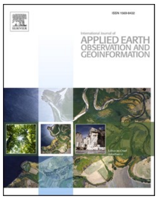
International Journal of Applied Earth Observation and Geoinformation 124 (2023) 103527

Contents lists available at ScienceDirect

International Journal of Applied Earth Observation and Geoinformation

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



Robust surface crack detection with structure line guidance

Yongjun Zhang [a,*], Yixin Lu [a], Yansong Duan [a], Dong Wei [a,*], Xianzhang Zhu [b,c], Bin Zhang [a], Bohui Pang [d]

[a] *School of Remote Sensing and Information Engineering, Wuhan University, Wuhan, Hubei 430072, China*
[b] *Changjiang Spatial Information Technology Engineering Co., Ltd, Wuhan, Hubei 430010, China*
[c] *Water Resources Information Perception and Big Data Engineering Research Center of Hubei Province, Wuhan, Hubei 430010, China*
[d] *R & D Department of Huaneng Lancang River Hydropower INC, Kunming, Yunnan 650216, China*

ARTICLE INFO

Keywords:
Crack detection
Structure-line
Crack measure
Gaussian function

ABSTRACT

Crack detection plays a pivotal role in civil engineering applications, where vision-based methods find extensive use. In practice, crack images are sourced from Unmanned Aerial Vehicles (UAV) and handheld photography, and the balance between the utilization of global and local information is the key to detecting cracks from images of different sources: the former tends to eliminate interferences with a global perspective, whereas the latter pays more attention to the description of local details of cracks. However, many existing methods primarily target crack detection in handheld photographs and may not perform optimally on UAV-generated images or those with variable backgrounds or from different sources. In response to this challenge, we propose a robust and innovative method called Crack Detection with Structure Line (CDSL). The primary steps of this method can be summarized as follows: first, based on local information, an indicator called the “crack measure” is derived to directly generate a continuous crack map for effective image binarization; then, based on global information, the crack map is simplified in a unified and analyzable form using structure lines to perform a robust optimization for high-precision crack detection. The experiments we conducted on two publicly available datasets showed that CDSL provided competitive crack detection performance and outperformed four classical or current state-of-the-art methods by at least 13.0 % in the UAV dataset we collected.

1. Introduction

Cracks represent a common concern in civil engineering, impacting the structural integrity of infrastructure such as pavements, bridges, and buildings (Chen et al., 2021; Shi et al., 2016; Zhou et al., 2019). Left unaddressed, cracks can accelerate the deterioration of these facilities and pose concealed hazards to public safety. Thus, it is important to detect and locate cracks in engineering applications for maintenance operations (Liu et al., 2019).

Vision-based methods are widely employed for crack detection due to their cost-effectiveness and reliability (Ai et al., 2023). In extensive-range inspections, data acquisition methods often encompass Unmanned Aerial Vehicle (UAV) photogrammetry and supplementary handheld photography. However, the images from these two sources exhibit notable disparities. As shown in Fig. 1, images captured by handheld cameras intentionally position cracks prominently to minimize interference. Crack detection in such general images primarily emphasizes local information, focusing on factors such as crack continuity and details. Conversely, UAV imagery is automatically obtained from a greater distance and may include more interfering elements like potholes, fallen leaves, and shadows (Li et al., 2022; Pastucha et al., 2022). The key to detecting cracks in UAV images is to distinguish cracks from these interfering elements based on global information. Therefore, methods designed for crack detection in general images may not yield satisfactory results when applied to UAV images, a method for detecting cracks across diverse image sources must be able to utilize both local and global information.

Currently, deep learning methods with extensive parameterization have the capability to process two types of information (Ji et al., 2023; Xu et al., 2022). However, the efficacy of deep learning methods hinges on the quality of their training datasets (Fig. 1c). Creating a representative training dataset is difficult and time-consuming due to the intricate and variable nature of UAV image backgrounds. Additionally, gathering sufficient data for specific scenarios presents a formidable

---

* Corresponding authors.

E-mail addresses: zhangyj@whu.edu.cn (Y. Zhang), luyixin@whu.edu.cn (Y. Lu), ysduan@whu.edu.cn (Y. Duan), weidong@whu.edu.cn (D. Wei), zxzorigin@whu.edu.cn (X. Zhu), bin.zhang@whu.edu.cn (B. Zhang), chenhao@lcjgs.chng.com.cn (B. Pang).

https://doi.org/10.1016/j.jag.2023.103527
Received 30 March 2023; Received in revised form 28 September 2023; Accepted 16 October 2023

Handheld photograph of a crack

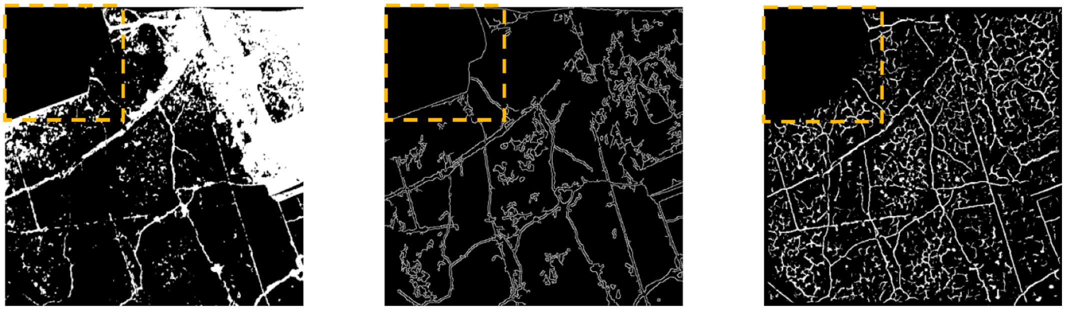
(a)

UAV image with ground truth

(b)

U-net (Ronneberger et al., 2015)

(c)



OTSU (Otsu, 1979)

(d)

Canny (Canny, 1986)

(e)

CDSL binarization

(f)

Fig. 1. Comparison of crack images. (a) is taken by a handheld camera; (b) is processed to remove the house (orange box)[11]; (c) shows that a U-net trained by a dataset[22] of concrete can only detect a few cracks in (b); (d) is the binarization of (b) using OTSU; (e) is the edge detection results using Canny; (f) is obtained by the CDSL crack measure. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

obstacle. Therefore, we have contemplated the utilization of traditional methods for crack detection.

For traditional methods, image binarization is an important step for generating a crack map that retains potential crack pixels while eliminating background pixels (Li et al., 2018). Commonly employed techniques involve threshold segmentation (Li et al., 2018; Safaei et al., 2022; Xie et al., 2023; Zhang et al., 2020) and edge detection algorithm (Canny, 1986; Frangi et al., 1998; Kanopoulos et al., 1988; Stehfest, 1970). The former relies on a global perspective but may result in a discontinuous crack map (Fig. 1d). The latter adopts a local approach but introduces ambiguity, leading to an incomplete crack map (Fig. 1e). Therefore, an effective image binarization algorithm should be derived.

Result optimization is another crucial step following image binarization, aiming to rectify issues of either over-localization or over-globalization (Ai et al., 2018; Wang et al., 2021). As shown in Fig. 1f, while the crack map is relatively complete and continuous, it inevitably excludes certain cracks (i.e., false negatives); or includes potholes that connect with genuine cracks (i.e., false positives). Therefore, it is necessary to devise an effective approach for addressing these omissions, structuring the crack map to distinguish cracks from interference, and implementing rigorous error elimination to achieve precise crack detection (Chen et al., 2021; Li et al., 2018; Shi et al., 2016).

To address these challenges, we introduce a novel crack detection method called Crack Detection with Structure Line (CDSL). Our CDSL method is based on the theory that cracks can be effectively represented and analyzed using local straight lines, referred to as “structure lines,” which encapsulate both the topological and semantic information of the cracks (as elaborated in Section 3). When compared with four classical or current state-of-the-art methods using two publicly available datasets, CDSL provided competitive crack detection performance and outperformed other methods by at least 13.0 % on the UAV dataset we collected. The primary contributions of this study are summarized as follows:

(1) We introduced an effective method, named CDSL, designed for the detection of cracks on concrete surfaces or asphalt pavements using general crack or UAV images.
(2) We derived an indicator known as the “crack measure” from local information and serves to generate a continuous and comprehensive crack map.
(3) We have proposed a crack representation, termed a “structure line,” that leverages global information. This representation facilitates mathematical analysis based on semantic structural feature constraints for crack detection.

2. Related works

2.1. Image binarization

Image binarization constitutes a crucial step in many traditional methods. The simplest contemporary method is threshold segmentation (Cheng et al., 2003; Liu et al., 2002). The methods for threshold selection include empirical selection, thresholding based on histogram entropy (Kapur et al., 1985), OTSU (Otsu, 1979), and iterative selection (Ridler and Calvard, 1978). However, complications may arise when interferences exhibit darker attributes than the cracks, leading to a reduction in the global threshold and consequent elimination of crack pixels. Local threshold methods have been proposed to solve this problem (Li et al., 2018). Nevertheless, these methods are limited to applying a “global” approach within specific blocks, and a local-intensity-based method will identify disjoint crack fragments (Zou et al., 2012).

Many contemporary methods employ edge-detection algorithms for crack detection (Ma et al., 2022; Nong et al., 2020; Zhou et al., 2019). Commonly used edge-detection algorithms include the Sobel filter (Kanopoulos et al., 1988), Canny filter (Canny, 1986), Laplace operator (Stehfest, 1970), and Hessian matrix (Frangi et al., 1998). However, edge detection methods often yield two edges for a single crack, leading to ambiguity, especially when the crack width varies significantly (i.e.,

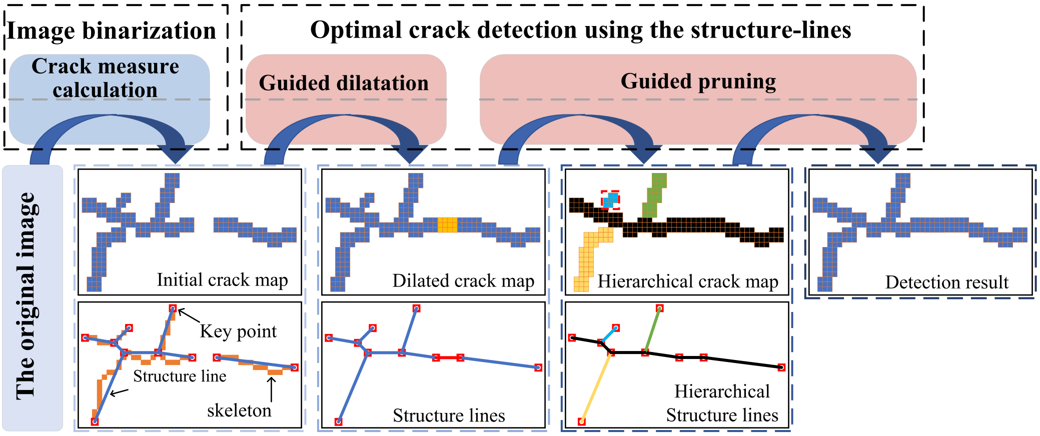


Fig. 2. CDSL flow diagram.

determining whether two adjacent edges belong to the same crack). This can result in incomplete crack representations (Ai et al., 2023). Feature detection through multiscale analysis function systems offers a potential solution to these issues, as it directly abstracts potential crack pixels, mitigating ambiguity and contributing to more comprehensive crack identification.

In recent years, numerous multiscale analysis function systems have been developed for the detection of various image features, including spots, edges, and ridges (Andrushia et al., 2021; Gibert et al., 2014). Our crack measure has been derived from a multiscale analyzing function system: by treating the crack as an inverted ridge for detection, continuous and complete crack pixels can be obtained directly without ambiguity.

2.2. Crack detection methods

Crack detection methods can be broadly classified into traditional and deep learning methods. Traditional methods are rule-driven (O'Mahony et al., 2020), interpretable, and amenable to optimization. However, the performance of different traditional methods can vary significantly because manually crafted rules may not cover all scenarios. Deep learning methods, on the other hand, are data-driven and extract features directly from the data itself to detect cracks (O'Mahony et al., 2020). They tend to perform better when an ample training dataset is available. However, they necessitate that the training set and test data conform to the same distributions. In some scenarios and at the beginning of a project, it is difficult to obtain sufficient data, and even when sufficient data is obtained, labeling the data is time-consuming and laborious (Farahani et al., 2021). Both methods find widespread application.

Many deep learning methods have been employed for crack detection in images from diverse scenarios (Feng et al., 2023; Liu et al., 2019; Ma and Li, 2022; Ren et al., 2023; Xiao et al., 2023; Xu et al., 2022). However, data collected via UAVs tend to be variable and limited in quantity. UAVs often operate from considerable distances and high altitudes to ensure safety during large-scale inspections. Consequently, the higher the UAV's flight altitude, the fewer images are available to cover the same area. Additionally, there is a paucity of publicly available datasets. Thus, the available data may not suffice to meet the training requirements, making deep learning methods less ideal for crack detection in UAV images.

In traditional methods, the processing emphasis after binarization is to refine the initial crack map (Ataiwe, 2023; Li et al., 2018; Wang et al., 2023). Various techniques have been employed for this purpose, such as mathematical morphology methods (Ai et al., 2018; Weng et al., 2019), active contour models (Nguyen et al., 2018), and graph theory-based methods (Payab et al., 2019; Zou et al., 2012). Each of these methods has its characteristics: Purely relying on the morphological method often fails to yield promising results, as it heavily depends on local information to enhance crack connectivity. Active contour models typically assume a constant crack width, which may not align with real-world scenarios. The graph theory-based methods were able to reduce false positives to some extent but performed weakly in detecting cracks with complex topologies (Ai et al., 2023).

Since we perform an effective image binarization to obtain a continuous crack map, the process of CDSL is more inclined to reduce false positives. Thus, we incorporate global structure information to utilize the graph theory-based method to mitigate false positives, meanwhile, we introduce the process like the morphological method to reduce false negatives.

3. Overview of the proposed method

As shown in Fig. 2, the workflow chart of CDSL comprises two primary components: image binarization using the crack measure and optimal crack detection using the structure lines.

For image binarization, we present different outcomes resulting from convolutions of a 1D crack with both odd- and even-order derivatives of Gaussian functions (Section 4.1). Subsequently, we derive a two-dimensional crack measure based on these convolutions to produce a continuous crack map (Section 4.2).

For optimal crack detection, we first streamline the crack map using the structure lines (Section 5.1); we then perform guided dilatation with the structure lines on the crack map to mitigate false negatives (Section 5.2); and finally, we conduct guided pruning to reduce the false positives and present the final detection result of CDSL (Section 5.3).

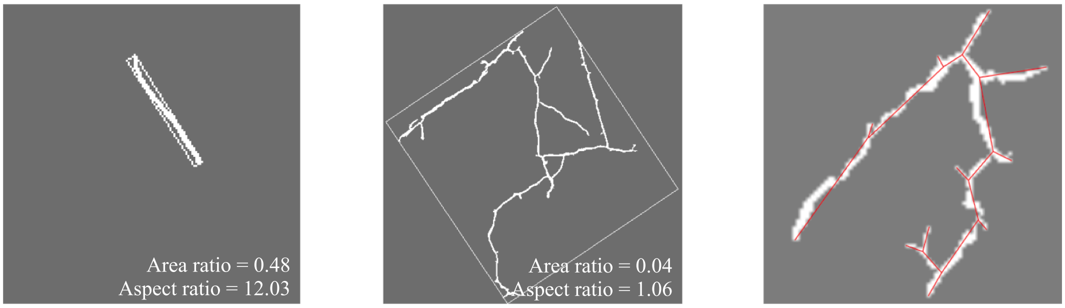
CDSL relies on several properties that are commonly observed in most cracks, as extensively documented in previous studies:

(1) The grayscale within the crack is notably lower than that of its surrounding pixels (Andrushia et al., 2021; Chen et al., 2021; Li et al., 2018; Shi et al., 2016).

---

[1] The steps to remove the building are as follows: first, we use continuous UAV image pairs to reconstruct the surface point cloud and recover 3D information from the images; then, we perform a plane segmentation to keep ground points (Zhu et al., 2021); finally, we calculate the pixel coordinates of the ground points by using 3D information of images and retain pixels while covering other pixels.

[2] U-net was trained by the positive set of "Concrete_Crack_Images_for_Classification". https://github.com/SalhaNasser2019/Concrete_Crack_Images_for_ClassificationLargNewNew.



(a) (b) (c)

Fig. 3. Morphological features of cracks. The crack is long and narrow with (a) a large aspect ratio or (b) a small area ratio. (c) Structure lines of cracks (red lines) with obvious direction changes. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

(2) Cracks tend to exhibit continuity, following a consistent trend and orientation, with gradual grayscale changes (Pantoja-Rosero et al., 2022; Zou et al., 2012).

(3) The crack shape is long and narrow (Li et al., 2018; Shi et al., 2016). Therefore, the crack exhibits a considerable length, often leading to a high aspect ratio for its enclosing rectangle (Fig. 3a) or a low area ratio (Fig. 3b).

In line with Property 2, the pixels constituting a crack are expected to be closely distributed around a trend line that connects the two ends of the crack. This trend line is referred to as the "structure line." Thus, under this assumption, a crack can be represented using local straight lines; a connected crack pixel-set with an obvious direction change is the connection of two distinct cracks; the two cracks are divided by their intersection and their structure lines consist of polylines (Fig. 3c).

4. Image binarization using the crack measure

Establishing a continuous and comprehensive crack map forms the foundation for achieving improved detection outcomes, where "1″ is assigned to potential crack pixels and "0" to others (Fig. 6c). In this section, with local information around pixels, a crack measure is derived based on Gaussian function derivatives to generate a continuous crack map.

4.1. Gaussian function and ideal 1D crack

As shown in Fig. 4h, Equation (5), which was derived from Gaussian derivatives, returned a maximum at the center of the crack (denoted as C). Based on this, the crack measure shown in (6) can be derived by extending (5) in two dimensions.

Let the normalized forms of the odd- and even-order Gaussian derivatives (Fig. 4c and d) be

$$\psi^o = \frac{G_0'(x)}{\|G_0'(x)\|_1}, \psi^e = \frac{G_0''(x)}{\|G_0''(x)\|_1}, \tag{1}$$

where

$$G_0(x) = e^{-x^2} \tag{2}$$

is the Gaussian function. (1) can be extended by

$$f^o_{x_0,j}(x) = n\psi^o(a^j(x - x_0)), \tag{3}$$

$$f^e_{x_0,j}(x) = n\psi^e(a^j(x - x_0)), \tag{4}$$

where a is a constant, j is a scale factor, x_0 is an offset factor, and n is an amplification factor. When (3) and (4) align C, the convolution $\langle C, f^e_{x_0,j}(x)\rangle$ will obtain the maxima (Fig. 4f), and the results are all zero for $\langle C, f^o_{x_0,j}(x)\rangle$ (Fig. 4e). The cusps at the center of the crack can be obtained by

$$f(x) = \langle C, f^e_{x_0,j}(x)\rangle - \left|\langle C, f^o_{x_0,j}(x)\rangle\right|. \tag{5}$$

The maximum of the cusps corresponds to the j which making the distance (denoted as w_j and shown in Fig. 4b) between the two zeros of (3) equal to the width of C (Fig. 4g and h).

4.2. Crack measure

By extending (5) to two dimensions, the crack measure is derived as

$$C(I, \boldsymbol{t}_0) = \sum_{j\in J}\langle I, m^e_{j,\theta^*,\boldsymbol{t}_0}\rangle - \sum_{j\in J}\left|\left\langle I, m^o_{j-j^o,\theta^*,\boldsymbol{t}_0}\right\rangle\right| - \beta\sum_{j\in J}\left|K_{g^e}(I, j, \boldsymbol{t}_0)\right| \tag{6}$$

where I denotes the grayscale map, $\beta > 0$ is a threshold, and $J = [j_{min}, j_{max}]$ is an integer set of j, the above three parameters are the input. The other parameters of crack measure were as follows:

$$m_{j,\theta,t_0}(\boldsymbol{t}) = g(\boldsymbol{A}_{a^j}\boldsymbol{R}_\theta[b(\boldsymbol{t} - \boldsymbol{t}_0)]), K_{g^e}(I, j, \boldsymbol{t}_0) = \int_{-\frac{w_0}{2}\times\frac{w_{j^*}}{w_j}}^{\frac{w_0}{2}\times\frac{w_{j^*}}{w_j}} \psi^e, \tag{7}$$

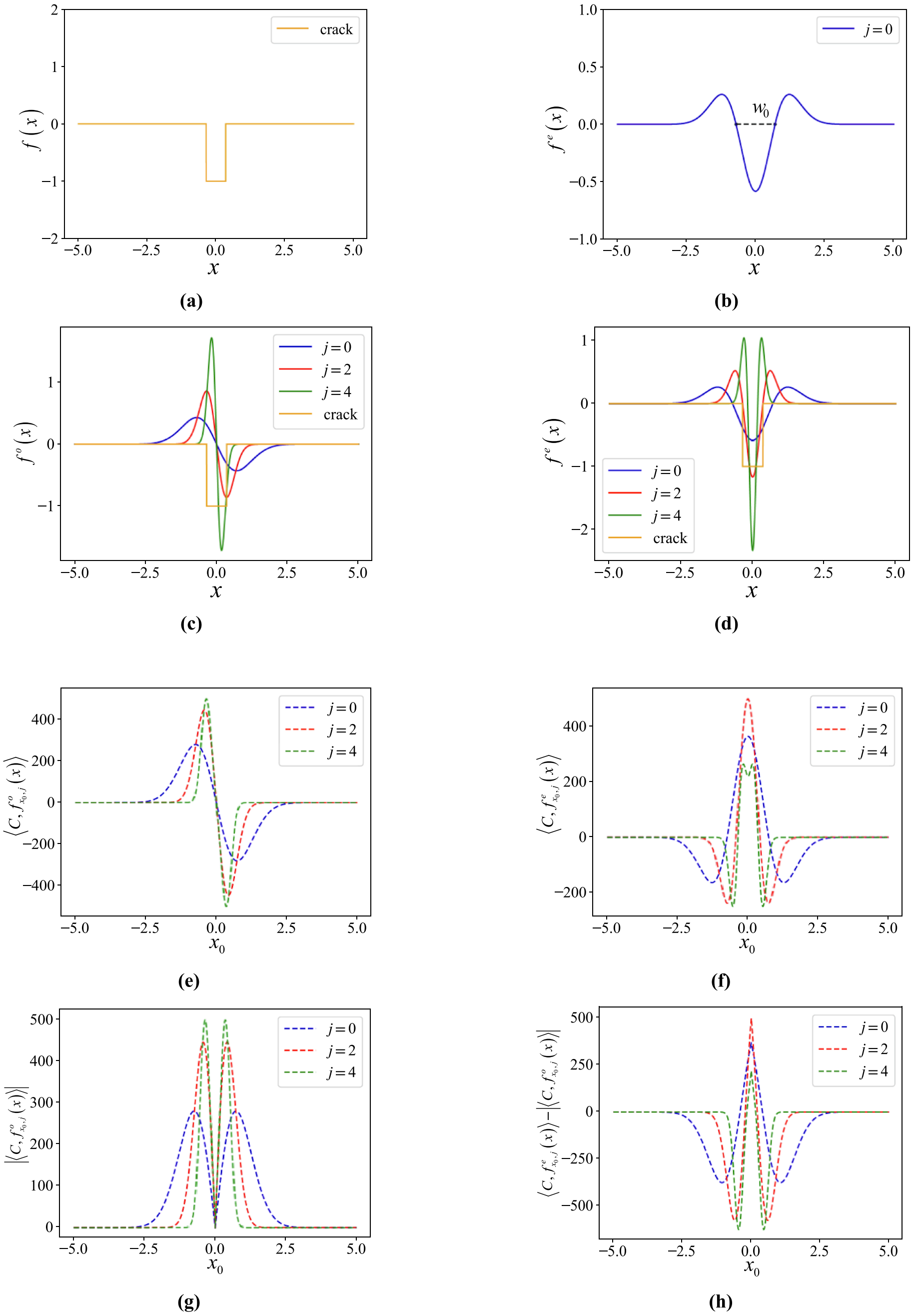
where $g(x, y) = \pi^{-\frac{1}{2}}\psi(x)G_0(y)$, x and y are the pixel coordinates, the superscripts of (6) correspond to (1); the parameters

$$\boldsymbol{A} = \begin{bmatrix} a^j & 0 \\ 0 & a^j \end{bmatrix}, \boldsymbol{R}_{\theta\in[0,2\pi]} = \begin{bmatrix} cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}, \boldsymbol{t} = [x, y]^T, \boldsymbol{t}_0 = [x_0, y_0]^T. \tag{8}$$

$b = 0.0001$ is the scale parameter of the coordinates that refines the function, and j^o denotes the scaling offset between even- and odd-symmetric molecules. Same as in the 1-D case, when m^e_{j,θ,t_0} aligns a crack and the width of its basis ψ^e (i.e., w_j) is equal to the width of the crack, $\langle I, m^e_{j,\theta,t_0}\rangle$ will return the maximum; and the parameters j^* and θ^* can be obtained by

$$(j^*, \theta^*) = argmax\left|\left\langle I, m^e_{j,\theta,t_0}\right\rangle\right|(j \in J, \theta \in [0, 2\pi]). \tag{9}$$

As shown in Fig. 5c, when $m^e_{j^*,\theta^*,t_0}$ aligns the crack, w_{j^*} is equal to the width of the crack. For a pixel p_m (black point) with a crack measure greater than zero, the real crack pixels are the red line of direction θ^* and length w_{j^*} that crosses p_m (Fig. 5). Assign "1″ to the pixel of all the lines, and assign "0" to the remaining pixels, an initial crack map is obtained (Fig. 6c). Inspired by Reisenhofer and King (2019), the crack measure improves the process of image convolution and threshold selection so that most background pixels are eliminated while cracks are retained. More details about the parameters can be found in Reisenhofer and King (2019).



(a) (b) (c) (d) (e) (f) (g) (h)

Fig. 4. Function images and convolutions, where $a = \sqrt{2}$ and $n = 1000$: (a) ideal 1D crack; (b) w_0; (c) and (d) $f^o_{x_0=0,j}(x)$ and $f^e_{x_0=0,j}(x)$ with the ideal crack, respectively; (e) results of $\langle C, f^o_{x_0,j}(x)\rangle$, which are zeros for all j when $x_0 = 0$; (f) results of $\langle C, f^e_{x_0,j}(x)\rangle$, when $x_0 = 0$, the functions are aligned with the crack, the width of the ideal crack is equal to w_2, and the maximum can be obtained by $\langle C, f^e_{x_0=0,j=2}(x)\rangle$; (g) absolute value of (e); (h) results of $\langle C, f^e_{x_0,j}(x)\rangle - \left|\left\langle C, f^o_{x_0,j}(x)\right\rangle\right|$, when $x_0 = 0$, the cusps can be obtained for all j.

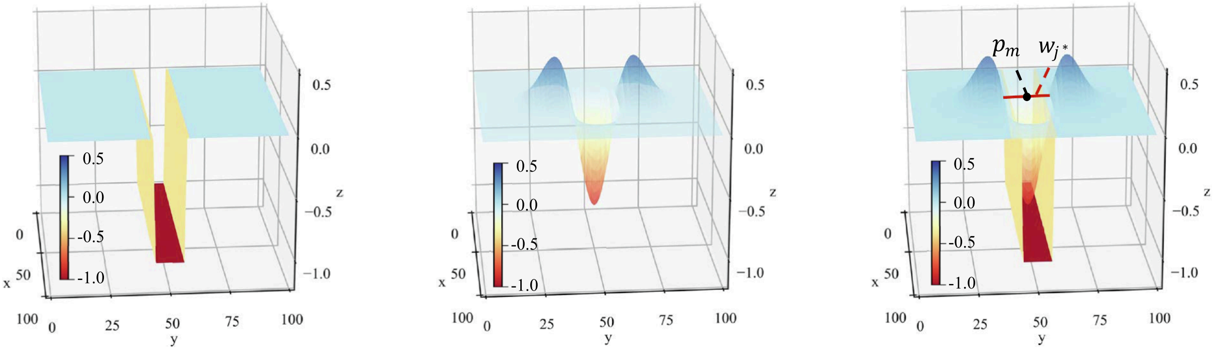
The crack measure directly obtains a univocal and continuous crack map (Fig. 6c). Furthermore, it is contrast-invariant (Reisenhofer and King, 2019), making it unaffected by variations in light and shadows. It also mitigates the ambiguity often associated with binarization methods relying on edge detection algorithms.

5. Optimal crack detection using structure lines

The crack measure, which generates a crack map based on local information, may lead to two types of errors: 1) the true crack cannot be detected successfully (false negative), and 2) the detected result is not the true crack (false positive). To globally optimize the crack map, we streamline it using structure lines and control the errors: trend-guided dilatation for false negative reduction and semantic-guided pruning for false positive elimination.

5.1. Structure-line abstraction

The crack map comprises disorganized pixel sets, which is inconvenient for mathematical analysis (Fig. 6c). To streamline the crack map for global optimization, we thin the crack map as one-pixel-wide

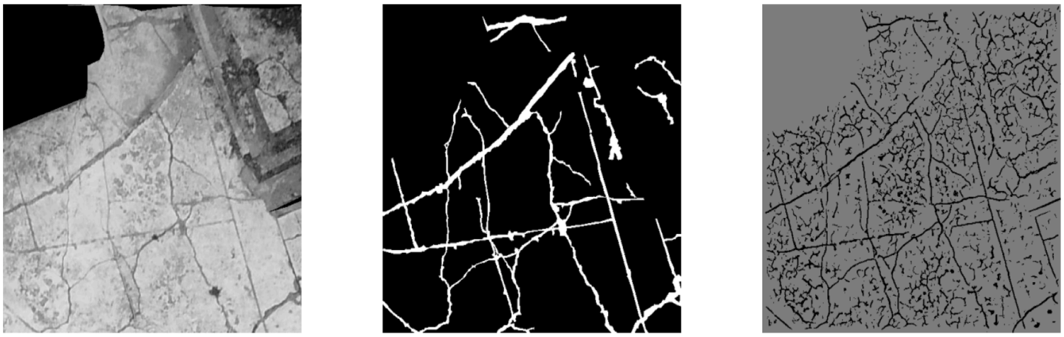


(a)

(b)

(c)

Fig. 5. Details of the crack measure. x and y are the coordinates of the pixel, and z is its grayscale. (a) Ideal crack; (b) function image of $m^{e}_{j^*,\theta^*,t_0}$; (c) superposition of (a) and (b).



(a)

(b)

(c)

Fig. 6. Crack map generated by CDSL. (a) Original image; (b) ground truth; (c) crack map calculated by crack measure. For (c), to increase visibility, we set the pixels of the background and cracks to “128” and “0”, but “0” and “1” in the calculation.

(a)

(b)

Fig. 7. (a) Skeleton and (b) structure lines of Fig. 6c.

skeletons (Hilditch, 1983) (Fig. 7a) and connect adjacent key points (the terminal point and intersection[3]) based on their connectivity to obtain the initial structure lines (Fig. 7b).

A crack map represented through structure lines can effectively convey the attributes of cracks, including their distribution, shape, growth trend and topological relationships. As lines constitute fundamental and standardized elements, various analytical operations can be directly applied to structure lines. In contrast to Fig. 6c, Fig. 7b presents the crack map in an organized format utilizing structure lines.

5.2. *Guided dilatation*

Based on the properties described in Section 3, we used structure lines to express the growth trend of a crack. As shown in Fig. 8, the structure-line of C_i consists of the terminal point $p^t_{C_i}$ and key point $p^k_{C_i}$, we consider the possible growth area of C_i to be the sector within the 5 pixels neighborhood of $p^t_{C_i}$ with the symmetry axis of $\overrightarrow{p^k_{C_i}p^t_{C_i}}$, and its opening angle less than $\pi/2$ (i.e., the sector $Ap^t_{C_i}B$); and C_i should be dilatated by connecting with C_j when they are “similar.”

---

[3] The terminal point and intersection are the points with only one non-zero neighbor and more than two non-zero neighbors on the skeletons, respectively. When there are several adjacent intersections, the one with the most “01 patterns” (Cheng et al., 2017) is chosen.

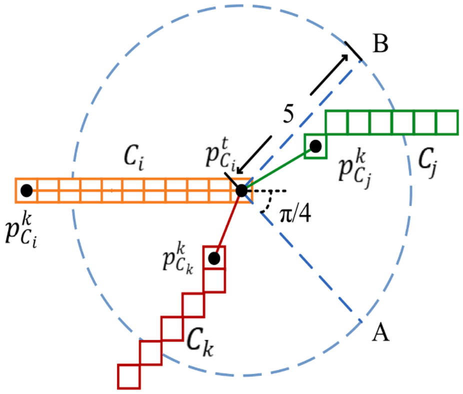


Fig. 8. Crack dilatation.

The dilatation process is expressed as follows: Let p be a key point of a crack and $P(p)$ be a set that consists of the surrounding pixels of p on the crack where p is located, we define the following three functions:

$$f_d(p_0, p_1) = \begin{cases} True, & |\overrightarrow{p_0 p_1}| \le 5 \\ False, & others \end{cases}, \tag{10}$$

$$f_\tau(p_0, p_1, p_2) = f_\tau(\tau) = \begin{cases} True, & \pi - \tau \le \pi/2 \\ False, & others \end{cases}, \tag{11}$$

$$T(P(p_0), P(p_1)) = \begin{cases} True, & consistent \\ False, & different \end{cases}, \tag{12}$$

where

$$\tau(p_0, p_1, p_2) = \begin{cases} \angle p_0 p_1 p_2, & 0 \le \angle p_0 p_1 p_2 \le \pi \\ 2\pi - \angle p_0 p_1 p_2, & others \end{cases}, \tag{13}$$

T is a t-test (Student, 1908), which is easy to perform using a public code. For each $p^t_{C_i}$, if there is a $p^k_{C_j}$ that makes $f_d\left(p^t_{C_i}, p^k_{C_j}\right)$, $f_\tau\left(p^k_{C_i}, p^t_{C_i}, p^k_{C_j}\right)$, and $T\left(P\left(p^t_{C_i}\right), P\left(p^k_{C_j}\right)\right)$ all True, dilatate C_i by connecting $p^t_{C_i}$ and $p^k_{C_j}$. For example, in Fig. 8, since $\overrightarrow{p^t_{C_i} p^k_{C_j}}$ is within π/4 of $\overrightarrow{p^k_{C_i} p^t_{C_i}}$, and the distance $\left|\overrightarrow{p^t_{C_i} p^k_{C_j}}\right|$ is less than 5 pixels, $p^t_{C_i}$ and $p^k_{C_j}$ are further validated by the t-test for connection; C_k is rejected directly since $p^k_{C_k}$ is outside of the sector $Ap^t_{C_i}B$.

We can now acquire a guided dilatation crack map (Fig. 9a), followed by the removal of sub-cracks with short skeletons. As shown in Fig. 9b, the structure lines of the crack map after guided dilatation and length filtration, exhibit greater continuity and conciseness compared to those in Fig. 7b.

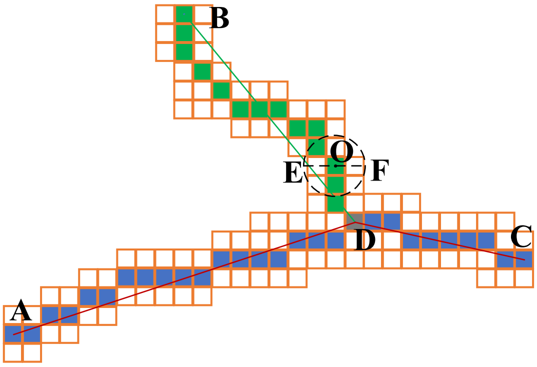


Fig. 10. Crack restoration, where the solid boxes are the skeleton and the empty boxes are the initial crack map.

5.3. *Guided pruning*

As local information alone may not effectively differentiate cracks from interferences or separate superimposed cracks, we hierarchize each sub-crack with semantic guidance of structure lines to segment cracks and prune hierarchical cracks to reduce the false positives. The hierarchization rules are as follows: 1) structure lines should be maximally extended; 2) subordinate structure lines must not contain repeated segments either within themselves or with their superior; and 3) adjacent structure lines should align within their mutual directional sector (Fig. 8).

For each sub-crack, the shortest paths between key points were calculated by considering the structure line as paths (Floyd, 1962); these paths were sorted in descending order by their lengths. Let $\chi(a,b) = \{p_i | i = a, 1, 2, \cdots, b\}$ be the shortest path from p_a to p_b; let $\mathrm{X} = \{\chi_i | i = 0, 1, \cdots, n\}$ be the set of shortest paths between all the points of a crack; let $\Phi = \left\{\phi_j | j = 0, 1, \cdots, n\right\}$ be the set that stores the paths that satisfy the rules; the process of hierarchization can be expressed by

$$f_h(\mathrm{X}, \Phi) = f_T(\chi_i) \cap f_{RE}(\chi_i, \Phi)_{\chi_i \in \mathrm{X}, i=0,1,\cdots,len(\mathrm{X})} \Rightarrow \begin{cases} if\ \ True, & append\ \ \chi\ \ to\ \ \Phi \\ else, & continue \end{cases}, \tag{14}$$

where

$$f_{RE}(\chi_i, \Phi)_{\chi_i \in \mathrm{X}} = \bigcap_{j=0}^{len(\Phi)} f_{re}\left(\chi_i, \phi_j\right)_{\phi_j \in \Phi}, \tag{15}$$

$$f_{re}\left(\chi_i, \chi_j\right) = \begin{cases} True, & no\ \ repeated\ \ segments \\ False, & others \end{cases}, \tag{16}$$

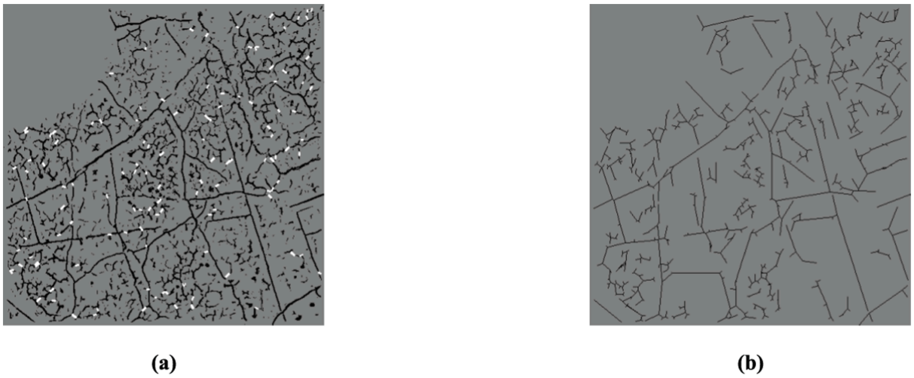


Fig. 9. Guided dilatation with structure-lines. (a) Guided dilatation result of Fig. 6c, where the white pixels are the dilatation pixels. (b) Structure lines of (a) after length filtration.

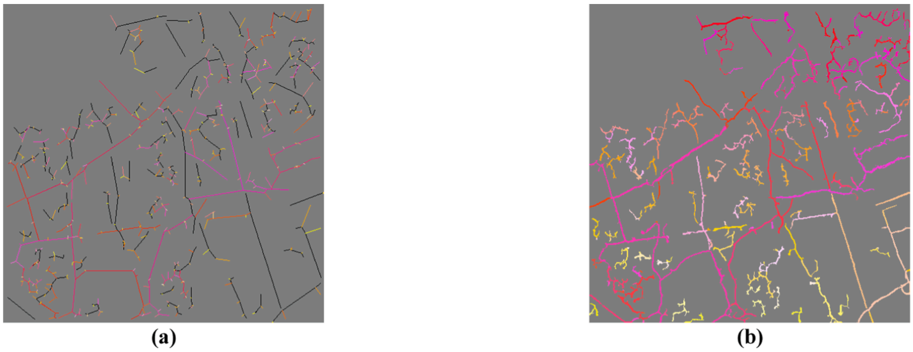


Fig. 11. (a) Hierarchical structure lines, where the black lines are the primary structure lines and the others are the subordinate structure lines of the sub-cracks. (b) Hierarchical crack map corresponding to (a).

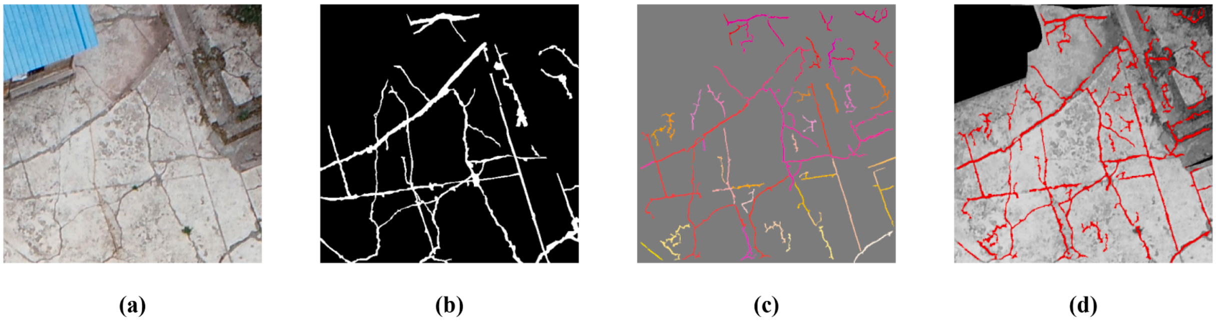


Fig. 12. Guided pruning result. (a) Original image; (b) ground truth; (c) final result of CDSL; and (d) overlay of (a) and (c). Note that (c) has fewer false positives than Fig. 11b.

and

$$f_{\mathrm{T}}(\chi_i)_{\chi_i \in \mathrm{X}} = \bigcap_{i=0}^{len(\chi)-2} f_\tau(p_i, p_{i+1}, p_{i+2})_{p_i \in \chi_i}. \tag{17}$$

(15) and (16) check whether the superior and subordinate structure lines are repeated; (17) check whether all adjacent structure lines of χ_i are located in each other's trend sector. The final hierarchization result is the reverse order of Φ.

After hierarchization, each sub-crack and skeleton were transformed into multiple hierarchical cracks corresponding to the hierarchical structure lines. As shown in Fig. 10, by locating a circle O that traverses along the skeleton and simultaneously touches "0″ pixels with its diameter ($\overline{EF}$), and subsequently extracting "1" pixels from the crack map within the circle, we can restore the hierarchical crack (Fig. 11). Each hierarchical crack is considered an individualized unit.

Let R_{C_j} be the enclosing rectangle of hierarchical crack C_j; let $A_r(R_{C_j})$ be the aspect ratio of R_{C_j}; and let $S_r(R_{C_j})$ be the area ratio of C_j to R. The filter functions are defined as

$$f_{A_r}(C_j, A_{thr}) = \begin{cases} True, & A_r(R_{C_j}) > A_{thr} \\ False, & others \end{cases}, \tag{18}$$

$$f_{S_r}(C_j, S_{thr}) = \begin{cases} True, & S_r(R_{C_j}) < S_{thr} \\ False, & others \end{cases}. \tag{19}$$

If both (18) and (19) are false, we regard C_j as a false positive and pruned to obtain the final result (Fig. 12). To sum up, four important parameters need to be set in CDSL, namely, J, β, S_{thr}, and A_{thr}.

6. Experimental results and discussion

To assess the performance of CDSL, we conducted a comparative analysis with four open-source methods: CrackForest (Shi et al., 2016), PYNQ (Zhang et al., 2020), DAUNet (Polovnikov et al., 2021), and TOPO-Loss combined with TernausNet (TPTN) (Pantoja-Rosero et al., 2022). CrackForest and PYNQ rely on the edge detection algorithm and threshold segmentation, respectively, and were employed to evaluate the performance of methods utilizing different binarization algorithms. In contrast, DAUNet and TPTN represent state-of-the-art deep learning methods designed for crack detection in complex backgrounds.

We employed three datasets of escalating complexity to evaluate CDSL. These datasets encompassed scenarios involving a single scene with multiple images, multiple scenes (i.e., concrete surfaces and asphalt pavement), and UAV images. Each dataset was randomly divided into training, validation, or testing subsets in a 6:2:2 ratio. The dataset specifications are as follows:

FCN-set (Yang et al., 2018**)** comprised 776 images of concrete cracks and ground-truths. Each image had an approximate size of 300 × 300 pixels. The images exhibited variations in crack widths, quantities, and backgrounds, thereby enhancing the complexity of crack detection.

Deepcrack (Liu et al., 2019**)** comprises 537 crack images with manually annotated labels of size 544 × 384 pixels. The dataset encompasses two distinct scene types: concrete surfaces and asphalt pavements, thereby posing a challenge to the method's generalization capability.

UAV-set was obtained utilizing a UAV employed for extensive landslide inspections. This dataset encompassed 100 images of size 256 × 256 pixels, and we manually marked the ground truth. It is difficult to obtain data in the landslide area and the background of images is complicated, which requires the method to detect cracks effectively with limited and complicated data. We have preprocessed the images to cover conspicuous buildings[1].

We used precision, recall, and F1-score to evaluate performance in crack detection:

$$precision = \frac{TP}{TP + FP},$$

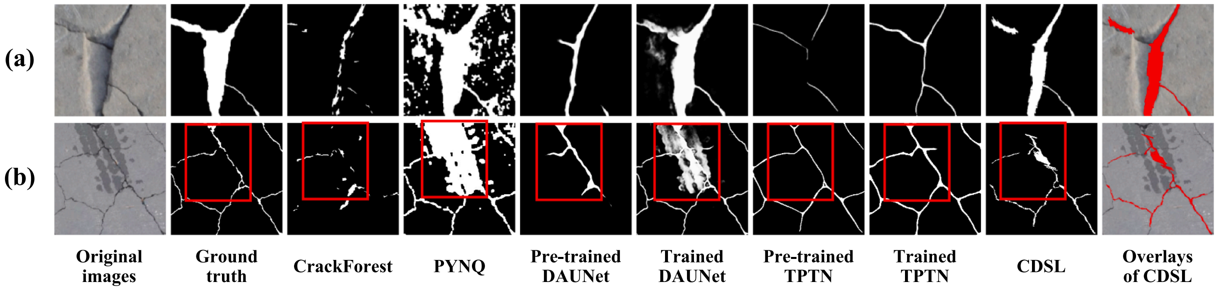


Fig. 13. Results using FCN-set.

Table 1
Results using FCN-set.

Method	Precision (%)	Recall (%)	F1-Score (%)	Time (s/image)
CrackForest (Shi et al., 2016)	59.3	66.6	62.8	0.24
PYNQ (Zhang et al., 2020)	21.1	**99.3**	34.8	0.01
DAUNet (Polovnikov et al., 2021)	97.4 / 85.0	63.6 / **97.1**	76.9 / **90.7**	0.06
TPTN (Pantoja-Rosero et al., 2022)	**98.6** / **97.5**	49.3 / 67.3	65.7 / 79.6	0.28
CDSL (ours)	92.9	75.4	**83.2**	0.37

Table 2
The results of Deepcrack.

Method	Precision (%)	Recall (%)	F1-Score (%)	Time (s/image)
CrackForest (Shi et al., 2016)	21.2	75.5	33.1	0.37
PYNQ (Zhang et al., 2020)	15.5	**95.6**	26.6	0.01
DAUNet (Polovnikov et al., 2021)	63.0 / **95.1**	81.2 / **89.9**	71.0 / **92.3**	0.07
TPTN (Pantoja-Rosero et al., 2022)	**98.6** / 91.4	54.8 / 73.9	70.5 / 81.7	0.30
CDSL (ours)	81.9	82.7	**82.3**	0.61

$$recall = \frac{TP}{TP + FN},$$

$$F1 = 2\cdot\frac{precision\cdot recall}{precision + recall}, \tag{20}$$

where *TP*, *FP*, and *FN* are the number of true positives, false positives, and false negatives, respectively, with a detected pixel being considered a *TP* when it is closer than three pixels from the ground truth. Obtaining ample training data can be challenging in certain scenarios, and traditional methods do not require extensive training. Hence, for deep learning methods, we provided two scores for the results of pre-trained and after-trained models (pre-trained model/after-trained model) to compare the cross-scene application capabilities of different methods, and show the performance improvement of deep learning methods with adequate training and the performance in complex scenes with few samples.

Each method was executed following the respective authors' recommendations. For CDSL, we set the parameters $\beta = 15$, $S_{thr} = 0.25$, and $A_{thr} = 4$ as default, and set $J = [16, 26]$ for the two open-source datasets and set $J = [22, 26]$ for the UAV-set, which exhibits distinct features compared to the open-source datasets. In the subsequent subsections, we present the performance and outcomes of each method on individual datasets in Sections 6.1 to 6.3. Subsequently, in Section 6.4, we delve into the strengths and limitations of each method based on the results. Finally, we address the parameters in Section 6.5.

6.1. Evaluation using FCN-set

CDSL was evaluated using the FCN-set, which includes a substantial collection of concrete crack images featuring diverse backgrounds. A qualitative comparison is provided in Fig. 13, while a quantitative assessment is presented in Table 1. CDSL achieved the third-highest precision (92.9 %), the second-highest recall (75.4 %), and the highest F1-score (83.2 %).

The results from CrackForest detected part of the cracks (Fig. 13a), resulting in low precision and recall. PYNQ detected most of the cracks; however, its results mistakenly identified water stains as cracks, leading to a high recall but the lowest precision (Fig. 13a and b). The performance of DAUNet and TPTN exhibited significant improvement following training on the dataset. Nevertheless, in the case of the after-training DAUNet, the results still included water stains, which contributed to decreased precision (Fig. 13b). While CDSL was also somewhat affected by the water stain (Fig. 13b), its results were more comprehensive and comparatively more accurate (Fig. 13a and b).

6.2. Evaluation using Deepcrack

We employed DeepCrack to evaluate CDSL using images of concrete surfaces and asphalt pavement. The quantitative evaluation is shown in Table 2. We selected the representative images showcased in Fig. 14 for qualitative comparison. CDSL secured the second position in both precision (81.9 %) and recall (82.7 %), while achieving the highest F1-score (82.3 %), which satisfied the requirement for crack detection in different scenarios.

The outcomes from CrackForest exhibited a notable presence of false positives (Fig. 14a) and misidentified areas with distinct edges as cracks (Fig. 14b), resulting in high recall but low precision. PYNQ successfully detected high-contrast cracks against a bright background (Fig. 14a) but faced challenges in images with lower overall grayscale. Some white patches appeared in the results of the pre-trained DAUNet but were resolved in the after-training DAUNet's results (Fig. 14a). TPTN demonstrated insensitivity to variations in crack width, leading to reduced recall. As shown in Fig. 14, CDSL exhibited adaptability to changes in crack width and remained unaffected by landmarks, resulting in improved performance.

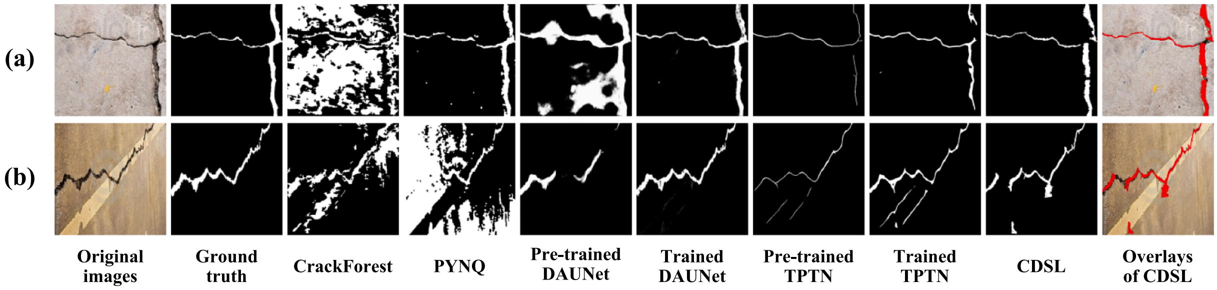


Fig. 14. Results using Deepcrack.

Table 3
Results using UAV-set.

Method	Precision (%)	Recall (%)	F1-Score (%)	Time (s/image)
CrackForest (Shi et al., 2016)	10.4	78.3	18.3	0.21
PYNQ (Zhang et al., 2020)	20.7	73.0	32.2	0.01
DAUNet (Polovnikov et al., 2021)	30.1 / 69.7	23.2 / **70.9**	26.2 / **70.3**	0.15
TPTN (Pantoja-Rosero et al., 2022)	**97.4** / **95.5**	43.8 / 52.8	60.4 / 68.0	0.31
CDSL (ours)	80.5	**86.2**	**83.3**	0.35

6.3. *Evaluation using UAV-set*

CDSL underwent verification across diverse scenes, with UAV images chosen as the test dataset, given their tendency to exhibit complex backgrounds. Due to the intricate nature of the UAV dataset and its limited size, we employed the pre-trained models provided by their respective authors as initial parameters to train and finetune deep learning models on the UAV dataset. Quantitative and qualitative evaluations are presented in Table 3 and Fig. 15, respectively. Notably, CDSL achieved an F1-score at least 22.9 % higher than that of the other four methods (a 13.0 % improvement when compared to the F1-Score of after-trained models).

The results from CrackForest exhibited relatively poor performance. PYNQ managed to detect a few cracks (the red box in Fig. 15c). DAUNet successfully identified thick cracks (Fig. 15b) but overlooked thinner ones (Fig. 15a and c). TPTN excelled in detecting the main stems of cracks but struggled with lighter-colored cracks (the red box in Fig. 15a) and areas with multiple cracks (the red box in Fig. 15c). As for CDSL, it mistakenly identified a portion of the curbstone as cracks, leading to a decrease in its precision (Fig. 15c). Nevertheless, CDSL effectively detected lighter-colored and partially grass-covered cracks (Fig. 15a). While it was influenced by spots and stains, it still managed to detect a greater number of cracks (Fig. 15b), and its results exhibited comprehensiveness when compared to those produced by the other four methods (Fig. 15c).

Table 4
Score comparison of Fig. 18 between different *J*.

J	Precision (%)	Recall (%)	F1-Score (%)
$[22, 26]$	34.6	85.8	49.3
$[16, 26]$	20.1	96.7	33.3

6.4. *Discussion of results*

The experiment showed the performance, shortcomings, and advantages of different methods. CrackForest, which relies on an edge detection algorithm, is affected by the quality of edges within the image. It excels in crack detection when the image features distinct edges for cracks (Fig. 13a). However, in cases where crack edges are less discernible (Fig. 15a) or when other interferences exhibit pronounced edges (Fig. 14b), its detection performance is less reliable. PYNQ based on global threshold segmentation offers simplicity and speed, but the fixed threshold limits its applicability across various scenarios and resulting in inconsistent performance across different images (Fig. 13, Fig. 14, and Fig. 15).

DAUNet and TPTN benefit from their complex and multiparameter models, enabling them to consider both global and local information. Their performance saw significant enhancements when trained on extensive data (Fig. 13 and Fig. 14, Table 1, and Table 2). Nevertheless,

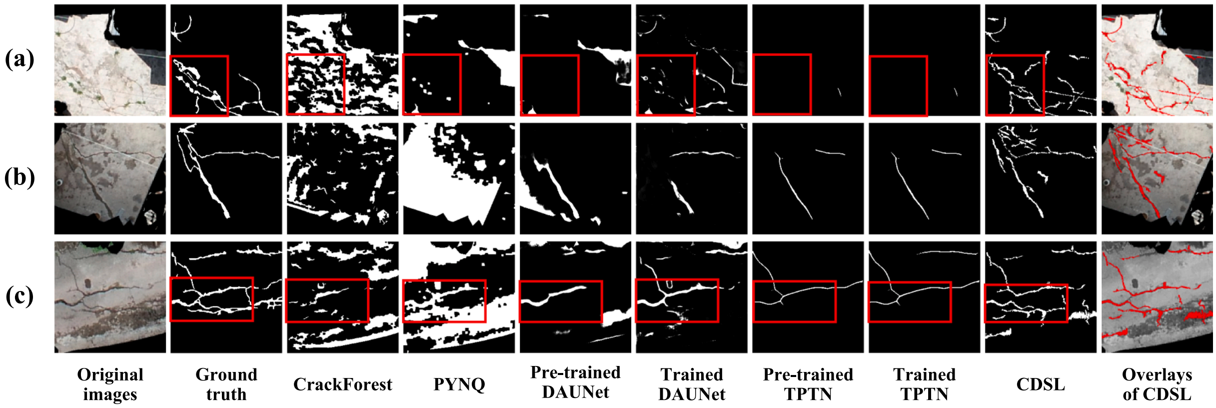
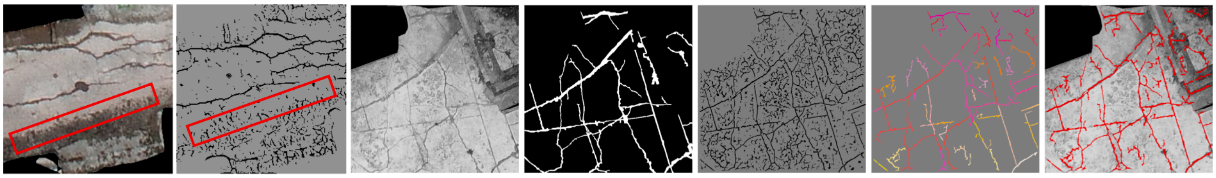


Fig. 15. Results using UAV-set.



(a) (b) (c) (d) (e) (f) (g)

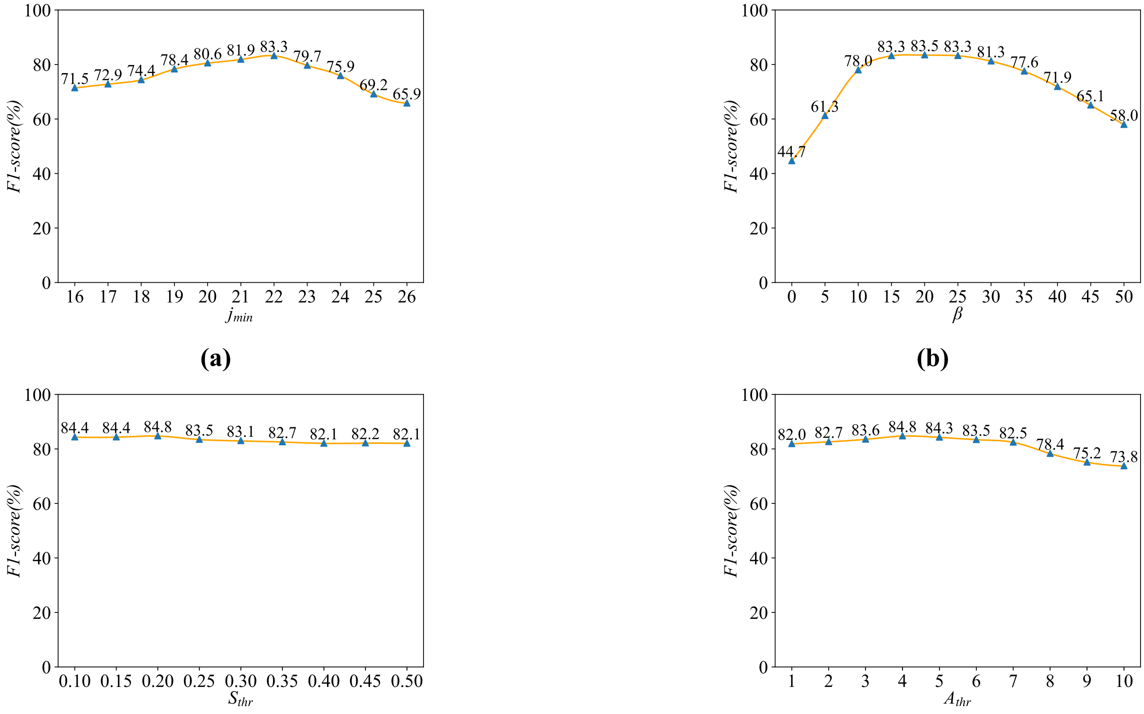
Fig. 16. Intermediate results for CDSL: (a) UAV image with obvious edges; (b) crack map of (a) detected by CDSL; (c) UAV image with some interferences; (d) ground truth; (e) crack map of (c) detected by CDSL; (f) final detection result; and (g) overlay.

Table 5
w_j for different j in pixels.

j	0	1	…	16	17	18	19	20	21	22	23	24	25	26
w_j	14,142	10,000	…	55	39	28	20	14	10	7	5	3	2	1

the efficacy of deep learning methods is contingent upon the representativeness of the training set, i.e., how closely it mirrors the characteristics of the test set. This limitation restricts their applicability in scenes with limited data diversity. The UAV dataset, obtained from a densely populated landslide-prone area, presented challenges in obtaining an extensive dataset due to logistical constraints. Owing to the inherently complex backgrounds of UAV images, even though the performance of the two deep learning methods displayed notable improvements post-training, they still struggled to yield consistently reliable detection results (Fig. 15 and Table 3).

CDSL is constrained by its rule-driven approach, which may lead to occasional detection errors (Fig. 15). Its computational time is slightly slower than other methods due to the multi-step optimization process (Table 1, Table 2, and Table 3). However, CDSL also offers distinctive advantages. Firstly, CDSL's crack measure does not detect edges but directly obtains potential crack pixels to generate a continuous and unambiguous crack map based on local information (Fig. 16a and b). Consequently, it exhibits reduced sensitivity to edge-related interference. The scale function of the crack measure enables CDSL to adapt to varying crack widths (Fig. 14a). Second, the global optimization of CDSL, which refines binarization results based on trends, aspect ratios, and area ratios of pixel sets, lessens its dependence on a fixed threshold (Fig. 15). Random interferences, such as potholes and leaves, may appear in the binarization results, but they are subsequently filtered out and exert minimal influence on the final detection performance (Fig. 16c-g). Finally, being a traditional method, CDSL is not contingent on the volume of available data, rendering its performance stable even when data are limited. While CDSL's performance on open-source datasets may not surpass that of a trained DAUNet, it consistently produces competitive detection results and notably outperforms trained deep-learning methods on more complex UAV-set (Table 3). Considering the challenges in procuring extensive training data at the early stages of



(a) (b) (c) (d)

Fig. 17. F1-score curves. In each figure, the fixed parameters were as follows. (a) $\beta = 15$, $S_{thr} = 0.25$, $A_{thr} = 4$; (b) $j_{min} = 22$, $S_{thr} = 0.25$, $A_{thr} = 4$; (c) $j_{min} = 22$, $\beta = 20$, $A_{thr} = 4$; (d) $j_{min} = 22$, $\beta = 20$, $S_{thr} = 0.2$.

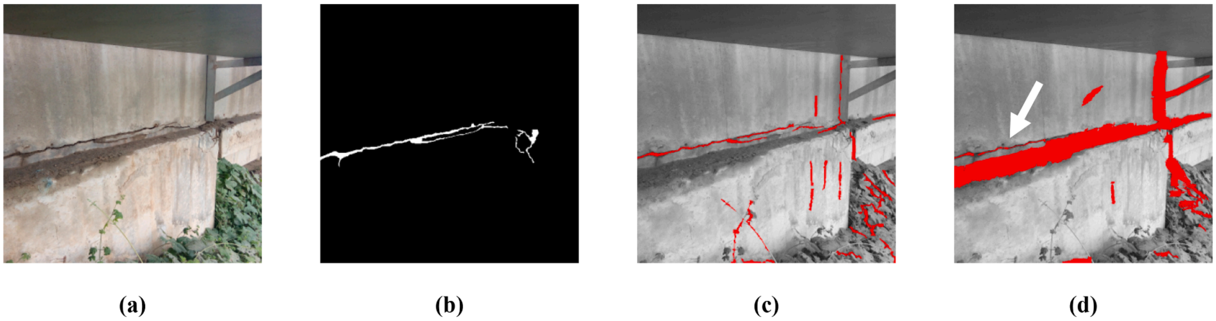


Fig. 18. Results of an image of a thin crack with different J: (a) input image; (b) ground-truth; (c) $J = [22, 26]$; and (d) $J = [16, 26]$.

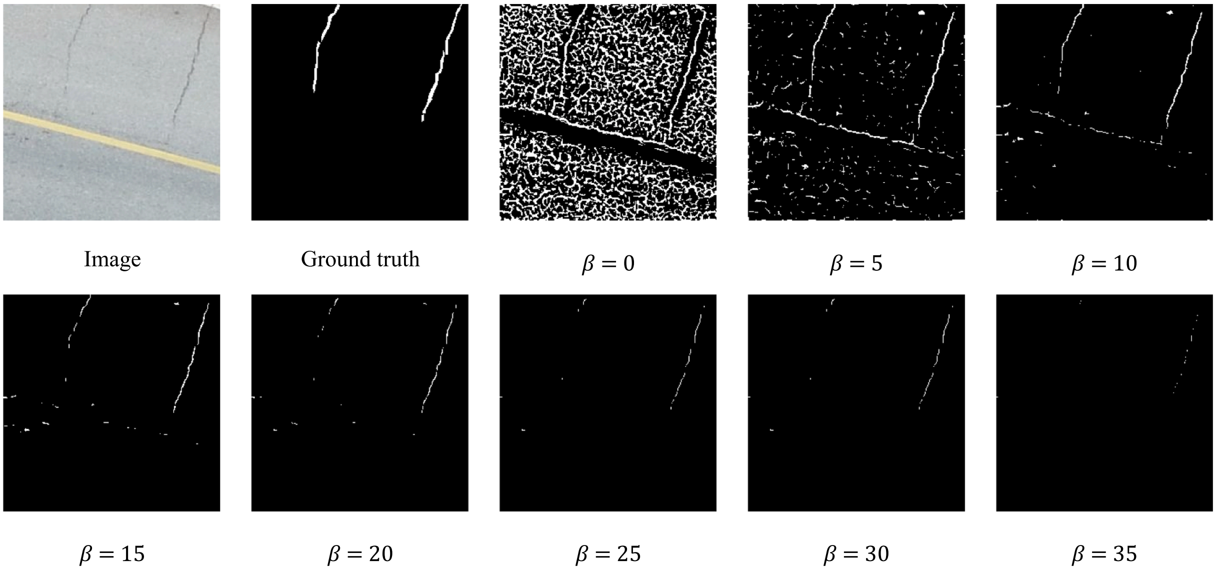


Fig. 19. Crack map of a UAV image calculated by the crack measure for different β.

most application scenarios, CDSL can deliver excellent detection results, and provide initial labels when data volumes become sufficient, thereby alleviating the manual data labeling burden.

From three experiments with a significant number of images, we conclude that CDSL can achieve competitive performance in crack detection from both general and UAV images.

6.5. Method enhancement and parameters discussion

In this section, we discuss the parameters using the UAV-set. For J, its sub-parameter j_{max} was fixed at 26 so that CDSL could detect cracks with a minimum width of 1 pixel (Table 5); thus, we only discussed j_{min}, which controls the maximum width of detectable cracks. The process was as follows: first, we fixed $\beta = 15$, $S_{thr} = 0.25$, and $A_{thr} = 4$ to calculate the F1-score of different j_{min}; then, β was tested by assigning j_{min} to the value that achieved a maximum F1-score, while the other two parameters were still fixed; finally, the similar operation was performed for the last two parameters.

In UAV-set, when $j_{min} \geq 19$, the F1-score is significantly improved and the maximum F1-score was achieved when $j_{min} = 22$. False detection may occur when an extremely small j_{min} is used to detect thin cracks. As shown in Fig. 18, a thin crack is covered by a long dark area when we set $j_{min} = 16$. Because the long dark area satisfies the properties described in Section 3, the CDSL cannot distinguish it from a crack. An unsuitable j_{min} will not lead to failure of the method, and CDSL can still detect cracks outside the dark area (white arrows in Fig. 18d). However, the precision can be improved when prior information on the approximate crack width range is obtained (Table 4).

The approximative widths of cracks corresponding to j were calculated via $w_j = w_0 \times a^{-j}$, where $w_0 = \sqrt{2}/b \approx 14142$ (Table 5). For close-taken open-source datasets, because their widths varied significantly, $j_{min} = 16$ was selected to enable CDSL to adapt to changes in crack width. For UAV images, because they were shot at a long distance, and most cracks were narrow, we set $j_{min} = 22$ to limit CDSL to detect cracks with a large width to proactively avoid errors like Fig. 18.

The parameter β is a contrast-based threshold that determines whether a pixel should be retained. The F1-score curve tended to flatten when $\beta \geq 10$ and decrease when $\beta > 30$ (Fig. 17). When β was too small, the crack map retained too many details; conversely, when β was too big, low contrast pixels of a crack were excessively eliminated (Fig. 19). Therefore, we generally choose β within 10 to 30.

Parameters S_{thr} and A_{thr} were used to prune the crack map. The F1-score varied slightly when $0.1 \leq S_{thr} \leq 0.5$ or when $A_{thr} \leq 7$, but decreased obviously when $A_{thr} > 7$ (Fig. 17c). The parameter A_{thr} reflects the degree to which a crack grows in the same direction. The longer the crack, the less likely it is to appear, and an overlarge A_{thr} will overmuch eliminate true cracks.

7. Conclusion

In this paper, we present a novel crack detection method called Crack Detection with Structure Line (CDSL), which leverages both geometric and textural characteristics inherent to cracks. CDSL first generates a crack map based on local information and extracts its structure lines.

These structural lines are then utilized to standardize the format of the crack map. Subsequently, guided dilatation and pruning operations are applied with the aid of these structure lines to achieve global optimization of the crack map. We conducted a comparative evaluation of CDSL's performance against four contemporary methods using extensive and complex image datasets. Our results affirm that CDSL is robust and can provide reliable results for crack detection in civil engineering applications. Furthermore, our proposed crack measure holds potential for integration into subsequent crack-detection methodologies, while our innovative structure-line guidance strategy offers valuable insights for the development of future linear feature-detection techniques. In future studies, we intend to explore the integration of structured geometric constraints into deep learning approaches to further enhance the efficacy of crack detection results.

CRediT authorship contribution statement

Yongjun Zhang: Conceptualization, Methodology, Project administration. **Yixin Lu:** Software, Validation, Writing – original draft. **Yansong Duan:** Project administration. **Dong Wei:** Formal analysis, Conceptualization, Methodology. **Xianzhang Zhu:** Data curation. **Bin Zhang:** Validation. **Bohui Pang:** Project administration.

Declaration of Competing Interest

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

Data availability

No data was used for the research described in the article.

Acknowledgements

This work was supported by The Science and Technology Major Project of Hubei Province under Grant (2021AAA010-3), China Yunnan Province Mahongqi Academician Free Exploration Project (202005AA160011), and Huaneng Lancang River Hydropower Inc. Science and Technology Project (HY2021/D11).

References

Ai, D., Jiang, G., Kei, L.S., Li, C., 2018. Automatic pixel-level pavement crack detection using information of multi-scale neighborhoods. IEEE Access 6, 24452–24463. https://doi.org/10.1109/ACCESS.2018.2829347.

Ai, D., Jiang, G., Lam, S.-K., He, P., Li, C., 2023. Computer vision framework for crack detection of civil infrastructure—A review. Eng. Appl. Artif. Intel. 117, 105478 https://doi.org/10.1016/j.engappai.2022.105478.

Andrushia, A.D., Anand, N., Arulraj, G.P., 2021. Evaluation of thermal cracks on fire exposed concrete structures using Ripplet transform. Math. Comput. Simul 180, 93–113. https://doi.org/10.1016/j.matcom.2020.07.024.

Ataiwe, T.N., 2023. Using Image Processing for Automatic Detection of Pavement Surface Distress. AJEST 2, 46–52. https://doi.org/10.55145/ajest.2023.01.01.006.

Canny, J., 1986. A computational approach to edge detection. IEEE Trans. Pattern Anal. Mach. Intell. 679–698 https://doi.org/10.1109/TPAMI.1986.4767851.

Chen, Q., Huang, Y., Weng, X., Liu, W., 2021. Curve-based crack detection using crack information gain. Struct. Control Health Monit. 28, e2764. https://doi.org/10.1002/stc.2764.

Cheng, Y.-C., Chen, H., Cheng, B.-C., 2017. Special point representations for reducing data space requirements of finger-vein recognition applications. Multimedia Tools Appl. 76, 11251–11271. https://doi.org/10.1007/s11042-016-3300-y.

Cheng, H., Shi, X., Glazier, C., 2003. Real-time image thresholding based on sample space reduction and interpolation approach. J. Comput. Civ. Eng. 17, 264–272. https://doi.org/10.1061/(ASCE)0887-3801(2003)17:4(264).

Farahani, A., Voghoei, S., Rasheed, K., Arabnia, H.R., 2021. A brief review of domain adaptation. Adv. Data Sci. Inf. Eng. 877–894 https://doi.org/10.1007/978-3-030-71704-9_65.

Feng, H., Ma, L., Yu, Y., Chen, Y., Li, J., 2023. SCL-GCN: Stratified Contrastive Learning Graph Convolution Network for pavement crack detection from mobile LiDAR point clouds. Int. J. Appl. Earth Obs. Geoinf. 118, 103248 https://doi.org/10.1016/j.jag.2023.103248.

Floyd, R.W., 1962. Algorithm 97: shortest path. Commun. ACM 5, 345.

Frangi, A.F., Niessen, W.J., Vincken, K.L., Viergever, M.A., 1998. Multiscale vessel enhancement filtering. In: MICCAI. Springer, pp. 130–137.

Gibert, X., Patel, V.M., Labate, D., Chellappa, R., 2014. Discrete shearlet transform on GPU with applications in anomaly detection and denoising. EURASIP J. Adv. Signal Process. 2014, 1–14. https://doi.org/10.1186/1687-6180-2014-64.

Hilditch, C., 1983. Comparison of thinning algorithms on a parallel processor. Image Vis. Comput. 1, 115–132. https://doi.org/10.1016/0262-8856(83)90063-X.

Ji, S., Zeng, C., Zhang, Y., Duan, Y., 2023. An evaluation of conventional and deep learning-based image-matching methods on diverse datasets. Photogram. Rec. 137–159 https://doi.org/10.1111/phor.12445.

Kanopoulos, N., Vasanthavada, N., Baker, R.L., 1988. Design of an image edge detection filter using the Sobel operator. IEEE J. Solid-State Circuits 23, 358–367. https://doi.org/10.1109/4.996.

Kapur, J.N., Sahoo, P.K., Wong, A.K., 1985. A new method for gray-level picture thresholding using the entropy of the histogram. CVGIP 29, 273–285. https://doi.org/10.1016/0734-189X(85)90125-2.

Li, L., Wang, Q., Zhang, G., Shi, L., Dong, J., Jia, P., 2018. A method of detecting the cracks of concrete undergo high-temperature. Constr. Build. Mater. 162, 345–358. https://doi.org/10.1016/j.conbuildmat.2017.12.010.

Li, C., Zhu, F., Guo, B., Wang, Z., Jiang, X., Wang, J., Liao, X., 2022. Power Line Extraction and Obstacle Inspection of Unmanned Aerial Vehicle Oblique Images Constrained by the Vertical Plane. Photogram. Rec. 37, 306–332. https://doi.org/10.1111/phor.12422.

Liu, Z., Suandi, S.A., Ohashi, T., Ejima, T., 2002. Tunnel crack detection and classification system based on image processing. In: Mach. vis. Appl. Ind. Insp. x. SPIE, pp. 145–152.

Liu, Y., Yao, J., Lu, X., Xie, R., Li, L., 2019. DeepCrack: A deep hierarchical feature learning architecture for crack segmentation. Neurocomputing 338, 139–153. https://doi.org/10.1016/j.neucom.2019.01.036.

Ma, L., Li, J., 2022. SD-GCN: Saliency-based dilated graph convolution network for pavement crack extraction from 3D point clouds. Int. J. Appl. Earth Obs. Geoinf. 111, 102836 https://doi.org/10.1016/j.jag.2022.102836.

Ma, X., Yue, D., Liu, R., Wang, R., Zhu, S., Wang, M., Yu, J., 2022. Road Curbs Extraction from Mobile Laser Scanning Point Clouds with Multidimensional Rotation-Invariant Version of the Local Binary Pattern Features. Photogram. Rec. 37, 410–434. https://doi.org/10.1111/phor.12431.

Nguyen, H.-N., Kam, T.-Y., Cheng, P.-Y., 2018. Automatic crack detection from 2D images using a crack measure-based B-spline level set model. Multidimens. Syst. Signal Process. 29, 213–244. https://doi.org/10.1007/s11045-016-0461-9.

Nong, C.R., Liu, Z.Y., Zhang, J., Zeng, Q.S., 2020. Research on Crack Edge Detection of Aircraft Skin Based on Traditional Inspired Network. In: 2020 2nd ITCA. IEEE, pp. 751–754.

O'Mahony, N., Campbell, S., Carvalho, A., Harapanahalli, S., Hernandez, G.V., Krpalkova, L., Riordan, D., Walsh, J., 2020. Deep learning vs. traditional computer vision, CVC 2019, Volume 1 1. Springer, pp. 128-144.

Otsu, N., 1979. A threshold selection method from gray-level histograms. IEEE Trans. Syst. Man Cybern. 9, 62–66.

Pantoja-Rosero, B., Oner, D., Kozinski, M., Achanta, R., Fua, P., Perez-Cruz, F., Beyer, K., 2022. TOPO-Loss for continuity-preserving crack detection using deep learning. Constr. Build. Mater. 344, 128264 https://doi.org/10.1016/j.conbuildmat.2022.128264.

Pastucha, E., Puniach, E., Gruszczyński, W., Ćwiąkała, P., Matwij, W., Midtiby, H.S., 2022. Relative Radiometric Normalisation of Unmanned Aerial Vehicle Photogrammetry-based RGB Orthomosaics. Photogram. Rec. 37, 228–247. https://doi.org/10.1111/phor.12413.

Payab, M., Abbasina, R., Khanzadi, M., 2019. A brief review and a new graph-based image analysis for concrete crack quantification. Arch. Comput. Meth. Eng. 26, 347–365. https://doi.org/10.1007/s11831-018-9263-6.

Polovnikov, V., Alekseev, D., Vinogradov, I., Lashkia, G.V., 2021. DAUNet: Deep augmented neural network for pavement crack segmentation. IEEE Access 9, 125714–125723. https://doi.org/10.1109/ACCESS.2021.3111223.

Reisenhofer, R., King, E.J., 2019. Edge, ridge, and blob detection with symmetric molecules. SIAM J. Imag. Sci. 12, 1585–1626. https://doi.org/10.1137/19M1240861.

Ren, M., Zhang, X., Chen, X., Zhou, B., Feng, Z., 2023. YOLOv5s-M: A deep learning network model for road pavement damage detection from urban street-view imagery. Int. J. Appl. Earth Obs. Geoinf. 120, 103335 https://doi.org/10.1016/j.jag.2023.103335.

Ridler, T., Calvard, S., 1978. Picture thresholding using an iterative selection method. IEEE Trans. Syst. Man Cybern. 8, 630–632. https://doi.org/10.1109/TSMC.1978.4310039.

Ronneberger, O., Fischer, P., Brox, T., 2015. U-net: Convolutional networks for biomedical image segmentation. In: MICCAI. Springer, pp. 234–241.

Safaei, N., Smadi, O., Masoud, A., Safaei, B., 2022. An automatic image processing algorithm based on crack pixel density for pavement crack detection and classification. Int. J. Pavement Res. Technol. 15, 159–172. https://doi.org/10.1007/s42947-021-00006-4.

Shi, Y., Cui, L., Qi, Z., Meng, F., Chen, Z., 2016. Automatic road crack detection using random structured forests. IEEE Trans. Intell. Transp. Syst. 17, 3434–3445. https://doi.org/10.1109/TITS.2016.2552248.

Stehfest, H., 1970. Algorithm 368: Numerical inversion of Laplace transforms [D5]. Commun. ACM 13, 47–49.

Student, 1908. The probable error of a mean. Biometrika 6, 1–25. https://doi.org/10.1093/biomet/6.1.1.

Wang, H., Li, Y., Dang, L.M., Lee, S., Moon, H., 2021. Pixel-level tunnel crack segmentation using a weakly supervised annotation approach. Comput. Ind. 133, 103545 https://doi.org/10.1016/j.compind.2021.103545.

Wang, W., Li, L., Zhang, F., 2023. Crack image recognition on fracture mechanics cross valley edge detection by fractional differential with multi-scale analysis. Signal, Image Video Process. 17, 47–55. https://doi.org/10.1007/s11760-022-02202-6.

Weng, X., Huang, Y., Wang, W., 2019. Segment-based pavement crack quantification. Autom. Constr. 105, 102819 https://doi.org/10.1016/j.autcon.2019.04.014.

Xiao, S., Shang, K., Lin, K., Wu, Q., Gu, H., Zhang, Z., 2023. Pavement crack detection with hybrid-window attentive vision transformers. Int. J. Appl. Earth Obs. Geoinf. 116, 103172 https://doi.org/10.1016/j.jag.2022.103172.

Xie, M., Zhang, X., Li, Y., Han, B., 2023. Automatic detection of thin oil films on water surfaces in ultraviolet imagery. Photogram. Rec. 38, 47–62. https://doi.org/10.1111/phor.12439.

Xu, Z., Guan, H., Kang, J., Lei, X., Ma, L., Yu, Y., Chen, Y., Li, J., 2022. Pavement crack detection from CCD images with a locally enhanced transformer network. Int. J. Appl. Earth Obs. Geoinf. 110, 102825 https://doi.org/10.1016/j.jag.2022.102825.

Yang, X., Li, H., Yu, Y., Luo, X., Huang, T., Yang, X., 2018. Automatic pixel-level crack detection and measurement using fully convolutional network. Comput.-Aided Civ Infrastruct. Eng. 33, 1090–1109. https://doi.org/10.1111/mice.12412.

Zhang, Y., Juan, Q., Zhiling, G., Kuncheng, J., Shiyuan, C., 2020. Detection of road surface crack based on PYNQ. In: 2020 IEEE ICMA. IEEE, pp. 1150–1154.

Zhou, X., Xu, L., Wang, J., 2019. Road crack edge detection based on wavelet transform. In: IOP Conference Series: Earth and Environmental Science. IOP Publishing, p. 032132.

Zhu, X., Liu, X., Zhang, Y., Wan, Y., Duan, Y., 2021. Robust 3-D plane segmentation from airborne point clouds based on quasi-a-contrario theory. IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 14, 7133–7147. https://doi.org/10.1109/JSTARS.2021.3093576.

Zou, Q., Cao, Y., Li, Q., Mao, Q., Wang, S., 2012. CrackTree: Automatic crack detection from pavement images. Pattern Recogn. Lett. 33, 227–238. https://doi.org/10.1016/j.patrec.2011.11.004.