Automation in Construction 123 (2021) 103535

Contents lists available at ScienceDirect

Automation in Construction

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

Robust pixel-wise concrete crack segmentation and properties retrieval using image patches

Yiqing Liu, Justin K.W. Yeoh *

Dept. of Civil and Environment Engineering, National Univ. of Singapore, 117576, Singapore

ARTICLE INFO

Keywords:
Crack segmentation
Crack properties retrieval
Computer vision
Noisy concrete surfaces

ABSTRACT

Crack identification is an essential task in periodic inspection and maintenance of buildings. The application of deep learning based computer vision techniques is increasingly popular, but suffer from challenges of insufficient performance on highly diverse field inspection scenarios as well as a requirement for large amounts of labeled training data. To address these limitations, this paper proposes a robust crack segmentation approach using image patches to detect and support further accurate retrieval of crack properties for integrity assessment. In the proposed approach, a local region-based active contour model is integrated with a convolution neural network and several post-processing morphological operations to derive a segmented crack map. Experimental validation shows significant improvement in terms of accuracy and robustness over previous work. Data labeling requirement is also comparatively lower. This paper enhances the current concrete inspection process, and lays the foundation for more data efficient methods of crack segmentation to be explored.

1. Introduction

Buildings are subject to deterioration due to external environmental factors, excessive usage or overloading. Consequently, the aging building stock potentially poses serious hazards to both public property and personnel safety. In Singapore, a series of recent mishaps has led to increased public concern over the condition of concrete building facades [1]. These recent cases highlight the urgent need to address the hazards posed by defective facades to the general public. In building maintenance, the presence of cracks is one of the most common and essential indicators in the diagnosis of building conditions [2,3]. Early detection of cracks through relevant effective measurement may prevent possible damage and collapse.

In current practice, routine crack inspection is carried out manually through visual examination of defects on the surfaces of building elements by professionals, making the entire procedure labor-intensive, time-consuming and error-prone. To improve efficiency as well as accuracy during inspection, there is an urgent need to explore automating the inspection process and transforming it to become less labor dependent. Adopting automation and artificial intelligence to aid in visual crack identification on surfaces will substantially enhance the current inspection regime, thereby increasing the accuracy of early identification of building failure. Examples of such automation and artificial intelligence include the application of systematic image acquisition and computer vision techniques [3]; these hold great promise in introducing a more rigorous and scientific approach to perform crack inspection.

In order to overcome the challenges of manual inspection processes, various digital technologies have been investigated to achieve crack detection and feature extraction in civil infrastructures, such as edge detectors (i.e. Canny filter, Sobel filter, etc.) [4,5], percolation processing [5–7], clustering-based methods [8] and 3D scene reconstruction [9–11]. These techniques are essential for effective and objective crack analysis and evaluation. To enable engineering assessment of the building façade, further crack properties quantification such as crack width and length measurement are necessary for decision making, and these are in turn, highly reliant on the detected cracks using the techniques described above.

While the aforementioned techniques have sought to address the limitations of the current crack inspection regime, the performance in real world applications is still questionable. Highly variable realistic contexts make image-based crack detection more challenging. Images captured commonly include various sources of noise such as blebs, illumination variations, shadings, unintended surface objects, surface stains and complicated background textures. The existence of noise on the surfaces will cause unexpected interference on extracting crack from images using image processing techniques. Recent studies have

* Corresponding author.
E-mail addresses: liuyiqing@u.nus.edu (Y. Liu), justinyeoh@nus.edu.sg (J.K.W. Yeoh).

https://doi.org/10.1016/j.autcon.2020.103535
Received 30 May 2020; Received in revised form 19 December 2020; Accepted 23 December 2020
Available online 3 January 2021

developed robust crack segmentation methods based on deep learning techniques [12–15]. However, these methods suffer from low data efficiency that in the context of this paper, refers to a requirement of large pixel-level labeled datasets for training the algorithms. These are known to be labor-intensive and time-consuming to prepare [16,17].

Specifically, this paper aims to address the issue of low data efficiency, by proposing a hybrid approach combining edge detection and deep learning that uses labeled image patches, which are more easily labeled, for training instead. Using this approach, a Convolution Neural Network (CNN) trained from labeled image patches is adopted to initialize multiple initial contours. These multiple initial contours are used to retrieve crack regions, so that a local region-based active contour model can be used to develop a crack map from the pixel-level crack segmentation. Thus, crack properties like width and length can be measured from this crack map.

In summary, this paper mainly makes the following contributions. First, a pixel-level crack segmentation approach was developed by integrating CNN with a local region-based active contour model (ACM), and shown to be more accurate than the state-of-the-art. Secondly, the robustness of the proposed approach was demonstrated experimentally when carried out on noisy concrete surface images. Crack properties derived from the proposed crack segmentation approach were also shown to be accurate even under such noisy conditions. Thirdly, a post-processing workflow was proposed to refine the segmentation results by leveraging the characteristics of ACM and CNN. Lastly, the proposed approach significantly improved the data efficiency and mitigated the tedious labeling process. Through these contributions, the current concrete inspection process can be enhanced; by requiring a lesser labeling effort, more data efficient methods for crack segmentation can be explored.

This paper starts with a review of current crack identification methods. Following that, the proposed crack segmentation and properties retrieval method using image patches is introduced. Experimental studies are carried out to validate the performance of proposed crack detection method under different noisy conditions. A comparison study with several state-of-art methods is also conducted for further demonstration. Finally, the paper discusses the research results and findings as well as future research directions.

2. Literature review

In recent years, various methods for crack detection by means of computer vision techniques have been proposed. The current developed methods are generally classified into two categories: 1) recognizing the presence of cracks in images (patch-level) and 2) segmenting crack regions from images (pixel-level). In the patch-level category, several studies have focused on detecting images containing cracks. Abdel-Qader et al. [18] proposed a crack detection approach on bridge surface based on Principle Component Analysis algorithm. However, the camera pose and distance caused significant influence on the accuracy of the results. Prasanna et al. [19] built an automated crack classification system using Support Vector Machines and histogram-based classification algorithms for concrete bridge deck inspection. The crack detection accuracy of the aforementioned method still required improvement.

With recent advances in deep learning techniques, CNN and other deep learning methods have become increasingly popular for patch-level crack detection. Feng et al. [20] proposed a deep active learning strategy where a deep residual network was first trained with small set of images, then the pretrained network was used to remove non-defect images to reduce time and resources input in the labeling process. Cha et al. [21] trained CNN to classify small image patches as crack or non-crack. The trained CNN was then used to test full images at different resolutions. In a comparison study conducted by Nhat-Duc et al. [22], CNN-based methods outperformed optimized edge-detection methods on pavement crack detection images. While CNN introduced a robust and efficient approach to classify crack images, it required a large number of labeled data for training, together with high computation and time costs. To overcome this challenging issue, transfer learning has been applied to the implementation of CNN for crack detection through transferring the weights from the pretrained network with well anotated image classification dataset to new network [23,24]. Dorafshan et al. [25] demonstrated the superiority of deep convolution neural network (DCNN) together with transfer learning for crack detection problems through comparison with several common edge detectors. Among the current methods developed, deep learning techniques achieved the best performance in crack and non-crack image classification regarding robustness. However, as patch-level crack detection only recognizes crack images or non-crack images, this lacks sufficient details for further engineering analysis to be carried out.

The second category of crack detection studies focused on segmenting cracks at pixel-level and producing an accurate crack map. This is necessary so that engineering analysis, such as severity assessment can be conducted. Yu et al. [26] developed a semi-automated crack detection system using a graph search method with manual inputs of start and end points, but their method required controlled lighting conditions to ensure accurate crack detection. To improve the robustness to noise, a percolation-based method was proposed by Yamaguchi et al. [6] for large concrete inspection. Unlike edge detectors, the crack region was retrieved based on localized connectivity relationships instead of pixel-to-pixel correspondences. However, their proposed method considered surface noise including surface stains and shadows but not texture patterns. Lattanzi and Miller [8] performed crack segmentation in field applications using Canny Edge and K-means clustering algorithm. However, the number of initial clusters as well as the presence of noise pixels that were similar to crack pixels easily resulted in misclassification of crack points. Fujita et al. [27] proposed a robust crack detection method from noisy concrete surfaces that consisted of two preprocessing steps and two detection steps. However, the performance was sensitive to locally adaptive thresholding. As observed from the above analysis, the aforementioned methods are mostly sensitive to the presence of noise on concrete surfaces including surface stains, unintended objects, background texture and lighting shadows.

To overcome the aforementioned challenges arising from noise on concrete surfaces, recent literature has explored learning-based approaches for pixel-level crack segmentation. Shi et al. [28] proposed crack detector and descriptor using random structured forests to discern crack from noise efficiently. Most recently, fully convolutional network (FCN) has been adopted for crack segmentation from images [12,29]. Liu et al. [14] firstly applied U-Net based on FCN structure for concrete crack detection task with better performance compared with previous FCNs in terms of both efficiency and robustness. Liu et al. [13] developed a deep hierarchical feature learning architecture comprising of extended FCN and Deeply-Supervised Nets (DSN) to predict pixel-level crack segmentation under various conditions. While these studies have showcased the strong capability of deep learning-based approaches for crack detection, the main challenge encountered is the requirement of large well-annotated training sets. The accurate pixel-level labels commonly involves large expense of manpower during the data preparation phase.

The aforementioned methods are developed to detect cracks from a single static image frame through image analysis. Several studies have tried to analyze and detect cracks by incorporating the crack propagation mechanism using multiple image frames. Dias-da-Costa et al. [30] proposed an image deformation approach to monitor crack propagation by correlating images containing cracks developed over certain time intervals. The proposed method is capable of detecting and characterizing surface discontinuities and is robust to lighting conditions. Kong and Li [31] developed a robust crack detection approach by searching differential features from a series of continuous images which is proven to be robust to ambient lighting conditions, crack-like edges and surface textures. Later, they treated crack breathing behavior as a robust indicator for crack identification and utilized image overlapping techniques

to detect fatigue cracks from non-crack edges on a low contrast metallic surface [32]. Chen et al. [33] proposed a novel approach for crack detection by aggregating information obtained from different video frames using local binary patterns (LBP), support vector machine (SVM) and Bayesian decision theory. While all these studies successfully achieved robust crack detection and propagation monitoring, they all required multiple images or video frames to infer crack propagation. However, multiples image frames of different phases of crack development may sometimes not be available to inspectors. In ad-hoc building inspections, defects may need to be assessed from single images.

As demonstrated from the above literature review, accurate crack segmentation under the interference of noise is a challenging problem to address. The adoption of deep learning techniques has achieved great performance in robust crack detection for both patch-wise and pixel-wise methods. However, the often labor-intensive and time-consuming semantic labeling process involved in pixel-level approaches makes their real-world applications more challenging. Compared with pixel-level approaches, patch-level methods are implemented more efficiently with a lesser labeling effort, but this leads to a lack of high level details for further quantification. The review shows that bridging this gap between patch-level and pixel-level methods may allow us to leverage the advantages of both methods to achieve accurate crack detection with greater robustness, and lesser effort during data preparation.

3. Proposed crack segmentation and properties retrieval method

To address the gaps identified in the literature review, a hybrid approach for robust crack segmentation and crack properties retrieval on noisy concrete surfaces is proposed. Fig. 1 features the overview of the proposed method. The proposed method consists of two stages: 1) pixel-level crack segmentation from labeled image patches; and 2) crack properties retrieval including crack length and width based on distance transformation from the crack map.

3.1. Achieving crack segmentation using image patches

In this section, the proposed crack segmentation method by integrating active contour model (ACM) and CNN is introduced. To the authors' knowledge, this is the first attempt to leverage the advantages of both techniques to achieve pixel-wise crack segmentation in terms of robustness and data labeling efficiency. To accurately segment crack regions in the image, a post-processing workflow is further developed that leverages the integration of CNN and ACM to achieve robustness in crack segmentation.

3.1.1. Active contour model (ACM)

The active contour model (ACM) is an effective image segmentation method that is widely studied and applied in computer vision domain [34,35]. It works by actively extracting areas of interest in images continuously and smoothly while flexibly handling changing topological structure. The basic principle of ACM is to actively drive an initialized contour to accurately approach the boundary of the object to be segmented through gradual evolution under predefined energy functional constraints. Based on these constraints, the existing ACM methods can be generally classified into two categories: 1) edge-based models and 2) region-based models. The edge-based models stop the moving contours at the expected boundaries by constructing edge stopping functions using image gradient information [36–38]. They are primarily used for efficient object segmentation of high contrast and quality images. However, they tend to be easily stuck in local minima and cause inaccurate detection of weak object boundaries under the existence of noise and inhomogeneous image intensity.

To overcome the problem of edge-based models, region-based models were developed utilizing statistical information to drive the curve evolution [39]. The region-based models can be primarily categorized into two types: global region-based models and local region-based models. Global region-based models utilize the statistical information inside and outside the contour to control the stopping criteria of the curve during evolution [40–42]. While they outperform edge-based models in terms of both robustness to noise as well as extraction of weak boundaries, they still suffer from the problem of inhomogeneous image intensity. Later, localized region-based models were proposed using statistical intensity information in localized regions instead of entire region of images. The performance greatly improved with respect to image intensity inhomogeneity as well as sensitivity to noise but was found to be highly reliant on the initial contour settings.

On concrete surfaces, noise in the form of background texture commonly appear globally throughout the image, while some isolated noise like surface stains and unintended objects may be located far away from the crack regions in the images. Global region-based models may detect those noise pixels which have similar intensity values as pixels in crack regions while local region-based models could effectively alleviate this problem through statistical analysis in local areas. Therefore, to enhance the robustness for crack segmentation, local region-based ACM is adopted in this paper. To address the limitation of local region-based ACM that is highly dependent on contour initialization, a pretrained CNN is proposed in this paper to locate multiple initial contours along the crack region.

The ACM adopted in this paper is a local statistical model introduced by Zhang et al. [43] which performs well in terms of intensity inhomogeneity and robustness to noise. Intensity inhomogeneity, as a measure of the inhomogeneity of intensity in an image, is mathematically defined as a smooth and spatially varying field multiplied by the constant true signal of the object in the image [44]. In their method, the inhomogeneous objects are modeled as Gaussian distributions of different means and variances. To investigate the local details in the image, the original image domain is transformed to another domain where the Gaussian distributions of object intensities are better separated. The means of the Gaussian distributions are adaptively calculated

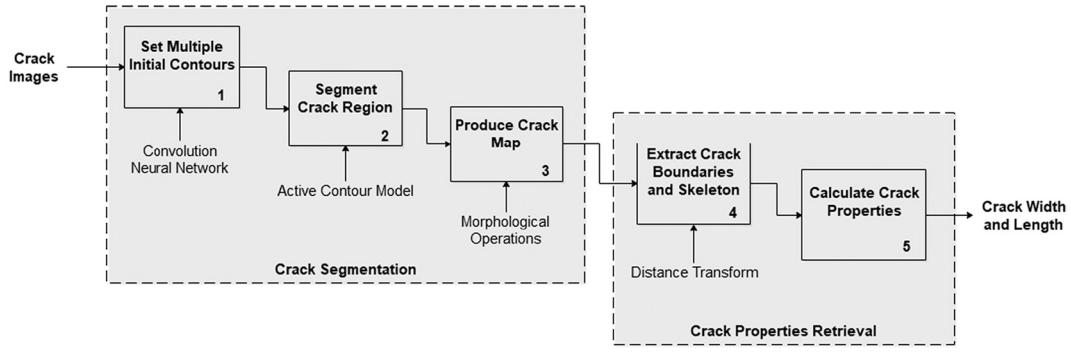

Fig. 1. Method overview.

by multiplying the original image signal with a bias correction within the local region. The initial contour is represented using the zero level set of the level set function. In the transformed region, the statistical energy function integrating the level set function, bias correction and the constant estimating the true signal of the object is defined for each local region. The segmentation proceeds with evolution of a higher dimensional level set formulation by means of minimization of predefined energy functional. This model is capable of assigning each pixel to multiple classes instead of only one class which improves segmentation performance with respect to noise and intensity inhomogeneity.

3.1.2. Multiple contours initialization using CNN

As summarized in the literature review, CNN-based methods are the best-in-class method in terms of patch-level crack detection. One of the main challenges of such deep learning techniques is the requirement of large-scale well-annotated dataset to learn more accurate and generalized models [45]. However, there is usually a lack of available data to train a deep CNN from scratch. Hence, transfer learning is often used. ImageNet provides a well-established database comprising more than 1.2 million natural images distributed over 1000 categories [46]. The CNN model trained on the ImageNet database serves as a basis for solving image classification problems in other domains. The underlying idea of transfer learning is to utilize the pretrained CNN model to learn specific features of the custom datasets [47]. In this paper, transfer learning is adopted to enhance the efficiency as well as accuracy of the CNN crack classifier.

VGG16 is a 16-layer CNN architecture widely used for image recognition. After initializing the VGG16 network with pretrained weights from ImageNet, it is trained on a dataset containing crack images. Since there exists low similarity between features of this crack dataset and ImageNet, fine-tuning the network is necessary to learning crack features effectively. Fig. 2 shows the architecture of the VGG16 network. The transferred generic features learned from the first three convolution blocks remain unchanged while the weights in the last two convolution blocks are trained to learn high-level knowledge from the crack images. Finally, the learned feature representations are fed into fully-connected layers followed by Softmax activation to construct a crack classifier.

In ACM, because of the non-convexity of energy functions or localized level set evolution, region-based models can easily fall into local minima. Additionally, the performance of region-based models varies with different localizations of initial contours. The best performance is commonly achieved when the initial contour is located in proximity to the area to be segmented within the image. Therefore, irregularities in crack shape and size may lead to brittleness in the contour evolution process. According to the experimental results in Fig. 3, the evolution was stuck in local minima after 1000 iterations while the crack region was completely segmented in just 200 iterations using multiple initial contours. Therefore, setting multiple initial contours localized in the neighborhood around a crack region enhances both the efficiency and accuracy of the curve evolution while eliminating unexpected recognition of noise in the image background.

To accurately locate the initial contours along the crack, the following deep learning-based patch-level crack detection method is utilized. A pretrained CNN on image patches cropped from several larger crack images is proposed to robustly approximate the crack region to be segmented in the target image. Because CNN requires sufficient patches for training, the number of larger images varies with different patch sizes. A larger patch size requires a significantly increasing number of original images to be cropped. Additionally, the cropped patches should contain adequate information and details to enable effective learning features. As shown in Fig. 4, patches of overly small sizes of 16×16 may contain limited information that could lead to degradation of training performance while larger patch sizes of 32×32 and 64×64 are better alternatives. Therefore, to reduce the number of original images needed as well as satisfy the training requirement, the image is cropped into patches of 32×32 pixels in the proposed method.

Fig. 5 illustrates how the proposed method is carried out. The image is first cropped into several small sub-images. Then the pretrained CNN serves as a crack classifier to carry out a first scan and filter out non-crack images in both horizontal and vertical sequences throughout the whole image space. Considering possible differences between training and test data, a selection criteria is proposed to choose crack sub-images as potential candidates for locations of good initial contours. In the first scan, misclassifications may occur in sub-images where the crack pixels appear at the edges. Therefore, a second scan beginning from a different start point is performed by CNN to enhance the classification as demonstrated in Fig. 5. Fig. 6 shows an example of the initial contour detection process, where P_i represents the predicted probability of a detected crack. As observed from the detection results shown in Fig. 6,

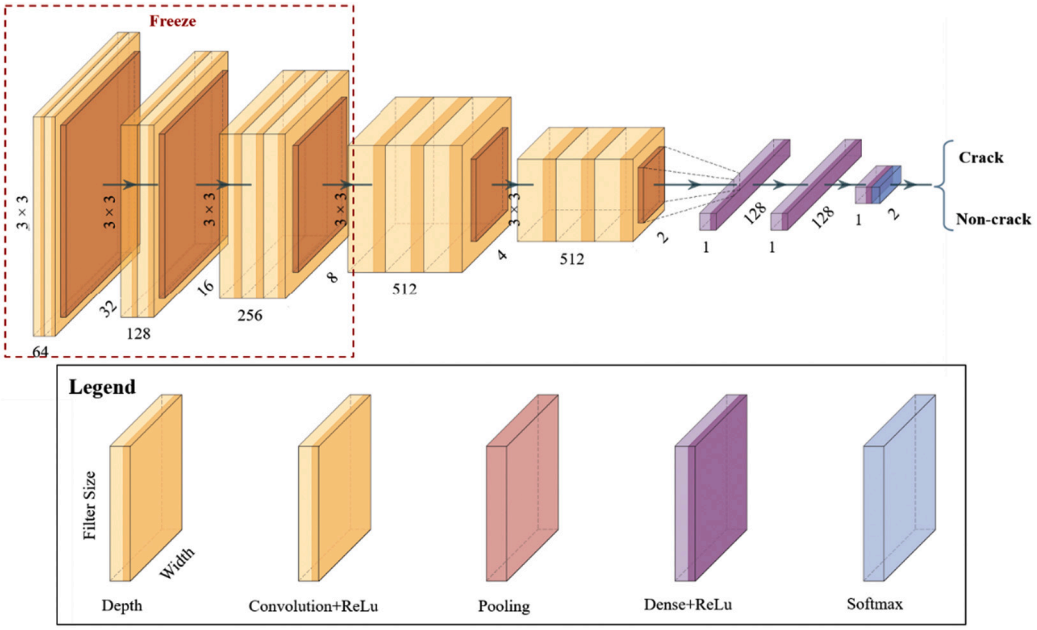

Fig. 2. The architecture of VGG16 with transfer learning.

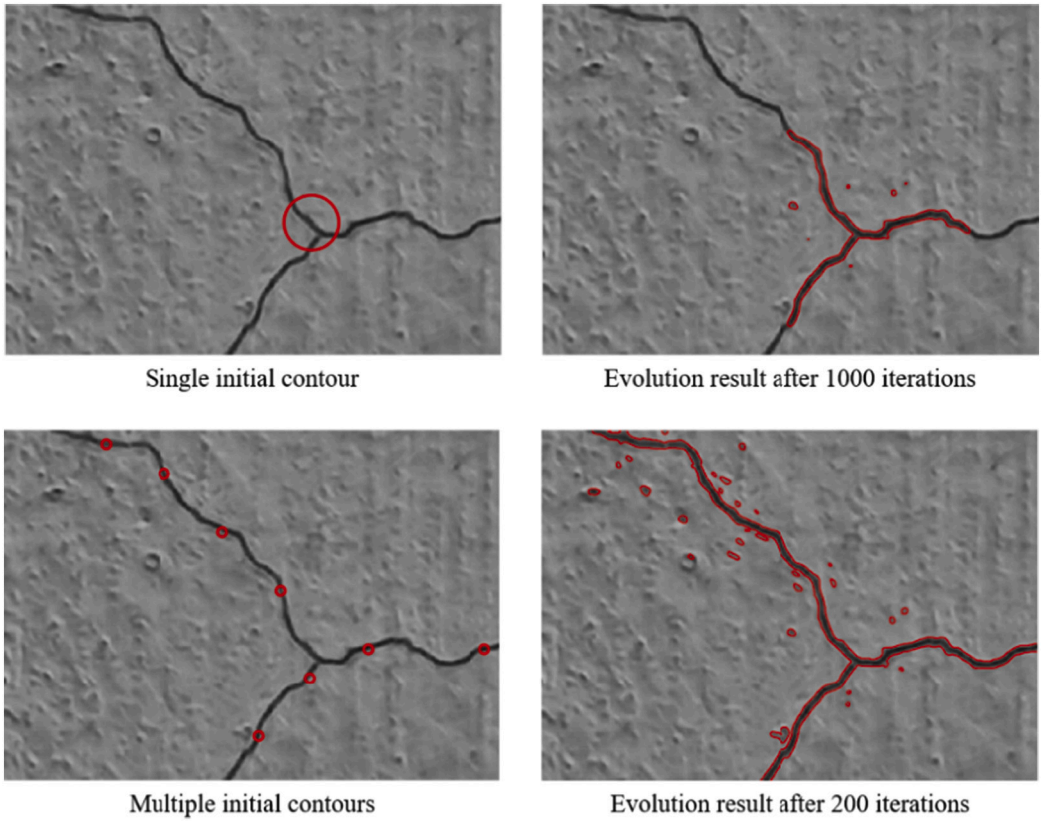

Fig. 3. Evolution results of different contour initializations.

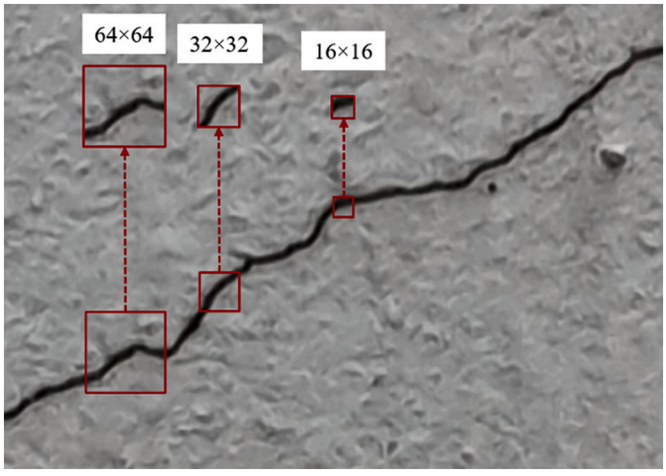

Fig. 4. Image patches in different sizes.

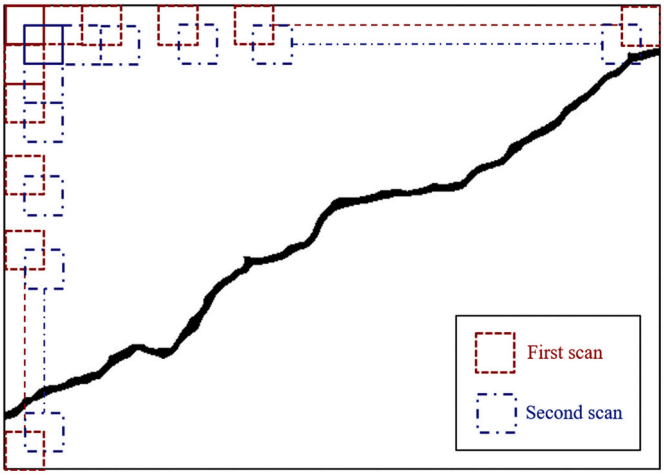

Fig. 5. The proposed framework of the localization of the initial contours.

all sub-images containing cracks are identified with probabilities of over 95%, while sub-image (4), (11) and (12) have predictions with significantly lower probability. This is because these sub-images contain few crack pixels that are at the boundary of the sub-image, and hence it may contain limited information to indicate the presence of cracks. Therefore, to achieve better initialization of the contours, a threshold based on 95% is suggested as a selection criteria to choose sub-images with a high probability of cracks.

3.1.3. *Post-processing morphological operations*

While the crack edges can be traced using ACM, the results may still contain some isolated noise surrounding the crack. Therefore, several morphological operations are further proposed to remove this noise and extract the crack region accurately, serving as post-processing steps. Fig. 7 illustrates the proposed post-processing steps following crack segmentation using ACM to remove the existing noise and extract the target region. Fig. 8 shows an example to illustrate the implementation of the post-processing steps proposed.

Fig. 8(a) and (b) shows the initial contours setting and segmentation boundary using ACM, respectively. As demonstrated in the preliminary segmentation results using ACM in Fig. 8(c), noise represented as enclosed boundaries may be located in the vicinity of the crack boundary or on the crack boundary. To remove these noise, an image filling technique is used to fill the enclosed noise boundaries (holes). Firstly, n enclosed holes are detected which are represented as $H = \{h_i, i = 1, 2, \ldots, n\}$. The areas of these holes are calculated and denoted as $S =$

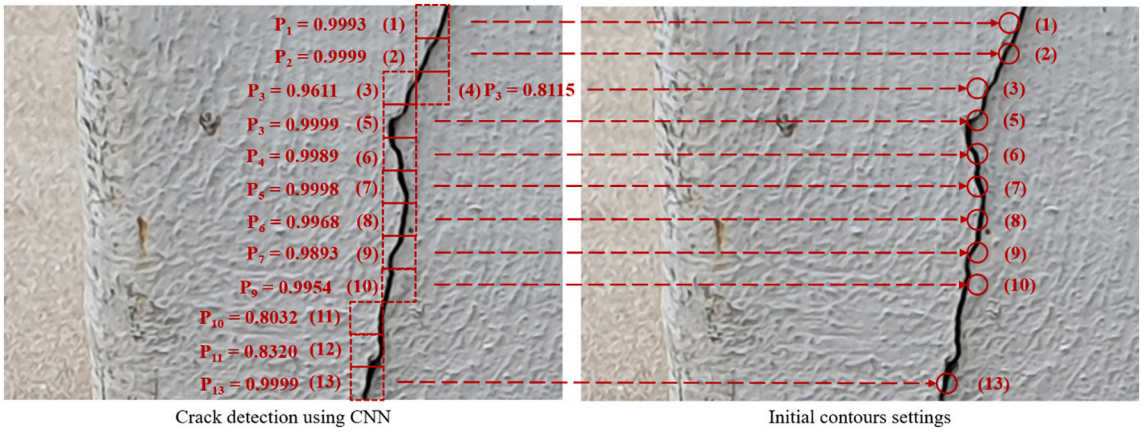

Fig. 6. The example process of initial contours localization.

$\{s_i, i = 1, 2, \ldots, n\}$. As observed from Fig. 8(c), the enclosed holes of noise surrounding the crack region are typically present in much smaller areas compared to the actual crack areas. By observation, the areas of these smaller holes of noise range from 1 to 39 pixels, while the areas of crack region range from 635 to 1209 pixels. Hence, holes with area less than 100 pixels are inferred to be noise, and will be filled. For holes with areas larger than 100 pixels, a further selection criteria is used. If the hole overlaps with previously detected crack patches, it is more likely to be a crack rather than noise. Conversely, holes without overlaps may be identified as noise to be filled. The image filling result is shown in Fig. 8 (d).

As observed from Fig. 8(d), most of the noise have been removed while some noise shown in black areas that are connected to the crack boundaries are not completely eliminated in the image. Despite these black areas being noise, the proposed approach considers these areas to be part of the boundary. To accurately segment the crack region, the crack region will be extracted without its boundaries. As a result, the noise connected to the boundaries will be eliminated, and will not affect the crack segmentation results.

In Fig. 8(e), the image space is segmented into four regions according to the crack boundaries. The areas of all regions are calculated and sorted in an increasing sequence represented as $A = \{a_j, j = 1, 2, \ldots, m\}$, *where* $a_1 < a_2 < \ldots < a_m$ and m is the number of regions. Because cracks develop in thin lines, the area of a crack region is normally much smaller than that of the background. Therefore, the region with smallest area is inferred to be a crack region, and extracted first. To avoid omitting isolated crack regions, a check of whether all crack patches contain an area of the inferred crack regions is done. If the criteria is not satisfied, this means there are more crack patches than inferred crack regions, indicating the presence of additional crack regions. In this case, the next smallest region will be inferred as another crack region, and the check applied again. The process of crack segmentation is completed when all the inferred crack regions are accounted for by the detected crack patches. Based on the calculated areas of the regions in Fig. 8(e), it can be observed the crack regions are the smaller regions with areas of a_1, a_2 and a_3 in the image, and these overlap with the crack patches earlier, as shown in Fig. 8(f). Consequently, a crack map without noise can be accurately produced from the proposed morphological operations (Fig. 8(g)).

In summary, the post-processing techniques are based on the crack characteristics, as well as the mechanism of ACM. Initially, the regions detected by ACM are enclosed with smooth and completely closed curves. Noise is observed to be represented as significantly smaller enclosed holes compared to the actual crack region. Thus the identified noise can be easily identified by setting a threshold, and subsequently removed using image filling techniques. Next, cracks are observed to commonly propagate in thin irregular lines, resulting in comparatively smaller areas compared to other background areas. Thus, the crack region can be inferred in the image space, and further verified by comparing against crack patches detected by CNN. In summary, the proposed morphological operator enhances the accuracy of detection, especially of cracks with a complex topology.

3.2. Retrieval of crack properties

The crack properties are measured in pixels from the produced crack map as shown in Fig. 9(a). Firstly, the crack boundaries are traced as shown in Fig. 9(b). Secondly, the crack skeleton is extracted from the crack map by determining the largest circle that fits within the crack region at each point. As illustrated in Fig. 9(c), the crack skeleton is composed of multiple centroids of circles tangent to the crack boundaries. Fig. 9(d) shows a map of the radius of these circles between crack boundaries with different colors indicating different radii lengths as indicated in the color bar. The largest circle is determined by finding the medial axis in the crack region using fast-marching distance transform approach [48] which is capable of precisely and efficiently computing the distance field in the image. After extracting the boundaries and skeleton, the topological configuration of the crack is determined by checking the connectivity of crack skeleton points. The crack junction point and end points of each segment are specified in red and green circles respectively, as indicated in Fig. 9(e). From these start and end points, the various crack branches are identified as shown in Fig. 9(f).

The crack properties are measured based on the extracted crack segment and distance information. All the crack dimensions are quantified in pixels. The crack length is approximated by calculating the number of points on the crack skeleton. The crack width of each point is estimated as twice the shortest distance between the boundary and point on the skeleton. In other words, the estimated crack width at each point on skeleton is the same as the diameter of the largest circle fitting in crack region at each position as shown in Fig. 9(c). The crack width at each point is used to compute the mean of all width values.

4. Experimental validation and discussion

The objective of this paper is to reduce the data requirements for crack segmentation, whilst achieving robustness to support the retrieval of crack properties under various noisy concrete surfaces. Hence, the efficacy of the proposed crack segmentation method is validated through following steps. Firstly, the datasets for training and testing the segmentation methods are introduced in details. Secondly, the metrics adopted to evaluate the performance of both crack detection and quantification are demonstrated. Thereafter, the experimental performance of the proposed crack segmentation method is validated on a noisy dataset and compared with state-of-the-art robust crack

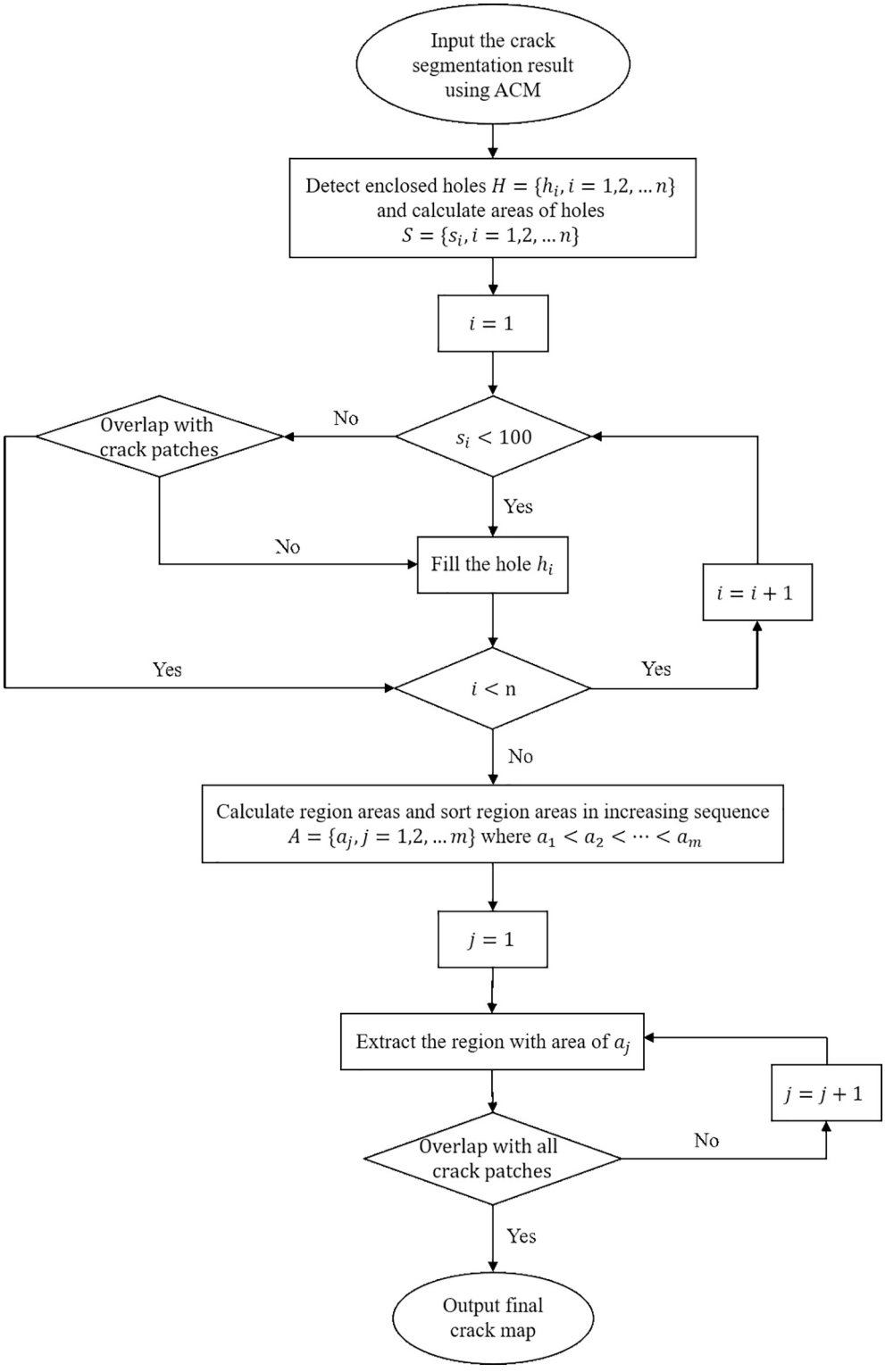

Fig. 7. Workflow of proposed post-processing steps.

segmentation methods in the literature.

4.1. Dataset

(1) Training dataset

The dataset for training the proposed method and compared methods is adopted from [13] which is an open-sourced dataset of crack images for crack detection and segmentation. The dataset consists a total of 537 crack images with pixel-wise segmentation labels in a size of 584×384 pixels. Fig. 10 shows the examples of crack images in the dataset. For the purposes of comparison, 470 original images with corresponding segmentations were selected to train the existing deep-learning based approaches.

To train the proposed method, 100 RGB images from the same dataset are cropped to several patches in a fixed size of 32×32 pixels.

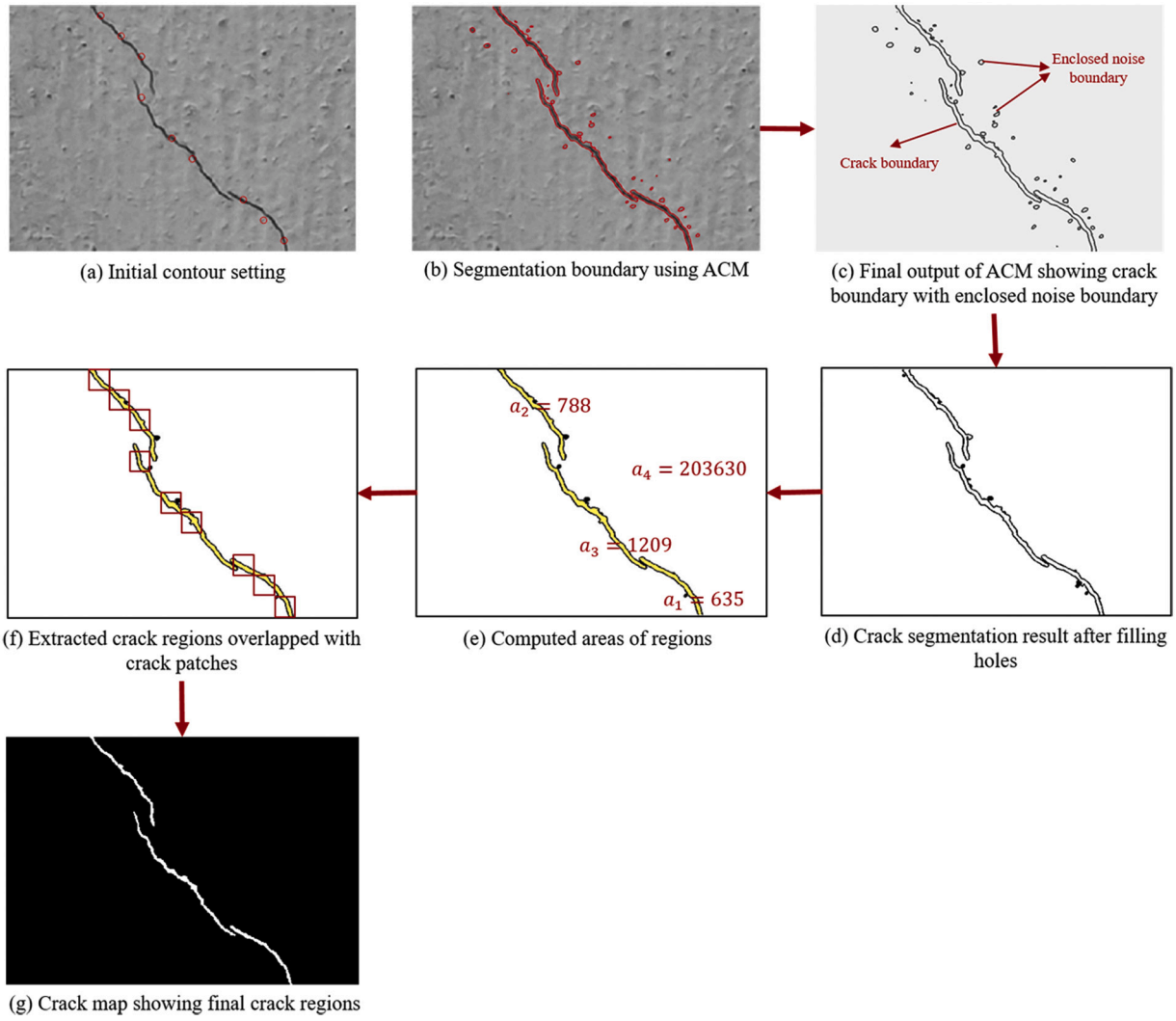

Fig. 8. Example of post-processing steps for crack segmentation.

The patch labels of crack and non-crack are derived from the presence of crack pixels within the image patch, and the pixel segmentation labels are subsequently ignored in the patches. Out of these cropped patches, a total of 3200 patches are selected that includes 1600 crack and 1600 non-crack images respectively. The image patches are split into training, testing and validation subsets which account for 80%, 10% and 10% of the dataset respectively.

(2) Testing dataset

In order to validate the performance of the proposed crack segmentation method, an additional custom dataset comprising 100 noisy crack images collected from real concrete building surfaces is introduced. Each image is set of a fixed size of 544 × 384 pixels, varying in crack sizes and patterns. In the test dataset, various noisy conditions on concrete surfaces are considered including lighting variations (shadings), background texture, unintended objects, surface blebs and stains, etc. The examples of noisy images are shown in Fig. 11.

4.2. Evaluation metrics

(1) Evaluation metrics of CNN

The performance of the CNN on crack detection is evaluated by four metrics – Accuracy, Precision, Recall and F1 Score – which are commonly used as evaluation metrics for machine learning algorithms. The above four evaluation metrics are calculated based on true positive (TP), true negative (TF), false positive (FP) and false negative (FN) which symbolizes correctly and incorrectly predicted images with positive and negative labels. The four metrics are formulated in the following Eqs. (1)–(4).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{4}$$

(2) Evaluation metrics of crack segmentation

The obtained crack segmentation results are evaluated through comparison with ground truth. The ground truth of the images are manually semantically annotated using Image Segmenter App in MATLAB®. The evaluation is carried out by comparing the pixels detected by the proposed approach and the points marked in the ground truth. The

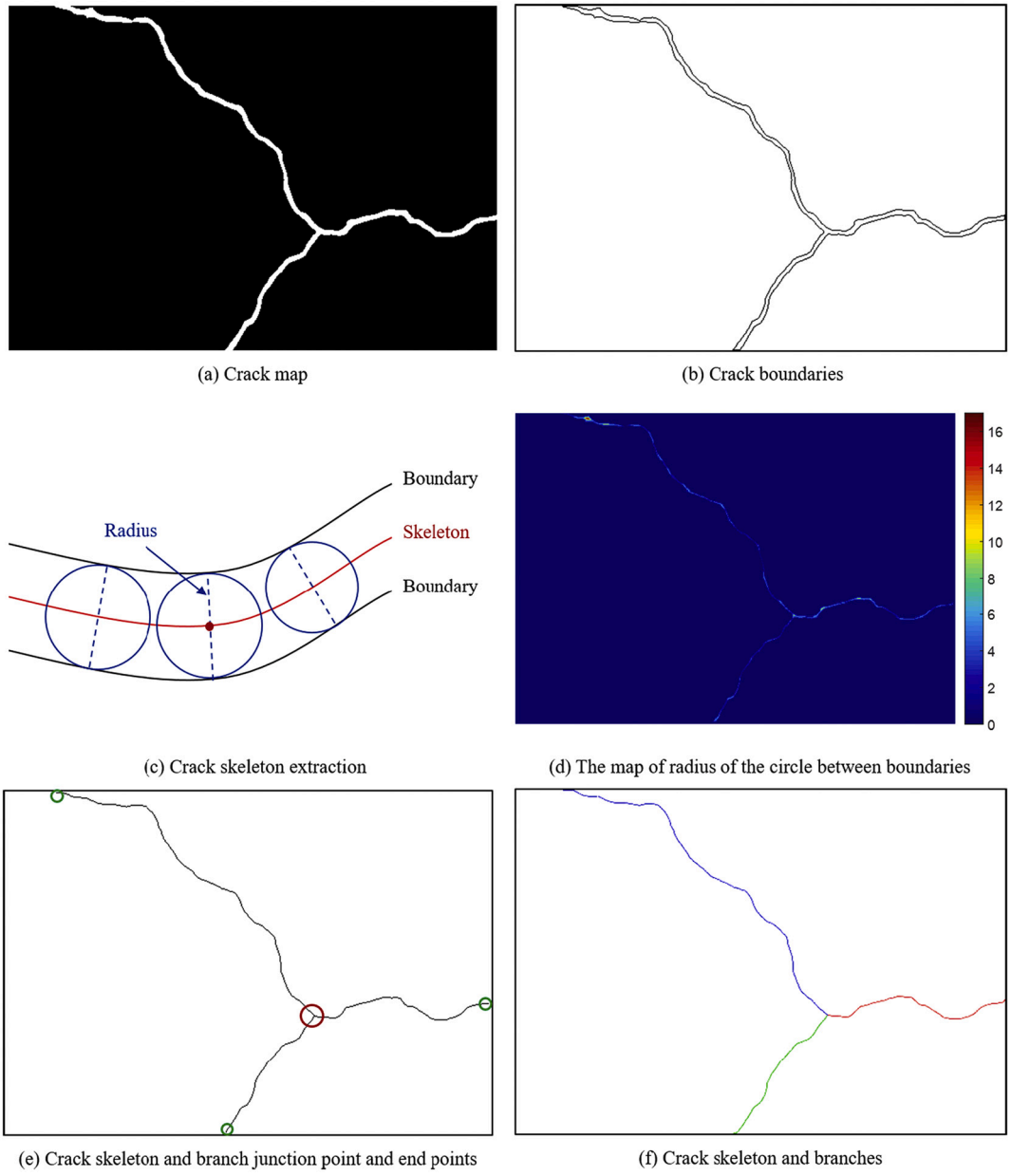

Fig. 9. Crack skeleton and boundary extraction method.

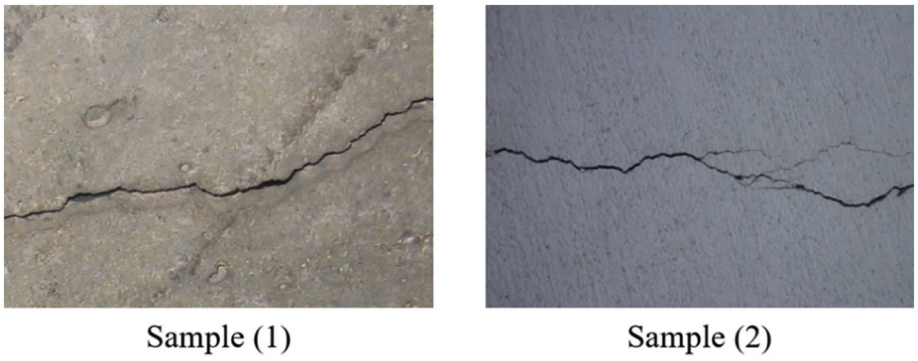

Fig. 10. Examples of crack images in the dataset.

Fig. 11. Examples of noisy crack images in the dataset.

correctly and incorrectly detected crack pixels as well as pixels that are not detected are identified as TP, FP and FN measurement criteria respectively. The performance of the proposed approach is further assessed by Precision, Recall and F1 Score which are three major metrics for image segmentation, as formulated in Eqs. (1)–(4). Additionally, intersection of union (IoU) which is most commonly used for efficient evaluation of image semantic segmentation performance is adopted to calculate the overlap ratio between the ground truth and predictions. In the binary crack segmentation case, mean IoU is formulated using TP, FP and FN as demonstrated in Eq. (5).

$$Mean\ IoU = \frac{TP}{TP + FP + FN} \tag{5}$$

(3) Evaluation metrics of crack properties measurement

The results of quantified crack properties are evaluated through comparison with the ground truth values of crack properties which are obtained by manual surveys in ground truth images. The crack width at each point and mean value of crack width in each image as well as crack length in each image are calculated. The relative errors for estimations are computed as essential indicators of the performance of the crack quantification approach as demonstrated in Eq. (6). The mean and standard deviation values of relative errors are further calculated.

$$Relative\ Error = \frac{|V_{True} - V_{Observed}|}{V_{True}} \tag{6}$$

4.3. Training CNN within proposed method

The training process was implemented on a single TITAN RTX GPU using Keras deep learning framework with Tensorflow backend. The training and testing results are demonstrated in Table 1 and Fig. 12. The model achieved a training accuracy of 99.75% and testing accuracy of 99.78%. Overfitting is a main problem that usually occurs in training the neural network where the network is trained to learn the image features so well that noise and other unnecessary details in the data start to negatively impact the performance. As a result, the network cannot generalize well from the training data to the unseen data. Conversely, the other criteria to judge the reliability of network performance is underfitting which indicates that the network lacks of the capability to learn the target characteristics well. Both overfitting and underfitting can be identified through training and validation performance curves. If the trend of validation performance starts decreasing at the point where the training performance starts exceeding the validation performance, the network is overfitting. On the other hand, underfitting is likely present when there is better validation performance than the training performance. As observed from Fig. 12, the training process was stable without presence of overfitting and underfitting. Therefore, the CNN was deemed to perform well on crack detection task which was capable of providing accurate estimations of the initial contours localizations.

Table 1
Results of CNN on crack detection.

Performance	Accuracy	Precision	Recall	F1 score
Train	99.75%	99.70%	99.81%	99.76%
Validation	99.74%	99.82%	99.65%	99.74%
Test	99.78%	99.72%	99.77%	99.70%

Additionally, we tested other state-of-art CNN models for image classification on our dataset including VGG19, ResNet50, ResNet101, Xception and Inception V3. All models were initialized with pre-trained weights on ImageNet, and fine-tuned for the crack classification task. The test results are tabulated in Table 2. According to the comparison results, it can be observed that VGG16 outperformed the other models in terms of all the metrics listed. Therefore, VGG16 was selected for setting the initial contours.

4.4. Validation of proposed method

In order to validate the capability of the proposed crack segmentation method, its performance was compared against two state-of-the-art deep learning-based methods for crack detection: (1) DeepCrack method [13], a deep hierarchical feature learning architecture; and (2) U-Net method [14] using deep CNN. All the methods are tested on the same concrete crack images introduced in the previous section. The parameter optimization of the referenced method was carried out to achieve the best performance on the test dataset. The performance of the segmentation method was discussed in terms of accuracy, robustness and data efficiency.

4.4.1. Comparison of accuracy against state-of-the-art

To measure the crack segmentation performance, the mean and standard deviation of precision, recall and F1 score values were calculated. The quantitative descriptive statistics of the experiment are

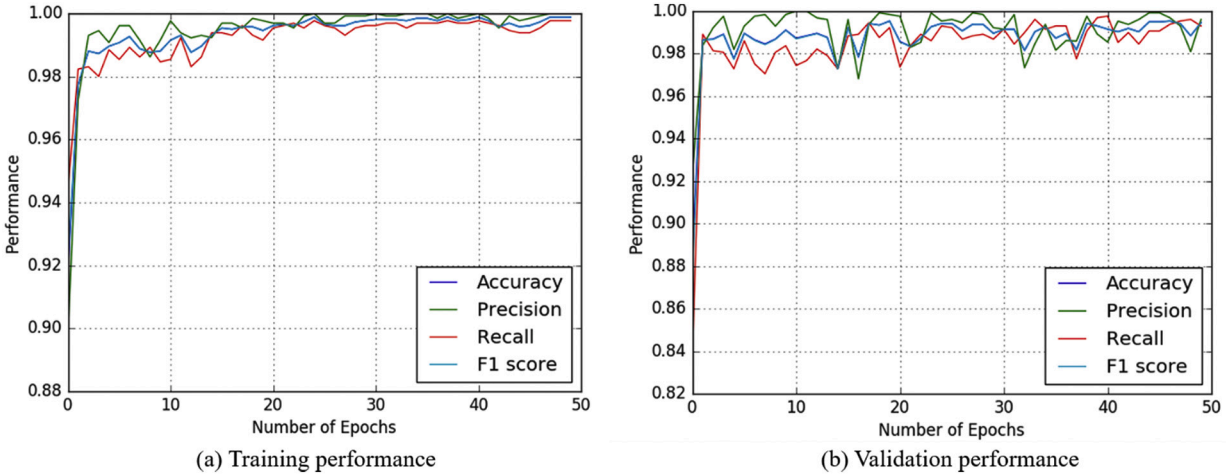

Fig. 12. Training and validation performance along epochs.

Table 2
Results of different CNNs on crack detection.

CNN	Accuracy	Precision	Recall	F1 score
VGG16	99.78%	99.72%	99.77%	99.70%
VGG19	99.33%	99.51%	99.18%	99.34%
ResNet50	98.92%	98.84%	99.00%	98.92%
ResNet101	91.46%	91.89%	89.87%	90.87%
Xception	98.17%	97.44%	98.79%	98.12%
InceptionV3	97.92%	97.10%	98.61%	97.85%

summarized in Fig. 13. The mean values of obtained precision, recall, F1 score and mean IoU values from proposed method were 0.8997, 0.9260, 0.9125 and 0.9179 respectively. From the results, it was observed the proposed method achieved overall improvement on both mean and standard deviation of the crack segmentation results obtained by other state-of-art methods with respect to four evaluation metrics. Additionally, compared to other methods, the experimental results of the proposed method shows lower variance, without the presence of outliers as shown in the boxplots.

To further confirm the observations, the significance of the differences between the proposed method and other methods are provided by paired *t*-tests, using a null hypothesis that there does not exist a significant difference between both samples. The larger the absolute value of the t-statistic, the smaller the *p*-value, and the greater the evidence against the null hypothesis (i.e. there is a significant difference between samples). The tests results between the proposed method and the other methods in terms of the four evaluation metrics were tabulated in Table 3. The results could be explained as further confirmation of the observation that the proposed method was superior to both current methods regarding the performance of crack segmentation on noisy scenarios due to its low *p*-values. While there exists an exception that there is no significant difference presented between the proposed method and DeepCrack in terms of recall (indicated by a higher p-value) shown in Table 3, the proposed method achieved a relatively higher mean recall value compared with DeepCrack. Therefore, the proposed method is shown to achieve more accurate crack segmentation as compared with other deep learning-based methods.

4.4.2. Comparison of robustness

The robustness of the proposed method is first assessed visually. Fig. 14 shows several examples of experimental results under various noisy conditions. From the observation of the various generated crack maps, other methods were limited in completely discriminating crack points from noise that were present in similar pixel density to the image space. As indicated in Fig. 14(a), the presence of dark areas surrounding the crack regions led to wrong identification of crack pixels near the crack boundaries by the DeepCrack algorithm. As a result, the segmented crack regions appeared wider than ground truth. Additionally, the presence of holes, blebs and unintended objects in the background also caused unexpected false positives in the segmentation results obtained from other methods as shown in Fig. 14(b)–(g).

In contrast, further visual assessment indicated the proposed method outperformed the other methods in terms of robustness to noise. It was observed the proposed method was capable of extracting crack regions completely under the influence of various noise without missing crack points. Hence, it could effectively alleviate the interference of noise on the background surface. Additionally, the proposed method was capable of extracting visibly thinner cracks from noisy surfaces as indicated in Fig. 14(g). The comparison results showed that the crack detected by the proposed method was thinner and closer to the ground truth in comparison to results produced by other methods. Therefore, the proposed method is capable of detecting cracks of various sizes from images accurately and robustly.

To further validate the performance of crack segmentation method quantitatively, the crack properties were derived under various noisy conditions. The crack width and length were measured based on the crack map obtained using the proposed crack segmentation method. The mean value of crack width at each point for each crack segment in images of the test set was calculated. The crack measurements were compared with results from manual surveys from the ground truth crack map in pixels. The calculated relative errors between estimations and ground truth values are shown in Fig. 15. The mean relative error of mean width and length was 0.0672 and 0.0251 respectively for test images with highly variable crack sizes. The results indicate that most of the measurements estimated by the proposed method were close to that done by manual surveys. Therefore, in accordance with the experimental results, the proposed systematic procedure of crack properties retrieval based on crack segmentation is able to provide reliable and desirable results for crack inspection.

Moreover, to demonstrate the effectiveness of the proposed crack segmentation method on achieving reliable crack quantification, an experiment to quantify crack widths using the proposed method is compared against in situ measurements. Five crack widths ranging from 0.5 mm to 2 mm were measured from images captured by a digital camera that was orthonormal to the concrete surface. The actual crack width (ground truth) was obtained using a crack width gauge shown in Fig. 16. To estimate the actual physical size, a pixel-physical conversion ratio was required to convert the pixel size in the image to the physical size (Fig. 17). This ratio can be computed from the relationship between physical size and sensor size as shown in Eq. (7).

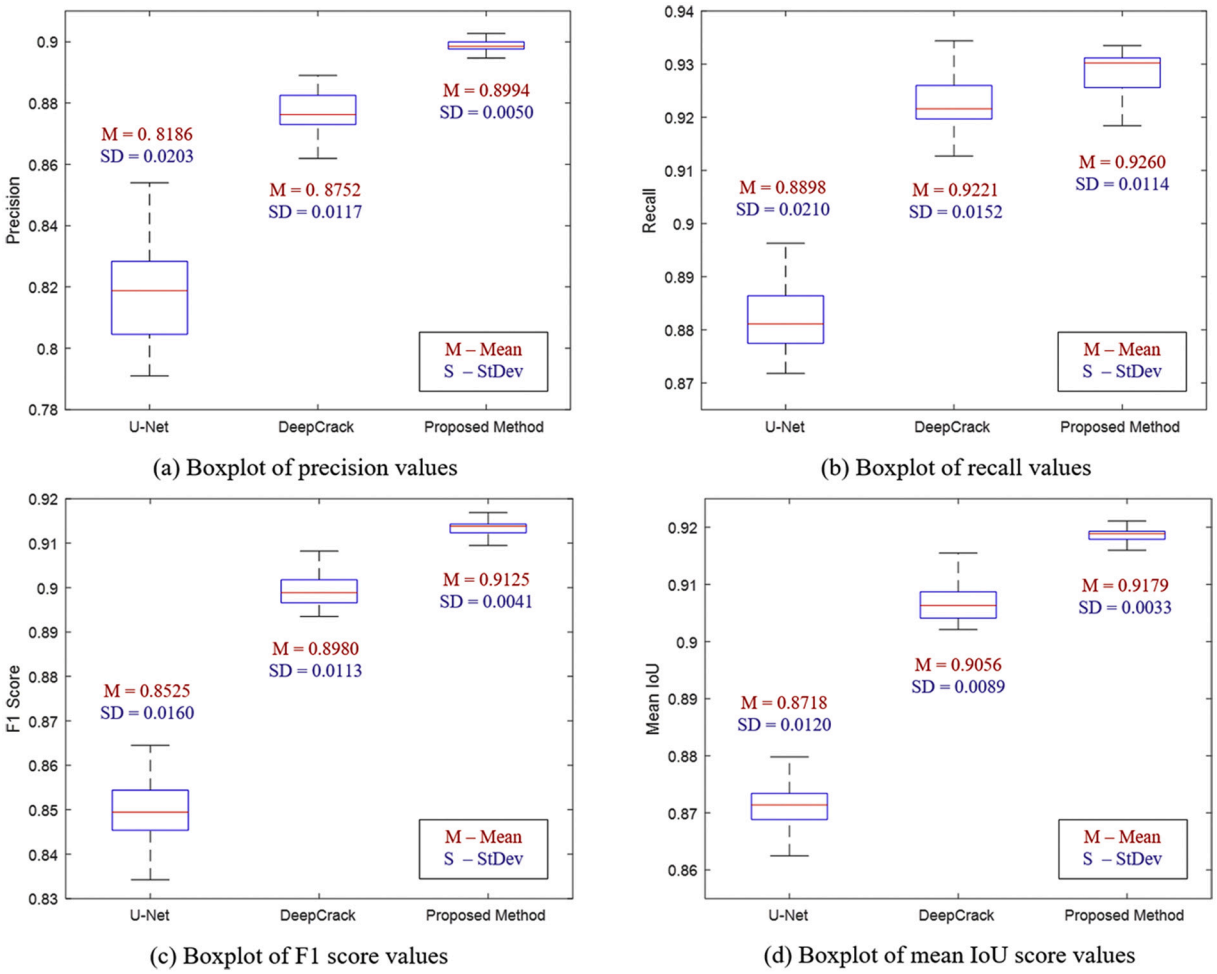

Fig. 13. Boxplots of precision (a), recall (b), F1 score (c) and mean IoU (d) values.

Table 3
Paired *t*-tests of precision, recall, F1 score and mean IoU values between proposed method and DeepCrack.

Comparison	Statistics	Precision	Recall	F1 score	Mean IoU
Proposed method – DeepCrack	T	15.8936	2.4402	15.2133	16.4847
	P Value	$5.3361e^{-29}$	0.0165	$1.1671e^{-27}$	$3.8341e^{-30}$
Proposed method – U-Net	T	46.0036	12.1035	31.2856	32.4731
	P Value	$1.2516e^{-68}$	$3.0841e^{-21}$	$3.9543e^{-53}$	$1.3682e^{-54}$

$$Physical\ size = \frac{Working\ distance \times Sensor\ size}{Focal\ length} = \frac{Working\ distance \times Image\ resolution}{Focal\ length} \times Pixel\ size \quad (7)$$

$$Pixel - Physical\ Ratio = \frac{Working\ distance \times Image\ resolution}{Focal\ length}$$

As the camera was perpendicular to the concrete surface, there was no geometric distortion of either the crack width gauge or crack in the image. As a result, the pixel-physical conversion coefficient of the crack and crack width gauge was constant. Based on the scale on the crack width gauge, the pixel-physical conversion coefficient could be calculated. Accordingly, the crack width measured in pixels from images using the proposed method could be converted to a real world measurement. The test results are tabulated in Table 4. It can be observed that the relative errors of physical width measurements vary approximately from 0.01 to 0.09 which are consistent with the measurement results in pixels. Thus, in accordance with the experimental results, the proposed procedure of crack properties retrieval is shown to be capable of providing reliable results for crack inspection. However, in real practice, significant sources of errors such as radial lens distortions may potentially cause geometric distortions that can lead to measurement error. Hence, significant calibration is necessary to ensure accurate measurement of the crack width, as was carried out in this experiment.

4.4.3. Comparison of data efficiency and time

The data efficiency was compared based on the performance the method achieved from the amount of data used for training and labeling [14]. The comparison results are tabulated in Table 5. The proposed crack detection method achieved higher mean IoU using 100 crack images as compared with U-Net and DeepCrack methods that were trained on 470 images. Thus, from the number of training images used, the proposed method performed more efficiently than other deep learning-based methods. Additionally, the training dataset used in the proposed method was labeled at patch-level while U-Net and DeepCrack were trained on images with pixel-level segmentations. It is known that pixel-based annotations requires significantly greater labor and time

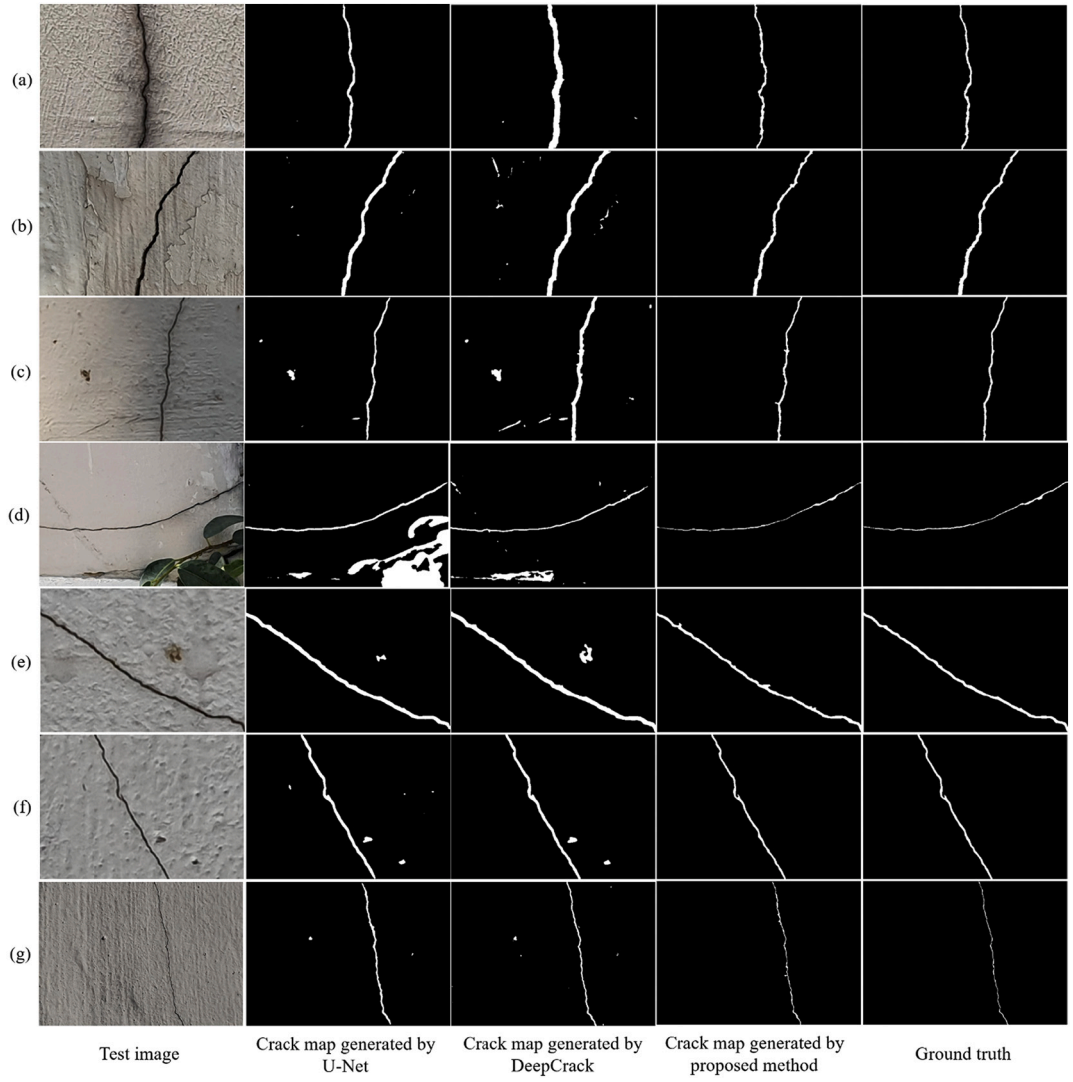

Fig. 14. Examples of crack segmentation results.

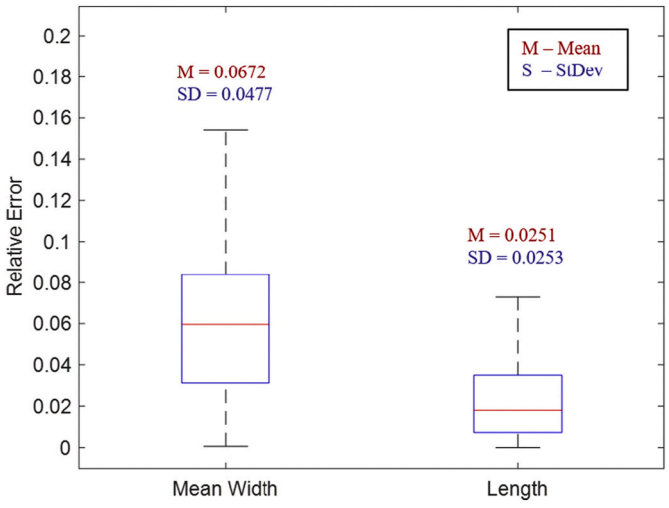

Fig. 15. Boxplot of relative error of mean width and length.

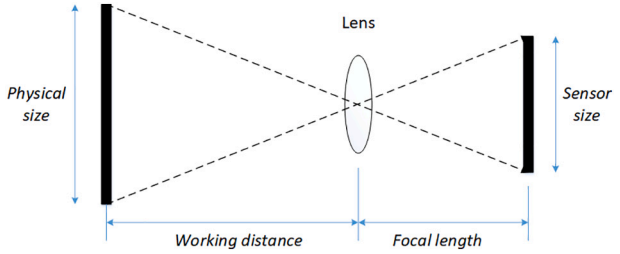

Fig. 16. Relationship between field size and sensor size.

cost compared than patch-based labeling process [16,17,49]. Therefore, the proposed method achieved better performance compared to these deep learning-based methods as it achieved better segmentation performance while involving significantly fewer images and consequently reduced labeling effort.

Additionally the computation times involved in the crack segmentation process for the three methods were also calculated. All computations were implemented using a single TITAN RTX GPU. Since all three methods involved training and testing processes, the computation time was defined as the sum of training and inference (testing) times for each method. The comparisons are listed in Table 6. Based on the results, it

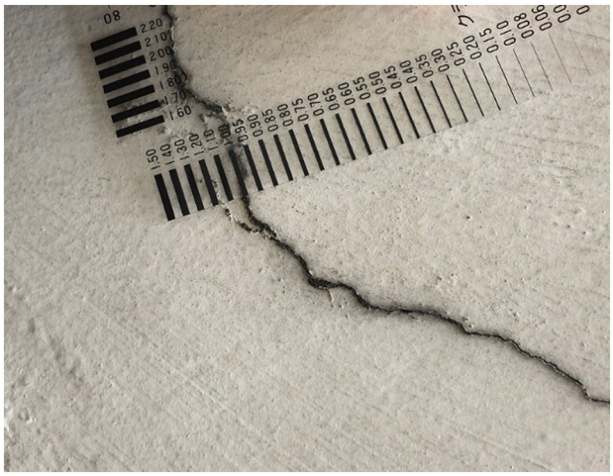

Fig. 17. Experimental setup for physical crack width measurement.

Table 4
Phsical widths measurement results of examples.

Example	Measurement (mm)	Ground truth (mm)	Relative error
a	1.760	1.900	0.0737
b	0.500	0.550	0.0909
c	0.650	0.600	0.0769
d	0.988	1.000	0.0120
e	0.919	1.000	0.0807

Table 5
Mean IoU and number of required image for different methods.

	No. of required images	Labeling effort	Mean IoU
U-Net	470	Pixel-wise	0.8718
DeepCrack	470	Pixel-wise	0.9056
Proposed method	100	Patch-wise	0.9179

Table 6
Comparsion of training and testing times among different methods.

	Training time (s)	Inference time (s)	Computation time (min)
U-Net	4199	80	1 h 11 min 19 s
DeepCrack	7437	26	2 h 4 min 26 s
Proposed method	90	10 (Initial contours setting) + 150 (Segmentation) = 160	4 min 10s

can be observed that deep learning models required much longer training times for pixel-wise segmentation task compared with the patch-wise classification in our proposed method. However, due to the evolution process in ACM, the testing process took longer compared to pre-trained CNN models for generating segmentation results. The total inference time for testing 100 images are 160 s. Each test image took 1.6 s for the proposed method to run through the ACM process and post-processing steps to predict the final crack map. Hence, the total computation time for testing larger scale datasets will correspondingly accumulate and increase.

Thus, there exists a tradeoff between the total computation time and the number of test images, and this tradeoff is evaluated. Herein, the training time in each method remains constant while the testing time increases as more images are tested. As shown in Fig. 18, there are breakeven points when the number of test images is less than 5136 and 6483 for DeepCrack and U-Net respectively. It is observed the proposed method takes less time compared with both U-Net and DeepCrack when there are fewer images to test. However, when the number of images is greater than the breakeven values of 6483 and 5136, the proposed method may not be as efficient. Hence, for testing larger datasets, the selection of the crack segmentation method may depend on the tradeoff between the accuracy and total computation time in realistic applications.

4.4.4. Limitations and future work

Based on the experimental validation results, the proposed method proved to achieve accurate crack segmentation robustly and efficiently. To further discuss the applicability of the proposed method, the limitations and future work are included here. Fig. 19 shows the application of the proposed method on noisy images which led to false positive crack recognition. In sample (a), the noise indicated in the solid circle on the test image contained highly similar pixel properties compared to crack pixels. It was connected to the target crack region that wrongly identified it as a crack by the proposed method. As a result, this was detected as part of the crack as shown in the dashed circle in the segmentation result. Additionally, in sample (b), the discoloration of concrete surface created noisy pixels along the crack area. These were judged to contain similar intensities to the true crack pixels, as indicated in the red solid circle on the test image. This led to the presence of false positives on the segmentation result. The noise due to dark points were wrongly classified as crack pixels as shown in the dashed circle. Consequently, the crack properties measurement, especially the width measurement, was adversely affected.

Additionally, the proposed method improves the data efficiency by reducing the effort and number of images for training. Hence, a patch size of 32×32 was used in the method to achieve crack segmentation based on the characteristics of the current dataset. However, the specific patch size that is suitable may differ from situation to situation, dependent upon conditions such as the amount of data available. Due to the data limitations, the fine-tuning approach of patch size under different requirements was not explored.

Moreover, compared with deep learning-based pixel-wise crack segmentation methods, the proposed method significantly reduces the training time during the preparation process. However, due to the evolution process involved in the ACM, it takes a longer time for the proposed method to make predictions compared with pretrained CNNs. The average inference time taken to evaluate each image is approximately 1.6 s running on a single TITAN RTX GPU. The total computational time for testing larger scale datasets will correspondingly accumulate and increase. Hence, based solely on computation time, the proposed method may not be as efficient as other crack segmentation methods for larger datasets.

Based on the discussions of potential limitations of the proposed method, two possible future research directions are proposed. Firstly, eliminating the aforementioned false positives should be studied to further improve the performance of crack segmentation and properties retrieval in field applications. A possible solution could be to further investigate and improve the ACM algorithm used in the proposed method to enhance its capability to discriminate noise connected to the regions of interest from the target pixels. Secondly, another future research is to identify an ideal patch size, and investigate its impact on performance under various conditions, which would further benefit practitioners in the field. Thirdly, the current inference time is approximately 1.6 s for each image running on a single TITAN RTX GPU. Methods to reduce the inference time for testing large datasets should also be investigated in the future. A possible solution to reduce inference time could be to accelerate the evolution process involved in the ACM. Finally, the proposed robust crack segmentation method creates an accurate crack map that is essential for further measurement and characterization. Such measurement and characterization plays an important role in condition assessment. Further crack characterization methods such as crack pattern analysis and crack propagation should be investigated.

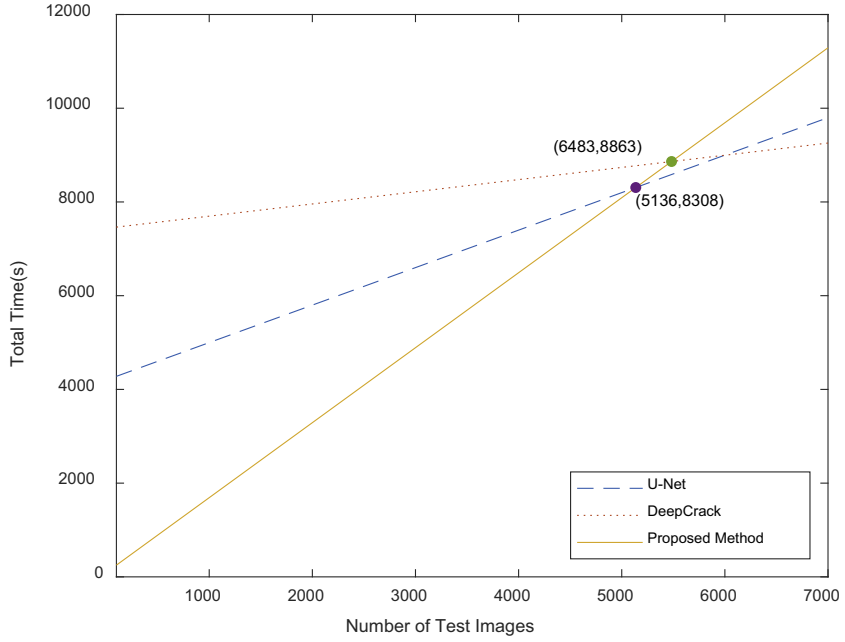

Fig. 18. Tradeoff between the number of test images and total time cost.

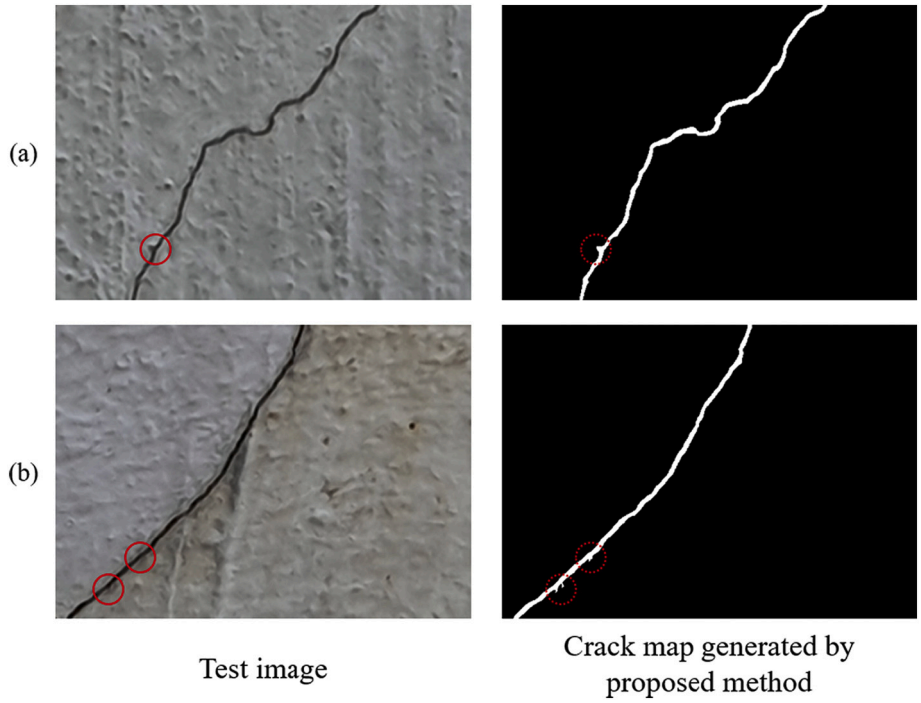

Fig. 19. Examples of inaccurate crack segmentation results from the proposed method.

5. Conclusion

In this paper, a robust image-based crack segmentation and properties retrieval method using image patches is introduced. The main objective of the proposed method is to accurately detect crack regions for further measurements using limited training data, whilst guaranteeing robustness to a high degree of noise variations present in the images, such as background texture, illumination variations, unintended objects, surface stains and blebs. In this approach, an integrated CNN with local region-based ACM is adopted to extract crack region smoothly and completely. The approach uses CNN to initialize multiple contours surrounding the crack region, and this allows the method to address the challenges of accurately and efficiently tracing irregularly shaped cracks. The ACM stage of the approach then propagates under the guidance of these initial contours settings to eliminate the misdetection of noise in the background. Finally, post-processing is introduced to produce the desirable crack map. The crack properties are further quantified based on the previous segmentation results.

The performance of the proposed crack segmentation method was validated through comparisons with related state-of-art crack detection methods on a highly-variable noisy dataset. The results demonstrated that the proposed method performed well with high accuracy,

robustness and data-efficiency in real applications. Furthermore, the crack width and length were measured in pixels based on the segmented cracks with various sizes. The experimental results illustrated the capability of the developed systematic process on accurate retrieval of crack properties in various conditions.

To address the analyzed limitations of the proposed method, possible future work could be further exploration to improve the crack segmentation performance in terms of reduction of false positives connected to the crack region. Additionally, how to fine-tune the patch size involved in the training process should be investigated under different conditions.

In closing, the contributions of this paper serve to enhance the current concrete inspection regime in buildings, by addressing problems related to accuracy of segmenting cracks in images, ensuring robustness to noise arising in concrete surfaces, as well as reducing the labeling effort needed. The results of this paper demonstrate that by combining deep learning methods with traditional techniques, more data efficient methods for crack segmentation can be explored.

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.

References

[1] D. Wong, Fixing Issue of Falling Facades in HDB Blocks, The Straits Times, 2018. https://www.straitstimes.com/singapore/housing/fixing-issue-of-falling-facades-in-hdb-blocks.

[2] N.-D. Hoang, Detection of surface crack in building structures using image processing technique with an improved Otsu method for image thresholding, Adv. Civil Eng. 2018 (2018) 1–10, https://doi.org/10.1155/2018/3924120.

[3] C. Koch, K. Georgieva, V. Kasireddy, B. Akinci, P. Fieguth, A review on computer vision based defect detection and condition assessment of concrete and asphalt civil infrastructure, Adv. Eng. Inform. 29 (2) (2015) 196–210, https://doi.org/10.1016/j.aei.2015.01.008.

[4] G. Li, S. He, Y. Ju, K. Du, Long-distance precision inspection method for bridge cracks with image processing, Autom. Constr. 41 (2014) 83–95, https://doi.org/10.1016/j.autcon.2013.10.021.

[5] S. Li, Y. Cao, H. Cai, Automatic pavement-crack detection and segmentation based on steerable matched filtering and an active contour model, J. Comput. Civil Eng. 31 (5) (2017), https://doi.org/10.1061/(asce)cp.1943-5487.0000695.

[6] T. Yamaguchi, S. Hashimoto, Fast crack detection method for large-size concrete surface images using percolation-based image processing, Mach. Vis. Appl. 21 (5) (2009) 797–809, https://doi.org/10.1007/s00138-009-0189-8.

[7] Z. Zhu, S. German, I. Brilakis, Visual retrieval of concrete crack properties for automated post-earthquake structural safety evaluation, Autom. Constr. 20 (7) (2011) 874–883, https://doi.org/10.1016/j.autcon.2011.03.004.

[8] D. Lattanzi, G.R. Miller, Robust automated concrete damage detection algorithms for field applications, J. Comput. Civ. Eng. 28 (2) (2014) 253–262, https://doi.org/10.1061/(asce)cp.1943-5487.0000257.

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

[10] M.R. Jahanshahi, S.F. Masri, Adaptive vision-based crack detection using 3D scene reconstruction for condition assessment of structures, Autom. Constr. 22 (2012) 567–576, https://doi.org/10.1016/j.autcon.2011.11.018.

[11] M. Torok Matthew, M. Golparvar-Fard, B. Kochersberger Kevin, Image-based automated 3D crack detection for post-disaster building assessment, J. Comput. Civ. Eng. 28 (5) (2014), A4014004, https://doi.org/10.1061/(ASCE)CP.1943-5487.0000334.

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

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

[14] Z. Liu, Y. Cao, Y. Wang, W. Wang, Computer vision-based concrete crack detection using U-net fully convolutional networks, Autom. Constr. 104 (2019) 129–139, https://doi.org/10.1016/j.autcon.2019.04.005.

[15] Y. Liu, J.K.W. Yeoh, Vision-Based Semi-Supervised Learning Method for Concrete Crack Detection, Construction Research Congress, 2020, pp. 527–536, https://doi.org/10.1061/9780784482865.056.

[16] G. Xu, Z. Song, Z. Sun, C. Ku, Z. Yang, C. Liu, S. Wang, J. Ma, W. Xu, CAMEL: a weakly supervised learning framework for histopathology image segmentation, in: Proceedings of the IEEE International Conference on Computer Vision, 2019, pp. 10682–10691, https://doi.org/10.1109/ICCV.2019.01078.

[17] Y. Wei, J. Feng, X. Liang, M. Cheng, Y. Zhao, S. Yan, Object Region Mining with Adversarial Erasing: A Simple Classification to Semantic Segmentation Approach, IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 6488–6496, https://doi.org/10.1109/CVPR.2017.687.

[18] I. Abdel-Qader, S. Pashaie-Rad, O. Abudayyeh, S. Yehia, PCA-based algorithm for unsupervised bridge crack detection, Adv. Eng. Softw. 37 (12) (2006) 771–778, https://doi.org/10.1016/j.advengsoft.2006.06.002.

[19] P. Prateek, D. Kristin, G. Nenad, B. Basily, Computer-vision based crack detection and analysis, in: Proc.SPIE Vol. 8345, 2012, https://doi.org/10.1117/12.915384.

[20] C. Feng, M.-Y. Liu, C.-C. Kao, T.-Y. Lee, Deep active learning for civil infrastructure defect detection and classification, Comput. Civil Eng. (2017) 298–306, https://doi.org/10.1061/9780784480823.036.

[21] Y.-J. Cha, W. Choi, O. Büyüköztürk, Deep learning-based crack damage detection using convolutional neural networks, Comput. Aid. Civil Infrastruct. Eng. 32 (5) (2017) 361–378, https://doi.org/10.1111/mice.12263.

[22] H. Nhat-Duc, Q.-L. Nguyen, V.-D. Tran, Automatic recognition of asphalt pavement cracks using metaheuristic optimized edge detection algorithms and convolution neural network, Autom. Constr. 94 (2018) 203–213, https://doi.org/10.1016/j.autcon.2018.07.008.

[23] K. Gopalakrishnan, S.K. Khaitan, A. Choudhary, A. Agrawal, Deep convolutional neural networks with transfer learning for computer vision-based data-driven pavement distress detection, Constr. Build. Mater. 157 (2017) 322–330, https://doi.org/10.1016/j.conbuildmat.2017.09.110.

[24] K. Zhang, H.D. Cheng, B. Zhang, Unified approach to pavement crack and sealed crack detection using preclassification based on transfer learning, J. Comput. Civil Eng. 32 (2) (2018), https://doi.org/10.1061/(asce)cp.1943-5487.0000736.

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

[26] S.-N. Yu, J.-H. Jang, C.-S. Han, Auto inspection system using a mobile robot for detecting concrete cracks in a tunnel, Autom. Constr. 16 (3) (2007) 255–261, https://doi.org/10.1016/j.autcon.2006.05.003.

[27] Y. Fujita, Y. Hamamoto, A robust automatic crack detection method from noisy concrete surfaces, Mach. Vis. Appl. 22 (2) (2010) 245–254, https://doi.org/10.1007/s00138-009-0244-5.

[28] Y. Shi, L. Cui, Z. Qi, F. Meng, Z. Chen, Automatic road crack detection using random structured forests, IEEE Trans. Intell. Transp. Syst. 17 (12) (2016) 3434–3445, https://doi.org/10.1109/tits.2016.2552248.

[29] X. Yang, H. Li, Y. Yu, X. Luo, T. Huang, X. Yang, Automatic pixel-level crack detection and measurement using fully convolutional network, Comput. Aid. Civil Infrastruct. Eng. 33 (12) (2018) 1090–1109, https://doi.org/10.1111/mice.12412.

[30] D. Dias-da-Costa, J. Valença, E. Júlio, H. Araújo, Crack propagation monitoring using an image deformation approach, Struct. Control Health Monit. 24 (10) (2017), https://doi.org/10.1002/stc.1973.

[31] X. Kong, J. Li, Vision-based fatigue crack detection of steel structures using video feature tracking, Comput. Aid. Civil Infrastruct. Eng. 33 (9) (2018) 783–799, https://doi.org/10.1111/mice.12353.

[32] X. Kong, J. Li, Non-contact fatigue crack detection in civil infrastructure through image overlapping and crack breathing sensing, Autom. Constr. 99 (2019) 125–139, https://doi.org/10.1016/j.autcon.2018.12.011.

[33] F.-C. Chen, M.R. Jahanshahi, R.-T. Wu, C. Joffe, A texture-based video processing methodology using bayesian data fusion for autonomous crack detection on metallic surfaces, Comput. Aid. Civil Infrastruct. Eng. 32 (4) (2017) 271–287, https://doi.org/10.1111/mice.12256.

[34] M. Kass, A. Witkin, D. Terzopoulos, Snakes: active contour models, Int. J. Comput. Vis. 1 (4) (1988) 321–331, https://doi.org/10.1007/BF00133570.

[35] V. Caselles, R. Kimmel, G. Sapiro, Geodesic active contours, Int. J. Comput. Vis. 22 (1) (1997) 61–79, https://doi.org/10.1023/A:1007979827043.

[36] Z. Guopu, Z. Shuqun, Z. Qingshuang, W. Changhong, Boundary-based image segmentation using binary level set method, Opt. Eng. 46 (5) (2007) 1–3, https://doi.org/10.1117/1.2740762.

[37] N. Paragios, R. Deriche, Geodesic active regions and level set methods for supervised texture segmentation, Int. J. Comput. Vis. 46 (3) (2002) 223–247, https://doi.org/10.1023/A:1014080923068.

[38] L. Chunming, X. Chenyang, G. Changfeng, M.D. Fox, Level set evolution without re-initialization: a new variational formulation, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition Vol. 1, 2005, pp. 430–436, vol. 431, https://doi.org/10.1109/CVPR.2005.213.

[39] T.F. Chan, L.A. Vese, Active contours without edges, IEEE Trans. Image Process. 10 (2) (2001) 266–277, https://doi.org/10.1109/83.902291.

[40] A. Tsai, A. Yezzi, A.S. Willsky, Curve evolution implementation of the Mumford-Shah functional for image segmentation, denoising, interpolation, and magnification, IEEE Trans. Image Process. 10 (8) (2001) 1169–1186, https://doi.org/10.1109/83.935033.

[41] J. Lie, M. Lysaker, T. Xue-Cheng, A binary level set model and some applications to Mumford-Shah image segmentation, IEEE Trans. Image Process. 15 (5) (2006) 1171–1181, https://doi.org/10.1109/TIP.2005.863956.

[42] L.A. Vese, T.F. Chan, A multiphase level set framework for image segmentation using the Mumford and Shah model, Int. J. Comput. Vis. 50 (3) (2002) 271–293, https://doi.org/10.1023/A:1020874308076.

[43] K. Zhang, L. Zhang, K.-M. Lam, D. Zhang, A Local Active Contour Model for Image Segmentation With Intensity Inhomogeneity, arXiv preprint, 2013. arXiv:1305.7053. https://arxiv.org/abs/1305.7053.

[44] C. Li, R. Huang, Z. Ding, J.C. Gatenby, D.N. Metaxas, J.C. Gore, A level set method for image segmentation in the presence of intensity inhomogeneities with

application to MRI, IEEE Trans. Image Process. 20 (7) (2011) 2007–2016, https://doi.org/10.1109/TIP.2011.2146190.
[45] A. Krizhevsky, I. Sutskever, G.E. Hinton, ImageNet classification with deep convolutional neural networks, Commun. ACM 60 (6) (2017) 84–90, https://doi.org/10.1145/3065386.
[46] J. Deng, W. Dong, R. Socher, L. Li, L. Kai, F.-F. Li, ImageNet: A large-scale hierarchical image database, in: IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 248–255, https://doi.org/10.1109/CVPR.2009.5206848.
[47] H.C. Shin, H.R. Roth, M. Gao, L. Lu, Z. Xu, I. Nogues, J. Yao, D. Mollura, R. M. Summers, Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning, IEEE Trans. Med. Imaging 35 (5) (2016) 1285–1298, https://doi.org/10.1109/TMI.2016.2528162.
[48] R. Van Uitert, I. Bitter, Subvoxel precise skeletons of volumetric data based on fast marching methods, Med. Phys. 34 (2) (2007) 627–638, https://doi.org/10.1118/1.2409238.
[49] D. George, W. Lehrach, K. Kansky, M. Lázaro-Gredilla, C. Laan, B. Marthi, X. Lou, Z. Meng, Y. Liu, H. Wang, A. Lavin, D.S. Phoenix, A generative vision model that trains with high data efficiency and breaks text-based CAPTCHAs, Science 358 (6368) (2017), https://doi.org/10.1126/science.aag2612 p. eaag2612.