Smart Structures and Systems, *Vol. 33, No. 2 (2024) 105-118*
https://doi.org/10.12989/sss.2024.33.2.105

UAV-based bridge crack discovery via deep learning and tensor voting

Xiong Peng [1], Bingxu Duan [1], Kun Zhou [2], Xingu Zhong*[1], Qianxi Li [2] and Chao Zhao [2]

[1] *Hunan Provincial Key Laboratory of Structures for Wind Resistance and Vibration Control, Hunan University of Science and Technology, Xiangtan 411201, Hunan, China*
[2] *School of Civil Engineering, Hunan University of Science and Technology, Xiangtan 411201, Hunan, China*

(*Received November 22, 2022, Revised December 30, 2023, Accepted January 4, 2024*)

Abstract. In order to realize tiny bridge crack discovery by UAV-based machine vision, a novel method combining deep learning and tensor voting is proposed. Firstly, the grid images of crack are detected and descripted based on SE-ResNet50 to generate feature points. Then, the probability significance map of crack image is calculated by tensor voting with feature points, which can define the direction and region of crack. Further, the crack detection anchor box is formed by non-maximum suppression from the probability significance map, which can improve the robustness of tiny crack detection. Finally, a case study is carried out to demonstrate the effectiveness of the proposed method in the Xiangjiang-River bridge inspection. Compared with the original tensor voting algorithm, the proposed method has higher accuracy in the situation of only 1-2 pixels width crack and the existence of edge blur, crack discontinuity, which is suitable for UAV-based bridge crack discovery.

Keywords: bridge crack; feature points; deep learning; tensor voting; unmanned aerial vehicle

1. Introduction

Crack is one of the main damages of concrete bridge. Compared with the traditional manual inspection methods, the UAV-based bridge inspection approach has significant advantages as means to provide faster, cheaper, safer, and more flexible image data acquisition way (Kim *et al*. 2017, Dorafshan and Maguire 2018, Kang and Cha 2018, Jung *et al*. 2019). Specifically, Bolouriana proposes a liDAR-equipped UAV path planning method considering potential locations of defects for bridge inspection (Bolourian and Hammad 2020). Wang presents a measurement method for cracks at the bottom of bridges based on tethered creeping UAV (Wang *et al*. 2020). Liu describes an image-based crack assessment method of bridge piers using unmanned aerial vehicles and three-dimensional scene reconstruction (Liu *et al*. 2020). Ribeiro proposes remote inspection method of RC structures using unmanned aerial vehicles and heuristic image processing (Ribeiro *et al*. 2020). Bhowmick uses vision and deep learning-based algorithms to detect and quantify cracks on concrete surfaces from UAV videos (Bhowmick *et al*. 2020). Ali proposes real-time multiple damage mapping using autonomous UAV and deep faster region-based neural networks for GPS-denied structures (Ali *et al*. 2021). The UAV is applied to measure the crack parameters for evaluating the performance of the bridge infrastructure by advanced photoelectric equipment.

In our previous study, the UAV-based machine system (consists of Sony ILCE-7RM2 camera, three-point laser rangefinder synchronized with the camera shutter) shown in Fig. 1(c) is applied to quantify bridge crack width (Peng *et al*. 2021b). According to the existing bridge technical condition evaluation standards (AASHTO 1970, Sterritt 2009, MOT 2011), cracks wider than 0.2 mm need to be detected. Thus, the accuracy of crack width measurement is reliable. In this case, the UAV system can take clear bridge crack images when flying about 2 m from the bottom of the bridge shown in Fig. 1(d) (Zhong *et al*. 2018). This crack images have clear edges, and it is continuous in the extension direction, with average width of more than 2 pixels shown in Fig. 1(e). However, the UAV-based bridge crack inspection still exists some challenges and problems. In the close-up bridge crack width quantification, the detection area of the equipment is only 0.465 square meters due to the limitation of imaging range shown in Fig. 1(d). If a small bridge with 30 meters long and 20 meters wide is detected, up to 1300 images need to be taken under the ideal condition of no overlap, and most of which are redundant and do not contain cracks. On the other hand, the farther away the UAV system is from the bridge surface, the less interference of the wind field on the bridge surface (Ali *et al*. 2021). Therefore, it is imperative to carry out a long-range crack discovery process before the crack width quantification for reducing a large amount of redundant data and improving detection efficiency. In this case, the minimum imageable pixel width of bridge crack image taken by UAV is only 1-2 pixels, and there are also some edge blurs shown in Fig. 1(f), which is difficult to discovery. It is necessary to propose a method of tiny bridge crack detection suitable for complex background environment, which can greatly improve the efficiency of UAV-based bridge inspection.

*Corresponding author, Ph.D., Professor,
E-mail: 1020086@hnust.edu.cn

ISSN: 1738-1584 (Print), 1738-1991 (Online)

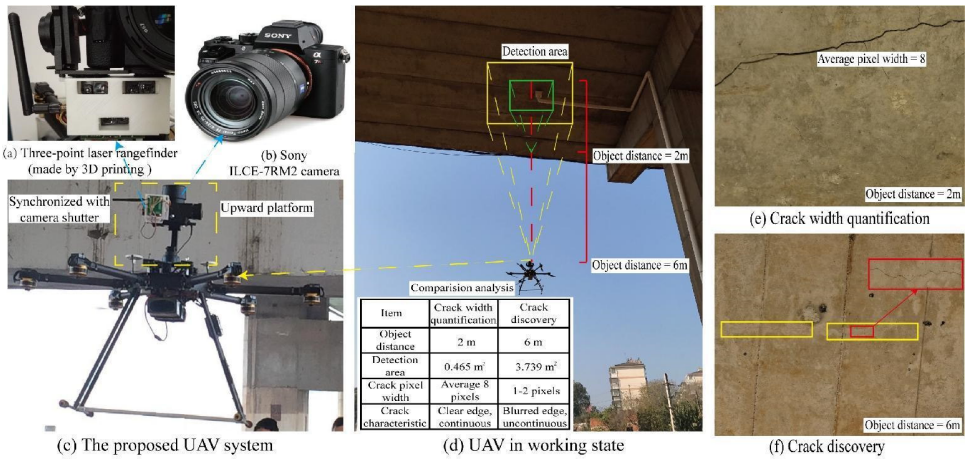


Fig. 1 UAV-based crack discovery

2. Related work of crack detection

Recently, deep learning method provide better detection accuracy since its first application in crack or damage detection (Cha *et al*. 2017, Zhang *et al*. 2022), which are divided into two categories: two-stage model and one-stage model (Dorafshan *et al*. 2018). For crack object detection, the two-stage target recognition network such as R-CNN (Girshick *et al*. 2014), SPP-Net (He *et al*. 2015), Fast-RCNN (Girshick 2015), Faster-RCNN (Zhou *et al*. 2022), R-FCN (Dai *et al*. 2016) have applied to detect the ROI of cracks. Specifically, Zheng applies Fast-RCNN network to amplify and extract the features of crack image (Zheng *et al*. 2020). Ju uses Faster-RCNN-based network to detect crack in complex backgrounds (Huyan *et al*. 2019). Deng presents new types of region-based CNN (R-CNN) crack detector with deformable modules (Deng *et al*. 2020). Although the two-stage neural network has good accuracy in crack location and identification, it needs to produce a large numbers of anchor boxes, which greatly reduces the detection speed of the network.

On the other hand, the one-stage networks such as YOLOv1-v5 (Redmon and Farhadi 2018, Bochkovskiy *et al*. 2020), SSD (Liu *et al*. 2016) and DSSD (Fu *et al*. 2017) is the regression and integration of local object detection result. It uses convolution neural network to judge each grid of the input image (Dong and Catbas 2021), and then the k-means clustering algorithm is applied to get the overall object recognition probability. To be specific, Deng proposes imaging-based crack detection method on concrete surfaces using You Only Look Once network (Deng *et al*. 2020). Yang applies Single Shot MultiBox Detector (SSD) deep learning framework on automatic pavement crack detection (Yang *et al*. 2020). Park uses YOLO algorithm for real-time crack detection (Park *et al*. 2020). Yu presents a real-time detection approach for bridge cracks based on YOLOv4-FPM (Yu *et al*. 2021). Further, in order to improve the accuracy of crack identification, these hybrid implementa-tion methods of deep learning with crack mapping are proposed (Liu and Yeoh 2021a). Liu applies images patches for robust pixel-wise concrete crack segmentation and properties retrieval (Liu and Yeoh 2021b). Jang proposes a deep learning–based autonomous concrete crack evaluation through hybrid image scanning (Jang *et al*. 2019). These algorithms have shown high detection accuracy in fast speed in concrete crack classification and localization with common background.

However, the background of bridge crack image is complex and the imaging conditions are uncertain. The bridge crack images taken by long-distance UAV-based inspection have its unique characteristics: (1) As shown in Fig. 1(f), the width of UAV-taken bridge cracks is only 1-2 pixels in the length direction, which is very unbalanced in the length and width direction, and only accounts for a small proportion in the whole image. Those imbalance makes it difficult for target detection based on neural network (Kang *et al*. 2020). Moreover, those pixel-level encoder to decoder networks such as FCN (Long *et al*. 2015), U-Net (Ronneberger *et al*. 2015) and Deeplab (Chen *et al*. 2017a, Chen *et al*. 2017b) also are difficult to extract this crack with high accuracy; (2) Crack images based on UAV imaging is often discontinuous in length direction, and the crack length is also random, but whether single-stage network or two-stage network, the size of bounding box is predefined, which cannot meet the requirements of crack detection which needs flexible bounding box sizes (Yu *et al*. 2021), (3) As shown in Fig. 1(f), the edge of UAV-taken bridge crack images is often blurred (Bae *et al*. 2020), and the gray field characteristics of many noises such as formwork and shadow are similar to the cracks (Peng *et al*. 2021a, b). The target detection method based on neural network is difficult to recognize tiny bridge crack with 1-2 pixels width in complex background under the condition of small-scale data set.

To address those problems, we propose a novel method of UAV-based tiny bridge crack discovery method combining deep learning and tensor voting. In this method, we consider the tiny crack detection as the local crack feature point calculation and its connection under the special transmission rules. Firstly, the SE-ResNet50 (Hu *et al*. 2018) neural network is used to judge whether the image grid is crack or not, and its center point is taken as the feature point approximately to represent this grid. Then, according to the Gestalt principle (Chen *et al*. 2020),

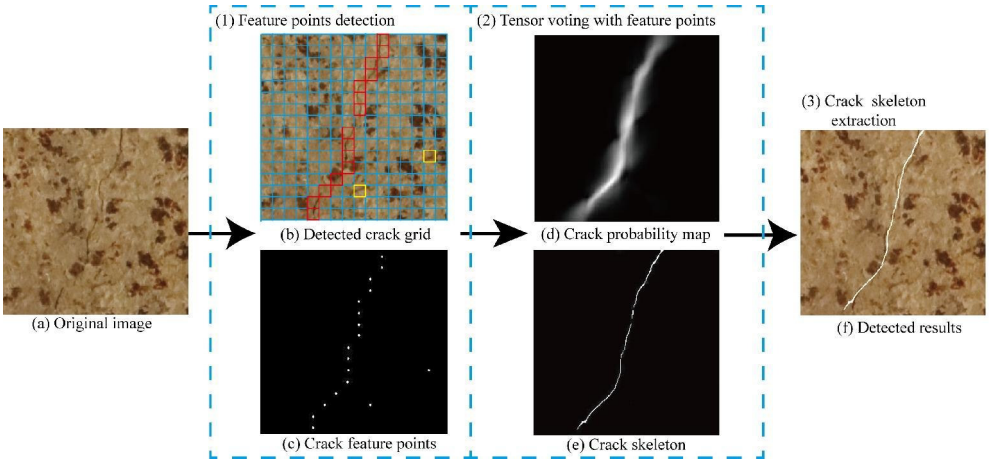


Fig. 2 Overview of tiny bridge crack detection

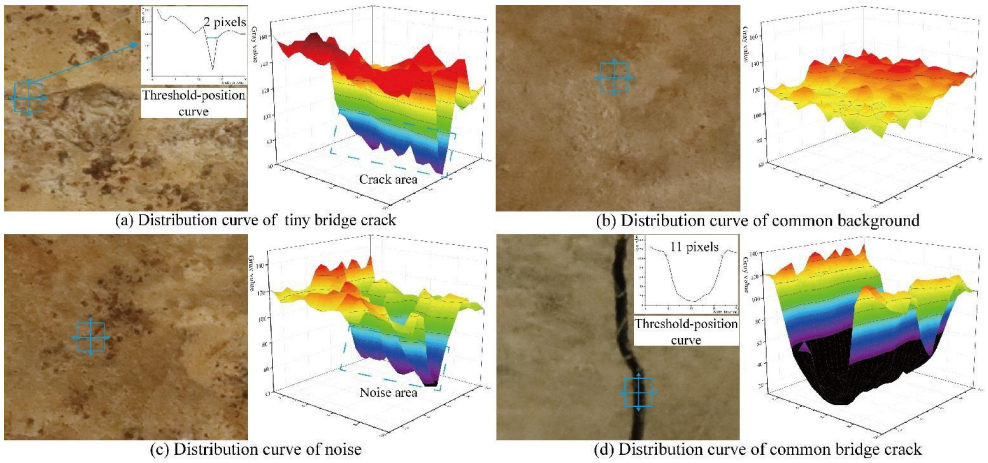


Fig. 3 The distribution curve of bridge crack and background

the information transmission rules between crack feature points and its surrounding points are formulated by tensor voting method (Mordohai and Medioni 2006, Guan *et al*. 2014, Lu *et al*. 2020) to form the probability significance map. Finally, the usefulness of this method is demonstrated in a case study for crack discovery compared with classical target detection network.

3. Methodology

To achieve reliable and rapid tiny bridge crack detection, we have proposed a novel method combining deep learning and tensor voting, which is introduced in the following subsections. As shown in Fig. 2, this framework involves four main tasks: (1) Crack feature points detection, (2) Tensor voting with feature points, (3) Crack target detection and skeleton extraction.

3.1 Crack feature points extraction

As shown in Fig. 3, there are three gray distribution curves, which are crack, common background and imprint noise. The actual bridge crack image has the characteristics of gray scale variation shown in Fig. 3(a), which is quite different from the common background and the clear crack shown in Fig. 3(d). However, in the local region, the gray distribution curves of imprint edge have the similarly characteristics with crack shown in Fig. 3(c). Therefore, the traditional feature description operators such as SIFT (Zhao *et al*. 2021) and Harris (Qin *et al*. 2019) are difficult to correctly describe the neighborhood features of crack feature points correctly. In order to realize the rapid and automatic extraction of crack feature points, the overview of crack feature points extraction is proposed as shown in Fig. 4. In this process, the input image shown in Fig. 4(a) is divided into some grides Fig. 4(b) with size of $n \times n$ (three channels). Firstly, the grides of crack image are fed into SE-ResNet-50 to training the classification model shown in Fig. 4(e). The SE-ResNet50 (Yu *et al*. 2021) neural network is a classical architecture with good target classification accuracy, it is used to judge whether the image grid is a crack or not in the proposed method. Finally, we highlight the image grides judged to be crack, and take its center point as the crack feature point approximately.

As shown in Fig. 5, the SE-ResNet network consists of three parts: squeeze, exception and reweight, which makes use of the dependence between different channel features to

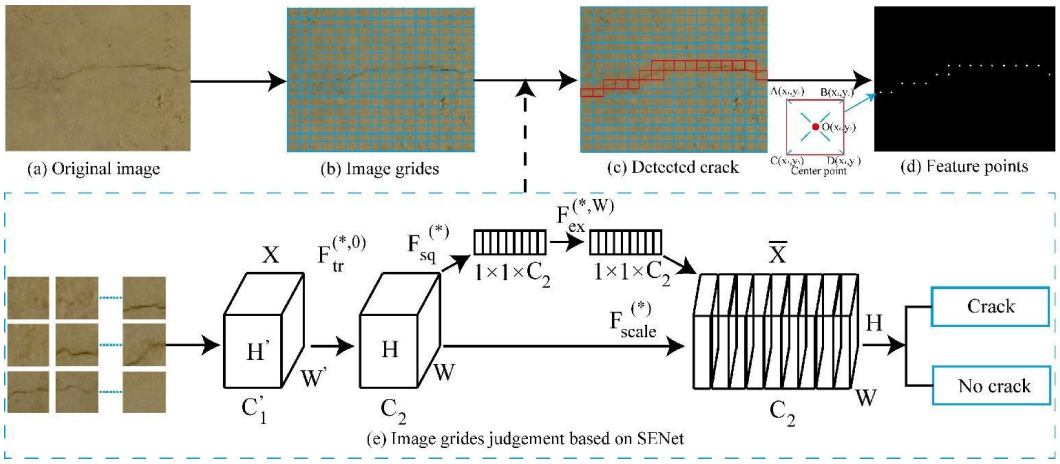


Fig. 4 Framework of crack grides judgement based on SE-ResNet50

improve the performance of processing feature selection and deletion. The whole process is as follows: (1) squeeze operation, which uses global average pooling to pool the feature layer of a single channel into a specific data; (2) Exception operation, through assignment operation, which gives each of the two full connection layers a weight value to build the dependence between different channels; (3) Reweight operation, which is used to normalize the weight to the range of 0-1 by sigmoid function. Finally, each channel is multiplied to make the weight weighted to the original feature. The mapping relationship is shown in Eqs. (1)-(3).

$$z = F_{sq}(u_c) = \frac{1}{W \times H}\sum_{i=1}^{W}\sum_{j=1}^{H} u_c(i,j) \tag{1}$$

$$s = F_{ex}(z, W) = \sigma(g(z, W)) = \sigma(W_2\delta(W_1, z)) \tag{2}$$

$$x_c = F_{scale}(u_c, s) = s_c \bullet u_c \tag{3}$$

Where in Eq. (1)-(3), u_c represents each characteristic channel, W1 and W2 are the weights of the full connection operation, and the activation function δ (Relu) is used between the full connections to deal with the nonlinearity, and finally the normalization function σ (Sigmoid) is used to output the weight vector and the original feature map corresponding to the multiplication operation, output the judgment result.

3.2 Crack feature points voting

Tensor voting is a perceptual reorganization method based on Gestalt principle (Mordohai and Medioni 2006), which uses second-order tensor to represent image pixels. It brings some data with geometric features into the voting process and realizes the mutual transmission of information between data points through different voting fields. According to the smoothness, proximity and continuity principles of tensor voting algorithm, points of the same object and its surrounding points are located in the same tensor field. On the contrary, the noise points are sudden and independent in the surrounding environment. Therefore,

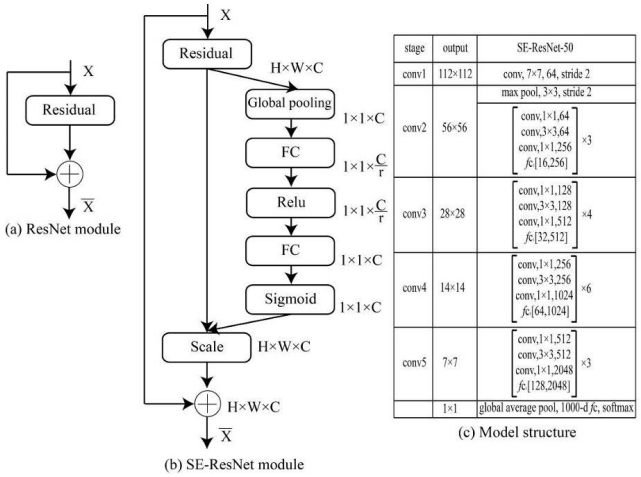


Fig. 5 Flow chart of SE-ResNet50

the tensor voting algorithm can connect typical crack feature points in the noise image. This algorithm consists of three steps: tensor coding, tensor voting and feature extraction. Firstly, the input image pixel point is encoded as a symmetric tensor. Then, the crack feature points are processed by ball voting and stick voting within the voting field. After the voting, their own votes are collected, and then combined with the initial tensor to obtain a new tensor value. Finally, the tensor is decomposed to extract the significant feature value.

(1) Tensor field

When calculating the voting field, the voting value of each point is calculated, and then the final voting field is composed of one voting value. The relative coordinate relationship between point Q and center point O in the voting field is shown in Fig. 6(a). Based on the decay function, the voting intensity value of any point in this area can be obtained to form a tensor voting field. The geometric relationship between the decay function and the voting field is shown in Eqs. (4)-(6)

$$DF(s, k, \sigma) = exp\left[-\left(\frac{s^2 + ck^2}{\sigma^2}\right)\right] \tag{4}$$

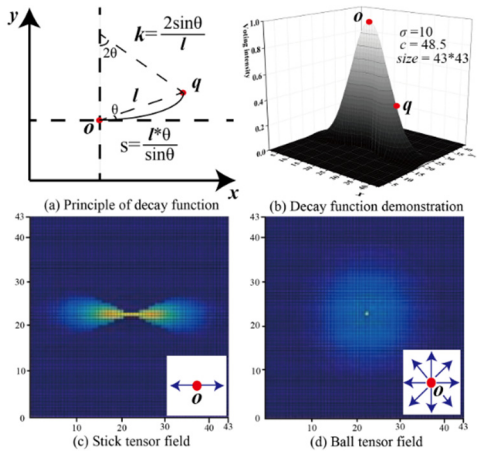


Fig. 6 Ball tensor and stick tensor

$$s = \frac{l \times \theta}{sin\,\theta} \tag{5}$$

$$k = \frac{2\,sin\,\theta}{l} \tag{6}$$

Where $DF(s,k,\sigma)$ is a significant decay function and the scale factor σ is the parameter that determines the size of the voting area. c is the parameter that controls the degree of decay. Further, the voting scale value of c in Eq. (4) is calculated as follows (Mordohai *et al*. 2006)

$$c = \frac{-16 \times (\log_2 0.1) \times (\sigma - 1)}{\pi^2} \tag{7}$$

The decay function characteristic diagram is shown in Fig. 6(b). Its abscissa and ordinate represent the coordinates of receiving points in the field, and the brightness information represents the voting intensity. The size of the voting intensity is related to the distance l. And the larger l, the smaller the value of the attenuation function and the smaller its influence on the voting results. If the pixel value of a point in the plane is 0, the information at this point is expressed as 0 by tensor; If the pixel value at a point in the plane is not 0, the information at this point is expressed as 1 by tensor. If σ = 10, then c = 48.5 is calculated according to Eq. (7), and the influence range of DF is 43 × 43. Multiply the non-zero tensor by the voting decay function to obtain the stick tensor voting domain shown in Fig. 6(c). Integrate the stick tensor domain at (0,2 π) in Eq. (8) to obtain the ball voting domain shown in Fig. 6(d) (Mordohai *et al*. 2006).

$$V_{b-f} = \int_0^{2\pi} V_{s-f}\, d\theta \tag{8}$$

(2) Tensor voting

Firstly, the image pixel shown in Fig. 7(a) is initialized as the ball tensor and the ball tensor voting is performed shown in Fig. 7(b). Then, the number of votes received at each point is analyzed to get the initial direction, and the initial direction is assigned to the stick voting field shown in Fig. 7(c). In the stick tensor voting, the stick voting process between any two points is shown in Fig. 7(d). Take the voting point A as the central point to generate the corresponding voting field, and vote all the receiving points in this field. Then the principle followed by the point B to receive the stick tensor from the point A is as follows

$$V(B) = DF(s,k,\sigma)\begin{bmatrix} -sin(2\theta) \\ cos(2\theta) \end{bmatrix}[-sin(2\theta) \quad cos(2\theta)] \tag{9}$$

Where θ Is the angle between line AB and the tangent line passing through point A. As shown in Fig. 7(e), the whole voting process is completed after stick tensor voting for all pixel points, and the crack probability significance map is obtained after tensor decomposition shown in Fig. 7(f).

(3) Tensor decomposition

As shown in Fig. 8(a), if point A obtains the final voting result through the above voting process. It can be further decomposed into and a ball tensor TB and a stick tensor TS shown in Fig. 8(a). Its definition in two-dimensional space is as follows

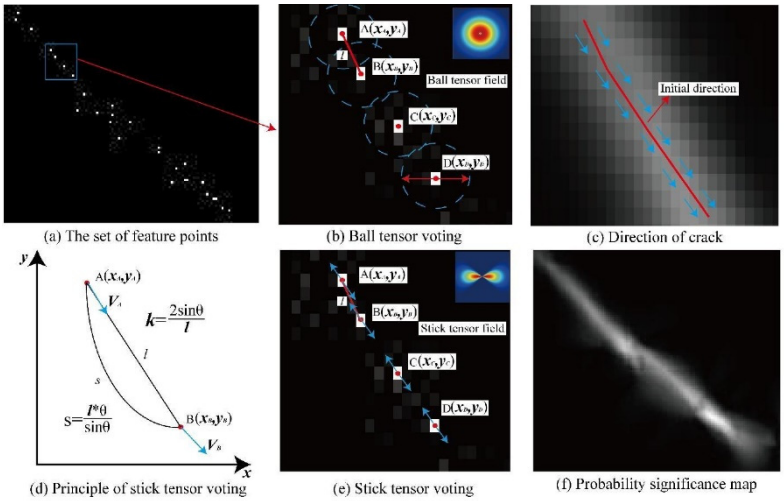
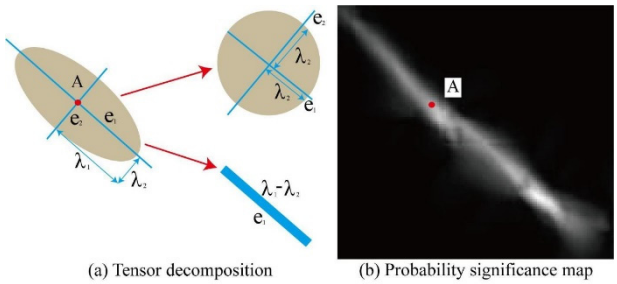


Fig. 7 The process of tensor voting with crack feature points



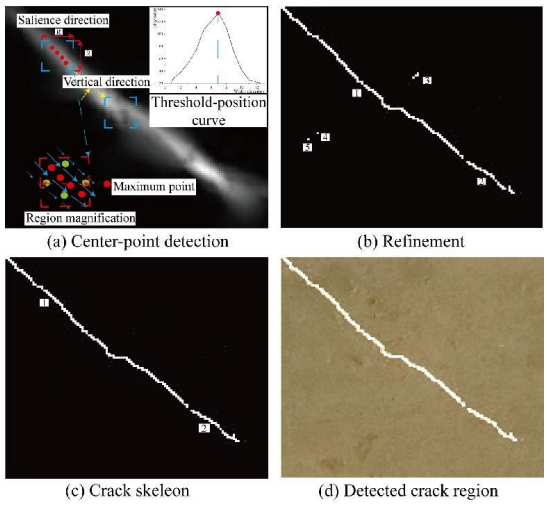
(a) Tensor decomposition (b) Probability significance map

Fig. 8 The process of tensor decomposition

$$T_o = [\vec{e}_1 \vec{e}_2] \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} \begin{bmatrix} \vec{e}_1 \\ \vec{e}_2 \end{bmatrix} = \lambda_1 \vec{e}_1 \vec{e}_1^T + \lambda_2 \vec{e}_2 \vec{e}_2^T \quad (10)$$

$$\begin{aligned} &\lambda_1 \vec{e}_1 \vec{e}_1^T + \lambda_2 \vec{e}_2 \vec{e}_2^T \\ &= (\lambda_1 - \lambda_2) \vec{e}_1 \vec{e}_1^T + \lambda_2 (\vec{e}_1 \vec{e}_1^T + \vec{e}_2 \vec{e}_2^T) \end{aligned} \quad (11)$$

Where e1 and e2 represent the normal and tangential directions of the point, λ_1 and λ_2 represents the size corresponding to the normal and tangential directions; $\lambda_1 - \lambda_2$ is the quantity representing the linear significance (Mordohai *et al*. 2006), which is corresponding to the value of each pixel in the image to obtain a probability significance map shown in Fig. 8(b).



(a) Center-point detection (b) Refinement

(c) Crack skeleon (d) Detected crack region

Fig. 9 Probability significance map refinement

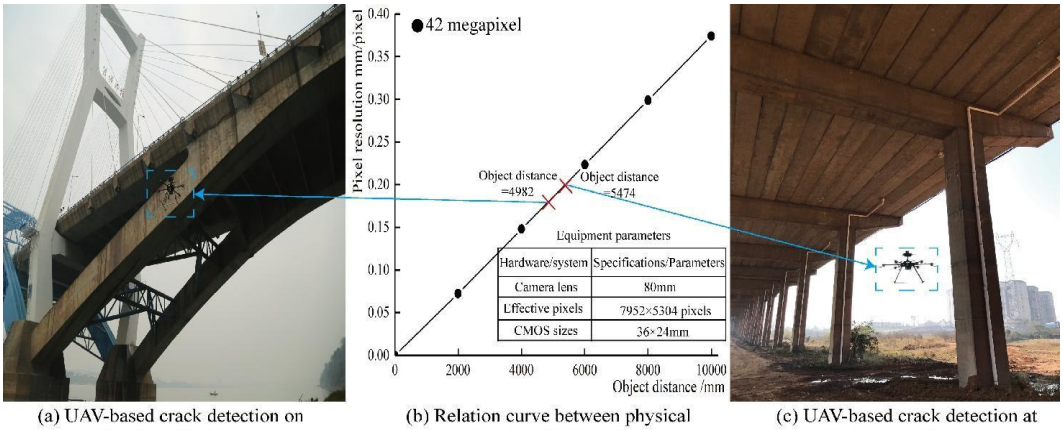
3.3 Crack target detection

After the tensor voting, the probability significant map is formed with crack feature points shown in Fig. 9(a). In order to detect the crack anchor boxes, the center point of crack in probability significance map is extracted by non-maximum suppression method. The specific process of crack target is as follows:

(1) Firstly, a region search range of $R \times R$ is given, and the endpoint in the probability significance map is taken as the starting point. If the pixel value of center point in the corresponding region is not zero, and the region center point is used as the datum point. Then the direction of the datum point obtained by tensor voting is used as the datum direction, and the region is searched according to the datum direction and perpendicular to the datum direction.

(2) Find out the maximum points in the region in both directions, and determine whether the found maximum points are consistent with the center point. If this extreme point is consistent with the center point, which is considered to a candidate point and retained shown in Fig. 9(a). On the contrary, transition to the next non-zero point for search and judgement.

Repeat the above steps until the whole image is traversed to obtain the refined crack probability map. As shown in Fig. 9(b), the maximum value in the normal direction of the voting field direction has been searched, and keep it until traversing the whole image for refining crack probability map to form crack skeleton shown in Fig. 9(c). Finally, the area connectivity morphological algorithm is used to process it. The area in the image is calculated and marked according to the connected area. The area with the connected area less than 5 is removed, and superimposed with the original image to get the crack anchor boxes shown in Fig. 9(d).



(a) UAV-based crack detection on the side of bridge (b) Relation curve between physical size of pixel and object distance (c) UAV-based crack detection at the bottom of bridge

Fig. 10 The process of UAV-based crack discovery

4. Case study

This paper has developed a novel UAV-based bridge crack detection method via deep learning and tensor voting. In order to verify the effectiveness of the above method, a case study of bridge crack discovery based on UAV is carried out. The proposed method is trained and tested using the collected data set, and the tested result is compared with the well-known networks and methods in detection accuracy and robustness. As shown in Fig. 10, the proposed UAV system with corresponding equipment parameters (consists of Sony ILCE-7RM2 camera, three-point laser rangefinder synchronized with the camera shutter) is applied to collect crack data at Xiang-Jiang River bridge. In this inspection process, the operator controls the UAV system at a low speed of 0.6 m/s within the range of 4 to 6 m from the bridge surface, whose pixel resolution is shown in Fig. 10(b). There are initially 1052 crack images collected from airborne imagery in this inspection process.

4.1 Data set build

In order to realize the recognition model training with tiny bridge crack in complex background, the crack classification data set is established. The data preprocessing mainly includes the following two aspects: image clipping and label making. The collected images are cut into image grides with the size of n × n pixels (n from 20 to 25, 30, 35, 40), and the crack image grides (consists of blurred edge, discontinuous condition) and background image grides are made into sample sets respectively shown in Figs. 11 and 12. After clipping, there are 1825 images in total shown in Table 1. According to the ratio of the training set to test set to verification set to = 8:2, the image sample data are divided into the training set (1460) and test set (365) shown in Table 1.

Table 1 The classification data set of bridge cracks

Items	Images size	Training	Validation	Total
Data set A	40×40	1460	365	1825
Data set B	35×35	1460	365	1825
Data set C	30×30	1460	365	1825
Data set D	25×25	1460	365	1825
Data set E	20×20	1460	365	1825

4.2 Training and validation

The hardware system used in the training is Intel core i7-7700@ 3.60 GHz 8-core processor with 64 GBs running memory (RAM) and it is accelerated by NVIDIA GeForce GTX-1080Ti graphics card, running in TensorFlow framework under Linux Ubuntu 20.04 system. The experimental programming language is Python 3.6.2. The batch size in the training process is set to 20 and the epoch is set to 100. Since the experiment is two classification tasks, the classification effectiveness of 2 categories of images is evaluated by cross-entropy loss and accuracy. The training curves of train accuracy, test accuracy and classification loss are shown in Fig. 13. The model began to converge from about epoch 20, and its convergence speed began to accelerate. The curve of training accuracy is close to the curve of test accuracy, and the curve of training loss rate is relatively smooth.

As shown in Fig. 14, the size of crack image in the dataset has a direct impact on the training loss and the accuracy of the model performance. When the grid size is

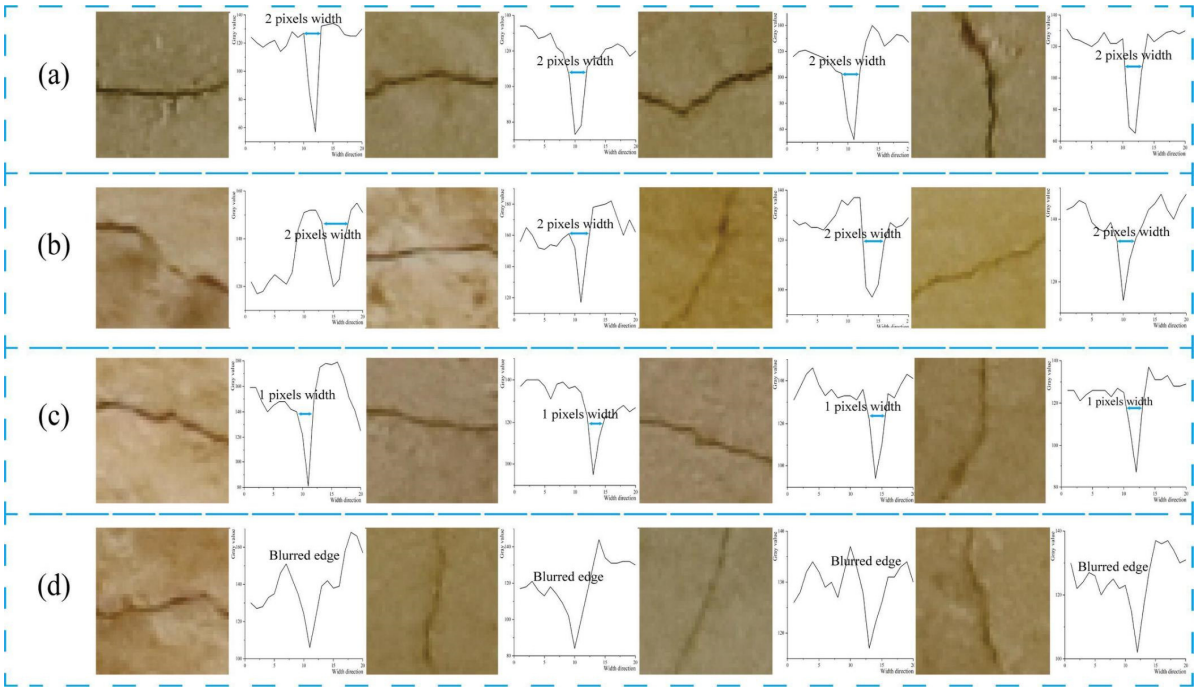


Fig. 11 Positive samples of crack data set: (a) 2 pixels width cracks with clear edges; (b) discontinuity crack; (c) 1 pixel width cracks with blur edges; (d) cracks with blur edges

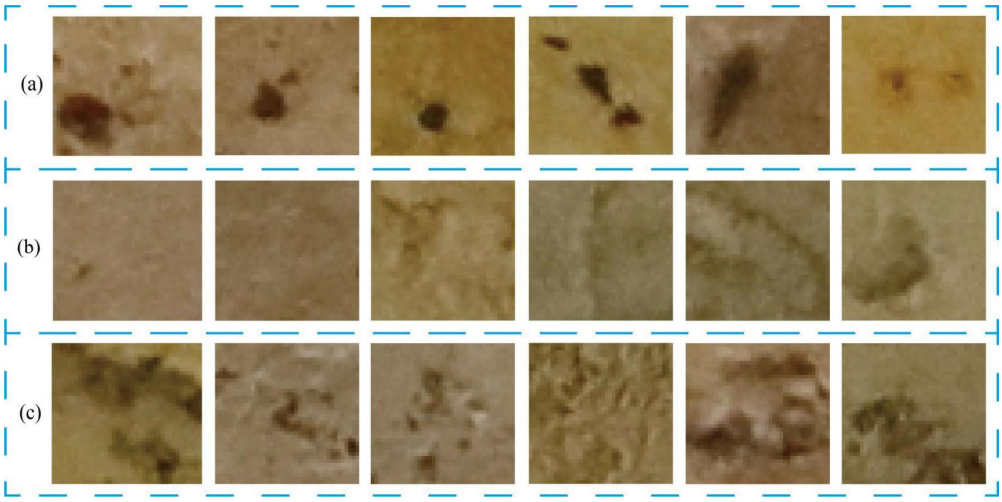


Fig. 12 Negative samples of crack data set: (a) hole; (b) common background; (c) imprint

less than 20, the training loss cannot converge normally, and the test accuracy is reduced to about 65%, which shows that the model cannot correctly classify cracks under too small grid size. On the other hand, when the grid size is greater than 40, the test accuracy decreases to about 75% indicating that the proportion of crack pixels is too low under too large grid size. Then, the ability of the model to classify cracks will be decreasing. The smaller image size means that the airborne crack image needs to be divided into more grides, which has a great impact on the effectiveness and running time of tensor voting. Therefore, the optimal grid size needs to be determined jointly with the parameters of tensor voting.

4.3 Crack feature points voting

The important parameter of this method is the voting scale σ in Eq. (4), which has a very important impact on the results and running time, and it is the only parameter that needs to be adjusted in the tensor voting algorithm. As shown in Fig. 15, the crack feature point images are voted with different voting scale σ (10 - 35) under the grid sizes of 25 pixels × 25 pixels. And the tensor voting field parameters with different voting scale σ (10 - 35) are calculated by Eq. (4) shown in Table 2. When the voting scale is too small, the algorithm will pay more attention to the details of the image, resulting in the failure to connect the crack feature points; However, if the voting scale is too large, although the crack feature point in the image can be connected, it will be affected by the noise points in the image, the detection results are inaccurate and the running time is long.

In order to select the appropriate σ and grid size, a series of σ between 20 and 35 are used to process the 50 images in the tensor voting verification crack data set (1200 pixels × 800 pixels). The numbers of feature point and the running time under the change of σ and gride size are shown in Fig. 16. With the increase of grid size, the detected feature points become less shown in Fig. 16(a), but the reduction running time of tensor voting algorithm is not obvious

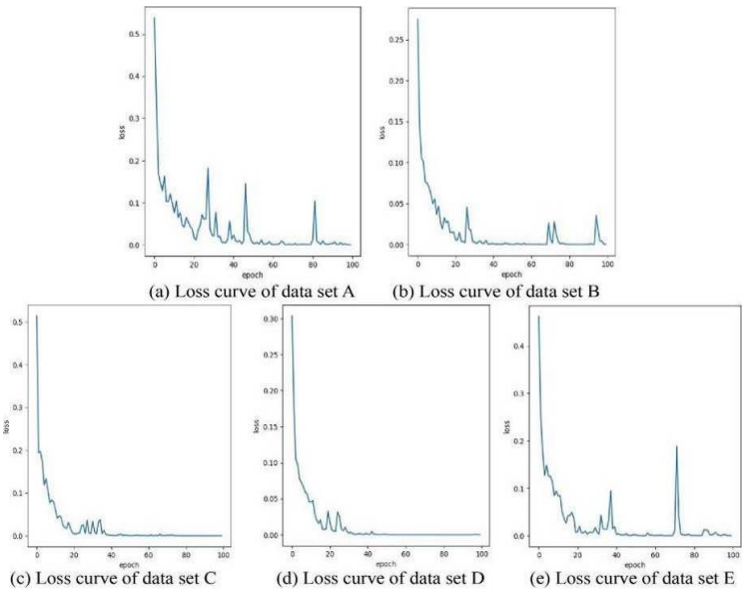


Fig. 13 Loss curve of training different crack data sets

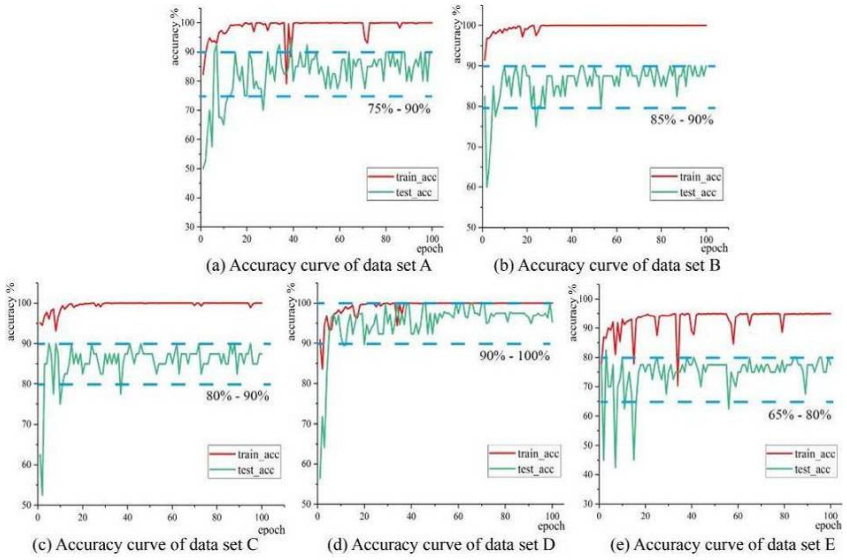


Fig. 14 Accuracy curves of training different crack data sets

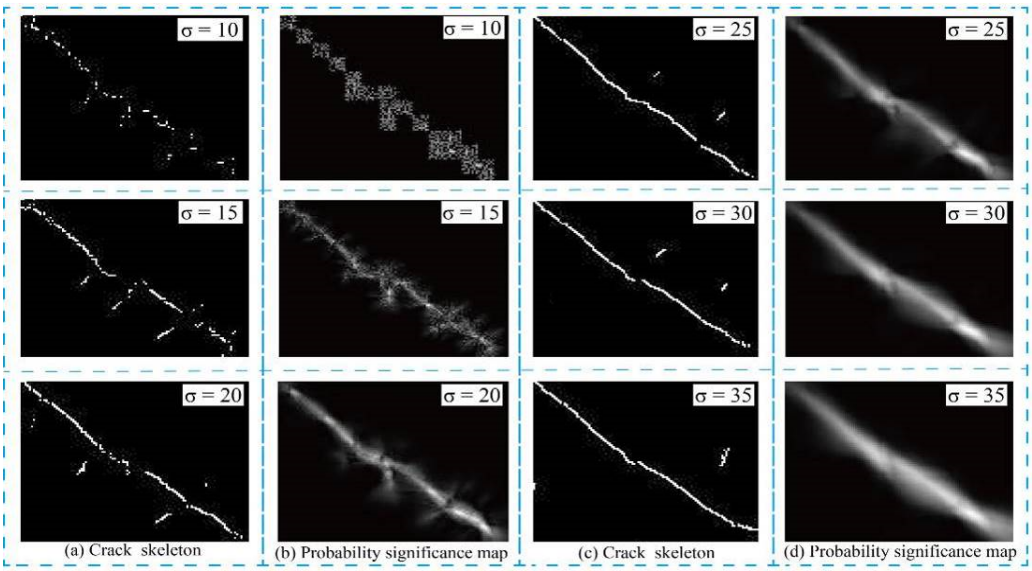


Fig. 15 Results with different voting scale

Table 2 Tensor voting field parameters

Item	$\sigma = 10$	$\sigma = 15$	$\sigma = 20$	$\sigma = 25$	$\sigma = 30$	$\sigma = 35$
c	48.47	75.39	102.32	129.25	156.17	183.10
Tensor voting field size	43×43	65×65	87×87	109×109	129×129	151×151

shown in Fig. 16(b).

The traditional AP (average precision) index is used to describe the ratio between the detected anchor frame and the ground truth. However, the bounding boxes from different algorithms may differ in terms of size and number, still encompasses the entire crack, which cannot correctly evaluate the effect of crack detection with different method. Therefore, in order to further select appropriate σ and grid size, the n_{ACG} (the prediction accuracy of crack grides) indexes are applied to evaluate the proposed algorithm. As shown in Fig. 17, the accuracy is expressed by nACG, and the calculation Eq. (12) is as follows

$$n_{ACG} = \frac{n_1 + n_2 + \ldots n_n}{n_{ground-truth}} \tag{12}$$

where $n_1 - n_n$ is detected crack length shown in Fig. 17(b), $n_{ground-truth}$ are the total number of cracks. Therefore, according to the highest point of the comprehensive evaluation index value, as shown in Fig. 17(d), it is

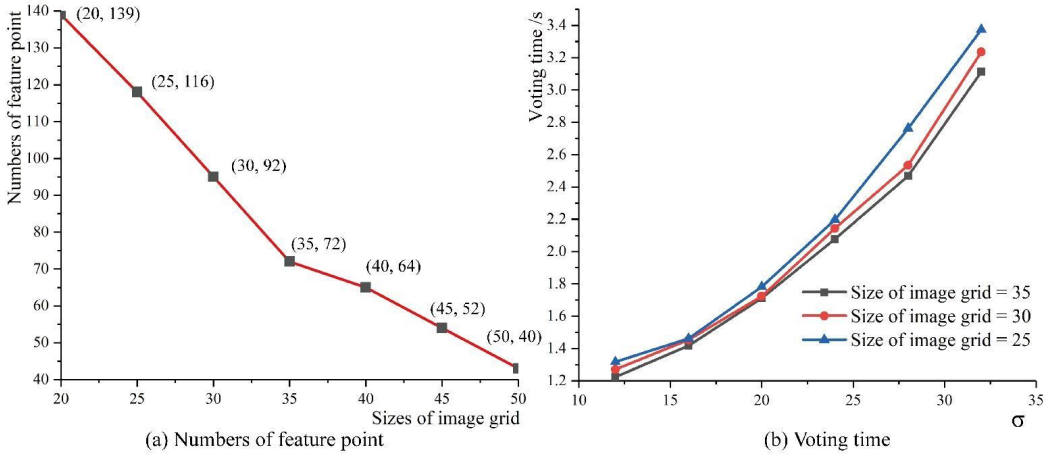


Fig. 16 Running time of tensor voting algorithm with different parameters

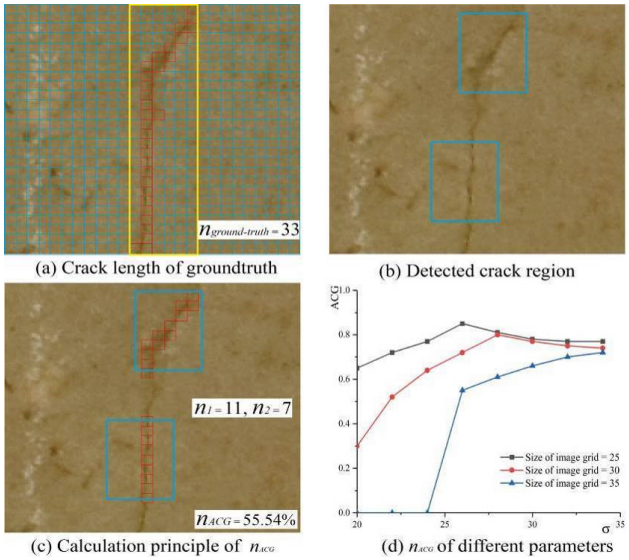


Fig. 17 Performance evaluation of method with different parameters

necessary to select the best one σ = 26 and the sizes of image grid = 25 × 25 pixels. As shown in Fig. 16(b), with the increase of the σ, the average running time of the algorithm also increases. When selecting σ = 26 and the sizes of image grid = 25 × 25 pixels, the corresponding average running time is about 2.3 s.

4.4 Algorithm validation

The proposed method is implemented according to the flow shown in Fig. 2, and the results of the key steps are shown in Fig. 18. The original crack image (1000 × 650 pixels) shown in Fig. 18(a) are divided into 1040 grides with size of 25 × 25 pixels. Firstly, the grides of crack image are fed into well-trained SE-ResNet50 classification model to judge whether the image grid is a crack or not shown in Fig. 18(b). Then we highlight the image grides that judged to be crack, and take its center point as the crack feature point approximately shown in Figs. 18(c) and (d). The σ is determined in section 4.4 with the value of 26. And the size of stick tensor field is 113 × 113 shown in Fig.

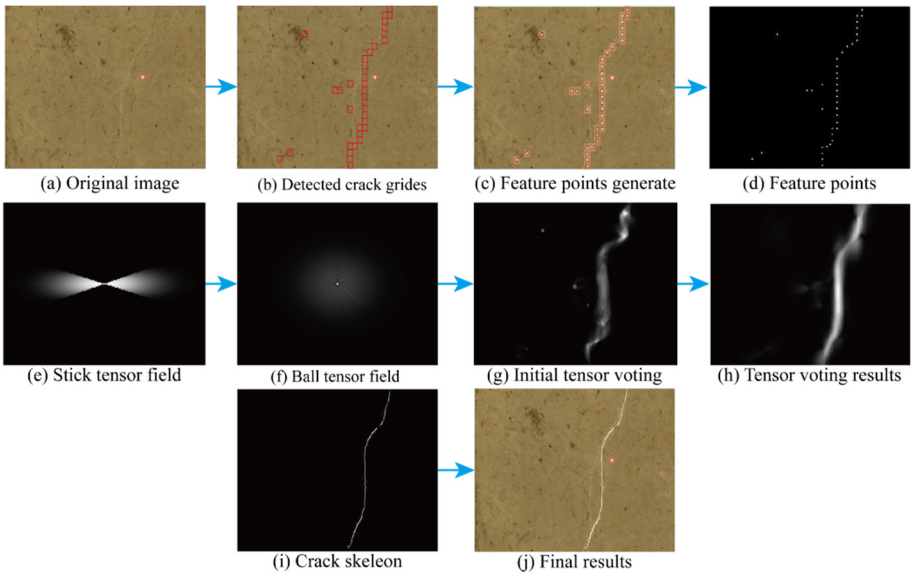


Fig. 18 The detail process of tiny crack recognition

18(e), and ball tensor is derived from the integration and rotation of the stick tensor shown in Fig. 18(f). The feature points are voted by the ball tensor to generate initial direction shown in Fig. 18(g). And then the feature points are voted in the initial direction with stick tensor to formed the crack probability map shown in Fig. 18(h). Finally, the crack skeleton is extracted by non-maximum suppression method shown in Fig. 18(i). And the (a)-(b) continuous cracks with clear edges, (c)-(d) continuous cracks with fuzzy edges, and (e) discontinuous cracks are detected, and the formation results are shown in Fig. 19. The results show that the algorithm proposed in this paper has good effect and good robustness.

4.5 Algorithm performance comparison

In order to verify the effectiveness of this method for tiny bridge crack discovery under complex background, the tiny crack image data sets are processed by tensor voting alone. Then the test data set are used to evaluate those three methods. The detail tiny crack recognition process of the proposed method with the optimal σ and the image gride sizes are shown in Fig. 19. And the evaluation results are shown in Tables 3 and 4.

Compared with the original tensor voting method, the accuracy and robustness of proposed method is higher. In this case, the n_{ACG} of our method has better robustness in special cases with blurred edges, which can achieve above 90% accuracy. And as shown in Fig. 19(a)-(e) and Table 4, our method has a good effect in the situation of only 1-2 pixels width crack and the existence of edge blur, crack discontinuity with about 90% accuracy. Therefore, the proposed method combining deep learning and tensor voting has high detection accuracy, fast detection speed, which can be used for the bridge crack discovery by UAV-based machine vision. And the proposed method combined with deep learning and tensor voting significantly reduces the number of voting seed points. The number of voting seed points is reduced from thousands to dozens and the voting time has been greatly reduced shown in Table 4.

Table 3 Comparison of n_{ACG} parameters for tested crack image

Methods	Image (a)	Image (b)	Image (c)	Image (d)	Image (e)
Proposed method	0.90	0.92	0.92	0.87	0.93
Tensor voting alone	0.72	0.85	0.72	0.66	0.61

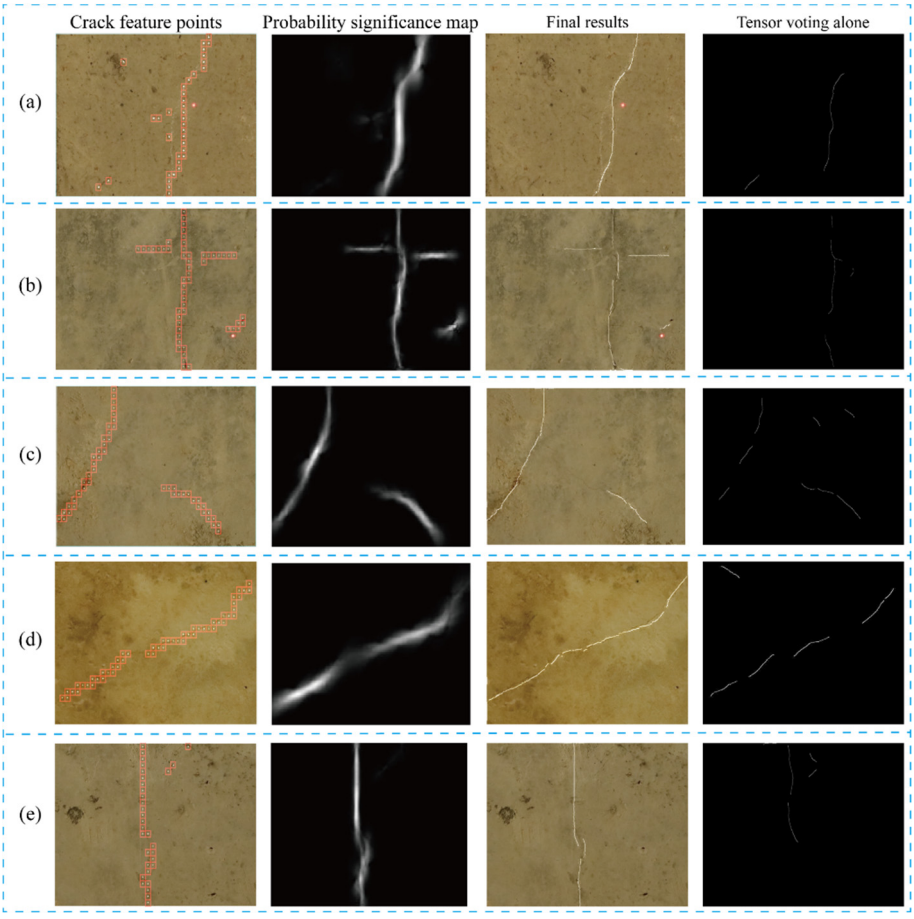


Fig. 19 The tiny crack recognition process of the proposed method: (a)-(b) continuous cracks with clear edges; (c)-(d) continuous cracks with blur edges; (e) discontinuity crack

Table 4 Comparison of average n_{ACG} parameters for proposed method and tensor voting alone

Methods	Separately time (FPS/s)	Total time (s)	Average n_{ACG}
Proposed method	35FPS (Detection) + 2.3 s (Voting)	2.329	0.91
Tensor voting alone	1.2 s (Voting) + 28 s (Voting)	29.2	0.71

5. Conclusions

This research proposes a novel bridge crack discovery approach via deep learning and tensor voting from UAV-based imagery. The proposed method is based on: (1) a classification neural network SE-ResNet50 that can descript and detect the divided crack image grid to generate feature points; and (2) the tensor voting algorithm that can define the direction and region of crack by calculating probability significance map. Further, the features points located in probability significance map are connected by non-maximum suppression to form detection anchor boxes, which can improve the robustness of tiny crack discovery. Finally, the 50 images validation data set are used to examine the performance of the proposed method.

Compared with tensor voting alone, this method has higher detection accuracy in this special situation of only 1-2 pixels width crack, the existence of edge blur and crack discontinuity. The tiny bridge crack discovery based on the proposed method is shown to have good performance, in which the n_{ACG} is about 89%. The time cost of our method is about 25FPS for crack detection per one 7952 × 5304 pixels images and 2.3 s in crack region detection per one 1000 × 650 pixels image.

Although the UAV-based machine vision for long-distance crack discovery and close-up crack width quantification can achieve high accuracy. However, the actual bridge structure health monitoring based on UAV platform still has some limitations and challenges. How to use the collected crack data, combined with multi-source heterogeneous data such as bridge structure form, bridge age, crack location and historical meteorological disasters, to establish a big data analysis and health monitoring system based on UAV platform needs to be further study in the future.

Acknowledgments

This study is supported by Chunhui Project Foundation of the Education Department of China (HZKY20220354) and Scientific Research Foundation of Hunan Provincial Education Department (23B0451), to which the authors are grateful.

References

AASHTO (1970), Manual for Maintenance Inspection of Bridges, Washington, DC, USA.

Ali, R., Kang, D., Suh, G. and Cha, Y.-J. (2021), "Real-time multiple damage mapping using autonomous UAV and deep faster region-based neural networks for GPS-denied structures", *Automat. Constr.*, **130**, 103831. https://doi.org/10.1016/j.autcon.2021.103831

Bae, H., Jang, K. and An, Y.-K. (2021), "Deep super resolution crack network (SrcNet) for improving computer vision–based automated crack detectability in in situ bridges", *Struct. Health Monitor.*, **20**(4), 1428-1442. https://doi.org/10.1177/1475921720917227

Bhowmick, S., Nagarajaiah, S. and Veeraraghavan, A. (2020), "Vision and deep learning-based algorithms to detect and quantify cracks on concrete surfaces from UAV videos", *Sensors*, **20**(21), 6299. https://doi.org/10.3390/s20216299

Bochkovskiy, A., Wang, C.-Y. and Liao, H.-Y.M. (2020), "Yolov4: Optimal speed and accuracy of object detection", arXiv preprint arXiv:2004.10934.

Bolourian, N. and Hammad, A. (2020), "LiDAR-equipped UAV path planning considering potential locations of defects for bridge inspection", *Automat. Constr.*, **117**, 103250. https://doi.org/10.1016/j.autcon.2020.103250

Cha, Y.J., Choi, W. and Büyüköztürk, O. (2017), "Deep learning-based crack damage detection using convolutional neural networks", *Comput.-Aided Civil Infrastr. Eng.*, **32**(5), 361-378. https://doi.org/10.1111/mice.12263

Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K. and Yuille, A.L. (2017a), "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs", *IEEE Transact. Pattern Anal. Mach. Intell.*, **40**(4), 834-848. https://doi.org/10.1109/TPAMI.2017.2699184

Chen, L.-C., Papandreou, G., Schroff, F. and Adam, H. (2017b), "Rethinking atrous convolution for semantic image segmentation", arXiv preprint arXiv:1706.05587.

Chen, H., Zhao, H., Han, D., Liu, W., Chen, P. and Liu, K. (2020), "Structure-aware-based crack defect detection for multicrystalline solar cells", *Measurement*, **151**, 107170. https://doi.org/10.1016/j.measurement.2019.107170

Dai, J., Li, Y., He, K. and Sun, J. (2016), "R-fcn: Object detection via region-based fully convolutional networks", *Adv. Neural Info. Process. Syst.*, **29**.

Deng, L., Chu, H.-H., Shi, P., Wang, W. and Kong, X. (2020), "Region-based CNN method with deformable modules for visually classifying concrete cracks", *Appl. Sci.*, **10**(7), 2528. https://doi.org/10.3390/app10072528

Deng, J., Lu, Y. and Lee, V.C.-S. (2021), "Imaging-based crack detection on concrete surfaces using You Only Look Once network", *Struct. Health Monitor.*, **20**(2), 484-499. https://doi.org/10.1177/1475921720938486

Dong, C.-Z. and Catbas, F.N. (2021), "A review of computer vision–based structural health monitoring at local and global levels", *Struct. Health Monitor.*, **20**(2), 692-743. https://doi.org/10.1177/1475921720935585

Dorafshan, S. and Maguire, M. (2018), "Bridge inspection: human performance, unmanned aerial systems and automation", *J. Civil Struct. Health Monitor.*, **8**(3), 443-476. https://doi.org/10.1007/s13349-018-0285-4

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

Fu, C.-Y., Liu, W., Ranga, A., Tyagi, A. and Berg, A.C. (2017), "Dssd: Deconvolutional single shot detector", arXiv preprint arXiv:1701.06659.

Girshick, R. (2015), "Fast r-cnn", *Proceedings of the IEEE*

International Conference on Computer Vision.

Girshick, R., Donahue, J., Darrell, T. and Malik, J. (2014), "Rich feature hierarchies for accurate object detection and semantic segmentation", *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Guan, H., Li, J., Yu, Y., Chapman, M., Wang, H., Wang, C. and Zhai, R. (2014), "Iterative tensor voting for pavement crack extraction using mobile laser scanning data", *IEEE Transact. Geosci. Remote Sens.*, **53**(3), 1527-1537. https://doi.org/10.1109/TGRS.2014.2344714

He, K., Zhang, X., Ren, S. and Sun, J. (2015), "Spatial pyramid pooling in deep convolutional networks for visual recognition", *IEEE Transact. Pattern Anal. Mach. Intell.*, **37**(9), 1904-1916. https://doi.org/10.1109/TPAMI.2015.2389824

Hu, J., Shen, L. and Sun, G. (2018), "Squeeze-and-excitation networks", *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Huyan, J., Li, W., Tighe, S., Zhai, J., Xu, Z. and Chen, Y. (2019), "Detection of sealed and unsealed cracks with complex backgrounds using deep convolutional neural network", *Automat. Constr.*, **107**, 102946. https://doi.org/10.1016/j.autcon.2019.102946

Jang, K., Kim, N. and An, Y.-K. (2019), "Deep learning–based autonomous concrete crack evaluation through hybrid image scanning", *Struct. Health Monitor.*, **18**(5-6), 1722-1737. https://doi.org/10.1177/1475921718821719

Jung, H.-J., Lee, J.H., Yoon, S.S. and Kim, I.H. (2019), "Bridge Inspection and condition assessment using Unmanned Aerial Vehicles (UAVs): Major challenges and solutions from a practical perspective", *Smart Struct. Syst., Int. J.*, **24**(5), 669-681. https://doi.org/10.12989/sss.2019.24.5.669

Kang, D. and Cha, Y.J. (2018), "Autonomous UAVs for structural health monitoring using deep learning and an ultrasonic beacon system with geo-tagging", *Comput.-Aided Civil Infrastr. Eng.*, **33**(10), 885-902. https://doi.org/10.1111/mice.12375

Kang, D., Benipal, S.S., Gopal, D.L. and Cha, Y.-J. (2020), "Hybrid pixel-level concrete crack segmentation and quantification across complex backgrounds using deep learning", *Automat. Constr.*, **118**, 103291. https://doi.org/10.1016/j.autcon.2020.103291

Kim, H., Lee, J., Ahn, E., Cho, S., Shin, M. and Sim, S.-H. (2017), "Concrete crack identification using a UAV incorporating hybrid image processing", *Sensors*, **17**(9), 2052. https://doi.org/10.3390/s17092052

Liu, Y. and Yeoh, J.K. (2021a), "Automated crack pattern recognition from images for condition assessment of concrete structures", *Automat. Constr.*, **128**, 103765. https://doi.org/10.1016/j.autcon.2021.103765

Liu, Y. and Yeoh, J.K. (2021b), "Robust pixel-wise concrete crack segmentation and properties retrieval using image patches", *Automat. Constr.*, **123**, 103535. https://doi.org/10.1016/j.autcon.2020.103535

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y. and Berg, A.C. (2016), "Ssd: Single shot multibox detector", In: *European Conference on Computer Vision*. https://doi.org/10.1007/978-3-319-46448-0_2

Liu, Y.F., Nie, X., Fan, J.S. and Liu, X.G. (2020), "Image-based crack assessment of bridge piers using unmanned aerial vehicles and three-dimensional scene reconstruction", *Comput.-Aided Civil Infrastr. Eng.*, **35**(5), 511-529. https://doi.org/10.1111/mice.12501

Long, J., Shelhamer, E. and Darrell, T. (2015), "Fully convolutional networks for semantic segmentation", *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Lu, J.Z., Hu, M.Q., Dong, J., Han, S.L. and Su, A. (2020), "A novel dense descriptor based on structure tensor voting for multi-modal image matching", *Chinese J. Aeronaut.*, **33**(9), 2408-2419. https://doi.org/10.1016/j.cja.2020.02.002

Mordohai, P. and Medioni, G. (2006), "Tensor voting: a perceptual organization approach to computer vision and machine learning", In: *Synthesis Lectures on Image, Video, and Multimedia Processing*, **2**(1), 1-136.

MOT (2011), Standards for technical condition evaluation of highway bridges, MOT Beijing, China.

Park, S.E., Eem, S.-H. and Jeon, H. (2020), "Concrete crack detection and quantification using deep learning and structured light", *Constr. Build. Mater.*, **252**, 119096. https://doi.org/10.1016/j.conbuildmat.2020.119096

Peng, X., Zhong, X., Chen, A., Zhao, C., Liu, C. and Chen, Y.F. (2021a), "Debonding defect quantification method of building decoration layers via UAV-thermography and deep learning", *Smart Struct. Syst., Int. J.*, **28**(1), 55-67. https://doi.org/10.12989/sss.2021.28.1.055

Peng, X., Zhong, X., Zhao, C., Chen, A. and Zhang, T. (2021b), "A UAV-based machine vision method for bridge crack recognition and width quantification through hybrid feature learning", *Constr. Build. Mater.*, **299**, 123896. https://doi.org/10.1016/j.conbuildmat.2021.123896

Qin, J., Li, H., Xiang, X., Tan, Y., Pan, W., Ma, W. and Xiong, N.N. (2019), "An encrypted image retrieval method based on Harris corner optimization and LSH in cloud computing", I*EEE Access*, **7**, 24626-24633. https://doi.org/10.1109/ACCESS.2019.2894673

Redmon, J. and Farhadi, A. (2018), "Yolov3: An incremental improvement", arXiv preprint arXiv:1804.02767.

Ribeiro, D., Santos, R., Shibasaki, A., Montenegro, P., Carvalho, H. and Calçada, R. (2020), "Remote inspection of RC structures using unmanned aerial vehicles and heuristic image processing", *Eng. Fail. Anal.*, **117**, 104813. https://doi.org/10.1016/j.engfailanal.2020.104813

Ronneberger, O., Fischer, P. and Brox, T. (2015), "U-net: Convolutional networks for biomedical image segmentation", *Proceedings of International Conference on Medical Image Computing and Computer-Assisted Intervention*, Munich, Germany, October.

Sterritt, G. (2009), Review of Bridge Inspection Competence and Training Project Rep, UK Bridges Board, London, UK.

Wang, H.-F., Zhai, L., Huang, H., Guan, L.-M., Mu, K.-N. and Wang, G.-p. (2020), "Measurement for cracks at the bottom of bridges based on tethered creeping unmanned aerial vehicle", *Automat. Constr.*, **119**, 103330. https://doi.org/10.1016/j.autcon.2020.103330

Yang, J., Fu, Q. and Nie, M. (2020), "Road crack detection using deep neural network with receptive field block", In: *IOP Conference Series: Materials Science and Engineering*.

Yu, Z., Shen, Y. and Shen, C. (2021), "A real-time detection approach for bridge cracks based on YOLOv4-FPM", *Automat. Constr.*, **122**, 103514. https://doi.org/10.1016/j.autcon.2020.103514

Zhang, C., Shu, J., Shao, Y. and Zhao, W. (2022), "Automated generation of FE models of cracked RC beams based on 3D point clouds and 2D images", *J. Civil Struct. Health Monitor*, **12**(1), 29-46. https://doi.org/10.1007/s13349-021-00525-5

Zhao, H., Yang, D. and Yu, J. (2021), "3D target detection using dual domain attention and SIFT operator in indoor scenes", *Visual Comput.*, **38**, 3365-3774. https://doi.org/10.1007/s00371-021-02217-z

Zheng, M., Lei, Z. and Zhang, K. (2020), "Intelligent detection of building cracks based on deep learning", *Image Vision Comput.*, **103**, 103987. https://doi.org/10.1016/j.imavis.2020.103987

Zhong, X., Peng, X., Yan, S., Shen, M. and Zhai, Y. (2018), "Assessment of the feasibility of detecting concrete cracks in images acquired by unmanned aerial vehicles", *Automat.*

Constr., **89**, 49-57. https://doi.org/10.1016/j.autcon.2018.01.005
Zhou, Q., Ding, S., Qing, G. and Hu, J. (2022), "UAV vision detection method for crane surface cracks based on Faster R-CNN and image segmentation", *J. Civil Struct. Health Monitor*., **12**, 845-855. https://doi.org/10.1007/s13349-022-00577-1