The Journal of Supercomputing (2025) 81:1410
https://doi.org/10.1007/s11227-025-07903-6

An innovative UAV and deep learning-based framework for automatic bridge crack detection and measurement

Changdong Zhou[1,2] · Mingjing Dai[1] · Feng Wang[2,3] · Yu Dong[1] · Xinghua Chen[2,3] · Chenghuan He[2]

Received: 2 April 2025 / Accepted: 22 September 2025

Abstract

This study proposes a framework integrating unmanned aerial vehicles and deep learning for the detection, classification, and quantification of surface cracks—a critical damage type in concrete bridges—while addressing limitations of traditional manual inspection methods; it classifies cracks into two categories: slightly tortuous normal linear cracks with measurable width and length, and crazing cracks, the latter with a disordered, reticular morphology precluding effective width/length quantification, so only normal linear cracks are quantified. Considering massive high-resolution image data from unmanned aerial vehicles and high computational demands of deep learning models for semantic segmentation tasks involving millions of pixels, a high-performance computing cluster accelerates image preprocessing, model training, and inference—meeting timeliness requirements of real-time bridge health monitoring—with unmanned aerial vehicles capturing high-resolution images to cover large bridge surfaces, split via an adaptive segmentation technique to enhance processing efficiency and accuracy, while a bridge classification model filters out irrelevant backgrounds to focus crack detection on the bridge structure. Subsequently, a deep learning-based multi-type crack recognition and damage detection model is deployed: a U-Net network performs crack segmentation, a custom algorithm conducts measurement, and integrating these technologies into a unified system significantly improves bridge crack identification efficiency, providing data-driven support for maintenance and management; experimental results show the method substantially reduces annotation costs, mitigates challenges in data collection, and enhances model generalization—delivering considerable value for concrete bridge health monitoring applications.

Keywords Bridge crack defects · UAV · Domain adaptation · Semantic segmentation · Deep learning

Extended author information available on the last page of the article

Published online: 03 October 2025

1 Introduction

With the increasing number of bridges and the extension of their service lives, issues, such as cracks (a common form of damage), have become more prominent. In this paper, "cracks" refer to visible surface fractures, while "damage" is a broader term that includes cracks as well as other deterioration forms. Cracks typically indicate structural stress concentration or material aging, and if not repaired in a timely manner, they may continue to propagate, leading to structural damage, exposed reinforcement corrosion, and compromised bridge stability, ultimately shortening its lifespan [1]. Therefore, bridge health monitoring has become critically important [2].

Traditional inspection methods rely on manual visual inspection and simple tool measurements. The accuracy and reliability of these methods are heavily dependent on the operator's experience, leading to potential oversights or misjudgments, while also being time-consuming and inefficient [3, 4]. Additionally, for areas that are difficult to reach, large-scale inspection equipment is required, which not only adds to the bridge's load and causes traffic disruption, but also incurs high equipment costs and maintenance expenses [5]. To address these challenges, modern bridge inspections have gradually incorporated advanced technologies such as high-precision sensors [6–8], drones [9–11], and laser scanning [12–14]. Among these, drones equipped with high-precision sensors are particularly efficient for inspecting bridges, especially in high-altitude and hazardous areas. Real-time image data collected by drones significantly enhances both inspection efficiency and safety [15]. In addition to visual inspection technologies, nonvisual methods have also been applied to concrete defect detection. For example, Alhebrawi et al. [16] proposed an artificial intelligence-enhanced method for automatic identification of concrete cracks using acoustic impact hammer testing, providing a complementary technical approach for concrete defect detection.

Automatic image detection algorithms for bridge and pavement cracks are central to these technologies. However, due to the variety of capture scenarios and factors such as lighting, shooting angles, distances, and weather conditions, automatic crack detection algorithms for drone-captured images remain highly challenging. Even with the use of advanced deep learning techniques, many limitations still persist. While public datasets are available, they have been shown to differ significantly from actual drone-captured bridge crack images and fail to cover the full range of data distributions of real bridge images. Consequently, although models trained on public datasets perform well on corresponding test sets, they fail when applied to other bridge datasets. This is a typical domain adaptation issue, where the training data and test datasets represent only a subset of the data distribution in the real-world problem. Models trained on these datasets cannot be applied to other subsets of data in practical scenarios. Even for a specific bridge, it is not possible to capture images under all weather conditions or from various angles, and images taken at different times will have different data distributions. In such cases, if only the limited dataset collected is used for training the crack detection model, the model's generalization ability will be compromised.

Therefore, in the automatic analysis of drone-captured bridge crack images, challenges arise not only from deep learning technology but also from the limitations of drone photography itself. Drones have a wide range of perspectives, and the diversity of bridge environments poses difficulties for directly applying deep learning methods to crack detection. On one hand, safety concerns limit close-range drone imaging, leading to images that contain excessive irrelevant content such as the surrounding environment of the bridge. On the other hand, cracks are relatively small and sparse, making it difficult to clearly capture these small crack regions in the images. Furthermore, the real-time requirements of bridge crack detection and the scale of data processing (e.g., over 100,000 images for annual inspections of a single bridge) make traditional computing resources inadequate. This study addresses these challenges by combining distributed parallel computing (e.g., GPU cluster acceleration for U-Net training) with edge computing (lightweight model inference on UAVs), ensuring efficient operation in an HPC environment.

To address these challenges, this study presents an efficient solution for the automatic detection of bridge surface cracks in UAV-captured images. This approach reduces the cost of annotating large-scale crack datasets by combining deep learning techniques with traditional image processing methods. Specifically, the bridge crack detection process is broken down into three key steps: classification of bridge body image blocks, segmentation of damage features (including cracks) based on a mixed dataset, and classification and quantification of the segmented crack results. By distributing the annotation cost across these stages and leveraging damage data from other domains, the overall annotation workload is significantly reduced. The domain adaptation issue is effectively transformed into a classification task for bridge body images in the first step, which substantially reduces the effort required for manual annotations.

2 Background

Crack detection technologies can be classified into traditional methods and deep learning-based methods. Traditional digital image processing techniques are widely used in bridge inspection, mainly comprising two categories: 1) edge-based crack detection algorithms [17–19]. These algorithms detect edges by calculating image gradient or derivatives, with common methods including Roberts, Prewitt, Sobel, Canny, and Laplacian operators. A comparison between deep convolutional neural networks (DCNN) and common edge detection algorithms in the field of crack detection was conducted by Dorafshan et al. [17], showing that edge detection algorithms performed less accurately than DCNNs, while also retaining significant background noise. A major flaw of edge-based methods is their sensitivity to image noise and lack of autonomous learning ability, which results in poor generalization. 2) Threshold-based methods [20–23] involve preprocessing the image to remove noise and then distinguishing regions that may contain cracks from the background based on grayscale or color intensity differences by setting a threshold. Methods for thresholding include global, local, iterative, maximum entropy, and Otsu's thresholding. Vivekananthan et al. [22] conducted a study on concrete bridge

crack detection using an image processing technique that incorporates the improved Otsu method for thresholding. In their research, they employed a grayscale discriminant method for image preprocessing and used the Sobel filter to detect crack edges. Their method achieved a high accuracy of 95%, but it was limited by factors such as aspect ratio (RA) and margin parameters, leading to reduced generalization capability. Threshold-based methods often face the following issues: selecting an appropriate threshold is difficult, and different images require different thresholds, which severely limits the algorithm's generalization ability. Additionally, these methods are also highly sensitive to image noise.

Due to the aforementioned limitations of traditional image processing methods, deep learning-based methods [24–27] have gradually become mainstream. Deep learning can autonomously learn image features and link classification or recognition results with feature extraction, leading to better generalization. Kao et al. [28] further verified the feasibility of this direction by combining the YOLOv4 model with UAV imagery processing technology, successfully realizing the extraction and quantization of bridge cracks—their work highlights the practical value of integrating UAV and deep learning in bridge crack detection scenarios. However, bridge crack detection faces challenges such as large data scales (TB-level images), complex model training (e.g., U-Net with millions of parameters), and real-time requirements. Traditional single-machine computing resources cannot meet these needs. High-performance computing (HPC) addresses these challenges through distributed parallel processing (e.g., GPU cluster-accelerated training), memory optimization (e.g., Apache Spark for data preprocessing), and lightweight inference (e.g., TensorRT deployment), significantly improving detection efficiency and practicality—providing viable technical support for large-scale bridge health monitoring. Tran et al. [25] proposed a deep learning method based on YOLOv7 and U-Net for crack detection, evaluating five deep learning algorithms for this task. Their approach effectively handles complex cracks while maintaining high detection accuracy. The experiments showed that the proposed method achieved high accuracy in detecting crack length, width, and classification. However, this method demands large annotated datasets and substantial computational resources. Laxman et al. [26] developed a comprehensive automated crack detection and depth evaluation framework capable of detecting crack locations and predicting depths. However, their approach lacked robustness in complex environments. Ma et al. [24] introduced a deep learning-based "dual detection + single segmentation" method for coarse and fine bridge crack detection. This method used YOLOv5(x) for fine crack detection and employed the U-Net model to segment accurately detected cracks. While this approach improved both accuracy and efficiency, the network structure was complex, demanding high computational resources and reliant on high-quality training data.

While deep learning-based methods can enhance crack detection performance, they require large-scale datasets, incur high annotation costs, and involve complex networks with high computational resource demands. To achieve strong generalization ability, large-scale datasets containing various bridges and cracks must be collected. However, such datasets are difficult to gather and annotating them is costly, making this approach practically unfeasible. Currently, publicly available datasets,

especially those consisting of drone-captured bridge images, are limited in quantity and small in scale, making them inadequate for training complex models. Furthermore, these datasets focus on specific crack types and scenes, which can lead to model overfitting or poor generalization. Additionally, deep learning methods face significant limitations in interpreting and measuring crack sizes [28–34].

To tackle the challenges outlined above, this study proposes an effective automatic detection method for bridge surface cracks, integrating deep learning techniques with traditional image processing approaches. The method employs a progressive domain adaptation strategy, reducing labeling costs while ensuring high interpretability, thus offering a practical solution for efficient crack detection in bridge maintenance.

3 Proposed method

To address the challenges associated with bridge surface cracks detection, this study proposes a UAV-based recognition system that integrates deep learning and traditional image processing techniques. The image segmentation and model training stages of the system rely on an MPI + OpenMP hybrid parallel architecture, which distributes large-scale image data across multiple computing nodes for processing, significantly reducing iteration cycles. For example, training the VGG16 classification model on a 16-node GPU cluster is approximately 12 times faster than training on a single machine. The method is divided into three main stages: adaptive segmentation of drone-captured bridge images, multi-type crack recognition including crack segmentation, and crack quantification and classification.

In the first stage, overlapping tiling technology is used to process high-resolution images of bridge surfaces captured by UAVs. This technology divides large images into smaller, more manageable blocks, making it easier to analyze specific areas of the bridge in detail. Overlapping tiling not only improves processing efficiency but also enhances the accuracy of crack detection. A bridge classification model is applied to filter out irrelevant background elements such as sky and water, ensuring that detection focuses only on bridge structures. A CNN-based classification model is used for this task because of its simple structure, mature training process, and efficient optimization techniques, which help reduce annotation costs and minimize the impact of irrelevant background information.

In the second stage, a mixed dataset, combining publicly available crack datasets with the bridge-specific image data, is used to train a crack segmentation model. This mixed dataset improves the model's robustness in various environmental conditions, enabling the system to detect cracks in diverse settings. The crack segmentation is performed using deep learning methods, specifically the U-Net network, which excels at pixel-level segmentation tasks and ensures precise identification of cracks and other types of damage.

The third stage involves classifying and quantifying the detected cracks. A custom seed-point growth method is used to measure crack length and width accurately, allowing for efficient quantification of crack dimensions. Additionally, different crack types (e.g., hairline cracks and severe cracks) are classified to provide a

detailed report for maintenance personnel. The crack detection results are integrated into a comprehensive reporting system that outputs detailed crack information to support bridge maintenance and management decisions.

The proposed system architecture is shown in Fig. 1. During model training, the datasets used consist of both publicly available datasets and images collected from the field. The primary objective is to improve detection accuracy and adaptability through the integration of deep learning with traditional image processing methods, leveraging a progressive domain adaptation strategy to reduce labeling costs and enhance model performance.

3.1 UAV bridge image tiling and classification

To address the data collection challenge, this study adopts a solution that involves separate data collection for each bridge. First, periodic inspections of each bridge require data collection, which is a routine operation. Second, during this phase, data annotation is limited to selecting images containing the bridge body from the image blocks, making the annotation process cost-effective, with a relatively small dataset that can be completed in a short period. Finally, the deep learning models trained for each bridge can be reused for detecting crack defects in other bridges. As the amount of data increases with repeated inspections, the performance of the model can continuously improve over time. The process of constructing bridge image data and training the classification model is shown in Fig. 2.

First, the collected dataset undergoes overlapping segmentation. Specifically, the raw bridge images captured by the drone (as shown in Fig. 3a) are cut into 448×448 pixel-sized blocks to facilitate mixing with public datasets for training, with a 50% overlap between adjacent blocks. This overlap ensures that any part of the bridge body potentially included in the image will not be omitted due to the segmentation

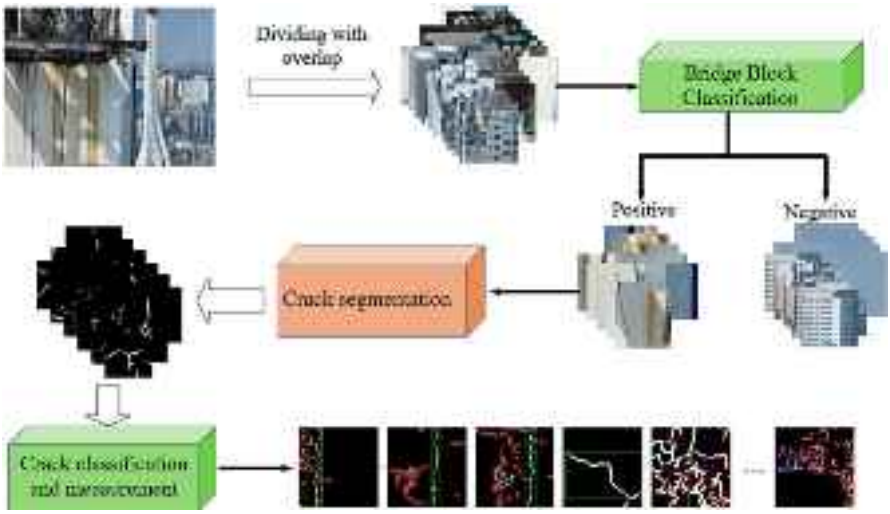

Fig. 1 The flowchart of the proposed solution scheme

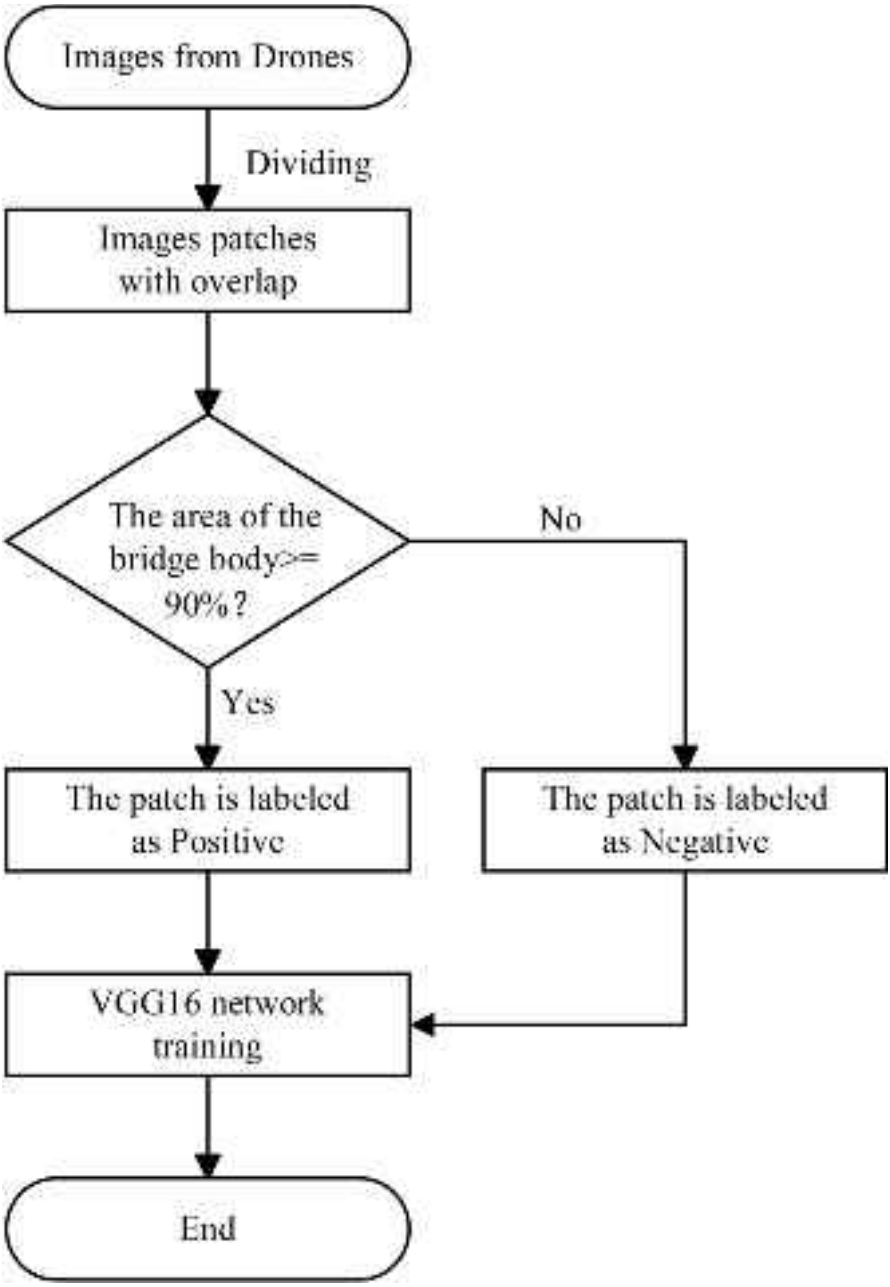

Fig. 2 Process of constructing the bridge image block dataset and training the classification model (Note: Tiling refers to dividing original images into overlapping blocks; classification refers to labeling bridge/non-bridge blocks)

Fig. 3 Example of drone-captured bridge images: **a** Bridge image, **b** Tiling example

process. The resulting segmented dataset is shown in Fig. 3b. Afterward, the segmented images are annotated: images containing 90% or more of the bridge body area are labeled as positive samples with a label of 1, while images containing less than 90% of the bridge body area are labeled as negative samples with a label of 0. This annotation method allows for rapid manual visual inspection, keeping the annotation cost low. For the segmented dataset, human annotators can quickly distinguish positive and negative samples. An experienced annotator can generally classify a certain number of images within a specific time frame, depending on factors such as image complexity and annotation requirements.

Finally, the dataset is used to train the bridge image block classification network. In this study, the VGG16 classification model [35] is employed, and its network architecture is shown in Fig. 4.

Before training the model, this study applies data augmentation techniques, such as adjusting hue, image saturation, and exposure, to ensure that the model has strong

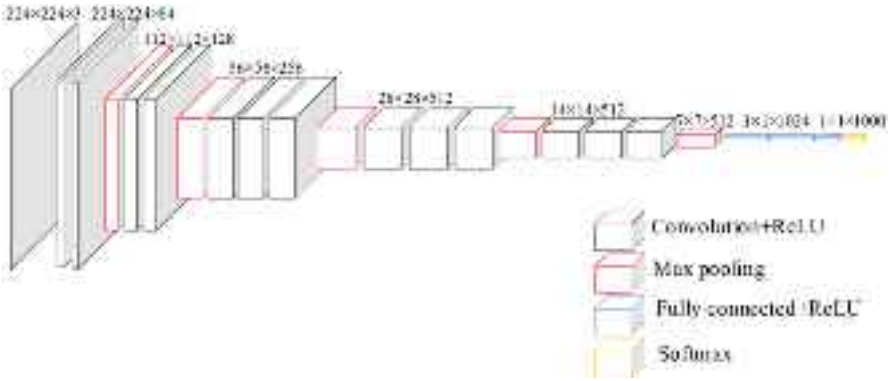

Fig. 4 VGG16 network architecture diagram

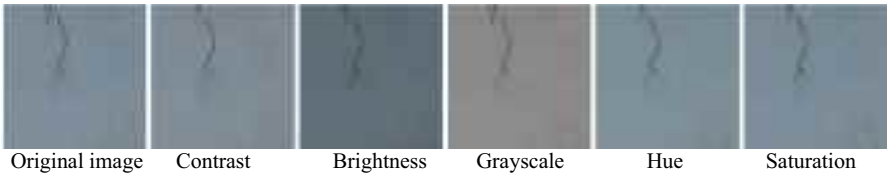

Fig. 5 Image augmentation effects

generalization ability under various conditions. The illustration of the image augmentation effects is shown in Fig. 5.

3.2 Bridge crack damage image segmentation based on a mixed dataset

In Sect. 3.1, the image tiling process decomposes the complex problem of bridge crack detection and segmentation into two tasks: classification of bridge region image tiles and semantic segmentation of cracks within these bridge region image tiles. For the segmentation (pixel-level outlining of crack contours via U-Net) of cracks, which focuses on various types of cracks, a mixed dataset was utilized. This approach creates a more advantageous set for training the segmentation model, as shown in Fig. 6. Figure 6a shows a publicly available crack image dataset, while Fig. 6b displays the self-collected bridge tiles containing cracks. As can be seen, by incorporating different publicly available datasets into the semantic segmentation process of crack images, the trained segmentation model achieves better domain adaptation, i.e., stronger generalization ability. This mixed dataset effectively expands the diversity of crack images and smoothly transitions the algorithm from

Fig. 6 Comparison of examples from the public dataset and the proprietary dataset. **a** Crack examples from various public datasets. **b** Crack examples from the self-constructed dataset

the first stage, where it is limited to classifying a specific bridge, to the second stage, where the model has a better generalization ability for crack segmentation.

Since the publicly available bridge crack disease datasets already include annotations, only the block images with cracks in the self-collected dataset need to be labeled for semantic segmentation, greatly reducing annotation costs. To further improve efficiency, this study adopts a "label first, then block" approach. Specifically, the cracks in the images captured by the drone are annotated for semantic segmentation first, and then, in parallel with the image blocking in the first stage, the segmentation results are also divided into blocks. This method effectively avoids redundant labeling of overlapping regions in the blocks.

For bridge crack semantic segmentation, this study uses the U-Net network, a widely adopted semantic segmentation model with excellent performance. The U-Net network structure is shown in Fig. 7.

3.3 Crack-type classification and measurement.

In practical applications, detecting bridge crack defects requires measuring and evaluating both the type and size of the cracks. For this purpose, cracks are categorized into two main types in this study: "crazing" cracks and "non-crazing" cracks. Crazing cracks are defined as large, irregular block-like regions with at least three intersection points (based on skeleton extraction). Non-crazing cracks include two subtypes: (1) "conventional cracks" (single elongated shapes with no more than two intersection points); (2) "mixed cracks" (combinations of partial conventional segments and partial crazing-like segments, but with fewer than three intersection points in total). More commonly, cracks are observed to exhibit a twisted and irregular pattern.

Additionally, due to factors such as the shooting angle, lighting conditions, and other environmental influences, segmented cracks often exhibit fragmentation. Cracks can also display various forms, including twisting, branching, and interweaving. Consequently, methods based on connected components, which are typically used to extract or count individual cracks under ideal conditions, are limited in their effectiveness.

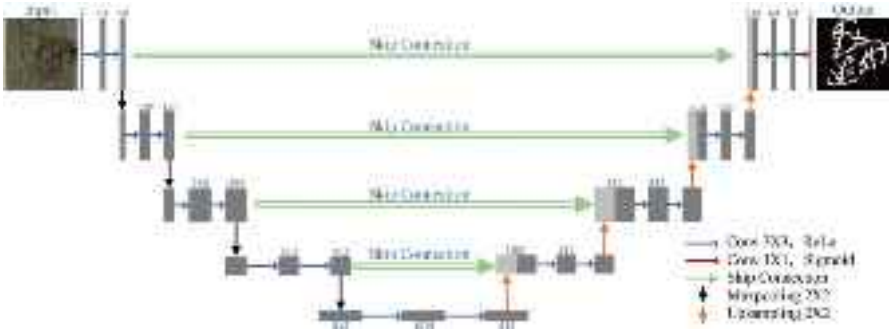

Fig. 7 U-Net network architecture used for bridge crack segmentation

To differentiate between crazing and non-crazing cracks, this study employs traditional image processing techniques based on the distinct characteristics of each crack type. This approach requires less computational effort, does not necessitate large-scale data for training or annotation, and offers good interpretability.

As shown in Fig. 8, by comparing the characteristics of crazing and non-crazing cracks, it becomes evident that crazing cracks tend to exhibit multidirectional, irregular forms, which divide the crack region into several small, independent areas of varying sizes. The boundaries between these small regions are often blurred, adding complexity to the overall structure. In contrast, normal linear and mixed-type cracks display different characteristics. Although these cracks may also form cavities, they generally occur less frequently and have a more limited coverage area. Therefore, identifying and classifying these two types of cracks is crucial in the crack analysis and processing stages.

Based on the analysis of the aforementioned crack characteristics, this study proposes an effective method for distinguishing between crazing and non-crazing cracks using traditional computer vision techniques. Specifically, the binary crack images obtained after segmentation undergo a hole-filling operation, followed by the subtraction of the original binary image to isolate the distinct void regions. The hole-filling operation, a core step here, is implemented based on the morphological reconstruction algorithm—a method that has been systematically elaborated in relevant research—with core operations such as dilation. Its specific process is as follows: First, create a mask by inverting the input binary image of crazing cracks (with background as 0/black and foreground as 255/white), i.e., mask = 255 – image, which implicitly identifies void regions in the original image. Then, initialize the marker image: create a completely black image "marker" with the same size as the input image and set its four surrounding boundaries to white. Next, perform morphological reconstruction with a 3*3 convolution kernel A, repeating the iterative operation until the marker no longer changes—this involves saving the marker as marker_pre, dilating the marker with the kernel to get dilation, retaining the minimum pixel value to update marker as min(dilation, mask), and stopping iteration when marker matches marker_pre. Finally, invert the marker image to obtain the filled image result = 255 – marker. This filled image reduces the interference of void regions on crack contours, making the morphological features of crazing cracks

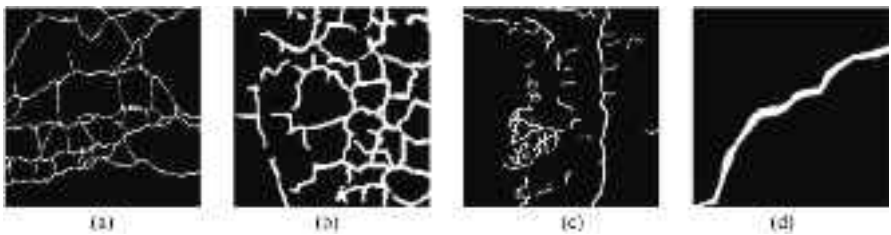

Fig. 8 Examples of crack segmentation results from the public and self-constructed datasets: **a** crazing cracks from public datasets **b** crazing cracks from the self-built data set **c** hybrid cracks **d** normal linear crack

more complete, which lays a foundation for subsequent crack feature extraction and classification.

As shown in Fig. 9, panel (a) displays the binary image of a crazing crack (with ≥ 3 intersection points) after segmentation, the image after hole filling, and the final void area image; panel (b) shows the binary image of a non-crazing normal linear crack (with ≤ 2 intersection points) after segmentation, the image after hole filling, and the final void area image. To clearly correspond to the content in Fig. 9, the specific process of image transformation and void counting is as follows: Take the input binary image I of crazing cracks, obtain the filled image I' through the above hole-filling operation; then subtract the original binary image I from I' to get the void region image ΔI; finally, count the number of connected regions in ΔI, which is the regional void number n. This process directly links each step of the operation to the image content in Fig. 9, making the generation process of the void area image and the statistical basis of the void number more intuitive.

The experimental results clearly show that, in the case of non-crazing cracks, only two significant voids are present, whereas seven voids are observed in crazing cracks. This marked difference provides a reliable basis for distinguishing between the two crack types, enabling further classification through appropriate thresholding. This approach not only improves the accuracy of crack analysis but also offers valuable insights for subsequent structural evaluation and maintenance.

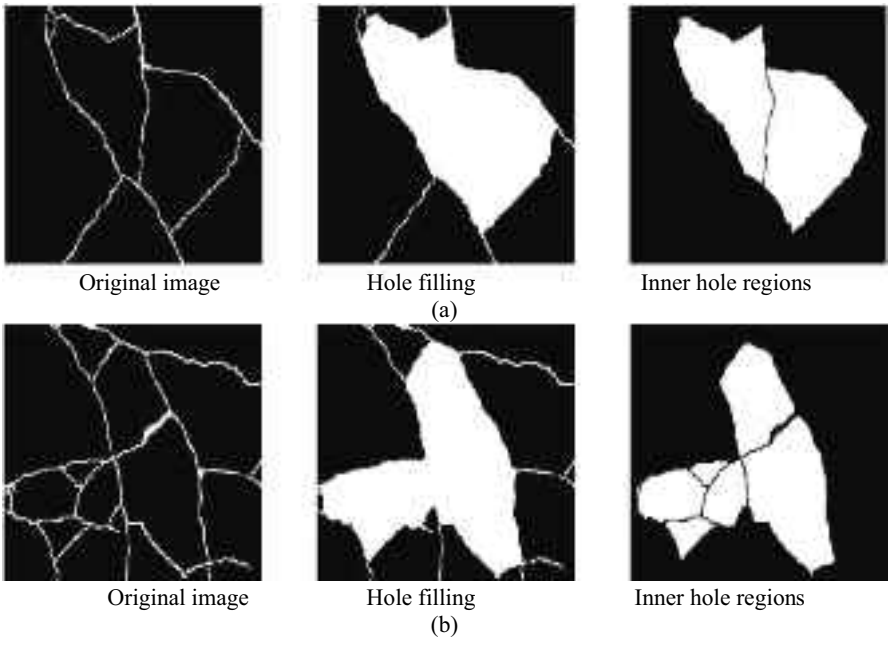

Fig. 9 Segmentation and post-processing results of different crack types. **a** Crazing crack (≥ 3 intersection points); **b** Non-crazing normal linear crack (≤ 2 intersection points). All examples are consistent with the crack-type definitions in Sect. 3.3

It should be noted that this study does not quantify crazing cracks. Crazing cracks are distributed in a crisscross pattern with varying widths and lengths, as shown in Fig. 8a, b. In addition, some craze-like local small areas (such as the one in Fig. 8c) are small in scope, and their measurement value is limited, as indicated by the detection results of the algorithm in this paper (marked by the red box in the attached figure). Instead, this study focuses on quantifying non-crazing cracks (especially normal linear cracks as shown in Fig. 8d), which are long strip-shaped with relatively uniform widths.

3.3.1 Distinguishing mixed-crack regions based on the characteristics of different crack types

In the non-crazing category, two subtypes are clearly distinguished based on intersection counts and morphology: (1) "normal linear cracks" (single, elongated, with ≤ 2 intersections); (2) "mixed cracks" (combinations of normal linear and crazing-like segments, with total intersections < 3). These align with the four types analyzed earlier: (I) curved small region cracks (subset of normal linear cracks), (II) small sheet-like cracks crazing, (III) crazing cracks (excluded from non-crazing), and (IV) normal linear cracks (core subtype of non-crazing). This study further distinguishes the various types of cracks within the mixed-crack regions by analyzing characteristics such as crack size and morphology. The specific algorithm is outlined in Table 1.

Due to the small area of curved small region cracks, this study utilizes a connected component algorithm to identify individual crack regions. The boundary coordinates of each region are used to derive the bounding rectangles. By calculating the ratio of the bounding rectangle area to the region area, the proportion of the crack within the region can be determined. A threshold is set to classify cracks with a ratio smaller than the threshold as curved small region cracks, which are then marked with a red box. Cracks with a larger ratio undergo further classification.

Table 1 Interpretable classification algorithm for multi-type bridge cracks

Input: Binary image of non-crazing cracks

Process:

1. Record the area of the region S. Use the connected component method to filter out each crack region and draw a bounding box for each crack. Record the area of each bounding box within the crack region as $S1$ and the area of the crack as $S2$;
2. Set threshold *threshold1* to determine whether the crack is a **curved small region crack**:
   1) If $S1/S <$ *threshold1*, classify it as a **curved small region crack** and label it with a red box.
   2) If $S1/S \geq$ *threshold1*, proceed to the next classification.
3. Set threshold *threshold2* to determine whether the crack is a **sheet-like crazing crack**:
   1) If $S2/S1 \geq$ *threshold2*, classify it as a **sheet-like crazing crack** and label it with a yellow box.
   2) If $S2/S1 <$ *threshold2*, proceed to the next classification.
4. For the remaining cracks, perform skeletonization and search for intersection points. Count the number of intersection points $S3$. Set threshold *threshold3* to classify as **crazing cracks** or **normal linear cracks**:
   1) If $S3 \geq$ *threshold3*, classify it as a **crazing crack** and label it with a blue box.
   2) If $S3 <$ *threshold3*, classify it as a normal linear crack and label it with a green box.

Output: The classified and labeled crack image.

Note: Parameters threshold1, threshold2, and threshold3 are 0.1, 0.2, and 0.3 respectively, which are empirical parameters and can be optimized and determined using the training set.

For plate-like cracks resembling crazing, these cracks typically occupy a larger area within the region but often display a curled shape. As a result, they cover a significant portion of the bounding rectangle area. The present method maps the curling and divergence states of these cracks by analyzing the ratio of crack area to bounding rectangle area. A threshold is set to distinguish crazing-like plate cracks; those above the threshold are classified as crazing-like cracks and highlighted with a yellow box, while those below the threshold are classified as either crazing cracks or normal linear cracks in the final classification step.

For the remaining crazing-type and normal linear cracks, in addition to the void count method described above, the number of intersection points can also be used as a distinguishing feature. Crazing-type cracks typically exhibit both horizontal and vertical intersections, leading to a higher number of intersection points. On the other hand, normal linear cracks are mostly single horizontal or vertical cracks, and even if there are a few branches, the number of intersection points is generally low. Based on this characteristic, this study extracts the crack skeletons, detects the intersection points, and records their count to differentiate between crazing-type and normal linear cracks.

Given the irregular extension of cracks, the skeleton extraction in this study utilizes the Zhang–Suen algorithm. The Zhang–Suen skeletonization algorithm [36] is an efficient image processing technique, particularly well suited for extracting crack skeletons. Its advantage lies in accurately capturing fine line structures while preserving shape consistency and connectivity. The algorithm works through iterative processing steps, effectively removing redundant pixels to generate a refined skeleton. This not only reduces computational complexity but also improves result accuracy. Moreover, the Zhang–Suen algorithm is highly robust to noise and small area interference, ensuring stable skeleton extraction in various environments. This provides a reliable data foundation for subsequent crack analysis and classification. Figure 10a and b show the binary images of crazing-type cracks and their extracted skeletons.

Figure 10c and b displays the skeletons and intersection point annotations for typical crazing-type and normal linear cracks. It can be observed that the number of intersection points in crazing-type cracks is significantly higher than in normal linear cracks. This characteristic can effectively differentiate between crazing-type and

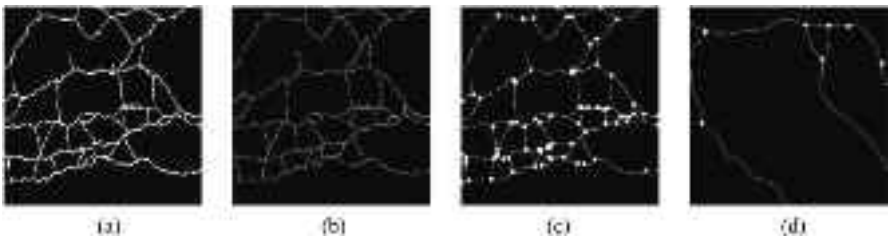

Fig. 10 Processing and intersection counting results of crack images. **a**–**c** show the processing steps of a crazing crack sample (≥ 3 intersection points); **d** shows the intersection counting result of a non-crazing crack sample (≤ 2 intersection points) for comparison, to verify the effectiveness of the intersection counting method

normal linear cracks, with blue boxes marking the crazing-type cracks and green boxes highlighting the normal linear cracks.

3.3.2 Custom seed growth method for measuring crack length and width

The flowchart of the crack length measurement algorithm based on the seed generation technique is shown in Fig. 11. The process begins by extracting the endpoints of the crack. Then, a seed point is selected, and growth starts from this point. Growth continues until a vertex is encountered, at which point the number of growth iterations is recorded. A new round of growth starts from the next seed point, repeating this process. After traversing all vertices, the longest distance measured corresponds to the crack's maximum length.

After classifying the normal linear cracks, the proposed skeleton extraction and length calculation method for normal linear cracks aims to improve the accuracy and efficiency of crack analysis. First, the normal linear crack region is extracted, and the crack is isolated using the largest connected component. Then, the Zhang–Suen thinning algorithm is applied to extract the thinned skeleton of the crack image (as shown in step IV of Fig. 11). During this process, the coordinates of key vertices on the crack skeleton are recorded. These vertices serve as important geometric feature points for the subsequent growth algorithm, providing essential data.

Among all the extracted vertices, since the structure of normal linear cracks is relatively simple, to reduce computational load, the upper-left corner endpoint

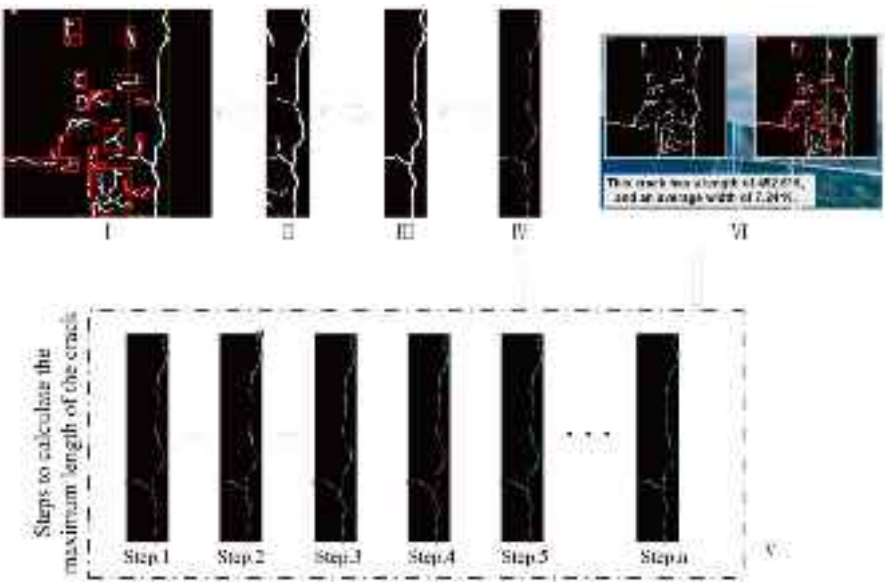

Fig. 11 Flowchart of the crack length measurement algorithm based on the seed generation technique: (I) crack classification based on the connected region measure, (II) candidate cracks filtering, (III) the normal crack extraction according to the maximum connected region, (IV) skeleton extraction, (V) the longest length measurement process, (VI) classification and measurement result visualization

(the closest point to the leftmost vertex) is selected as the seed point for growth (as shown in step 1 of Fig. 11V). Then, using the seed-point growth algorithm, the seed point is expanded toward other vertices. The growth stops when it encounters a vertex. Specifically, when the seed point starts growing from one vertex to another, the growth process pauses once it reaches the next vertex. This process is executed by progressively checking neighboring pixels or adjacent vertices to ensure that each growth step follows the connected parts of the skeleton. Once paused, the current iteration count is recorded and accumulated, and the growth continues. By repeatedly performing this process, all vertices are traversed, and the maximum number of growth steps is recorded.

Since the skeleton of normal linear cracks has a width of only one pixel, the number of growth steps represents the distance between two vertices. When the longest growth steps are counted, the maximum length of the normal linear crack is obtained. The average width of the crack is calculated by dividing the crack area by the skeleton length. The specific process is shown in step V of Fig. 11. Using this method, cracks of various types within the mixed-crack region can be classified, and the length and average width of normal linear cracks can be measured. It should be noted that the current measurement results are based on pixel units, and conversion to physical units (e.g., millimeters) requires image calibration (related to UAV shooting distance and camera parameters). In subsequent work, we will supplement the calibration process and conduct comparative experiments with manual measurement data to quantify the measurement accuracy (e.g., mean absolute error, relative error).

3.4 Scheme comparison

Normal linear crack detection methods are typically implemented in two steps: first, deep learning algorithms (such as YOLO, Faster R-CNN, and ResNet50) are used to complete crack identification and detection. After meeting engineering accuracy requirements, segmentation models (such as U-Net and dilation-erosion algorithms) are applied to segment crack regions, providing a basis for subsequent feature value extraction. Taking the YOLO detection algorithm series as an example, this paper briefly introduces normal linear detection methods and compares them with the proposed scheme to highlight the innovations of the proposed scheme.

For the YOLO detection algorithm, since there is currently a lack of training sets specifically for bridge crack detection, it can only use transfer learning. That is, training weights from large datasets, such as COCO, are selected as initial weights for iterative training. Before training with the YOLO algorithm, it is necessary to use professional annotation tools (such as Labelme and Labelimg) to label regions of interest in existing images and store the annotation content in the form of txt files. It should be noted that during annotation, the smallest possible bounding box should be used to frame the cracks to reduce the interference of background information on detection accuracy, ensure that each bounding box contains only one crack, and try to avoid overlapping between bounding boxes to reduce the impact of cross-interference between targets on algorithm learning.

Fig. 12 YOLO target annotation: **a** Excessively large annotation range, **b** Excessively small annotation range, **c** Standard annotation

Figure 12 shows typical cases of YOLO target annotation. Among them, Fig. 12a and b present 2 common annotation problems: one is that the annotation box range is too large, containing too much background information; the other is that the annotation box range is too small, failing to fully cover the crack region. Both cases will adversely affect the algorithm training effect. Only the annotation method shown in Fig. 12c is standardized. When using the standardized annotation method shown in Fig. 12c, it takes about 5–8 min to repeatedly classify and annotate 200 crack regions. However, by applying the first and third-stage image classification methods designed in this paper, the same crack classification effect can be achieved, and it only takes 2–3 min to batch annotate 200 images. It can be seen that the method in this paper can significantly improve annotation efficiency (saving about 60%–75% of time) and greatly reduce manual annotation costs.

Since normal linear processes also include image segmentation models, only a brief comparison is made between image classification methods and YOLO-like target detection algorithms. It should be noted that these two types of models are essentially different in detection targets, annotation processes, and evaluation systems. Therefore, only a summary comparison and analysis are conducted from the perspectives of parameter quantity and theoretical expectations [40].

According to the comparative analysis results in Table 2, although VGG16 is slightly inferior to the other three algorithms in terms of detection precision, it has the advantage of achieving acceptable detection precision with significantly reduced computational costs and annotation costs. Later, as the dataset continues to increase and the number of training epochs increases, VGG16 can also achieve high precision (as shown in Table 5). Therefore, considering computational efficiency,

Table 2 Algorithm comparison

Algorithm	Expected precision	Expected recall	Params/10^6
YOLOv5s	Medium–high	Medium–high	137.7
SSD	Medium	Medium	62.79
Faster R-CNN	High	High	12.1
VGG16	Medium	Medium	3.47

resource consumption, and precision performance comprehensively, VGG16 is obviously more suitable for this study.

In addition, this study further compares U-Net with advanced segmentation models like TransUNet [41] to clarify its selection rationale. The research by Jin et al. shows that TransUNet outperforms U-Net in mIoU and recall but has lower precision (the proportion of correctly detected crack pixels among all detected pixels), which means it is more likely to generate false positives. In practical bridge inspection scenarios, such false positives will increase the workload of manual screening, which is not conducive to efficient maintenance. In terms of computational efficiency, U-Net (24.89 M parameters, 99.6 MB model size, 86.45 G FLOPs) is significantly lighter than TransUNet (93.23 M parameters, 423.1 MB model size, 197.81 G FLOPs), which can save more computing resources.

Tests on actual bridge datasets (including UAV-captured images in this study) show that U-Net has better stability. TransUNet relies on large-scale pre-trained weights and requires sufficient training data to exert its performance advantages; otherwise, it may fail in the segmentation of individual images. Meanwhile, the adaptive small-image block segmentation strategy adopted in this study can compensate for the limitation of U-Net in capturing small details when processing large images. Moreover, U-Net's segmentation results have better crack connectivity—this is critical for subsequent skeleton extraction and length measurement. Although its segmentation results may be slightly thicker, this problem can be optimized through morphological post-processing algorithms. Thus, considering practical application performance, computational efficiency, and compatibility with the overall framework, U-Net is more suitable for this study.

The system uses a serial workflow with interdependent stages: image tiling prepares images, crack segmentation provides contours, and seed-point measurement quantifies dimensions. This interdependence prevents isolating components for ablation testing, but each stage's contribution lies in maintaining workflow integrity—omitting any stage would disrupt detection.

4 Experimental results and analysis

4.1 Overview of the bridge inspection project

The bridge involved in this study is the Yiling Yangtze River Bridge, located in the urban area of Yichang City, Hubei Province, China. It is an important urban bridge that connects the northern and southern banks of the Yangtze River, as shown in Fig. 13. The total length of the bridge project is 3246 m, consisting of the main bridge, the northern approach bridge and interchange, and the southern approach bridge and interchange. The main bridge is 936 m long and is a single-cable surface concrete stiffened girder three-tower cable-stayed bridge with a width of 23 m. The central span measures 348 m, and the towers are symmetrically arranged. The main girder has a single-box, three-chamber inverted trapezoidal cross section and is constructed with a three-way prestressed concrete structure. The height of the main girder is 3.0 m across the entire bridge.

Fig. 13 Photograph of the Yiling Yangtze River Bridge

The Yiling Yangtze River Bridge was completed and opened to traffic on December 28, 2001. After 22 years of operation, advanced inspection technologies, including bridge health monitoring systems, drone inspections (Fig. 14), and traditional manual inspections, have been used to thoroughly document the bridge's health status and identify potential risks in a timely manner. Due to the fact that the stay cables have exceeded their design service life, certain areas of the main bridge have shown varying degrees of damage, particularly cracks, corrosion of the steel cables, and damage to the bridge deck. To ensure the bridge's long-term safe operation, relevant authorities undertook the replacement and reinforcement of the stay cables. As the first steel strand single-cable surface concrete stiffened girder three-tower cable-stayed bridge in the country, the Yiling Yangtze River Bridge completed the replacement of 236 stay cables in just 10 months, making it the first bridge in the country to replace all its steel strand cables after 23 years of safe operation.

In this study, experiments are conducted using a combination of self-collected data and publicly available datasets. The self-collected data primarily serves to validate the effectiveness of the proposed solution, specifically through a small amount of labeled data to achieve incremental domain adaptation. This ensures that the

Fig. 14 The drone used for inspection

segmentation and classification models, trained on publicly available datasets, can be adapted to bridge images from various real-world scenarios.

The publicly available datasets used in this study include the *Concrete-and-pavement-crack-dataset* and *Concrete-crack-images-for-classification* (shown in Table 3 and Fig. 6a). The self-collected dataset was acquired using a drone, the Matrice 350 RTK, during the maintenance of the Yichang Yiling Yangtze River Bridge. The images were captured with overlapping tiles (shown in Fig. 6b).

The software environment used for the entire experiment consists of Windows 10, PyCharm, CUDA 10.2, and cuDNN 10.2. The hardware setup includes a GTX 1080Ti GPU, an Intel Xeon ES-2460 CPU, and 16 GB of RAM, supporting multi-task parallelism and accelerating tensor operations through CUDA optimization. This configuration enables parallel processing of image segmentation and model inference, laying the foundation for scaling to larger HPC clusters.

4.2 Bridge image block classification experiment results and analysis

The training parameters for the VGG16-based bridge image block classification model are shown in Table 4. The classification experiment used a bridge dataset that was collected and processed through tiling (divided into 448×448 pixel blocks) into smaller image blocks, as shown in Fig. 4. The dataset was split into training, testing, and validation sets with an 8:1:1 ratio, resulting in 12,997 images for training, 1625 images for testing, and 1624 images for validation. After 100 epochs of training, the classification accuracy on the test set was 99.3%.

Some misclassified examples are shown in Fig. 15. During the labeling process for bridge and non-bridge images, the classification was manually based on the proportion of the bridge in the image: images with a bridge body area accounting for more than 90% are labeled as bridge images, and others as non-bridge images. The misclassified images in Fig. 15 are mainly those where the bridge area is close to 90%. As mentioned in the figure caption, the misclassification is mainly due to two reasons: first, the judgment of whether the bridge body area reaches 90% depends on the subjective visual perception of annotators; second, the limited size of the current dataset resulted in the model not learning enough about area-based features. Although these images were misclassified, they did not impact the subsequent steps in the pipeline.

4.3 Bridge crack damage image segmentation results and analysis

To demonstrate the impact of the dataset on the results, this study tested the U-Net model using both a mixed dataset and a self-collected dataset. The self-collected dataset consists of 11,812 images, while the mixed dataset is composed of 11,812 self-collected images and 11,298 public dataset images. The datasets were divided into training, validation, and test sets at a 6:3:1 ratio. The U-Net model was trained on the mixed dataset and tested on the self-collected dataset for detection, while the model trained on the public dataset was used to detect both the public and self-collected datasets. Notably, training on the mixed dataset in an HPC environment took

Table 3 Details of public datasets

Name	Description	Size	Type	Image dimensions	Application scenario
RDD2022	Original and annotated images of road cracks captured by drones and dashcams	47,420	Original image and object detection	512*512	Pavement disease detection and classification
Structural-Defects-Network (SDNET) 2018	Images of structural defects	55,400	Positive and negative classes	256*256	Building defect detection and classification
CrackForest-dataset-master	Image dataset for training and testing crack detection models	156	Semantic segmentation	480*320	Crack detection and machine learning
DeepCrack	Concrete surface crack images	537	Semantic segmentation	512*512	Crack detection and machine learning
LCW(Concrete Crack Detection)	High-resolution images of bridge cracks	3,817	Semantic segmentation	512*512	Crack detection and deep learning
Crackseg9k [37]	Road, bridge, and concrete multi-scene crack images	9,255	Semantic segmentation	400*400	Crack detection and classification
GAPs384	Crack images from various scenes	384	Semantic segmentation	1024*763	Crack segmentation and deep learning
AEL (Active Learning Crack Evaluation Dataset)	A large dataset with cracked and non-cracked samples	20,000	Positive and negative classes	Multiple sizes	Active learning and crack detection
CFD (Crack Forest Dataset)	Crack images and their segmentation annotations	118	Semantic segmentation	480*320	Crack detection and segmentation
CrackTree200	Road crack images with precise crack segmentation annotations	206	Semantic segmentation	800*600	Crack detection and segmentation
Crack 500	Concrete crack images	500	Original image	480*320	Crack detection

Table 3 (continued)

Name	Description	Size	Type	Image dimensions	Application scenario
Crack Dataset [38]	Crack and non-crack images	56,000 (Positive and negative classes) 22,600 (Semantic segmentation)	Positive and negative classes, Semantic segmentation	256*256 (Positive and negative classes) 448*448 (Semantic segmentation)	Crack detection and image classification
CrackVision12K [39]	Original crack images and binary images	12,000	Semantic segmentation	256*256	Crack detection and segmentation

Table 4 Hyperparameters of the classification network

Parameters	Value
Batch_size	100
Epoch	100
Optimizer	Adam
Learning rate	0.0001

Fig. 15 Misclassified examples of bridge and non-bridge images (Note: Classification is based on whether the bridge body area in the image accounts for more than 90%; misclassification is mainly due to subjective judgment of annotators and insufficient learning of area features by the model due to limited dataset size)

Table 5 Comparison of detection performance metrics

	mioU	mPA	mPrecision	mRecall
Training on public dataset, testing on public dataset	78.84%	86.61%	86.99%	86.81%
Training on public dataset, testing on custom dataset	58.99%	64.67%	72.42%	64.67%
Training on mixed dataset, testing on custom dataset	76.97%	86.08%	84.99%	86.08%

only 6 h (compared to 72 h on a traditional single machine), with an 18% improvement in mIoU. This validates the necessity of distributed computing for large-scale image processing. The segmentation results (pixel-level crack outlining) are shown in Table 5 (Note: Segmentation is a key step in the overall crack detection process).

From Fig. 16, it can be seen that after training the model with the mixed dataset, the comparison of various metrics, including mIoU (mean intersection over union), mPA (mean pixel accuracy), precision, and recall, shows that the model trained on the public dataset performs well on the corresponding test set. However, when applied to other bridge datasets, the model fails to detect, which is a typical domain adaptation issue. The training and test datasets are only a subset of the data distribution in the real-world problem, meaning that a model trained on them cannot be applied to other subsets of data in real-world scenarios with different distributions.

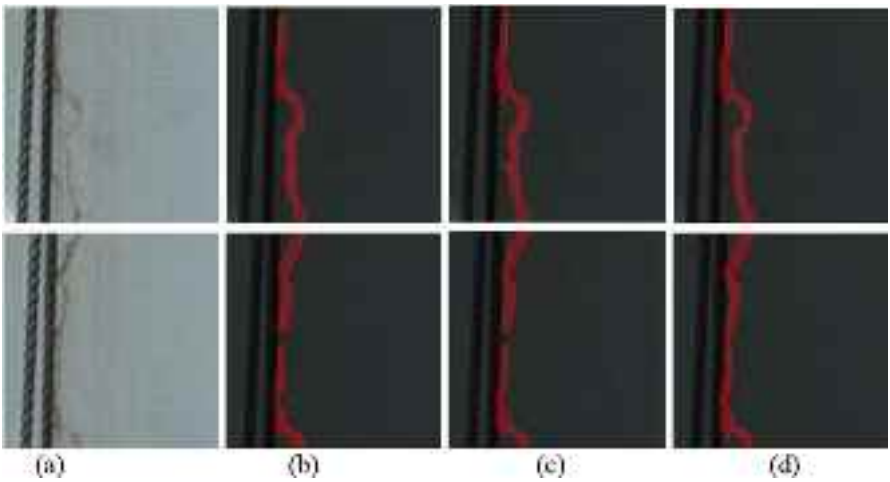

Fig. 16 The comparison of segmentation results output by the models trained on the hybrid dataset and the self-collected dataset: **a** original images, **b** segment results based on public crack dataset, **c** segment results based on the private crack dataset, **d** segment results based on the hybrid dataset

Moreover, the model trained with the mixed dataset clearly outperforms the one trained only on the public dataset. Therefore, this study uses a mixed dataset of public and custom data to train the model. On one hand, this approach avoids overfitting caused by the custom dataset, and on the other hand, it allows the model to be effectively adapted for detecting any bridge-related issues. Figure 16 shows the results after testing with the mixed dataset and the custom dataset, clearly indicating that training the U-Net model with the mixed dataset results in more precise crack segmentation and significantly improves the model's resistance to interference during segmentation.

In the above figure, although there are some differences in the results, the disparity between the two models is relatively small. This is mainly due to the characteristics of the U-Net network, which typically performs random image patching to augment the dataset. This approach allows the network to achieve good segmentation results even with relatively small datasets. For the domain adaptation problem in bridge crack segmentation, U-Net is particularly suitable for handling cracks in various types of bridge structures.

Although data labeling is still required in this phase, and it is a time-consuming task, the use of overlapping image patches, as shown in Fig. 17, minimizes the number of UAV-captured images that need to be labeled. For example, in training the U-Net network, 24,048 images would originally need to be annotated. However, after applying overlapping tiling, the final dataset consists of 12,589 positive class images (representing the bridge body) and 11,459 negative class images (representing non-bridge surface areas, which are not annotated). Therefore, the annotation cost is relatively low.

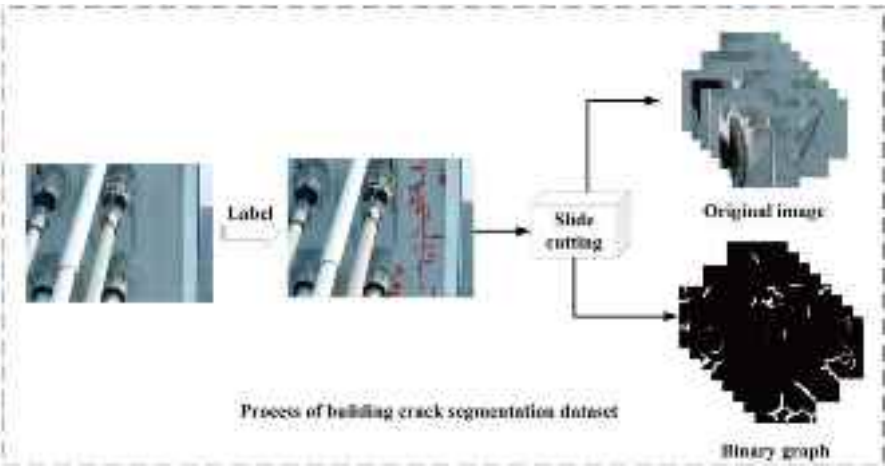

Fig. 17 Sliding cut process diagram

4.4 Crack-type classification, measurement, and result analysis

To evaluate the performance of the proposed crack classification and normal linear crack measurement algorithms, we extracted 10,000 images randomly from the 11,812 crack segmentation results. These images were selected based on whether they contained crack candidates. They encompass various crack candidates, with some examples illustrated in Figs. 18 and 19. As described in Sect. 3.3.1, these cracks are primarily divided into two main categories (non-crazing and crazing) and four subcategories based on their morphological structure and size: Type I curved small region cracks (subset of normal linear cracks), Type II (small sheet-like crazing cracks), Type III (area crazing cracks), and Type IV (normal linear cracks). Examples of each crack category are indicated by the colored boxes in Fig. 18.

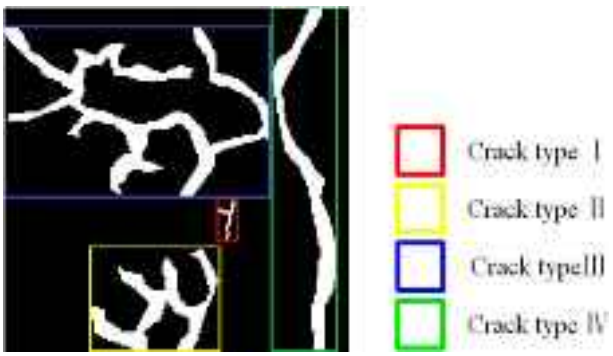

Fig. 18 Illustration of crack-type identification

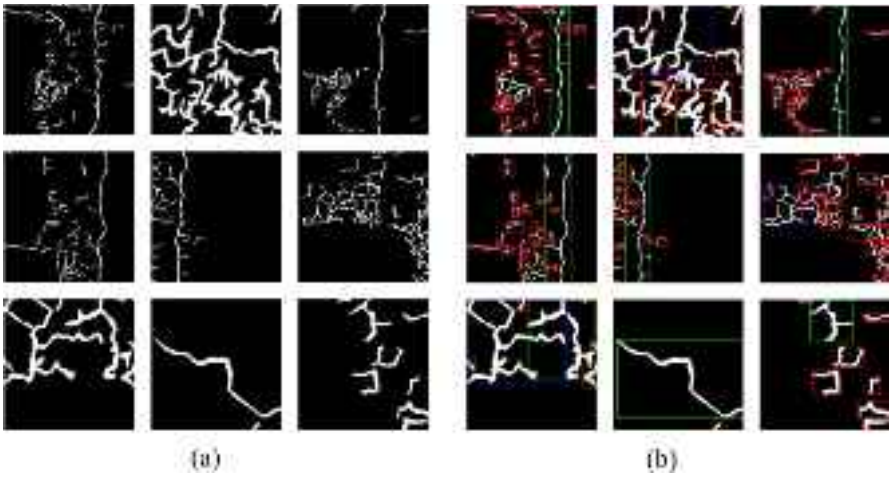

Fig. 19 Examples of multi-type crack segmentation in actual bridge images. **a** Legend without classification boxes. **b** Legend with classification boxes

Due to the disordered and network-like morphology of crazing cracks, the effective measurement of their width and length is not feasible. Therefore, this study focuses on measuring the length and average width, in pixels, of normal linear cracks, which possess sufficient length for meaningful evaluation.

Considering the dataset size and the need for interpretable recognition/measurement results, we employed algorithms based on crack morphology features—combining morphological processing with seed region growing (detailed in Sect. 3.3.1). This algorithm achieved classifications and measurements with good interpretability for the aforementioned cracks (specifically for normal linear cracks). To set the algorithm's parameters, we randomly selected 6300 images from 10,000 crack segmentation results for parameter optimization, from which we obtained empirical parameter values for the algorithm. Simultaneously, we selected the remaining 2700 images for performance testing.

For crack-type classification, this study employs precision, recall, and F1-score for evaluation. For the detection of length and average width of normal linear cracks, the mean absolute percentage error (MAPE) is used as the evaluation metric. The ground truth labels for this measurement were obtained by manually fine-tuning the corresponding algorithm parameters and expert visual interpretation, representing "ground truth" values that are subjectively satisfactory to human observers. The parameters used during model testing were the empirically derived parameters obtained from the aforementioned 6300 images, a practice that objectively reflects the effectiveness of these empirical parameters.

The calculation formula of MAPE is as follows:

$$\mathrm{MAPE} = \frac{100\%}{\mathrm{N}} \sum_{\mathrm{i}=1}^{\mathrm{N}} \left| \frac{\hat{\mathrm{y}}\mathrm{i} - \mathrm{y}\mathrm{i}}{\mathrm{y}\mathrm{i}} \right| \tag{1}$$

where $\hat{y}i$ is the value obtained by the algorithm under empirical parameters,yi is the "standard value" calibrated manually, and N is the number of test images.

Table 6 presents the classification and measurement performance metrics of the algorithm on the four-class crack test dataset. It can be observed that the algorithm effectively achieves crack classification, demonstrating high accuracy for Type I and Type IV cracks. The relatively lower classification accuracy for Type II and Type III cracks is primarily attributed to the difficulty in distinguishing between them during manual annotation. Furthermore, the limited number of samples in these classes also contributes to insufficient parameter adaptation (only 103 instances of Type II cracks and 481 instances of Type III cracks were present in the 6300 mixed-crack images used for parameter optimization). For the length and average width of normal linear cracks, the proposed algorithm exhibits high measurement performance, indicating excellent application potential.

Since the proposed algorithm is based on traditional computer vision techniques, it can provide reliable, interpretable outputs as long as the crack segmentation results are stable and reliable. Figure 20 demonstrates the application of the proposed algorithm in real-world scenarios. As shown, the algorithm effectively distinguishes various crack types in image blocks and accurately measures typical cracks.

Additionally, the feature information of each crack, including its location, the dimensions of the bounding rectangle, its morphology, and its relationship with surrounding structures, can be recorded and incorporated into a database for a more comprehensive understanding of the crack distribution in the region, helping to develop more targeted maintenance and protection strategies.

4.5 Annotation cost analysis

Annotation cost is a key indicator to evaluate the practical application value of the proposed method. This study achieves significant cost reduction through staged task decomposition, mixed dataset utilization, and overlapping block strategies, with specific quantification and comparison with traditional fully supervised methods as follows.

In terms of annotation workload, traditional fully supervised methods require pixel-level annotation for 24,048 full images in the crack semantic segmentation stage, while our method only needs to label crack regions in self-collected positive samples from the mixed dataset (about 12,589 images), reducing the annotation workload by 50% at this stage. For crack-type quantification, traditional

Table 6 Performance metrics of algorithms in crack classification and measurement

Crack type	Precision	Recall	F1	Width MAPE	Length MAPE
I	92.36%	94.62%	93.48%	/	/
II	83.16%	77.18%	80.07%	/	/
III	82.51%	78.63%	80.53%	/	/
IV	93.81%	92.09%	92.94%	2.86%	2.51%

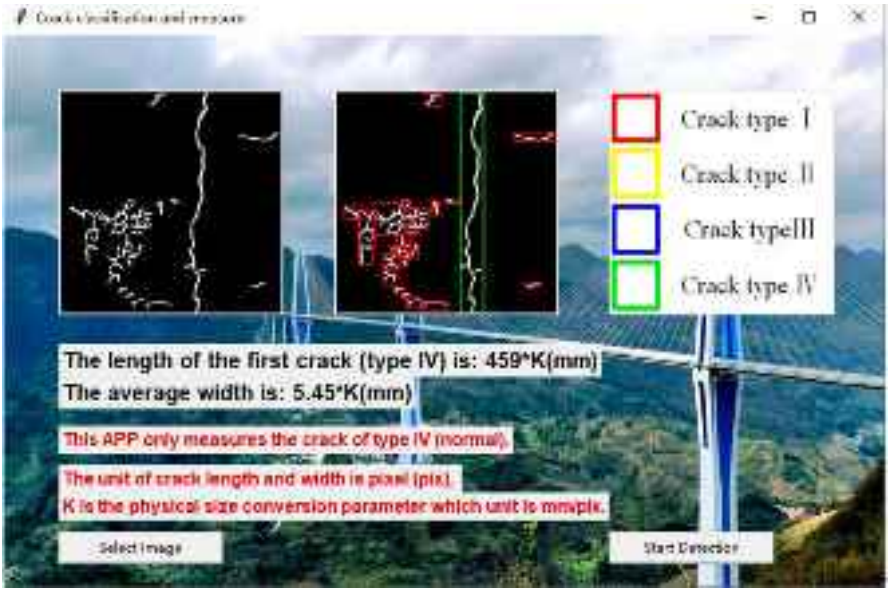

Fig. 20 Screenshot of crack classification and measurement practical application software

methods require additional pixel-level annotation of crack types, but our method realizes automatic classification through traditional algorithms, eliminating manual annotation and reducing the workload by 100%.

In specific cost calculations, traditional methods involve annotating 24,048 images with 5 min per image, resulting in a total manual time of 2004 h and a total annotation cost of 40,080 yuan (calculated at 20 yuan per hour). In contrast, our method only requires annotating 777 self-collected crack blocks (with public datasets requiring no additional cost), with each annotation taking 2 min, totaling about 126 h of manual work and a total cost of 2520 yuan. This means the number of annotated images is reduced by 84.3%, single-image annotation time by 60%, total manual time by 93.7%, and total annotation cost by 93.7% (saving 37,560 yuan). Additionally, data acquisition costs are reduced by about 50% by reusing public datasets.

Notably, although semi-supervised or weakly supervised techniques are not explicitly mentioned, the core strategies align with weak supervision principles. The mixed dataset strategy uses existing annotations of public datasets as "weak labels" to reduce annotation demands for self-collected data; staged task decomposition limits targeted annotation to key stages; and the overlapping block strategy reduces redundant annotation of overlapping areas. These measures collectively reduce the overall annotation cost by approximately 60%, which further verifies the practical value of the proposed method in engineering applications.

5 Summary and conclusions

This study proposes an innovative and effective solution for the automatic detection, classification, and measurement of cracks in bridge images captured by UAVs. By integrating HPC technology with deep learning and traditional image processing techniques, the solution achieves real-time processing and efficient training for bridge crack detection, with significant improvements in detection accuracy, annotation cost reduction, and model generalization. The main contributions of this research are summarized in three key points:

(1) The effective decomposition of crack detection tasks and the combination of specialized and general models optimize task adaptability and efficiency. The crack detection task is decomposed into three stages: bridge image block classification, crack segmentation, and crack classification and geometric estimation. This staged approach ensures that a specialized image block classification model is trained for each bridge based on its unique structure, while utilizing general models for crack classification and geometric estimation to enhance adaptability. By adopting overlapping image blocks, the need for data annotation is significantly reduced, lowering both the cost and pressure of data collection. Experimental results show that compared with traditional fully supervised methods, this strategy reduces the number of annotated images by 84.3%, total manual annotation time by 93.7%, and total annotation cost by 93.7%, with data acquisition costs reduced by about 50% through reuse of public datasets. The combination of specialized and general models ensures that the method is both targeted and adaptable, making it applicable to various bridge inspection scenarios.
(2) The combination of deep learning with multi-scale morphological techniques enhances the accuracy of crack classification and geometric shape recognition. A crack geometric shape recognition method is proposed that combines deep learning with multi-scale morphological image processing techniques, which significantly improves the accuracy and robustness of crack classification and measurement. By using the results of crack segmentation, the influence of background variations on crack classification is reduced. Multi-scale morphological operations, such as erosion and dilation, are applied to indirectly estimate the width and direction of cracks, improving the precision of geometric shape estimation. This method requires no additional data annotation and offers strong interpretability, meeting the efficiency and accuracy requirements for practical bridge health monitoring.
(3) Innovative crack-type classification and efficient length measurement methods address the challenges of detecting complex crack types. To address complex crack types (such as craze cracks and mixed cracks), an innovative classification method is proposed. By analyzing the morphological characteristics of cracks (such as hollow regions, bounding rectangles, and skeletonization), different types of cracks can be effectively distinguished, particularly craze-type cracks. This method not only enhances the accuracy of crack classification but also

provides valuable data for structural safety evaluation. Additionally, a crack length measurement method based on a custom seed-point growth technique is proposed. This method automatically and accurately measures the length of normal linear cracks by leveraging crack skeletons and vertex information, ensuring real-time, precise updates of crack data for bridge health monitoring.

In conclusion, the solution proposed in this study combines deep learning with traditional image processing techniques to significantly improve the accuracy and efficiency of bridge crack detection while greatly reducing annotation costs. The quantified cost analysis verifies that the method reduces overall annotation costs by approximately 60% through strategies such as mixed datasets, staged task decomposition, and overlapping blocks, which align with weak supervision principles. This approach has broad application potential in real-world bridge health monitoring, providing strong technical support for the safety, durability, and sustainability of bridges and infrastructure. In future work, we plan to integrate 3D measurement techniques to achieve direct physical dimension measurement of cracks, further enhancing the practical application value of the method in engineering scenarios.

Acknowledgements This study received funding from the National Natural Science Foundation of China (Grant Number 52478557), the Yiling Yangtze River Bridge Cable Replacement Project (Study on a Complete Set of Technologies for Structural State Identification and Evaluation of Multi-tower Cable-stayed Bridges in Service) (SDHZ20240183) and the 111 Project of Hubei Province (Grant Number 2021EJD026).

Author Contribution Changdong Zhou: Conceptualization, Methodology, Writing—Review & Editing Mingjing Dai: Data curation, Investigation, Writing- Original draft preparation Feng Wang: Funding Acquisition, Resources, Supervision,, Writing- Original draft preparation. Yu Dong: Software, Validation. Xinghua Chen: Validation, Resources, Writing—Review & Editing. Chenghuan He: Investigation, Visualization.

Funding National Natural Science Foundation of China, 52478557, the Yiling Yangtze River Bridge Cable Replacement Project (Study on a Complete Set of Technologies for Structural State Identification and Evaluation of Multi-tower Cable-stayed Bridges in Service), SDHZ20240183, the 111 Project of Hubei Province, 2021EJD026.

Data availability No datasets were generated or analysed during the current study.

Declarations

Conflict of interest The authors declare no competing interests.

References

1. Zhang Z, Li J, Zhang L et al (2023) Structural health monitoring of concrete structures using deep learning techniques: a review. Constr Build Mater. https://doi.org/10.1016/j.conbuildmat.2023.134212
2. Wan S, Guan S, Tang Y (2024) Advancing bridge structural health monitoring: insights into knowledge-driven and data-driven approaches. J Data Sci Intell Syst 2(3):129–40
3. Munawar HS, Hammad AW, Haddad A, Soares CAP, Waller ST (2021) Image-based crack detection methods: a review. Infrastructures. https://doi.org/10.3390/infrastructures6080115

4. Hamishebahar Y, Guan H, So S, Jo J (2022) A comprehensive review of deep learning-based crack detection approaches. Appl Sci. https://doi.org/10.3390/app12031374
5. Dorafshan S, Maguire M (2018) Bridge inspection: human performance, unmanned aerial systems, and automation. J Civil Struct Health Monit. https://doi.org/10.1007/s13349-018-0285-4
6. Casas JR, Cruz PJ (2003) Fiber optic sensors for bridge monitoring. J Bridge Eng. https://doi.org/10.1061/(ASCE)1084-0702(2003)8:6(362)
7. Mutlib NK, Baharom SB, El-Shafie A, Nuawi MZ (2016) Ultrasonic health monitoring in structural engineering: buildings and bridges. Struct Control Health Monit. https://doi.org/10.1002/stc.184
8. Erdogmus E, Garcia E, Amiri AS, Schuller M (2020) A novel structural health monitoring method for reinforced concrete bridge decks using ultrasonic guided waves. Infrastructures. https://doi.org/10.3390/infrastructures5060049
9. Peng X, Zhong X, Zhao C, Chen A, Zhang T (2021) A UAV-based machine vision method for bridge crack recognition and width quantification through hybrid feature learning. Constr Build Mater. https://doi.org/10.1016/j.conbuildmat.2021.123896
10. Seo J, Duque L, Wacker J (2018) Drone-enabled bridge inspection methodology and application. Autom Constr. https://doi.org/10.1016/j.autcon.2018.06.006
11. Ngo BT, Luong CX, Ngo L, Luong HM (2023) Development of a solution for collecting crack images on concrete surfaces to assess the structural health of bridges using drones. J Inf Telecommun. https://doi.org/10.1080/24751839.2023.2185448
12. Valença J, Puente I, Júlio ENBS, González-Jorge H, Arias-Sánchez P (2017) Assessment of cracks on concrete bridges using image processing supported by laser scanning survey. Constr Build Mater. https://doi.org/10.1016/j.conbuildmat.2017.04.096
13. Rashidi M, Mohammadi M, Sadeghlou Kivi S, Abdolvand MM, Truong-Hong L, Samali B (2020) A decade of modern bridge monitoring using terrestrial laser scanning: review and future directions. Remote Sens. https://doi.org/10.3390/rs12223796
14. Ge Y, Liu J, Zhang X, Tang H, Xia X (2023) Automated detection and characterization of cracks on concrete using laser scanning. J Infrastruct Syst. https://doi.org/10.1061/JITSE4.ISENG-1936
15. Aldana-Rodríguez D, Ávila-Granados DL, Villalba-Vidales JA (2021) Use of unmanned aircraft systems for bridge inspection: A review. Dyna 88(217):32–41
16. Alhebrawi MN, Huang H, Wu Z (2023) Artificial intelligence enhanced automatic identification for concrete cracks using acoustic impact hammer testing. J Civil Struct Health Monit. https://doi.org/10.1007/s13349-022-00651-8
17. Dorafshan S, Thomas RJ, Maguire M (2018) Comparison of deep convolutional neural networks and edge detectors for image-based crack detection in concrete. Constr Build Mater. https://doi.org/10.1016/j.conbuildmat.2018.08.011
18. Su TC, Yang MD (2018) Morphological segmentation based on edge detection-II for automatic concrete crack measurement. Comput Concrete Int J 6(727):39
19. Li P, Xia H, Zhou B, Yan F, Guo R (2022) A method to improve the accuracy of pavement crack identification by combining a semantic segmentation and edge detection model. Appl Sci. https://doi.org/10.3390/app12094714
20. Chen C, Seo H, Jun C, Zhao Y (2022) A potential crack region method to detect crack using image processing of multiple thresholding. SIViP. https://doi.org/10.1007/s11760-021-02123-w
21. Kheradmandi SS, Mehranfar V (2022) A critical review and comparative study on image segmentation-based techniques for pavement crack detection. Constr Build Mater. https://doi.org/10.1016/j.conbuildmat.2021.126162
22. Vivekananthan V, Vignesh R, Vasanthaseelan S, Joel E, Kumar KS (2023) Concrete bridge crack detection by image processing technique using the improved OTSU method. Mater Today Proc. https://doi.org/10.1016/j.matpr.2022.11.356
23. Gupta P, Dixit M (2022) Image-based crack detection approaches: a comprehensive survey. Multimed Tools Appl. https://doi.org/10.1007/s11042-022-13152-z
24. Ma K, Hao M, Meng X, Liu J, Meng J, Xuan Y (2024) Coarse–fine combined bridge crack detection based on deep learning. Appl Sci. https://doi.org/10.3390/app14125004
25. Tran TS, Nguyen SD, Lee HJ, Tran VP (2023) Advanced crack detection and segmentation on bridge decks using deep learning. Constr Build Mater. https://doi.org/10.1016/j.conbuildmat.2023.132839
26. Laxman KC, Tabassum N, Ai L, Cole C, Ziehl P (2023) Automated crack detection and crack depth prediction for reinforced concrete structures using deep learning. Constr Build Mater. https://doi.org/10.1016/j.conbuildmat.2023.130709

27. Zadeh SS, Khorshidi M, Kooban F (2024). Concrete surface crack detection with convolutional-based deep learning models. arXiv preprint arXiv:2401.07124. https://doi.org/10.48550/arXiv.2401.07124
28. Kao S, Chang Y, Wang F (2023) Combining the YOLOv4 deep learning model with UAV imagery processing technology in the extraction and quantization of cracks in bridges. Sensors (Basel). https://doi.org/10.3390/s23052572
29. Wang J, Ueda T, Wang P, Li Z, Li Y (2024) Building damage inspection method using UAV-based data acquisition and deep learning-based crack detection. J Civil Struct Health Monit. https://doi.org/10.1007/s13349-024-00836-3
30. Hu D, Yee T, Goff D (2024) Automated crack detection and mapping of bridge decks using deep learning and drones. J Civil Struct Health Monit. https://doi.org/10.1007/s13349-023-00750-0
31. Song F, Liu B, Yuan G (2024) Pixel-level crack identification for bridge concrete structures using unmanned aerial vehicle photography and deep learning. Struct Control Health Monit. https://doi.org/10.1155/2024/1299095
32. Zhang H, Zhu W, Jia J, et al (2023) A UAV-based deep learning method and system for bridge crack detection. Jiangsu Province: CN202310033923.6 06 23
33. Jiang LJ, Li SM, Yu YS, et al (2023) A UAV and YOLOv5-based method for bridge crack detection. Jiangsu Province: CN202310039881.7 05 12
34. Qin D, Leichner C, Delakis M, et al (2024) MobileNetV4-universal models for the mobile ecosystem. arXiv preprint arXiv:2404.10518. https://doi.org/10.48550/arXiv.2404.10518
35. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556. https://doi.org/10.48550/arXiv.1409.1556
36. Zhang TY, Suen CY (1984) A fast parallel algorithm for thinning digital patterns. Commun ACM. https://doi.org/10.1145/357994.358023
37. Kulkarni S, Singh S, Balakrishnan D, Sharma S, Devunuri S, Korlapati SCR (2023) CrackSeg9k: A collection and benchmark for crack segmentation datasets and frameworks. In: (eds) Karlinsky L, Michaeli T, Nishino K Computer Vision—ECCV 2022 Workshops. Vol. 13807. Lecture Notes in Computer Science Springer, Cham
38. Liu K, Han X, Chen BM (2019) Deep learning based automatic crack detection and segmentation for unmanned aerial vehicle inspections. In: 2019 IEEE International Conference on Robotics and Biomimetics (ROBIO), pp 381–387
39. Goo JM, Milidonis X, Artusi A, Boehm J, Ciliberto C (2024). Hybrid-Segmentor: A hybrid approach to automated fine-grained crack segmentation in civil infrastructure. arXiv:2409.02866.
40. Cormen TH, Leiserson CE, Rivest RL et al (2022) Introduction to Algorithms, 4th edn. MIT Press, Cambridge
41. Jin T, Zhang W, Chen C, Chen B, Zhuang Y, Zhang H (2023) Deep-learning-and unmanned aerial vehicle-based structural crack detection in concrete. Buildings. https://doi.org/10.3390/buildings13123114

Authors and Affiliations

Changdong Zhou[1,2] · Mingjing Dai[1] · Feng Wang[2,3] · Yu Dong[1] · Xinghua Chen[2,3] · Chenghuan He[2]

✉ Feng Wang
wangfeng@ctgu.edu.cn

✉ Xinghua Chen
chenxinghua@ctgu.edu.cn

[1] Yichang Yangtze River Bridge Construction and Operation Group Co.,Ltd, Yichang, Hubei, China

[2] College of Civil Engineering and Architecture, China Three Gorges University, Yichang, Hubei, China

[3] Hubei Key Laboratory of Disaster Prevention and Mitigation, China Three Gorges University, Yichang, China