, and LabelMe was used to mark crack areas on the image for training, validation and test of GCN and FCN. The datasets were randomly divided into training set, validation set, and test set according to

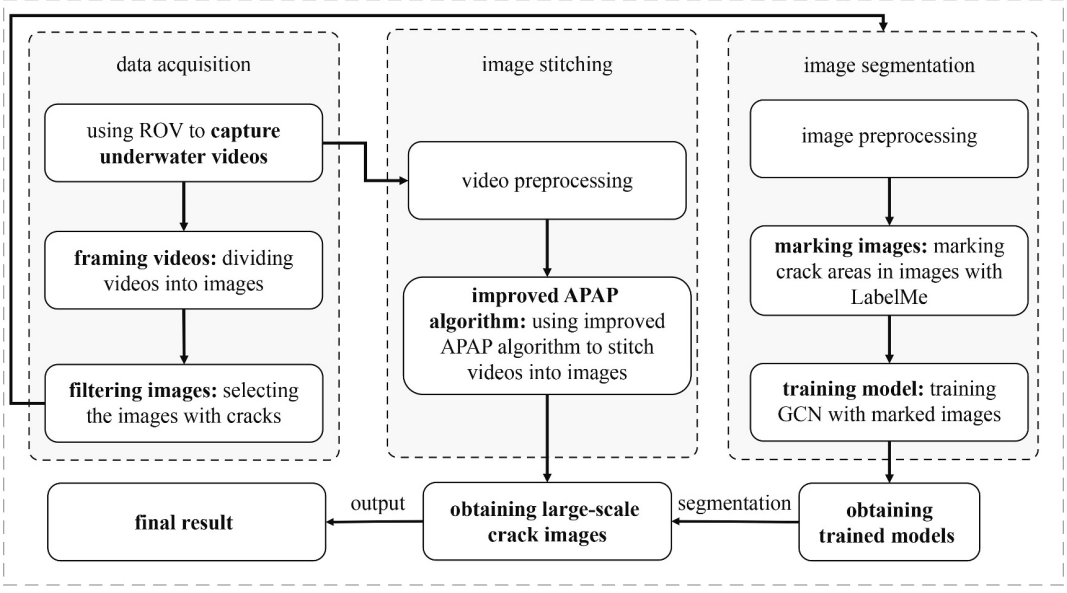

Fig. 1 Research process of this paper.

the ratio of 8:1:1. Finally, the stitched images were segmented and the large-scale underwater crack patterns were reconstructed by invoking the final training results.

The above processes was performed on Matlab2020 and Python 3.9. The computer specifications for code writing and program running were as follows: OpenCV 2.4.2 was used as the visual library for underwater crack detection. Image stitching was mainly carried out on HP Star14 X360 platform. PyTorch 3.8 was used as framework for DL in our study, and LabelMe was used to mark images. GCN and FCN were trained on NVIDIA Tesla V100 32GB GPU and NVIDIA GeForce RTX 2080Ti GPU.

2.1 Data acquisition

Data were collected from different underwater structures. These structures were a dam in Hubei Province, China, Tunnel A and Tunnel B, both in Hebei. Seabotix vLBV300 was mainly used to detect underwater cracks of the dam, and the camera used was a 650 TV line high-definition color camera. Tunnel A and Tunnel B were mainly detected by Dolphin One, and the camera used was Shark Marine. The detailed parameters of ROVs and cameras are shown in Table 1.

By framing the video, we obtained 4097 + 12701 + 8534 underwater images from the three sites, a total of 25332. After selection, a dataset containing 957 underwater crack images was established. After processing, the final dataset was obtained.

2.2 Image stitching

This paper proposes an improved APAP algorithm to stitch underwater videos directly. This algorithm can be divided into three steps in the process: keyframes detection, feature points matching, and image stitching, as shown in Fig. 2.

Table 1 The detailed operating parameters of ROVs

ROV	parameter	value
Seabotix vLBV300	max diving depth (m)	300
	max speed (km/h)	5.5
	size ($L \times W \times H$) (mm)	625 × 390 × 390
	weight (kg)	18.1
	the visual angle of camera (°)	65
	image resolution	720 × 480
	sensitivity	0.1lux@f2.0
	work area	Dam
Dolphin One	max diving depth (m)	100
	max speed (km/h)	3.0
	size ($L \times W \times H$) (mm)	457 × 338 × 254
	weight (kg)	11
	image resolution	1920 × 1080
	work area	Tunnel A and Tunnel B

The keyframes extraction was based on the frames' similarity to ensure that the similarity of each keyframe is not too low. In this study, the phash algorithm was used to calculate the similarity between frames. By phash, the Hamilton distance between two images could be obtained [57]. The greater the Hamilton distance between frames, the smaller their similarity.

Firstly, every frame was resized to the same pixel dimensions, 32 × 32, and converted to grayscale images (the purpose of which was to reduce the difference between the size and proportion of these images, only retaining the images' basic information). In order to decrease the calculated quantities and run the program conveniently on the CPU, discrete cosine transform (DCT) was used to transform gray images. The expression of DCT is:

$$F(u) = c(u)\sum_{i=0}^{N-1} f(i)\cos\left[\frac{(i+0.5)\pi}{N}u\right], \tag{1}$$

$$c(u) = \begin{cases} \sqrt{\dfrac{1}{N}}, & u = 0, \\ \sqrt{\dfrac{2}{N}}, & u \neq 0, \end{cases} \tag{2}$$

where $f(i)$ is the original image data, $F(u)$ is the coefficient after DCT transformation, N is the points' number of original image data, and $c(u)$ is the compensation factor that makes the DCT transformation matrix orthogonal.

Next, based on the result of the DCT calculation, a hash value composed of 64 bits was made. Then, the hash value of two images was compared to calculate the Hamilton distance between them. Hamilton distance between frames was used as the criterion for extracting keyframes in our study. The process of extracting keyframes is shown in Fig. 3, with the frame $n-1$ set as image A, frame n set as image B, allowing calculation of the Hamilton distance between image A and image B. Image B (frame n) was considered as the keyframe when the distance between B and A was larger than an artificially set threshold. Then image B became image A and the process was repeated. In this way, all keyframes were extracted from videos.

For feature points matching, the method used in this work was SIFT. There are three main processes for SIFT algorithm to achieve feature matching [46]: extracting some prominent feature points in two images; describing these feature points (for example, the location, the direction, and the number, etc.); matching them, as shown in Fig. 4.

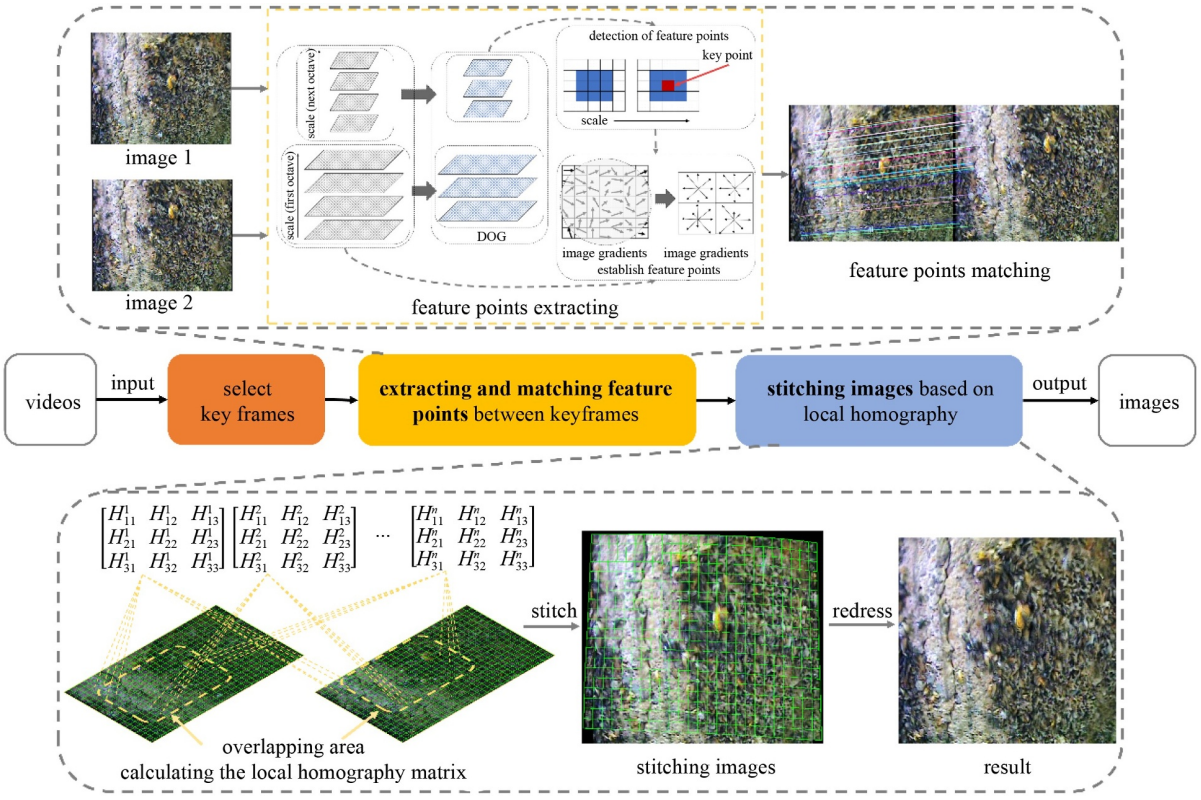

Fig. 2 The overview of improved APAP algorithms.

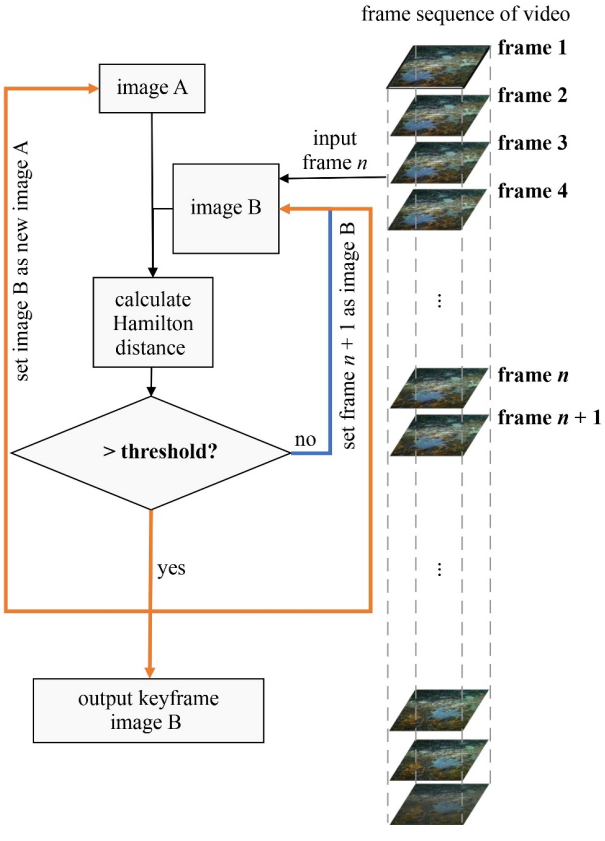

Fig. 3 Extracting keyframes from video.

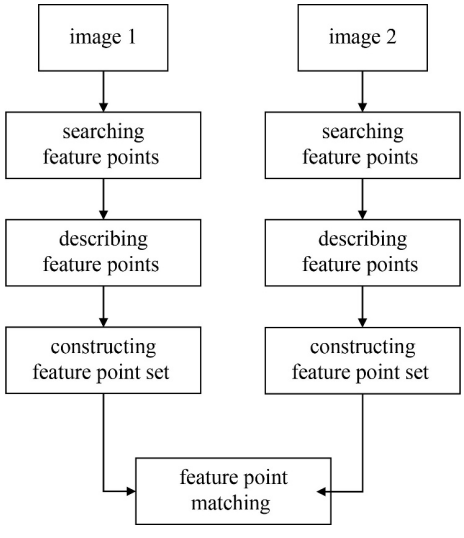

Fig. 4 Feature points extracting and matching based on SIFT.

The SIFT algorithm mainly searches for feature points in scale space. First, the keyframe $K(x,y)$ is convoluted using a Gaussian kernel function to obtain the projection $L(x,y,\sigma)$ in different scales; the formula is:

$$L(x,y,\sigma) = G(x,y,\sigma) * K(x,y), \tag{3}$$

$$G(x,y,\sigma) = \frac{1}{2\pi\sigma^2}\exp\left(-\frac{(x-x_i)^2+(y-y_i)^2}{2\sigma^2}\right), \tag{4}$$

where $G(x,y,\sigma)$ is a scale-varying Gaussian kernel function; (x,y) is a spatial coordinate of pixel; σ is the scale coordinate that determines the images' smoothness, the overview characteristics of the image corresponding to a large scale, and the detail characteristics of the image corresponding to a small scale. σ values correspond to coarse scales (low resolution) and fine scales (high resolution). Combining the original image with the projection to get an image pyramid and a difference of Gaussian scale (DOG), the DOG function is:

$$D(x,y,\sigma)=L(x,y,k\sigma)-L(x,y,\sigma), \tag{5}$$

Feature points will be found on the DOG, and described on the image pyramid. SIFT considers that the feature points are essentially the extreme points of the DOG function, which means feature points are composed of local extreme points in the DOG. In SIFT, extreme points of the DOG function are those points that are larger or smaller than the surrounding pixel point in scale domain.

To detect extreme points, local characteristics of the image were used to assign a baseline direction to each critical point. The gradient and direction distribution characteristics of other neighborhood pixels were counted with the feature point as the origin and 3σ as the radius to determine the gradient and direction of the feature point. The formulas for calculating the pixel gradient and direction are:

$$m(x,y)=\sqrt{(L(x+1,y)-L(x-1,y))^2+(L(x,y+1)-L(x,y-1))^2}, \tag{6}$$

$$\theta(x,y)=\cot\frac{L(x,y+1)-L(x,y-1)}{L(x+1,y)-L(x-1,y)}. \tag{7}$$

Finally, a descriptor was created for each feature point, a set of vectors was used to describe the feature point, and a subset of descriptions containing all feature points was created. Based on the subset of feature point descriptions, the feature point in image A which was nearest to the feature point in image B was searched and matched.

Since the study focused on the crack area, too many matching points will affect the stitching effect of the crack area. Therefore, the random sample consensus (RANSAC) was used to remove some matching points that may have affected the final stitching result. The core idea of RANSAC is that: for a fitting problem, there are two kinds of data points, one affects the fitting effect (outer point) and the other is conducive to the fitting of function (inner point). RANSAC aims to find out and eliminate the outer points through continuous random sampling.

Direct linear transform (DLT) was used to estimate the perspective transformation matrix for the remaining feature points, and a global homography matrix was obtained. The calculation method of homography matrix is:

$$\begin{pmatrix}x_b\\y_b\\w_b\end{pmatrix}=\boldsymbol{H}\begin{pmatrix}x_a\\y_a\\w_a\end{pmatrix},\ \boldsymbol{H}=\begin{bmatrix}h_1&h_2&h_3\\h_4&h_5&h_6\\h_7&h_8&h_9\end{bmatrix}, \tag{8}$$

where $\boldsymbol{H}$ is the homography matrix; $\begin{pmatrix}x_a\\y_a\\w_a\end{pmatrix}$ and $\begin{pmatrix}x_b\\y_b\\w_b\end{pmatrix}$ are the camera model matrix of the two pictures to be stitched. Then by dividing the images to be stitched into grids and taking the center points of each grid, the distance and weights between the interior points on the source map and center points could be determined. Putting the weightings into the A matrix of the DLT algorithm and building a new $W*A$ matrix, the local homography matrix of the current grid could be naturally obtained. Then, the stitched image was obtained by traversing each grid and mapping it to the panoramic canvas using the local homography matrix.

In practice, the video only needed to be imported directly into the program. Our algorithm first divided the video into frames, and then compared the distances between frames. Setting the threshold, the program output all key frames that met the requirements as shown in Fig. 3. In the end, these key frames were stitched based on SIFT.

2.3 Image segmentation

2.3.1 Fully convolutional networks

FCN, first proposed by Long et al. [37] in 2015, is the first image semantic segmentation system based on DL. FCN can accept any input size and produce appropriate output through efficient reasoning and learning. The network structure of FCN is divided into two parts: full convolution and deconvolution. The full convolution part replaces the last full connection layer of the CNN network with convolution to extract features and form a heatmap. The purpose of deconvolution part is to upsample the heatmap so that the output results are consistent with the original size. In this work, the backbone of FCN was replaced with Resnet101 and the attention mechanism was inserted in Resnet101 to ensure reliability compared with GCN.

2.3.2 Graph convolutional network

This study used a new semantic segmentation algorithm, GCN. Moving on from traditional semantics segmentation algorithms such as FCN, U-net, and Deeplab, a new convolution method, graph convolutional, was used in GCN, enabling the model to learn deeper information

about the data [58]. In addition, the attention mechanism was inserted into the backbone to make the GCN in this work able to access more crack area information during training. The GCN of this paper [58] can be divided into two parts: backbone part and graph convolution part, as shown in Fig. 5.

The backbone used in this paper is Resnet101, which mainly consists of 33 convolution blocks, two pooling layers, and one full connection layer. Each convolution block of Resnet101 contains three convolution cores connected by residuals to ensure that no network degradation or loss of information occurs during training. To connect with the graph convolution part, the full connection layer was removed so that the output of Resnet101 is a dense information feature map with 2048 channels. Also, considering the small proportion, by size, of crack areas in images, the attention mechanism was inserted into each convolution block, as shown in Fig. 6. By inserting the attention mechanism, the feature map information could mainly focus on the crack area. This helped the graph convolution part to learn the key information better.

The output of the backbone is a feature map with multiple channels. This study considers that not only the pixel of the feature map has correlation, but also those different channels have correlation. Therefore, the graph convolution part had two branches, which convoluted the output of backbone from channel and feature. In the channel branch, 2048 channels of the feature map were convoluted to determine which channels were important and which were unimportant, by two 1 × 1 graph convolution kernels. After further aggregation and compression, the weightings of these channels were obtained. In the feature branch, the pixel of the feature map was convoluted by three 1 × 1 graph convolution kernels, to obtain the coordinates and correlation information of pixels in non-Euclidean space. Finally, the result of two branches was aggregated with the output of backbone to get the segment result of underwater crack.

Due to the underwater environment, the image data collected was often distorted and did not have translation invariance. Therefore, this paper made a hypothesis that the underwater image data is better described with non-Euclidean data. So, a new convolution method: graph convolution was adopted in the graph convolution part. Graph convolution is a special kind of convolution that can handle data in non-Euclidean space and extract deeper data features.

$G = (V, E, W)$ was used to represent the data [59]. V is node-set, E is edge-set, and W is the weighted adjacency matrix. The node corresponds to the pixel of the images, which records the color, brightness and other information of the object. In contrast, the edge corresponds to the relationship between pixels and records the shape and texture of the object. For normal images, the arrangement of nodes and side rules can be achieved by smoothing the data on the data and learning the deep information of the data through convolution kernel. However, as shown in Fig. 7, for non-Euclidean data, if the graph convolution is processed in the same way, a lot of information is missed.

So, this study builds a graph convolution method based on Fourier transformation. Graph convolution uses

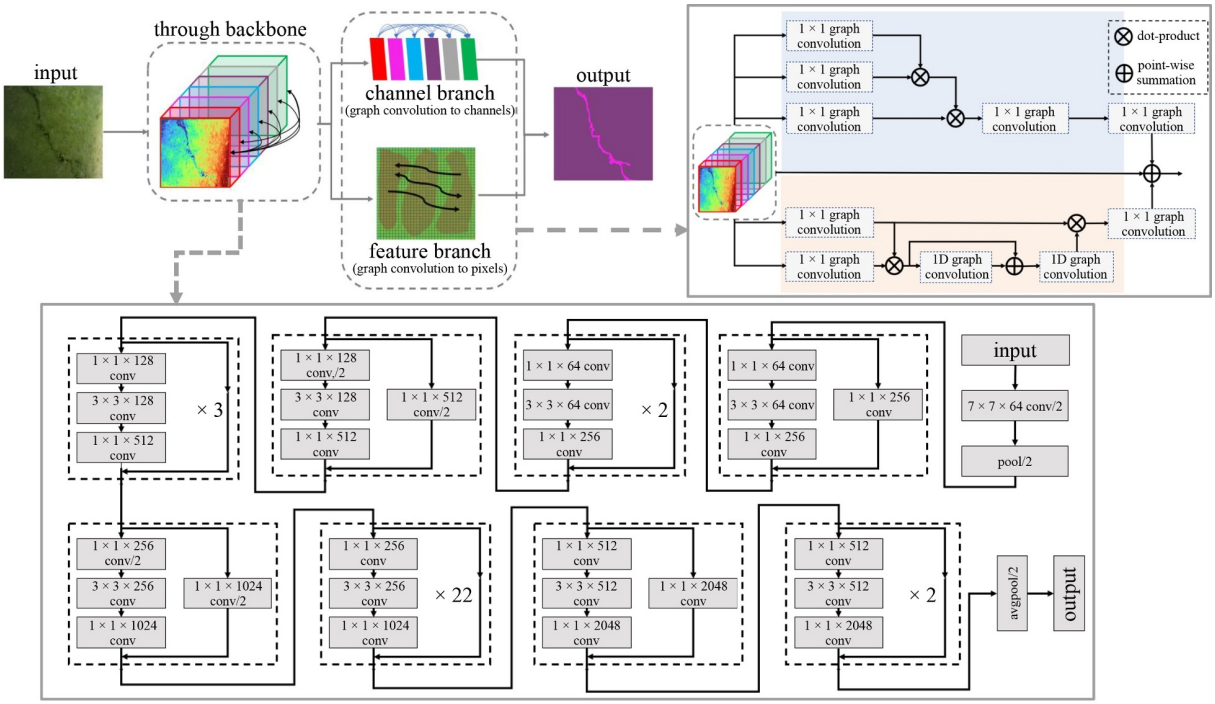

Fig. 5 The overall structure of GCN model.

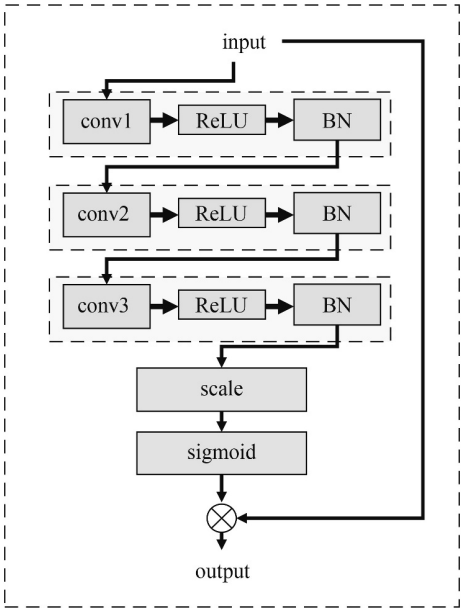

Fig. 6 The attention mechanism inserted in the convolution block of backbone.

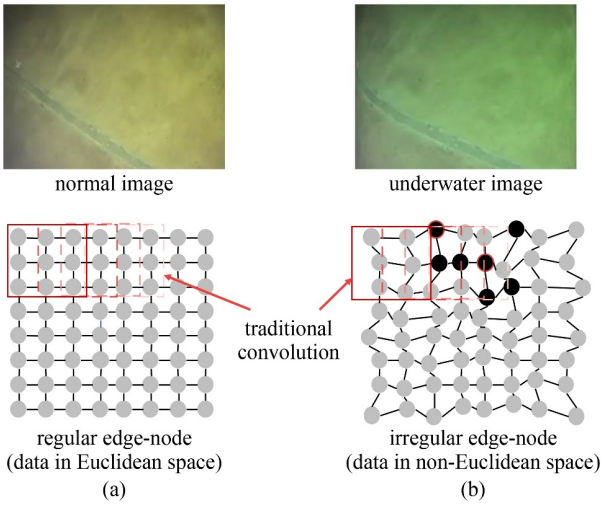

Fig. 7 Traditional convolution kernels convolute normal and underwater images. (a) Normal image and its convolution process; (b) underwater image and its convolution process. Note: The gray node represents the pixel of the image; the black node represents the missed node during convolution.

Fourier transformation and Laplace matrix to transform non-Euclidean data into frequency domain, obtain the graph's spectrum, and convolve the spectrum of the graph, as shown in Fig. 8. Furthermore, the graph Laplacian could be diagonalized as L:

$$L = U\Lambda U^{\mathrm{T}}, \tag{9}$$

where $U = [u_1, u_2, \ldots, u_n]$ is the complete set of orthonormal eigenvectors; $\Lambda = \mathrm{diag}([\lambda_1, \lambda_2, \ldots, \lambda_n])$ is the non-negative eigenvalues of L. Then, the data could be transformed into the spectral domain by Fourier transformation:

$$\hat{x} = U^{\mathrm{T}}x, \tag{10}$$

where x is our data, and $\hat{x}$ is the projection of our data in the spectral domain. Correspondingly, the process of converting data from the spectral domain to graph could be represented as:

$$x = U\hat{x}. \tag{11}$$

According to convolution theorem, graph convolution could be written as:

$$x *_{\mathrm{G}} y = U\left((U^{\mathrm{T}}x) \odot (U^{\mathrm{T}}y)\right), \tag{12}$$

where $(U^{\mathrm{T}}y)$ is the convolution filter in spectral domain.

We implemented GCN using Pytorch. The hyperparameter information of our program is shown in Table 2. In our program, we adopted a polynomial learning rate decay schedule where the initial learning rate was multiplied by $\left(1 - \frac{iter}{total\_iter}\right)^{0.9}$. The loss function used in this paper was Cross Entropy Loss Function, the activation function was ReLU. We also used synchronized

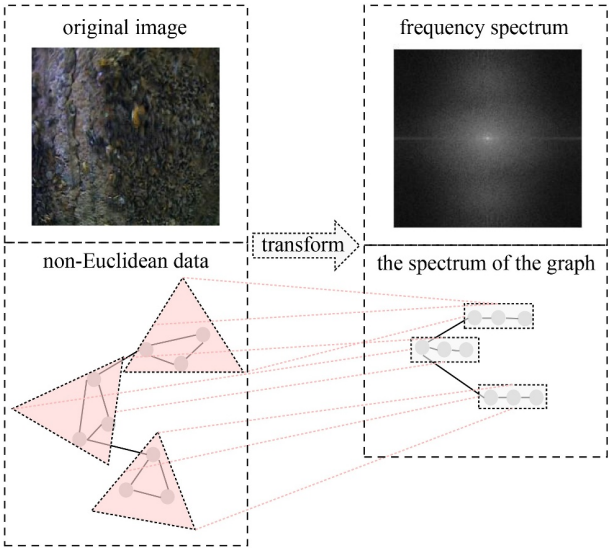

Fig. 8 The graph data is transformed into frequency domain based on Fourier transformation.

Table 2 The hyperparameters of GCN

hyperparameter	value
the initial learning rate	0.001
momentum	0.9
weight decay coefficients	0.0001
epoch	3500
batchsize	8
classifier	Softmax

batch normalization for better estimation of the batch statistics.

2.3.3 Evaluation index

This study mainly evaluated GCN and FCN from three indices: *m-IOU*, *F*1, and accuracy. *m-IOU* was mainly used to evaluate the segmentation effect of the trained model in dealing with other data and judge the generalization and stability of the model. The closer to 1, the better the model effect. The calculation formula is:

$$m\text{-}IOU = \frac{TP}{TP+FP+FN}, \tag{13}$$

where *TP* represents the number of correct pixels extracted when calling the trained model to extract the crack area; *FP* represents the number of error pixels extracted; *FN* represents the number of pixels in the crack area that have been misjudged.

*F*1 combines the indicators of Precision and Recall, representing the model's balance value with the constraints of recall and prediction. It is often used to compare the actual application of models. *F*1 could reflect the overfitting phenomenon of the model. The calculation formula is:

$$F1 = \frac{2\times(Precision\times Recall)}{(Precision+Recall)} = \frac{2\times TP}{2TP+FP+FN}. \tag{14}$$

Accuracy indicates how many of all pixels are accurately identified as crack areas. The calculation formula is:

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN}, \tag{15}$$

where *TN* represents the number of pixels in which the non-crack area is divided into non crack areas.

3 Results and analysis

3.1 Image stitching results

Based on the improved APAP algorithm, the image of collected videos were stitched.

522 keyframes were extracted from three videos. There were 66 keyframes with underwater cracks to which image stitching was done.

Through the SIFT algorithm, feature points of each image were extracted and matched roughly. The SIFT algorithm could extract many feature points, but most of these feature points were useless. Therefore, only a few fine matching point pairs were retained after RANSAC. As shown in Table 3, Tunnel A had the highest number of well-matching points and the longest stitching time. The Dam (a) area had the least number of well-matching points and the shortest time. But this does not mean that the stitching time was related to the number of well-matching points. Tunnel B had only 625 well-matching points, but its stitching time was longer than that of Dam (b). We think that this was probably because the total number of pixels in Tunnel B was more than that for Dam (b).

Based on the exact matching result, the local homograph matrix was used for image fusion. By iteratively fusing these images, the final stitching result was obtained after adjustment, as shown in Fig. 9.

3.2 Image semantic segmentation results

Unlike our improved APAP algorithm, the training of GCN and FCN was carried out directly on frames. By framing these videos, a total of 4097 + 12701 + 8534 = 25332 underwater images were obtained. After selection, 957 images containing underwater cracks were collected. After cropping, de-ghosting, and secondary selection, a

Table 3 Image matching and stitching in different areas

area	match points pairs		time (s)
	rough matched points pairs	good match points pairs	
Dam (a)	3911	316	37.75
Dam (b)	9093	1098	64.12
Tunnel A	4346	1893	187.94
Tunnel B	2984	625	98.12

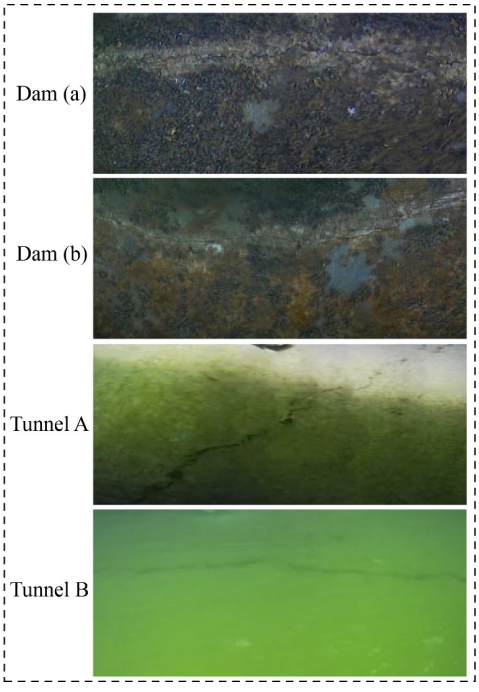

Fig. 9 The stitching result of large-scale underwater cracks in different areas.

dataset for GCN and FCN training, validation, and test were obtained.

After training, the trained GCN and FCN models were used to segment the stitched underwater image, and results are shown in Fig. 10. The loss curves of GCN are shown in the Fig. 11. The result indicates that GCN could accurately segment the underwater crack in images and was not affected by noises such as water, lighting conditions, aquatic plants, shadows, floating dust, etc. GCN detected most of the underwater cracks and segmented the actual crack pixels as much as possible. Compared with the segmentation result of FCN, the segmentation result of GCN was finer, and GCN had better effect on slim cracks. The segmentation result of FCN was coarser, less sensitive to slim cracks, and susceptible to the underwater environment.

Our study also calculated the proportion of the crack area in whole images, and the proportion of the crack area extracted by GCN and FCN in images, as shown in Table 4. By comparison, it can be seen that the result of GCN was closest to the actual value, and FCN was larger.

3.3 Image semantic segmentation evaluation

In order to more accurately evaluate the effect of GCN and FCN, three indices on the test set were compared, as shown in Table 5.

It can be seen that compared with FCN, GCN offered significant improvement. The *m-IOU* value of GCN was 25.02% higher than that of FCN, and the *F*1 value was 15.71% higher than that of FCN. GCN showed better generalization ability and practical application effect than FCN. However, their accuracies were not much different—both were more than 90%. This was probably due to the

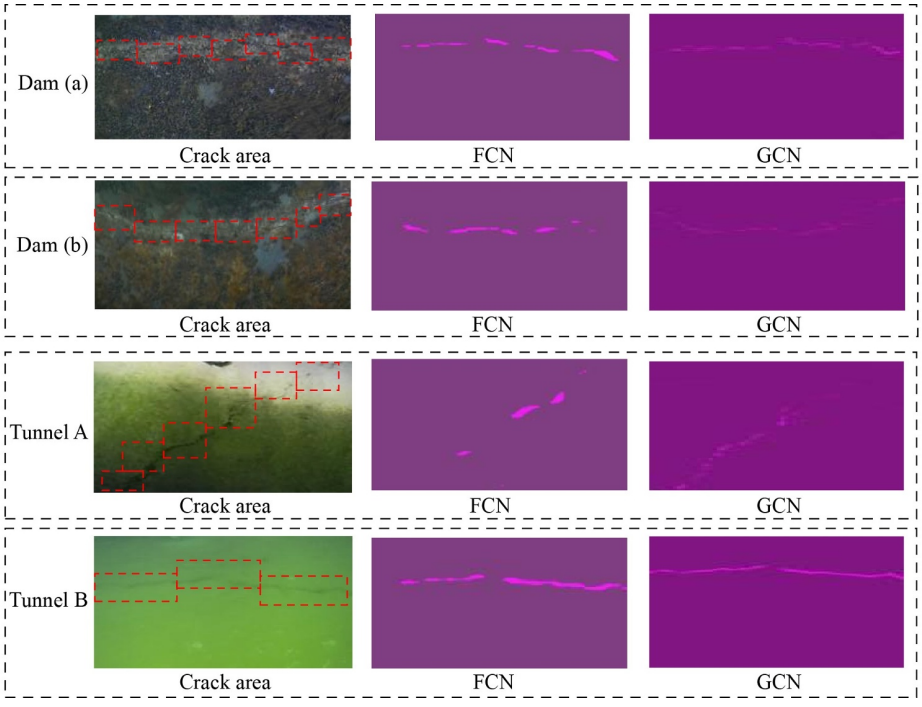

Fig. 10 The segmentation results of GCN and FCN in different areas.

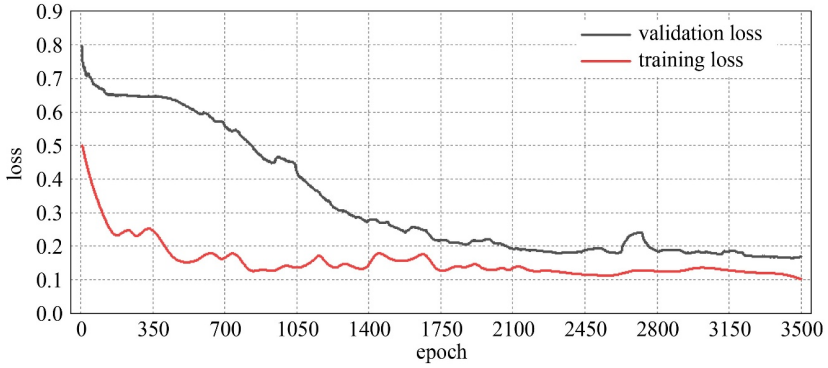

Fig. 11 Loss curves for each epoch.

calculation of accuracy. Table 4 shows that the proportion of crack area in the whole image was tiny. That is, the proportion of non-crack areas in the image was large. When calculating accuracy, both *TP* and *TN* are included. Because the non-crack area is large, the final *TN* value is also very high and far higher than *TP*, *FP* and *FN*. So, the accuracy tends to be 1, which indicates that accuracy is not an appropriate criterion for the scenario used in this paper.

4 Discussion

4.1 Threshold selection in improved APAP algorithm

Since the difference (distance) between two adjacent frames in the video is usually small (especially for ROV, which moves slowly in water) [60], if images are stitched directly frame by frame, it will not only increase the running burden of the computer, but also affect the final stitching result and decrease the efficiency.

Moreover, the ROV's navigation is not uniform and straight because of human operations and complex underwater environments. Therefore, the area scanned by the camera is different in different time periods. So, extracting keyframes should not be based on the video timing but on the severity of scene changes in the video; the more dramatic the scene changes, the larger area scanned by the camera, the more keyframes should be extracted. In our opinion, when the similarity between two adjacent frames is small (the distance between two adjacent frames is large), it means that the scene changes violently in this time period (there are more keyframes in this period).

As shown in Fig. 3, the improved APAP algorithm is based on a threshold when extracting keyframes. This threshold is set artificially, and it is different in different videos. Our study compared the effect of different thresholds on key frame extraction results. Figure 12(a) shows that the number of keyframes extracted from the video sequence decreases as the threshold increased. But the keyframe was not extracted from the video when the threshold was larger than a certain range.

The ratio of keyframes number extracted and the total frames number in video could be identified as the extraction rate. From Fig. 12(b), it can be seen that with the increase of threshold, the extraction rate decreased, and the threshold was different in different regions. The lower the extraction rate, the fewer keyframes extracted, and the fewer images to be stitched, the faster the algorithm. However, this also meant that the distance between two adjacent keyframes was greater.

As shown in Fig. 13, as the distance between images increased, the number of matching points pairs reduced, and the final stitching result could be gradually distorted. In summary, the selection of thresholds was neither too large nor too small. The threshold needed to be set according to the video quality and the stitching requirements.

Table 4 The proportion of crack area in images

area	original image	FCN	GCN
Dam (a)	5.60‰	10.65‰	6.39‰
Dam (b)	3.36‰	8.74‰	4.11‰
Tunnel A	6.76‰	7.76‰	6.49‰
Tunnel B	11.99‰	15.75‰	10.95‰

Table 5 Comparison of FCN and GCN on test set

method	*m-IOU*	*F*1	accuracy
FCN	51.18%	67.70%	99.40%
GCN	75.20%	83.41%	94.30%

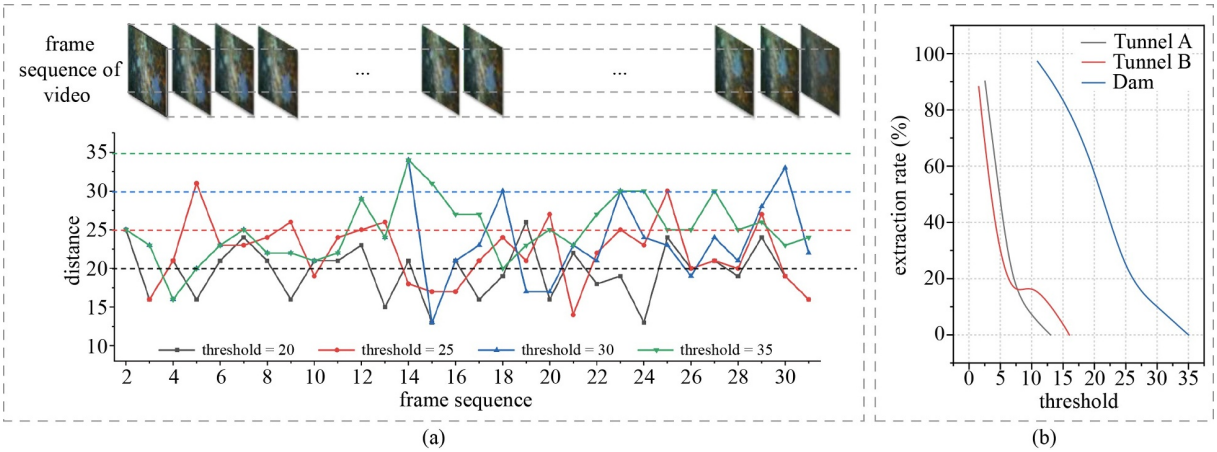

Fig. 12 Effect of threshold on keyframes detection. (a) Extracting keyframes from video based on different thresholds; (b) extraction rate and threshold.

4.2 Effect of the underwater environment on image segmentation

Because the underwater environment is very complex, acquired underwater crack images are affected by many factors, which produces great difficulty in surveying underwater cracks. If we use graph, $G=(V,E,W)$, to represent an image, the nodes correspond to the pixel of the images, which records the color, brightness, etc. of the underwater structure, and the edge corresponds to the relationship between pixels and records the shape, location, etc. of the underwater structure. Therefore, there are two main impacts of the underwater environment on data: the impact on nodes and the impact on edges.

For the impact of nodes, it is mainly manifested in that the image does not reflect the true color of the underwater structure. Many studies [52,61,62] have shown that scattering, refraction, and absorption are unavoidable when light travels in water, as shown in Fig. 14. There is a lot of floating dust in the water, and light is scattered by these impurities. At the same time, underwater cameras often have water shields. Moreover, the medium from the lens to the imaging point is air, so the light will refract when passing through the lens. Because water molecules strongly resonate with photons in the infrared, yellow and ultraviolet bands, there is a strong spectral effect when light is transmitted in water. The energy in yellow, ultraviolet, and infrared bands of light is largely absorbed by water. Therefore, as shown in Fig. 15, underwater structures are mainly imaged in the green band; the underwater image brightness in the green band is higher than that in the red and blue bands; the color information

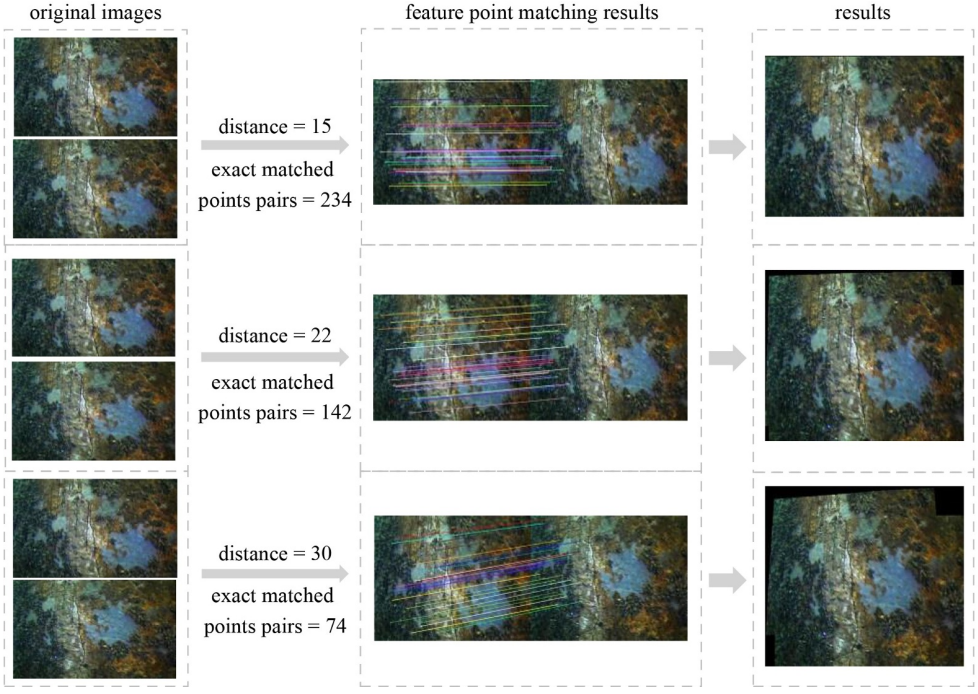

Fig. 13 The influence of the distance between images to be stitched on the stitching result.

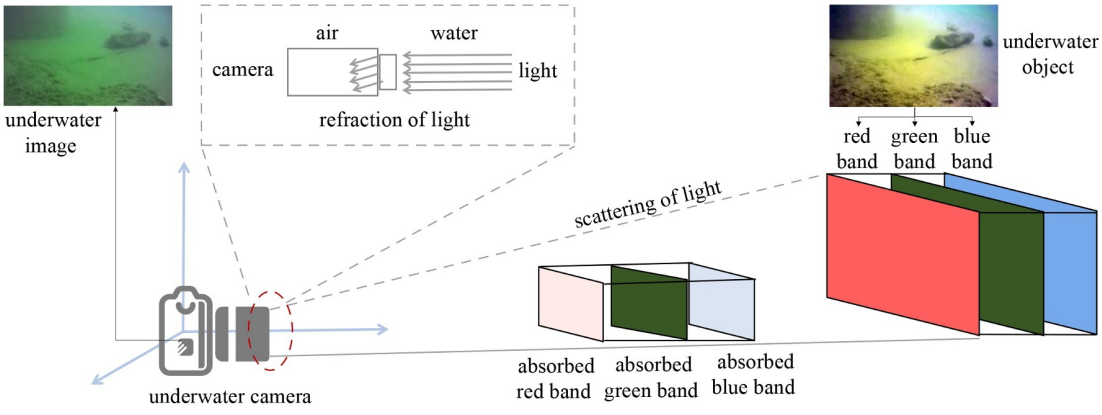

Fig. 14 The influence of underwater environment on images: scattering of light, refraction of light, absorption of light in different bands.

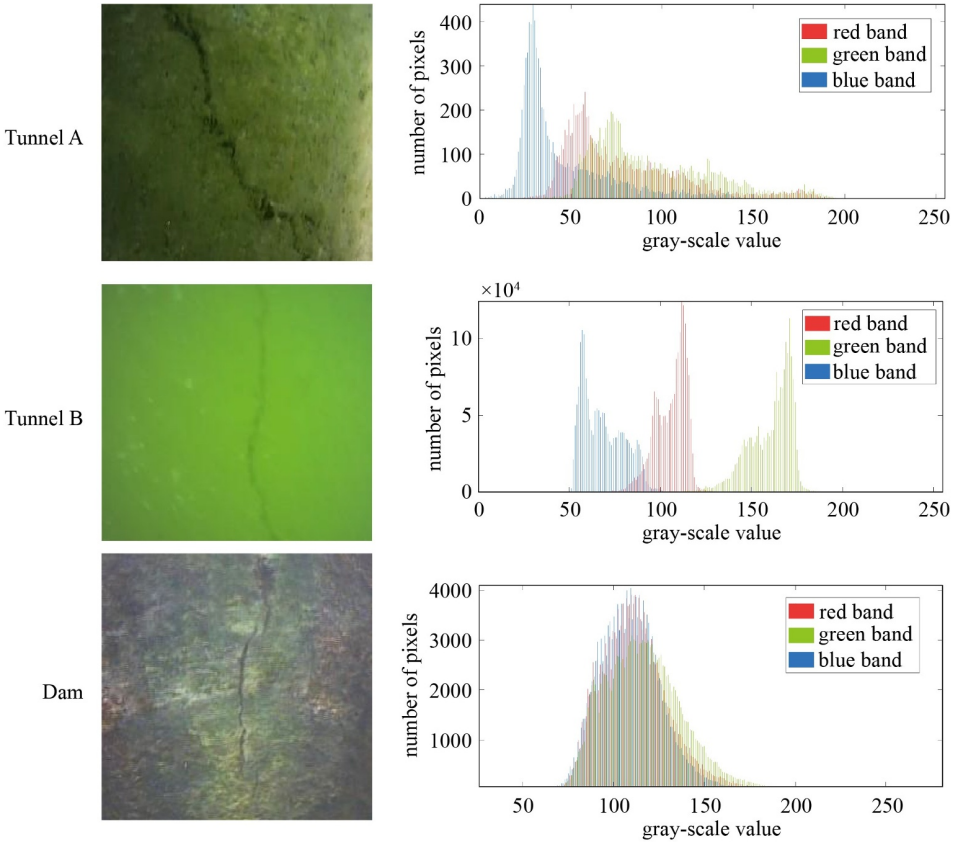

Fig. 15 Underwater images are mainly imaged in green.

of images is incomplete.

The impact of edges is mainly manifested as image distortion. For the underwater environment, due to the use of wide-angle lenses, or fisheye lenses, and the refraction of light caused by multiple media, underwater images are often distorted. Our study tested the distortion of underwater images in Tunnel A, Tunnel B and dam. Figure 16 shows the distorted degree of any point on the image relative to the image center. Moreover, the SMIA TV Distortion of these underwater images were calculated: the distortion of Tunnel A was –5.17%, the distortion of Tunnel B was –2.3%, and the distortion of dam was –29.5%. Significantly, Tunnel A and Tunnel B were monitored by the same camera, but the distortion in the two environments was different. These observations indicate that the distortion of underwater image was mainly barrel distortion, and different underwater environments had different effects on image distortion. Another factor is that the surface of some underwater structures is sometimes a curved surface rather than flat. Images are flat, which means curved surfaces are compressed and distorted during imaging, as shown in Fig. 17.

Generally speaking, image creation involves a structure-to-plane projection. When collecting underwater images with ROV, many factors affect the result, such as refraction, scattering, type of lens, suspended solids, etc. Calculable underwater structures are essentially projected onto a distorted plane. Although the data is still image, the pixel correlation has been distorted, as shown in Fig. 18. So, it would be more appropriate to describe them with non-Euclidean data. Therefore, GCN converged more easily than FCN during training, obtaining higher *m-IOU* and *F*1 values.

In fact, there are other networks [63,64] besides Resnet101 that can work as backbone for FCNs and GCNs. As shown in Table 6, this study compared the performance of GCN and FCN in different backbones and the result shows that Resnet101 is indeed more effective than other networks.

The segmentation results of FCN and GCN under different water depths were also compared, and these results are shown in Fig. 19. It can be seen that with the increase of water depth, FCN was affected by the surrounding environment, and the error increased; many non-crack areas are divided into crack areas. GCN can still maintain good segmentation results and fewer errors. This indicates that GCN is less affected by water depth change and has good stability.

4.3 The order of segmentation and stitching

According to some studies [43,54], stitching the processed images can effectively reduce the stitching

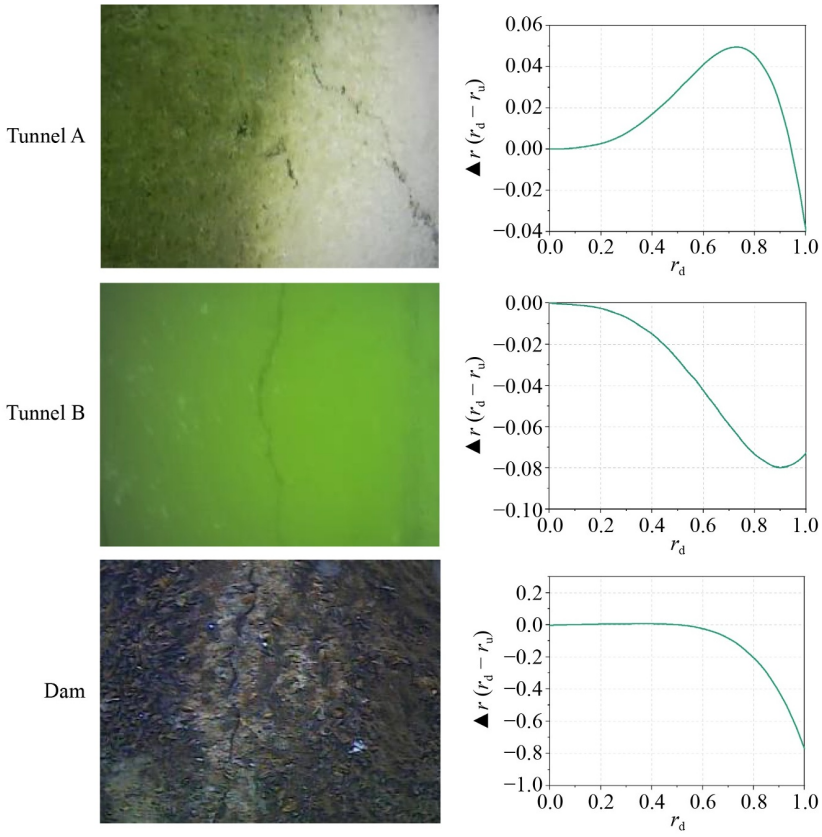

Fig. 16 Image distortion in different areas. Note: r_d represents the distance from a point on the image to the image center, which is also called as image distortion radius; r_u represents the distance from a point on the undistorted image to the image center, which is also called as image undistorted radius; ▲r represents the difference between the distortion radius and the undistorted radius from a point to the center of the image.

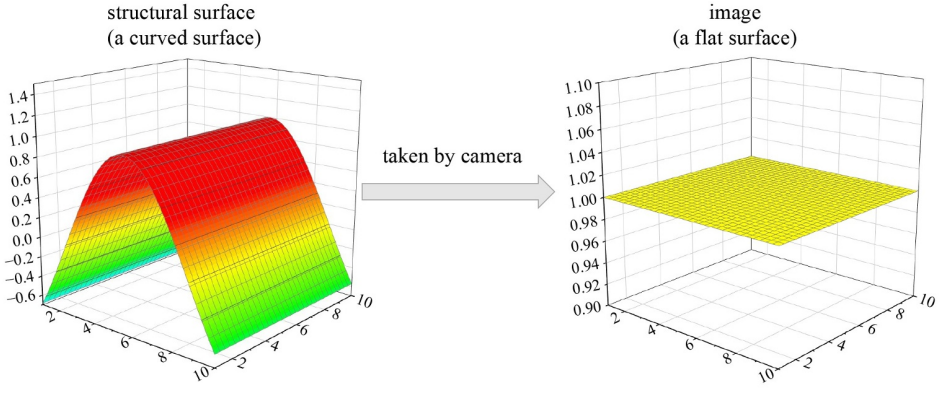

Fig. 17 Three-dimensional underwater structures are compressed into a flat surface during imaging.

time. This paper also compared the order of segmentation and stitching when extracting large-scale cracks, as shown in Fig. 20.

It can be seen that the final result of segmentation-first was similar to that of stitching-first. Moreover, stitching the segmented image can reduce the time-consuming. However, in this paper, stitching and segmentation are two steps of a process. Therefore, it is necessary to analyze the total time consumption. As shown in Table 7, although segmentation first reduced the time, it increased the segmentation time. Overall, stitching first saves time. The total time formula could be expressed as:

$$T_{\text{total}} = T_{\text{seg}} + T_{\text{sti}}, \tag{16}$$

where T_{total} represents the total time; T_{seg} represents the segmentation time; T_{sti} represents the stitching time. Further, T_{seg} is determined by the model and the pixel number. The larger the model size, the more pixels to be processed, the longer the T_{seg} is. T_{sti} is determined by the pixel number and the feature point number.

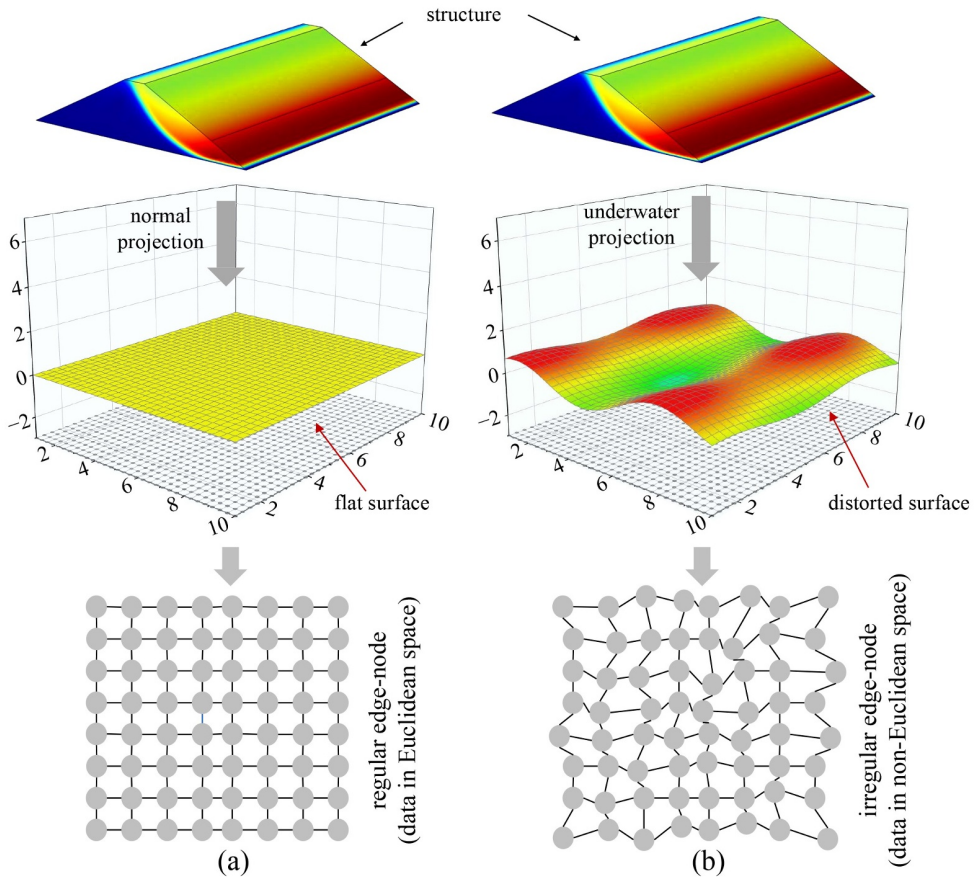

Fig. 18 The reason why underwater pictures can be described with non-Euclidean data. (a) The imaging process of normal images, the relationship between nodes and edges; (b) the imaging process of underwater pictures, the relationship between nodes and edges in pictures.

Table 6 Comparison of different backbones

method	backbone	*m-IOU*
GCN	Resnet101	0.75
	EfficientNet	0.55
	MobilenetV3	0.49
FCN	Resnet101	0.51
	Vgg16	0.39

Although segmentation-first reduced the number of feature points processed during stitching, it also increased the segment pixel number. Due to the large size of GCN, the segmentation time became longer and the total time was increased. Stitching-first reduced the number of pixels to be segmented, so the total time was shorter. Therefore, the relationship between the order and T_{total} is uncertain, which needs to be determined according to the model size and the number of image pixels.

5 Conclusions

Most underwater cracks are large-scale, but an underwater camera has a small field of view and cannot get the complete shape of underwater cracks; this paper presents an underwater large-scale crack detection method based on image stitching and image semantics segmentation.

This paper proposes an improved APAP algorithm, which can directly extract keyframes in the video for image stitching. The experimental result shows that: the improved APAP algorithm can adapt to different underwater environments; the number of keyframes extracted is far less than the total number of video frames, which greatly simplifies the data; APAP can extract a large number of feature points from complex underwater pictures; the use of RANSAC algorithm can reduce useless matching points; there is no obvious seam and ghosting in the stitching result, and the result is ideal.

Based on previous studies, this study considers that: due to the complexity of the underwater environment, the irregularity of the underwater structure, the scattering and absorption of light by water, the presence of suspended matter, the refraction of light, the use of fisheye cameras, and so on, the underwater images are essentially distorted and the relationship between pixels is irregular. Therefore, it is more appropriate to describe and process underwater images using non-Euclidean data.

For image semantics segmentation, the use of GCN to

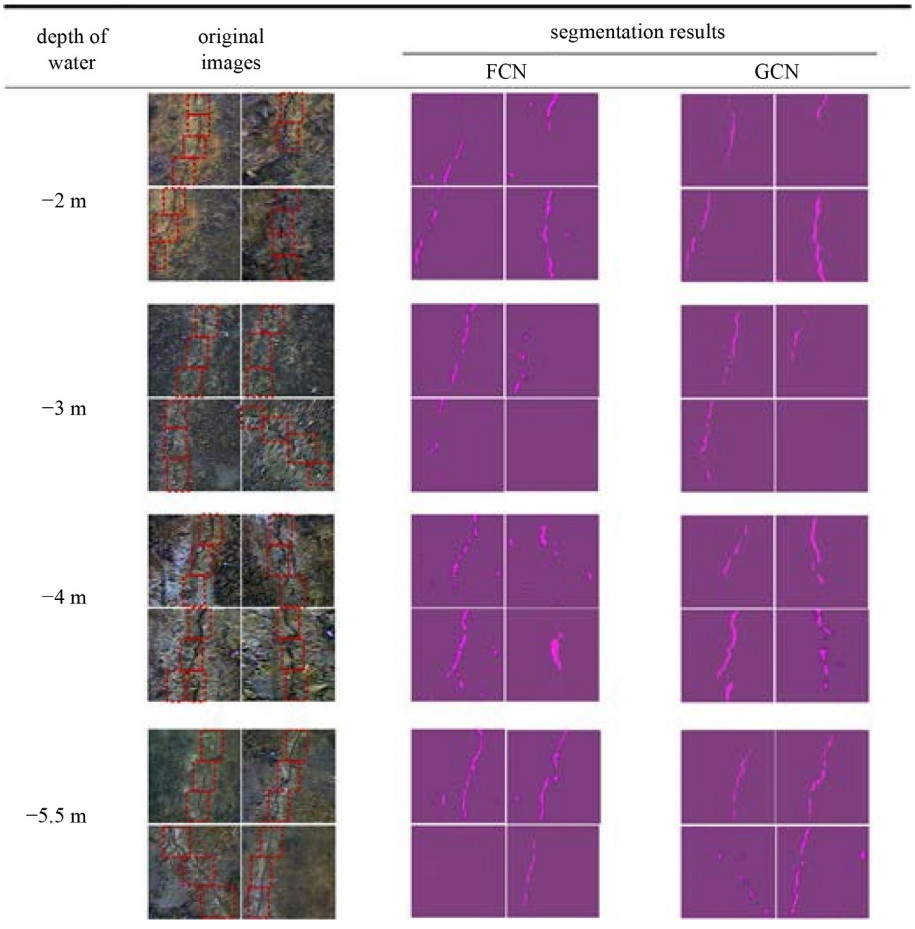

Fig. 19 The effect of different water depths on segmentation results.

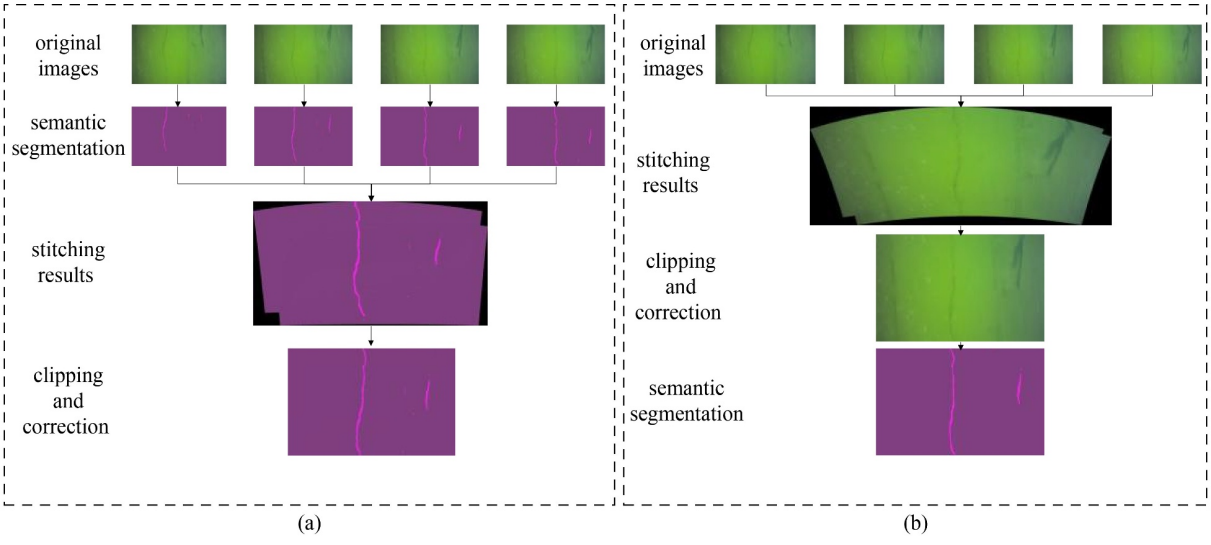

Fig. 20 The effect of the order of segmentation and stitching on final results. (a) Segmentation first; (b) stitching first.

Table 7 Time-consuming comparison of segmentation first and stitching first

the order	T_{seg} (s)	T_{sti} (s)	T_{total} (s)
segmentation first	71.42	2.49	73.91
stitching first	55.95	6.88	62.83

segment underwater cracks is proposed in this study. By inserting the attention mechanism into the Resnet101, the backbone part could retain more crack information during training. By inserting the dual channel graph convolution module, GCN could process the non-Euclidean data and extract the high-dimensional features of underwater

images. The experimental results show that: GCN has good effect in segmenting different underwater cracks; after training, *m-IOU* and *F*1 have reached 75.20% and 83.41%; GCN always has high segmentation accuracy for cracks in different areas, different water depths and different degrees of distortion, which proves that GCN has good generalization ability. Compared with FCN, this study proves that GCN has better performance and potential in underwater image processing.

Acknowledgements Thanks to South to North Water Diversion Central Route Information Technology Co., Ltd. for providing the underwater videos of the water conveyance tunnels for research purposes. Thanks to CISPDR Corporation for providing the underwater video of the dam for research purposes. This work was supported by the National Natural Science Foundation of China (Grant Nos. 51979027, 52079022, 51769033 and 51779035).

References

1. Li W, Chen G, Ge J, Yin X, Li K. High sensitivity rotating alternating current field measurement for arbitrary-angle underwater cracks. NDT & E International, 2016, 79: 123−131
2. Fang H, Duan M. Off-shore Operation Facilities: Equipment and Procedures. Boston: Gulf Professional Publishing, 2014, 537–686
3. Sun J, Xue C, Yu Y. Research on feature-based underwater image mosaic technology. Ship Electronic Engineering, 2017, 37(6): 118−121
4. Rabczuk T, Belytschko T. Cracking particles: A simplified meshfree method for arbitrary evolving cracks. International Journal for Numerical Methods in Engineering, 2004, 61(13): 2316−2343
5. Zhang Y M, Yang X Q, Wang X Y, Zhuang X Y. A micropolar peridynamic model with non-uniform horizon for static damage of solids considering different nonlocal enhancements. Theoretical and Applied Fracture Mechanics, 2021, 113: 102930
6. Zhang Y, Mang H A. Global cracking elements: A novel tool for Galerkin-based approaches simulating quasi-brittle fracture. International Journal for Numerical Methods in Engineering, 2020, 121(11): 2462−2480
7. Martinez J, Rey J, Hidalgo M C, Garrido J, Rojas D. Influence of measurement conditions on the resolution of electrical resistivity imaging: The example of abandoned mining dams in the La Carolina District (Southern Spain). International Journal of Mineral Processing, 2014, 133: 67−72
8. Xue X, Yang X. Earthquake safety assessment of an arch dam using an anisotropic damage model for mass concrete. Computers and Concrete, 2014, 13(5): 633−648
9. Zhang Y M, Zhuang X Y. Cracking elements method for dynamic brittle fracture. Theoretical and Applied Fracture Mechanics, 2019, 102: 1−9
10. Zhang Y M, Zhuang X Y. Cracking elements: A self-propagating Strong Discontinuity embedded Approach for quasi-brittle fracture. Finite Elements in Analysis and Design, 2018, 144: 84−100
11. Rezaiee-Pajand M, Tavakoli F H. Crack detection in concrete gravity dams using a genetic algorithm. Proceedings of the Institution of Civil Engineers. Structures and Buildings, 2015, 168(3): 192−209
12. Su H, Li J, Hu J, Wen Z. Analysis and back-analysis for temperature field of concrete arch dam during construction period based on temperature data measured by DTS. IEEE Sensors Journal, 2013, 13(5): 1403−1412
13. Lai S L, Lee D H, Wu J H, Dong Y M. Detecting the cracks and seepage line associated with an earthquake in an earth dam using the nondestructive testing technologies. Journal of the Chinese Institute of Engineers, 2014, 37(4): 428−437
14. Luo D, Yue Y, Li P, Ma J X, Zhang L L, Ibrahim Z, Ismail Z. Concrete beam crack detection using tapered polymer optical fiber sensors. Measurement, 2016, 88: 96−103
15. Shi P, Fan X, Ni J, Khan Z, Li M. A novel underwater dam crack detection and classification approach based on sonar images. PLoS One, 2017, 12(6): e0179627
16. Xiong P, Xingu Z, Chao Z, Anhua C, Tianyu Z. A UAV-based machine vision method for bridge crack recognition and width quantification through hybrid feature learning. Construction and Building Materials, 2021, 299: 123896
17. Bang S, Park S, Kim H, Kim H. Encoder–decoder network for pixel-level road crack detection in black-box images. Computer-Aided Civil and Infrastructure Engineering, 2019, 34(8): 713−727
18. Xu G. Research and implementation of concrete apparent crack detection algorithm based on deep learning. Thesis for the Master's Degree. Shanghai: Shanghai Jiao Tong University, 2020 (in Chinese)
19. Belytschko T, Lu Y Y, Gu L. Element-free Galerkin methods. International Journal for Numerical Methods in Engineering, 1994, 37(2): 229−256
20. Ukai M. Development of image processing technique for detection of tunnel wall deformation using continuously scanned image. Quarterly Report of RTRI, 2000, 41(3): 120−126
21. Ukai M. Advanced inspection system of tunnel wall deformation using image processing. Quarterly Report of RTRI, 2007, 48(2): 94−98
22. Lu G F, Zhao Q C, Liao J G, He Y B. Pavement crack identification based on automatic threshold iterative method. In: Seventh International Conference on Electronics and Information Engineering. Nanjing: International Society for Optics and Photonics, 2017

23. Talab A M A, Huang Z C, Xi F, Liu H M. Detection crack in image using Otsu method and multiple filtering in image processing techniques. Optik (Stuttgart), 2016, 127(3): 1030−1033
24. Xiao Y, Li J. Crack detection algorithm based on the fusion of percolation theory and adaptive canny operator. In: 2018 37th Chinese Control Conference (CCC). Wuhan: IEEE, 2018, 4295−4299
25. Feng C C, Zhang H, Wang H R, Wang S, Li Y L. Automatic pixel-level crack detection on dam surface using deep convolutional network. Sensors (Basel), 2020, 20(7): 2069
26. Nguyen-Thanh V M, Anitescu C, Alajlan N, Rabczuk T, Zhuang X Y. Parametric deep energy approach for elasticity accounting for strain gradient effects. Computer Methods in Applied Mechanics and Engineering, 2021, 386: 114096
27. Nguyen-Thanh V M, Zhuang X Y, Rabczuk T. A deep energy method for finite deformation hyperelasticity. European Journal of Mechanics. A, Solids, 2020, 80: 103874
28. Guo H W, Zhuang X Y, Rabczuk T. A deep collocation method for the bending analysis of Kirchhoff plate. Computers, Materials & Continua, 2019, 59(2): 433−456
29. Zhuang X Y, Guo H W, Alajlan N, Zhu H, Rabczuk T. Deep autoencoder based energy method for the bending, vibration, and buckling analysis of Kirchhoff plates with transfer learning. European Journal of Mechanics. A, Solids, 2021, 87: 104225
30. Guo H W, Zhuang X Y, Chen P W, Alajlan N, Rabczuk T. Stochastic deep collocation method based on neural architecture search and transfer learning for heterogeneous porous media. Engineering with Computers, 2022, 1−26
31. Samaniego E, Anitescu C, Goswami S, Nguyen-Thanh V M, Guo H, Hamdia K, Zhuang X, Rabczuk T. An energy approach to the solution of partial differential equations in computational mechanics via machine learning: Concepts, implementation and applications. Computer Methods in Applied Mechanics and Engineering, 2020, 362: 112790
32. Anitescu C, Atroshchenko E, Alajlan N, Rabczuk T. Artificial neural network methods for the solution of second order boundary value problems. Computers, Materials & Continua, 2019, 59(1): 345−359
33. Kang F, Wu Y R, Li J J, Li H J. Dynamic parameter inverse analysis of concrete dams based on Jaya algorithm with Gaussian processes surrogate model. Advanced Engineering Informatics, 2021, 49: 101348
34. Kang F, Liu X, Li J J. Temperature effect modeling in structural health monitoring of concrete dams using kernel extreme learning machines. Structural Health Monitoring, 2020, 19(4): 987−1002
35. Cha Y, Choi W, Büyüköztürk O. Deep learning-based crack damage detection using convolutional neural networks. Computer-Aided Civil and Infrastructure Engineering, 2017, 32(5): 361−378
36. Kim B, Cho S. Automated vision-based detection of cracks on concrete surfaces using a deep learning technique. Sensors (Basel), 2018, 18(10): 3452
37. Long J, Shelhamer E, Darrell T. Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015, 3431–3440
38. Dung C V, Anh L D. Autonomous concrete crack detection using deep fully convolutional neural network. Automation in Construction, 2019, 99: 52−58
39. Zhang X X, Rajan D, Story B. Concrete crack detection using context-aware deep semantic segmentation network. Computer-Aided Civil and Infrastructure Engineering, 2019, 34(11): 951−971
40. Zhang C B, Chang C C, Jamshidi M. Concrete bridge surface damage detection using a single-stage detector. Computer-Aided Civil and Infrastructure Engineering, 2020, 35(4): 389−409
41. Liu J W, Yang X, Lau S, Wang X, Luo S, Lee V C S, Ding L. Automated pavement crack detection and segmentation based on two-step convolutional neural network. Computer-Aided Civil and Infrastructure Engineering, 2020, 35(11): 1291−1305
42. Zhang Y, Yuen K V. Crack detection using fusion features-based broad learning system and image processing. Computer-Aided Civil and Infrastructure Engineering, 2021, 36(12): 1568−1584
43. Wang L L, Spencer B F, Li J J, Hu P. A fast image-stitching algorithm for characterization of cracks in large-scale structures. Smart Structures and Systems, 2021, 27(4): 593−605
44. Wu L J, Lin X, Chen Z C, Lin P J, Cheng S Y. Surface crack detection based on image stitching and transfer learning with pretrained convolutional neural network. Structural Control and Health Monitoring, 2021, 28(8): e2766
45. Burt P J, Adelson E H. The laplacian pyramid as a compact image code. IRE Transactions on Communications Systems, 1983, 31(4): 532−540
46. Lowe D G. Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision, 2004, 60(2): 91−110
47. Bay H, Ess A, Tuytelaars T, Van Gool L. Speeded up robust features (SURF). Computer Vision and Image Understanding, 2008, 110(3): 346−359
48. Rublee E, Rabaud V, Konolige K, Bradski G. ORB: An efficient alternative to SIFT or SURF. In: IEEE International Conference on Computer Vision. 2016, 2564–2571
49. Zhu Z, German S, Brilakis I J. Detection of large-scale concrete columns for automated bridge inspection. Automation in Construction, 2010, 19(8): 1047−1055
50. Won J, Park J W, Shim C, Park M W. Bridge-surface panoramic-image generation for automated bridge-inspection using deepmatching. Structural Health Monitoring, 2021, 20(4): 1689−1703
51. Zhang R W, He D H, Li Y P, Huang L, Bao X J. Synthetic imaging through wavy water surface with centroid evolution. Optics Express, 2018, 26(20): 26009−26019
52. Qin G Q. Research on underwater photogrammetry for surface easurement of satellite antenna in simulated zero-gravity conditions. Dissertation for the Doctoral Degree. Zhengzhou: The PLA Information Engineering University, 2011 (in Chinese)
53. Zaragoza J, Chin T J, Tran Q H, Brown M S, Suter D. As-projective-as-possible image stitching with moving DLT. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2014, 36(7): 1285−1298
54. Liu Q. Research on aerial image stitching technology based on improved SURF and APAP. Thesis for the Master's Degree. Dalian: Dalian University of Technology, 2021 (in Chinese)
55. Landrieu L, Simonovsky M. Large-scale point cloud semantic segmentation with superpoint graphs. In: IEEE/CVF Conference

on Computer Vision and Pattern Recognition. Salt Lake City, UT: IEEE, 2018, 4558−4567

56. Yuan Y H, Wang J D. Ocnet: Object context network for scene parsing. 2018, arXiv:1809.00916
57. Ding K M, Chen S P, Meng F. A novel perceptual hash algorithm for multispectral image authentication. Algorithms, 2018, 11(1): 1−14
58. Zhang L, Li X T, Arnab A, Yang K Y, Tong, Y H, Torr P H S. Dual graph convolutional network for semantic segmentation. 2019, arXiv:1909.06121
59. Simo-Serra E, Trulls E, Ferraz L, Kokkinos I, Fua P, Moreno-Noguer F. Discriminative learning of deep convolutional feature point descriptors. In: 2015 IEEE International Conference on Computer Vision. Santiago: IEEE, 2015, 118−126
60. Yang J X, Zhang Y B, Huang L H, Guo D C, Yang Y K. A novel diamond search algorithm for fast block motion estimation. In: International Conference on Image Processing and Pattern Recognition in Industrial Engineering. SPIE, 2010, 737−743
61. Zhao X H, Wang Y, Du Z S, Ye X F. Research on the image enhancement technology of underwater image of super cavitation vehicle. In: 2019 IEEE International Conference on Mechatronics and Automation (ICMA). Tianjin: IEEE, 2019, 1520−1524
62. Zhao X W, Jin T, Chi H, Qu S. Modeling and simulation of the background light in underwater imaging under different illumination conditions. Acta Physica Sinica, 2015, 64(10): 104201
63. Baheti B, Innani S, Gajre S, Talbar S. Eff-unet: A novel architecture for semantic segmentation in unstructured environment. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. Seattle, WA: IEEE, 2020
64. Yuan Y H, Chen X L, Wang J D. Object-contextual representations for semantic segmentation. In: European Conference on Computer Vision. Springer, 2019