Smart Structures and Systems, *Vol. 30, No. 4 (2022) 385-395*
https://doi.org/10.12989/sss.2022.30.4.385

Physical interpretation of concrete crack images from feature estimation and classification

Eunbyul Koh [1a], Seung-Seop Jin [2b] and Robin Eunju Kim*[1]

[1] *Department of Civil and Environmental Engineering, Hanyang University, 222, Wangsimni-ro, Seongdong-gu, Seoul 04763, Korea*
[2] *Department of Structural Engineering Research, Korea Institute of Civil and Building Technology, 283, Goyang-daero, Ilsanseo-gu, Goyang-si, Gyeonggi-do 10223, Korea*

(*Received December 23, 2021, Revised May 31, 2022, Accepted July 8, 2022*)

Abstract. Detecting cracks on a concrete structure is crucial for structural maintenance, a crack being an indicator of possible damage. Conventional crack detection methods which include visual inspection and non-destructive equipment, are typically limited to a small region and require time-consuming processes. Recently, to reduce the human intervention in the inspections, various researchers have sought computer vision-based crack analyses: One class is filter-based methods, which effectively transforms the image to detect crack edges. The other class is using deep-learning algorithms. For example, convolutional neural networks have shown high precision in identifying cracks in an image. However, when the objective is to classify not only the existence of crack but also the types of cracks, only a few studies have been reported, limiting their practical use. Thus, the presented study develops an image processing procedure that detects cracks and classifies crack types; whether the image contains a crazing-type, single crack, or multiple cracks. The properties and steps in the algorithm have been developed using field-obtained images. Subsequently, the algorithm is validated from additional 227 images obtained from an open database. For test datasets, the proposed algorithm showed accuracy of 92.8% in average. In summary, the developed algorithm can precisely classify crazing-type images, while some single crack images may misclassify into multiple cracks, yielding conservative results. As a result, the successful results of the presented study show potentials of using vision-based technologies for providing crack information with reduced human intervention.

Keywords: crack assessment; crack detection; crazing; image processing; multiple crack; single crack

1. Introduction

Concrete, the most common material in the construction industry, is prone to have cracks within their service life, requiring systematic and regular inspection. Some of those cracks may degrade the serviceability and safety of the structure, while some other types may be less effective. Thus, in examining concrete cracks, the quantitative judgment of crack types, in terms of the severity, needs to be made for proper decisions on repair and maintenance plans. Traditional crack investigations usually rely on visual assessments, followed by sophisticated measurement equipment on areas to be examined in detail. However, such approaches may yield problems associated with high cost, laborious human intervention, time-intensive, and localized assessments (Chang *et al*. 2003, Jahanshahi *et al*. 2011, Moore *et al*. 2001, Jung *et al*. 2019).

With the recent advances in computer vision-based techniques, the use of images (or videos) for crack investigations has been sought intensively anticipating those as a potential automated inspection approach. In an earlier period of the efforts, classical filter approaches are adapted for identifying edges of cracks (Abdel-Qader *et al*. 2003). Such approaches revealed that the cracks can be detected from portable devices, while the accuracy may be greatly affected by the filter characteristics (Maini and Aggarwal 2009, Nishikawa *et al*. 2012). Then to enhance the accuracy, deep learning-based algorithms have been proposed in concrete crack inspections. Most of the existing computer vision-based damage detection techniques today falls into either of those two categories.

Classical filter approaches transform an image using edge detection algorithms to only filter the crack edges. This class of techniques in earlier times was realized by processing the image via Sobel, Canny, Otsu's, and Hough transform methods (Canny 1986, Hough 1962, Otsu 1979, Sobel and Feldman 1968). One of the major advantages is their simplicity in the implementation. Then, to increase the adaptability and to reduce the noises effectively, pre-processing filters using contrast enhancement, smoothing, Gaussian filter, and/or median filter are employed (Biggs and Andrews 1997, Deng and Cahill 1993, Parker 1991, Wang *et al*. 2019). For example, Abdel-Qader *et al*. (2003) and Wang and Huang (2010) compared the existing edge detection methods and demonstrated the applicability of the approach (Abdel-Qader *et al*. 2003, Wang and Huang 2010). To further enhance the results by reducing the

*Corresponding author, Assistant Professor,
E-mail: robinekim@hanyang.ac.kr
[a] Ph.D. Student, E-mail: ebkoh@hanyang.ac.kr
[b] Ph.D., E-mail: seungsab@kict.re.kr

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

background noises and the effects of lights, several edge detecting filters are proposed as an integrated procedure (Ahmad *et al*. 2020, Liu *et al*. 2020). Talab *et al*. (2016) detected cracks by combining Sobel and Otsu method without pre-preprocessing the image (Talab *et al*. 2016). Qiang *et al*. (2016) proposed an approach that is based on an iterative threshold segmentation and the Canny method (Qiang *et al*. 2016). Hoang (2018) implemented a gray intensity adjustment method, in which the threshold is automatically estimated from the Otsu method (Hoang 2018). To further achieve more actionable information, Wang *et al*. (2019) used an improved Otsu method and identified the direction of crack (whether the crack is in vertical, diagonal, or transversal), along with the width and the length estimation from the pixel distances (Wang *et al*. 2019). Then, more advanced filters were adopted to enhance the performance (Han *et al*. 2020, Xuan and Hong 2017, Zheng *et al*. 2020). Shahrokhinasab *et al*. (2020) investigated the Digital Image Correlation method by comparing multiple images from the same scene, and reduced the background noises (Shahrokhinasab *et al*. 2020). Classical filter-based approaches are still widely adopted and being improved with their merits on portability, while the research direction these days are claimed to employ mathematic-based evaluation and to realize autonomous investigations (Spencer *et al*. 2019).

The advancement of deep-learning algorithms for computer vision-based technologies in various fields has driven research efforts to apply those methods for detecting structural defects (Ye *et al*. 2019). Among various deep-learning algorithms, a multilayer Neural Network (NN) has been adopted for autonomous learning of crack features. One of the most actively used methods is Convolutional Neural Networks (CNNs), while other variations of NNs have also been also investigated (Cha *et al*. 2017, Kalchbrenner *et al*. 2014, Krizhevsky *et al*. 2012). Mostly, with the primary objectives being able to detect cracks, the realizations can be categorized into i) classification, which determines crack or non-crack zone by process image patch, ii) object detection, where the and method outputs bounding boxes around the crack locations iii) segmentation, which detects cracks in pixels (Cha *et al*. 2017, Hsieh and Tsai 2020, LeCun *et al*. 2015). Successful results in detecting cracks with the aforementioned algorithms at high precision (over 85%) can be found in various literature (Narazaki *et al*. 2020, Spencer Jr *et al*. 2019). However, the primary focus of previous research has been made on detecting the existence of cracks and reducing background noise (Liu and Yeoh 2021, Qi *et al*. 2021, Lee *et al*. 2021, Li *et al*. 2022, Wang and Xiang 2021).

To dates, only a few studies have proposed algorithms that classifies concrete surface cracks. In the early stage, crack width, length, and orientation have been extracted and interpreted as the main indicators of crack severity (Adhikari *et al*. 2014, Zhu *et al*. 2011, Wang *et al*. 2021). For instance, Flash *et al*. (2020) used the Keras classifier with the Otsu image processing algorithm to estimate crack width presenting about 96% accuracy. However, their algorithm is limited to a single crack application, showing degraded results for images with intersecting cracks. Then, because crack patterns are also important for assessing the extent of the crack, Liu and Yeoh (2021) adopted a Deep Image Structure and Texture Similarity (DISTS) index to recognize crack patterns and differentiate whether the cracks are structural or non-structural. DISTS index compares a similarity of a statistically calculated texture parameters between input images. Although the DISTS-based algorithm yields about 98% accuracy, the index is intrinsically affected by the surface texture and the training database. Thus, algorithms that can analyze and interpret the characteristics of crack types need to be further developed.

Thus, in this paper, an algorithm is proposed with primary focus made (i) to differentiate whether an image contains structural crack or non-structural crack (crazing); and (iii) to detect radial crack zone, which multiple cracks in different directions meet. The overall procedure and selected test images are described in Section 2. In Section 3, detailed processes and key parameters for detecting radial crack are discussed. And methodologies for classifying structural crack and crazing are proposed. Following that, the performance of the proposed algorithm is validated in Section 4 using 227 open database images. The results of the presented study successfully demonstrated the potential use of computer vision-based technologies for not only detecting the existence but also yielding physical crack interpretation that can assist the more practical maintenance.

2. Overall procedure and preprocessing test images

2.1 Framework

This section develops an algorithm that categorizes cracks based on crack image interpretation. As shown in Fig. 1, the overall procedure is composed of four major steps: (i) Preprocessing (Acquisition & Edge detection);

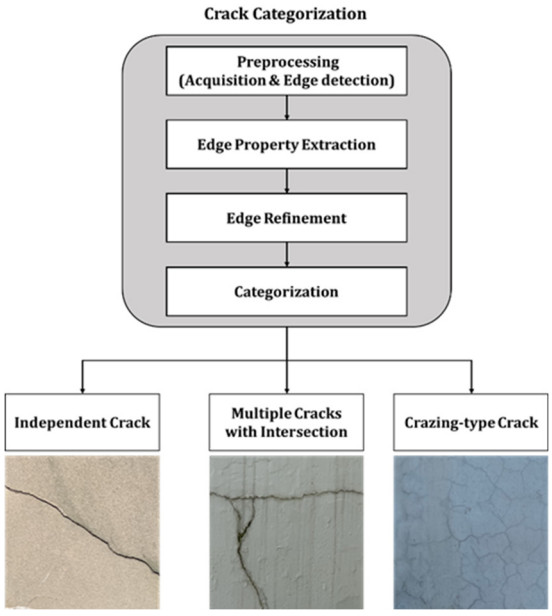


Fig. 1 Overall framework

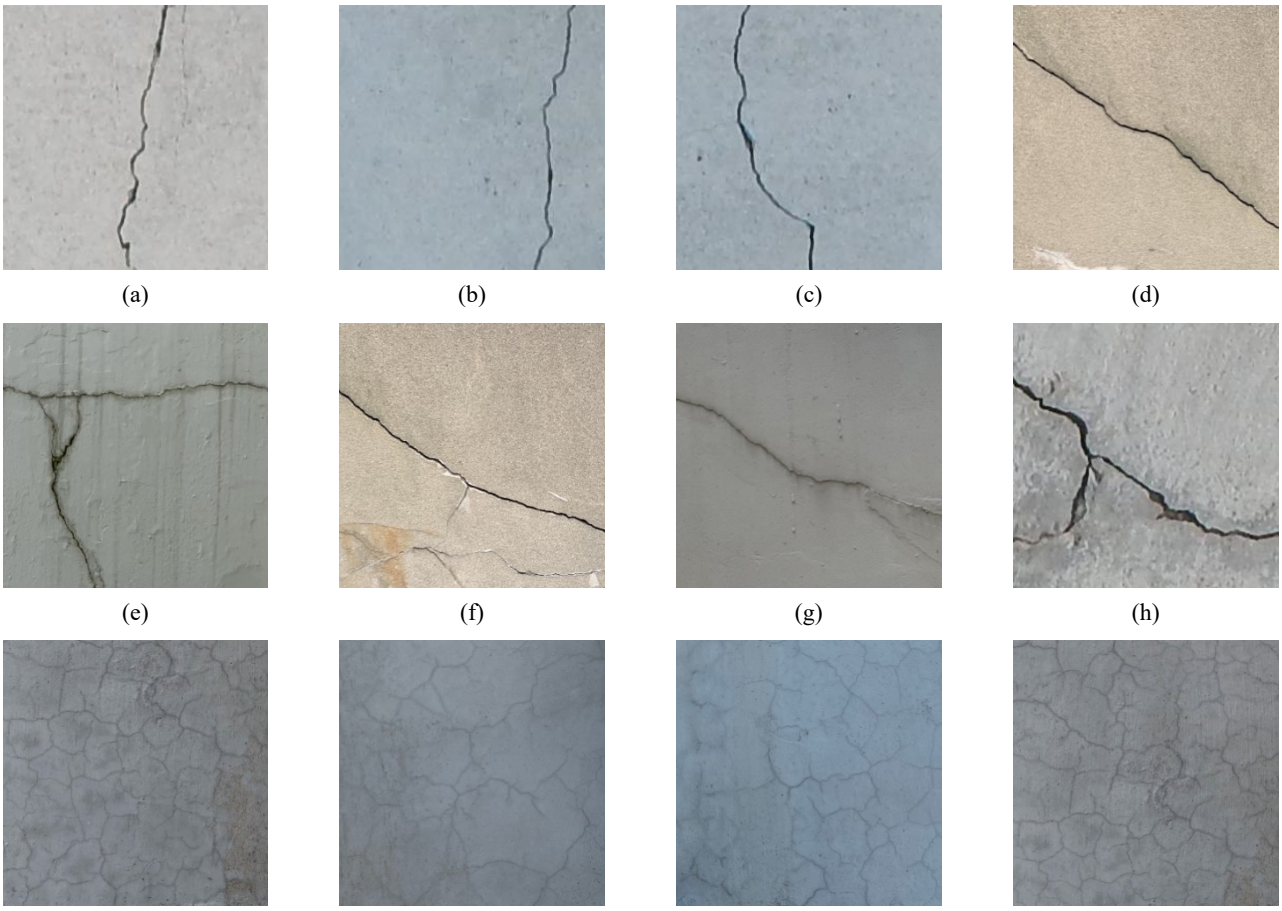


Fig. 2 Example images of independent cracks: (a)-(d) single crack; (e)-(f) multiple cracks; (i)-(l) crazing-type cracks

(ii) Edge Property Extraction; (iii) Edge Refinement; and (iv) Categorization. In the first step, the 3-dimensional input image (in RGB format) is resized and transcoded into the 2-dimensional gray-scale format. Then, the grayscale image is binarized with a local-binarization algorithm in the edge detection step. Because the concrete surface contains non-crack related noises, due to surface irregularities or background, binarized pixels are refined in the second step as well. In the categorization step, binarized pixels are isolated firstly and then re-bonded based on each isolated crack's attributes. At the same time, crack intersecting zones are searched using a radial box, which calculates how many cracks with different angles are co-located. As a result, a crack image is categorized whether it contains a single crack, multiple cracks with intersections, or crazing. Each step of the proposed framework is discussed in detail in the following sections.

2.2 Data preparation

Two types of database are prepared for (i) developing the framework and (ii) validating the framework. In the algorithm development phase, a total of 93 images which contains both structural cracks (50 images) and crazing-type cracks (43 images) were used. Those images are photographed by the authors: Structural cracks were taken at building walls in Hanyang University, Seoul, South Korea. More specifically images were carefully taken to contain 17 single crack images and 33 multiple cracks images with at least one intersection. Representative images for those categories are shown in Figs. 2(a)-(d) and Figs. 2(e)-(h), respectively. Non-structural cracks were taken from a specimen provided by the Korea Institute of Civil Engineering and Building Technology (KICT). Note that the specimen from KICT is a part of the Seodaemun Overpass in South Korea, which was used from 1971 to 2015. Due to its age and demolishing and hauling jobs, various types of cracks have been created. Sample images of the crazing case are shown in Figs. 2(i)-(j). When collecting the image, the authors used a cell phone camera, iPhone XR, 12 MP wide camera, positioning at a fixed distance of 1.5m from the target object, with the resolution of 4,032 × 3,024. Then, each image is cropped at the center by the size of 1,000 × 1,000 pixels, which corresponds to the actual size of 50 × 50 cm.

The other dataset used in the validation phase is retrieved from an open dataset, SDNET2018 (Maguire *et al.* 2018). SDNET2018 contains over 56,000 images, of which size is 256 × 256 pixels taken from bridge decks, walls, and pavements. Among those, images from walls that contain clear single or multiple cracks are selected for validation.

3. Methodology

3.1 Preprocessing (Acquisition & Edge detection)

The foremost step to acquire the information of the crack is resizing and binarization. To enhance the effectiveness of the process, the input images are further resized into 500 × 500 pixels. When resizing, bicubic interpolation is applied which the output value is a weighted average of pixels in the nearest 4-by-4 pixels. When resized, the quality of the image is mathematically lowered, allowing filtering effect on the noise due to surface roughness. Note that this sizing ratio is found to have the smallest size that yields a robust outcome for the proposed algorithm. The RGB images are then transcoded to grayscale with the Matlab® (Math Works 2021) built-in unction, rgb2gray. Default intensity values of each R, G, and B channel are used.

The two-dimensional grayscale image is further binarized, followed by the edge detection step. Among various edge detecting algorithms, the Canny algorithm is used here due to its simplicity (Canny 1986). The Canny algorithm blurs the input image with the Gaussian filter and then a mask finds the intensity gradient of the image for finding edges. Using double thresholds, edges are separated into strong and thin edges. Finally, the algorithm finds all thin edges that are connected with a strong edge in any of the 8-directions. The algorithm has been realized using an image processing toolbox embedded in Matlab® (Math Works 2021). In this study, the upper and the lower limits are found to yield the best results in the overall process, which are 0.25 and 0.01, respectively. By applying the Canny algorithm, the crack edges are nominally well detected. However, due to the nature of concretes whose surfaces are uneven and the colors are not uniform, non-cracks are also detected as shown in Fig. 3. Thus, those non-crack-related outputs from the Canny algorithm need to be refined to enhance the results.

3.2 Edge property extraction

In Matlab, various measures of a binary image can be extracted using a function `regionprops`. This function returns the user-selected features for each 8-connected component. In this presented work, following 3 features are calculated and used as the edge properties; i) Circularity, ii) Extent, and iii) Solidity. The circularity is calculated as the ratio between the area of the field and its perimeter (P_c) (Milan *et al*. 1998, Souza and Menegalli 2011, Kröner and Carbó 2013)

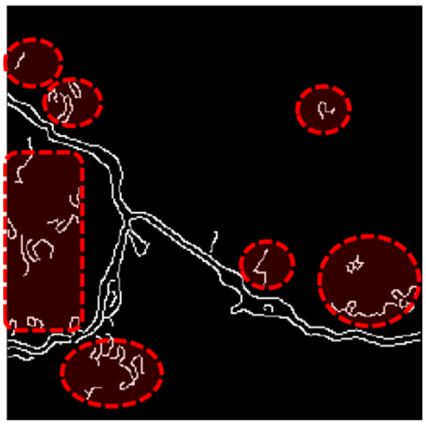


Fig. 3 Canny filter results for image Fig. 2(h)

$$C_C = \frac{4\pi A_C}{(P_C)^2} \tag{1}$$

where A_c is the area of the connected pixels in the input image, labels based on each field, as below

$$A_C = \sum_{i=1}^{N_{field}} P_{xy} \tag{2}$$

Here N_{field} is the total number of pixels (P_{xy}) in the field. Note that the subscript x and y refer to row and column in the image. Also, the physical implication of A_c indicates the area of an object in an input image, and the area in a binary image implies the actual number of pixels in the region, returned as a scalar. The object is defined here as a connected group of foreground pixels (8-value two-dimensional connectives along the horizontal, vertical, or diagonal direction) in a binary image. The value of C_c varies from 0 to 1, which near 0 value implies that the field is close to long and narrow shape, and similar to a circle for $C_c \approx 1$. The extent (E_c) is the ratio of A_c to the area in the total regional box (A_B). Note that A_B is the smallest box area that contains the field.

$$E_c = \frac{A_C}{A_B} \tag{3}$$

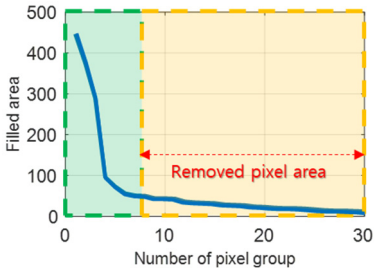
Lastly, the solidity (S_c) examines the density of the field, which the value is calculated from the following equation

$$S_c = \frac{A_C}{A_{convex}} \tag{4}$$

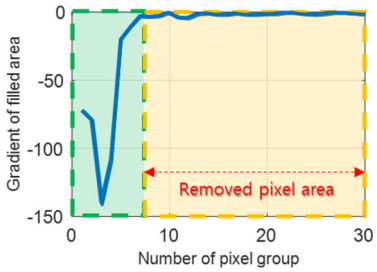
here, A_{convex} specifies the number of pixels of the filled-in hull in each field (The Math Works 2021). S_c varies within 0 to 1, yielding near 1 for a convex polygon. Although these extracted features are well found for the selected field, the information about the location of the field cannot be returned. Thus, in this paper, `bwboundaries` is used subsequently to export the pixel locations of the field. Note that by linking the locational information with results from `regionprops`, one can not only preserve the global locations but also ease the computation in the subsequent edge detection procedure.

3.3 Edge refinement

To eliminate the non-crack edges, due to concrete surface irregularities, a pixel refinement strategy is proposed on the detected edges based on the area of connected edges. First, A_c is sorted by its size. As illustrated in Fig. 4, typical A_c shows an exponential drop. A larger A_c is likely to be crack edges, while the smaller



(a) Filled area graph



(b) Gradient of filled area graph

Fig. 4 Edge Area for Fig. 2(h)

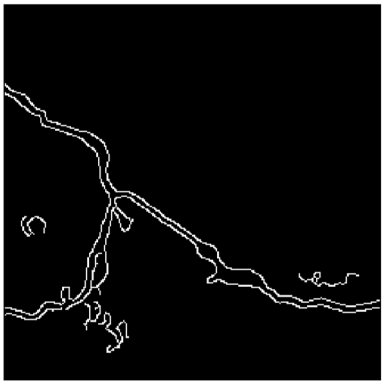


Fig. 5 Refined images for Fig. 2(e)

A_c may indicate non-crack edges. When k^{th} A_c is $(A_c)_k$ and k–1^{th} is $(A_c)_{k-1}$, $(A_c)_k - (A_c)_{k-1}$ becomes –1 when they have only a one-pixel difference. Considering the randomness of the crack, converging trends in the difference may be less realistic to be assumed as cracks. Thus, ΔEdge is defined by the gradient of A_c

$$\Delta Edge = \frac{(A_c)_k - (A_c)_{k-1}}{2}, \quad k = 2, \cdots, N_{field} \qquad (5)$$

where N_{edge} is the total number of the field. ΔEdge is plotted in Fig. 4(b) showing a gradual increase. Using Eq. (1) and Fig. 4(b), fields are then removed for the edges where ΔEdge converges to –0.5 or 0. Such a strategy implies that if areas from two different fields have the same or one-pixel difference, the area becomes an indicator of non-crack edges. The resulting refined images are shown in Fig. 5. As can be seen, non-crack fields that were presented in Fig. 3 are well removed. Especially, pores of the concrete surfaces are fairly refined demonstrating the effectiveness of pixel refinement strategy. Note that the features of the refined field are also eliminated accordingly for further evaluation.

3.4 Categorization

This section develops Categorization process that evaluates cracks from the identified crack edges. A flowchart of the proposed method is illustrated in Fig. 6. Given a preprocessed image, the geometrical analysis is applied firstly, which connects crack properties with locational information. In the crack properties, Hough transformation is utilized along with the features extracted from Section 3.2, which this function sorts the cracks with the same angle. Hough detects group points that share the same absolute angle (θ) and distance (ρ) from an origin. Thus, detected points imply a possible line from the categorized image matrix. Several functions are applied to further interpret Hough results. For example, `numpeaks` exports the maximum number of lines to identify, where the default value is 10, `Fillgap`, implies the minimum spacing between two identified lines to avoid overlap, with the default being 20. Also, MinLength is the minimum length of a line with a default value of 20. A detected line is ignored if it is shorter than MinLength. Then, as shown in Fig. 7, all the properties are combined with locational information to yield labelled cracks with property information embedded.

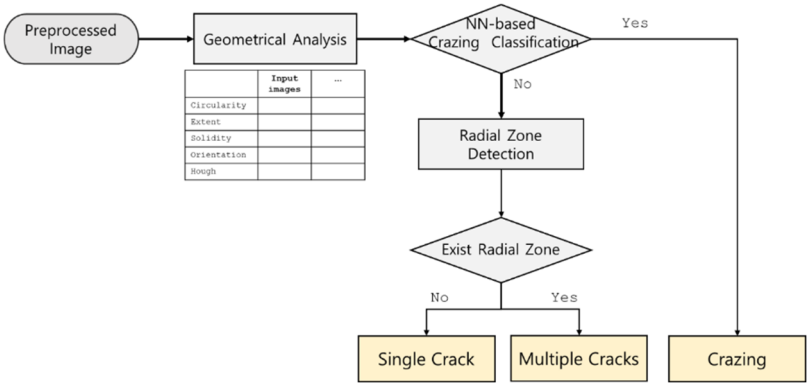


Fig. 6 Flowchart of crack type classification

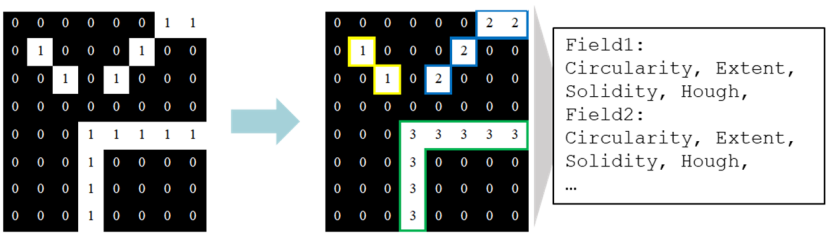


Fig. 7 Labeled crack with embedded property information

Using a table constructed for crack attributes, the fully connected neural network (NN) is applied to classify whether the properties of the input image contain crazing cracks. Note that in the classification, a wide neural network is selected to categorize the image as crazing if the number of crack fields are large and crack edges contain various angles: When the image is classified into crazing, the report is generated with no further process. On the other hand, for the input properties classified into structural cracks, the radial zone detection step is further processed.

Subsequently, the concept of the bounding box is employed to detect the locations that contain the crack intersection, which is defined as the radial zone herein. The bounding box with a specified pixels explores the categorized matrix (resulting in a transformed image matrix) from the top-left to the bottom-right. Then the box examines if the region contains more than three distinct label IDs (identification number as illustrated in Fig. 7). If this condition is met, the box is indicated as a radial zone. After this scrutinizing step, radial zones are unified. Note that the size of a bounding box is small (20×20 pixels) compared to an input image used in this framework. Thus, unified, the resulting boxes become polygonal shapes. Such polygonal shape is beneficial to fit only the crack intersection and less prone to yield oversized detections. As illustrated in Fig. 6, if no radial box is detected from this step, the image is categorized into a single crack image, otherwise categorized into multiple cracks. Note that the size of the bounding box affects the results of radial zone: For example, if the box is too large, false detection may occur for two distinct cracks. Thus, herein, the size of the bounding is carefully set as 20×20 pixels, which yields

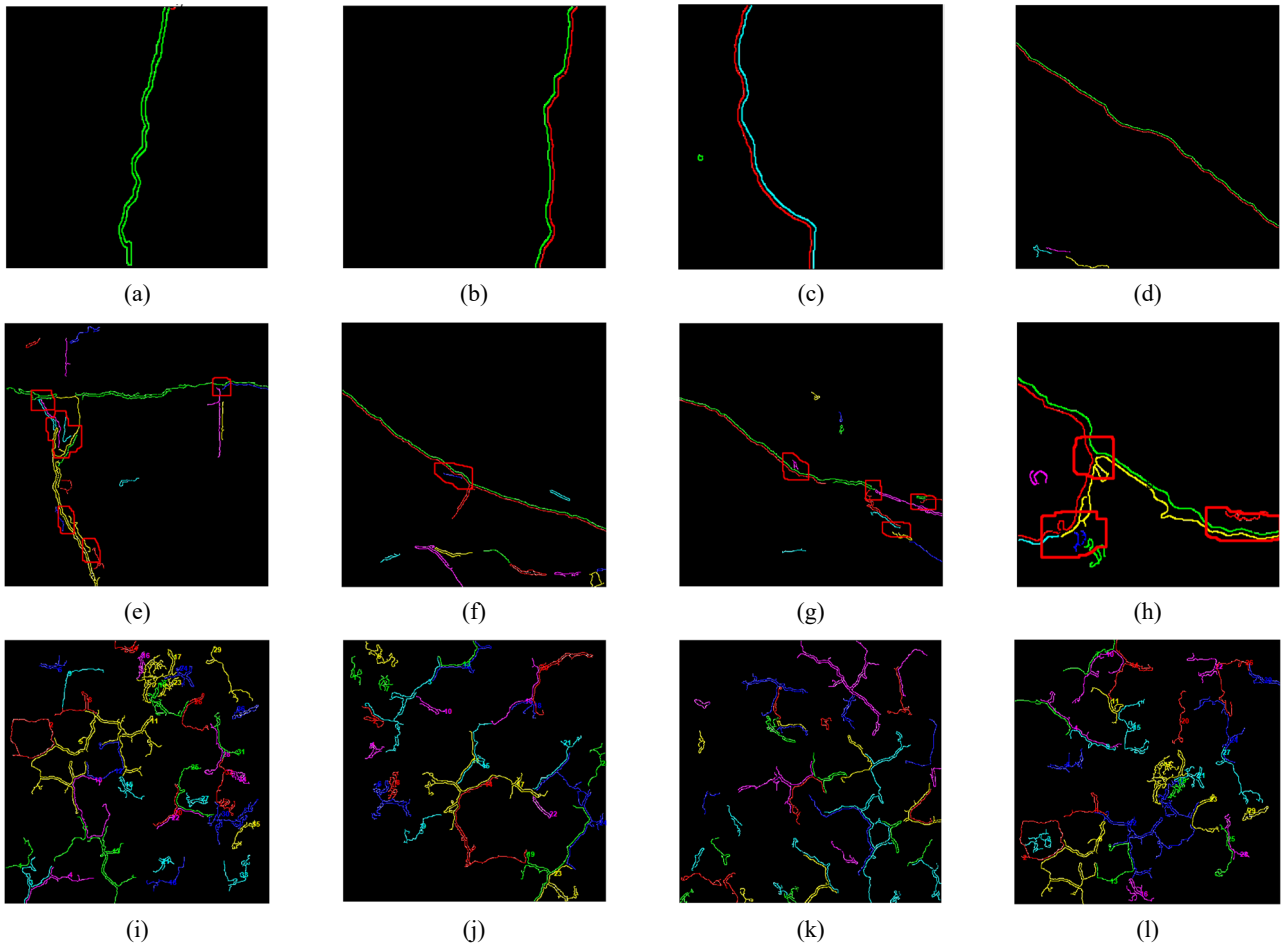


Fig. 8 Crack type classification results

good performance for input image sizes used in this paper.

The results of the proposed framework are summarized in Fig. 8. Different colors in detected results indicate the categorized fields, and a red box implies the radial zone. As can be seen, single crack images in Figs. 8(a)-(d) are well detected with no radial zone found. Each line-shape crack is detected as a continuous field. Multiple cracks in Figs. 8(e)-(h) are also well classified, which the size of red boxes indicates the severity of the radial crack. Although those crack images have been reported as multiple cracks, the performance within each image are closely examined using precision and recall

$$\text{Precision} = \frac{\text{true positive}}{\text{predicted positive}} = \frac{TP}{TP + FP} \tag{6}$$

$$\text{Recall} = \frac{\text{true positive}}{\text{actual positive}} = \frac{TP}{TP + FN} \tag{7}$$

here TP is true positive, FP is false positive, and FN is false negative. Precisions for Figs. 8(e)-(h) are 40%, 100%, 25%, and 33%, while recall values are found as 67%, 25%, 100%, and 100%, respectively. In three cases, recall showed higher value than precision, which may imply that the proposed algorithm yields conservative identification in finding radial zones, more false positive than false negative. Finally, Figs. 8(i)-(l) are well classified as crazing-type images. Note that the resulting images only field indications, because the algorithm does not search for radial zones once detected as crazing from NN-based classification. Fig. 9 compares the crack attributes for single, multiple, and crazing, which are input parameters of the classifier. As can be seen, average C_C, E_C, and S_C show smaller values for crazing-type images, assuring the clear classification.

4. Algorithm validation

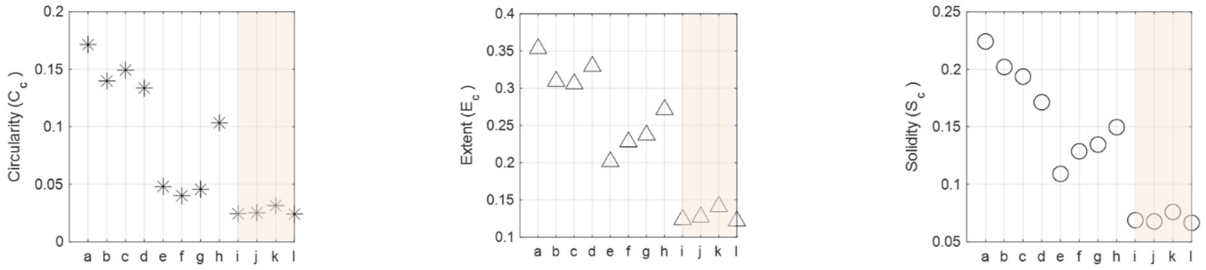
The performance of the proposed algorithm is validated in this section using a database, which is constructed for images taken by the researchers and selected images from SDNET 2018. In total, 227 images are used and each is labelled into single, multiple, and crazing cracks. Then, to reduce the overfitting, the dataset is randomly split into two sets; the training set consists of 80% (181 images), and the remaining 20% serves as a validation set (46 images). The distribution of dataset is summarized in Table 1.

Prediction results for the training and test dataset are summarized as confusion matrices in Fig. 11. A confusion matrix is an efficient tool that visualizes the performance of the algorithm. Each row of the matrix indicates the true class, while the column represents the predicted class, with a probability of correct or incorrect predictions are noted in cells. The classification using the training dataset performed quite well with highly accurate prediction and low misclassification as can be seen in Fig. 11(a). Note that the NN-based classifier, a shallow neural network, using Matlab® (Math Works 2021) is developed here to sort all crazing-type images accurately. The architecture of the proposed model is shown in Fig. 10; the number of fully connected layers is 1, and the size of the first layer is 10. Also, the activation function used is ReLU, the repetition limit of 1000, and the normalization strength was set to zero.

Using the developed NN-based model, the structural cracks are also well classified, showing 88.9% accuracy for the single crack case. When examined the confusion matrix for the test dataset, crazing-type cracks are all well sorted, while one multiple cracks image was misclassified to crazing. Other than this case, single and multiple cracks showed good accuracy achieving 84.6% and 90.9% prediction, respectively. Closely examining the misclassifications, some single-type cracks are sorted as multiple cracks with rare opposite cases: This result implies that the proposed algorithm yields conservative results.

Table 1 The number of images used for validation

Crack type	Training		Test		Total
	Field	SDNET2018	Field	SDNET2018	
Single	13	77	7	19	116
Multiple	27	30	3	8	68
Crazing	34	-	9	-	43



(a) Circularity (b) Extent (c) Solidity

Fig. 9 Crack attributes detected for images Fig. 8(a)-(l)

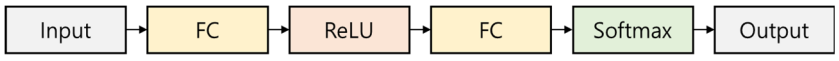
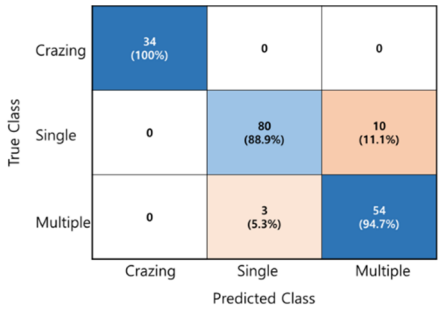
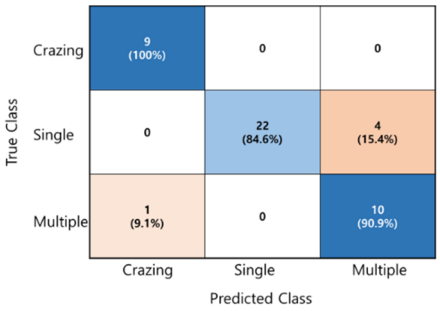


Fig. 10 Neural network structure (FC: Fully connected layer)



(a) Training prediction



(b) Test prediction

Fig. 11 Confusion matrices

Then to further evaluate the performance of the proposed algorithm on datasets, five metrics including accuracy, precision, recall, F1 score, and False Positive Rate (FPR), which all are commonly used in assessing the classification quality are compared. Those indicators are obtained as follows

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{8}$$

$$F1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{9}$$

$$\text{FPR} = \frac{FP}{FP + FN} \tag{10}$$

Table 2 Classification performance for the training set

	Accuracy	Precision	Recall	F1score	FPR
Crazing	1	1	1	1	-
Single	0.89	0.96	0.89	0.92	0.03
Multiple	0.95	0.84	0.95	0.89	0.08

Table 3 Classification performance for the test set

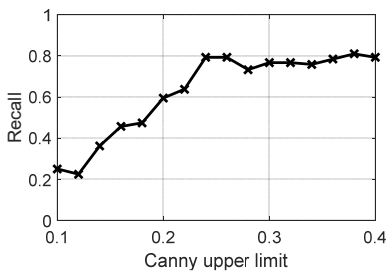
	Accuracy	Precision	Recall	F1score	FPR
Crazing	1	0.9	1	0.95	0.03
Single	0.85	1	0.85	0.92	-
Multiple	0.91	0.71	0.91	0.8	0.11

Note that accuracy implies the ratio of correctly predicted observation, F1 is the weighted average of precision and recall. Results for the training set and test set are summarized in Tables 2 and 3, respectively. In general, high accuracy results are observed for classifying structural crack and crazing-type cracks. In sorting structural cracks, multiple cracks show over 90% accuracy, whereas the single crack showed 84.6-88.8% accuracy. However, FPR for a single crack is fairly low implying that the proposed algorithm is likely to report as a severe crack (containing candidate radial cracks). The overall accuracy of the entire database is 92.8% for the training set and 89.13% for the test set. Although the accuracy for the test set is less than 90%, such a result may be due to a limited number of images in the test set.

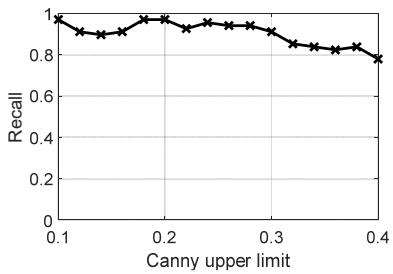
5. Sensitivity analysis and model comparison

In the proposed algorithm, the accuracy is affected by the quality of the edge detection results from the Canny filter. Thus, in this section, sensitivity analysis to determine the filter parameters in the Canny algorithm is presented, followed by an accuracy and processing speed comparison between an algorithm from literature to demonstrate the efficacy of the presented work.

In Canny algorithm, edge detection is controlled by two thresholds: the upper limit and the lower limit. In general, setting the upper threshold too low will cause a noisy result, whereas a high limit will induce spurious and undesirable edge fragments. To demonstrate such an effect and to validate the determined threshold, the results are compared by varying the upper limit from 0.1 to 0.4. Note that the



(a) Single crack



(b) Multiple cracks

Fig. 12 Recall value with different Canny

lower limit also varied, to have 40% of the upper limit, i.e., from 0.04 to 0.01. Fig. 12 plots the recall for classifying single and multiple cracks images; In total 116 single crack images are used and 68 multiple cracks images are evaluated. FNs in single crack classifications are calculated if any radial zone is found. For multiple cracks zones, FNs are counted if no radial zone is detected in the image. As can be seen, single cracks show better results with higher upper limits, implying that clearer edges are obtained. On the other hand, the multiple cracks show the opposite results due to spurious edges in the higher threshold. To achieve sufficient accuracy for both types of cracks, the threshold has been set as 0.25 in this study.

Next, to demonstrate the efficiency of the proposed algorithm, the running time is compared with DISTS developed by (Ding *et al*. 2020). Although DISTS is based on a deep-learning method, the authors in Liu and Yeoh (2021) adopted the algorithm to sort crazing-type crack and structural cracks. The suggested algorithm achieved up to 96% accuracy and classified the structural cracks as single crack and multiple cracks in DISTS.

The calculation was made using a desktop computer with Intel® Core™ i5-6500 CPU @ 3.20 GHz and 16 GB RAM. In DISTS, an input image is resized into 256×256 pixels, and compared with images in a database. When a single image is located in the image base (1-to-1 case), the calculation time took 1.289 sec, and 26.68 sec if the input is compared with 36 images. On the other hand, the proposed algorithm only required 3.734 sec. Thus, by using the physical interpretation of an image, the calculation time has been dramatically reduced.

However, the proposed algorithm is associated with some limitations. For instance, the results of the classification still depend upon the accuracy of edge detection. Surface irregularities or hand-writing on the concrete surface may be recognized as candidate crack in our approach, possibly sorting no crack image into crazing type. To reduce such error, enhancing the edge detection case or incorporating deep-learning-based edge detection must be devised. Although surface irregularities are remained for future studies, due to the scope of the presented study, one of the deep-learning-based edge detection, Holistically-Nested Edge Detection (HED) is adopted in the edge detecting process and discussed in Appendix (Xie and Tu 2015).

6. Conclusions

Computer vision-based concrete crack detection has shown great potentials for autonomous monitoring of a structure with reduced human intervention. To dates, a number of studies have been performed and successfully detected cracks using various image techniques. However, algorithms that can classify the types of cracks for inspection purposes are rare. Thus, the presented research develops an integrated image processing procedure that provides information on crack types, including single crack, multiple cracks, and crazing-type crack. To this end, the algorithm has been proposed by combining processes of edge detection, edge property extraction, edge refinement, and classification. In the edge detection process, the Canny algorithm is applied to detect candidate cracks. Then, various edge properties including circularity, extent, and solidity are extracted as a distinct field for the binarized edge. Edge refinement is followed to remove background noise based on the gradient of the field area. In the categorization, geometrical attributes further including Hough transformation are combined and an NN-based classification is applied to sort crazing-type crack and structural cracks. For the structural cracks, a radial zone is sought to further classify the image into a single crack or multiple cracks case. Once developed for selected images, the proposed algorithm is further validated using a database containing 227 crack images. Based on the experimental findings, the following results can be drawn: The algorithm showed high accuracy in identifying crazing-type images.

The overall accuracy of the proposed approach showed 92.8% for training and 89.1% for test datasets. Within the misclassification cases for structural crack images, single cracks are more recognized as multiple cracks, whereas the opposite case is rare. When closely examined the detected radial zones in multiple cracks images, high recall values are obtained implying that the overall process yields conservative results. Thus, in summary, the presented processes showed agreeable accuracy for providing abstracted information of a crack image.

Acknowledgments

This work was supported by the Korea Agency for Infrastructure Technology Advancement (KAIA) grant funded by the Ministry of Land Infrastructure and Transport (Grant 22CTAP-C164093-02). The authors appreciate the supports.

References

Abdel-Qader, I., Abudayyeh, O. and Kelly, M.E. (2003), "Analysis of edge-detection techniques for crack identification in bridges", *J. Comput. Civil Eng.*, **17**(4), 255-263. https://doi.org/10.1061/~ASCE!0887-3801~2003!17:4~255!

Adhikari, R., Moselhi, O. and Bagchi, A. (2014), "Image-based retrieval of concrete crack properties for bridge inspection", *Automat. Constr.*, **39**, 180-194. https://doi.org/10.1016/j.autcon.2013.06.011

Ahmad, A.R., Osman, M.K., Ahmad, K.A., Anuar, M.A. and Yusof, N.A.M. (2020), "Image segmentation for pavement crack detection system", *Proceedings of the 10th IEEE International Conference on Control System, Computing and Engineering* (*ICCSCE*), Penang, Malaysia, August, pp. 153-157. https://doi.org/10.1109/ICCSCE50387.2020.9204935

Biggs, D.S. and Andrews, M. (1997), "Acceleration of iterative image restoration algorithms", *Appl. Optics*, **36**(8), 1766-1775. https://doi.org/10.1364/AO.36.001766

Canny, J. (1986), "A computational approach to edge detection", *IEEE Transact. Pattern Anal. Mach. Intell.*, (6), 679-698. https://doi.org/10.1109/TPAMI.1986.4767851

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

Chang, P.C., Flatau, A. and Liu, S.C. (2003), "Health monitoring

of civil infrastructure", *Struct. Health Monitor.*, **2**(3), 257-267. https://doi.org/10.1177/147592170303616

Deng, G. and Cahill, L.W. (1993), "An adaptive Gaussian filter for noise reduction and edge detection", *Proceedings of IEEE Conference Record Nuclear Science Symposium and Medical Imaging Conference*, San Francisco, CA, USA, October-November, pp. 1615-1619. https://doi.org/10.1109/NSSMIC.1993.373563

Ding, K., Ma, K., Wang, S. and Simoncelli, E.P. (2020), "Image quality assessment: Unifying structure and texture similarity", *IEEE Transact. Pattern Anal. Mach. Intell.*, **44**(5), 2567-2581. https://doi.org/10.1109/TPAMI.2020.3045810

Flah, M., Suleiman, A.R. and Nehdi, M.L. (2020), "Classification and quantification of cracks in concrete structures using deep learning image-based techniques", *Cement Concrete Compos.*, **114**, 103781. https://doi.org/10.1016/j.cemconcomp.2020.103781

Han, L., Tian, Y. and Qi, Q. (2020), "Research on edge detection algorithm based on improved sobel operator", *Proceedings of 2019 International Conference on Computer Science Communication and Network Security* (*CSCNS2019*), Vol. 309, p. 03031. https://doi.org/10.1051/matecconf/202030903031

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

Hough, P.V. (1962), Method and means for recognizing complex patterns; Google Patents.

Hsieh, Y.A. and Tsai, Y.J. (2020), "Machine learning for crack detection: Review and model performance comparison", *J. Comput. Civil Eng.*, **34**(5), 04020038. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000918

Jahanshahi, M.R., Masri, S.F. and Sukhatme, G.S. (2011), "Multi-image stitching and scene reconstruction for evaluating defect evolution in structures", *Struct. Health Monitor.*, **10**(6), 643-657. https://doi.org/10.1177/1475921710395809

Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S. and Darrell, T. (2014), "Caffe: Convolutional architecture for fast feature embedding", *Proceedings of the 22nd ACM International Conference on Multimedia*, Orlando, FL, USA, November, pp. 675-678. https://doi.org/10.1145/2647868.2654889

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

Kalchbrenner, N., Grefenstette, E. and Blunsom, P. (2014), "A convolutional neural network for modelling sentences", arXiv preprint arXiv:1404.2188. https://doi.org/10.48550/arXiv.1404.2188

Krizhevsky, A., Sutskever, I. and Hinton, G.E. (2012), "Imagenet classification with deep convolutional neural networks", *Commun. ACM*, **60**(6), 84-90. https://doi.org/10.1145/3065386

Kröner, S. and Carbó, M.T.D. (2013), "Determination of minimum pixel resolution for shape analysis: Proposal of a new data validation method for computerized images", *Powder Techonol.*, **245**, 297-313. https://doi.org/10.1016/j.powtec.2013.04.048

LeCun, Y., Bengio, Y. and Hinton, G (2015), "Deep learning", *nature*, **521**(7553), 436-444. https://doi.org/10.1038/nature14539

Lee, C.Y., Xie, S., Gallagher, P., Zhang, Z. and Tu, Z (2015), "Deeply-supervised nets", *Artif. Intell. Statist.*, 562-570.

Lee, J.H., Yoon, S., Kim, B., Gwon, G.H., Kim, I.H. and Jung, H.J. (2021), "A new image-quality evaluating and enhancing methodology for bridge inspection using an unmanned aerial vehicle", *Smart Struct. Syst., Int. J.*, **27**(2), 209-226. https://doi.org/10.12989/sss.2021.27.2.209

Li, Z., Huang, M., Ji, P., Zhu, H. and Zhang, Q. (2022), "One-step deep learning-based method for pixel-level detection of fine cracks in steel girder images", *Smart Struct. Syst., Int. J.*, **29**(1), 153-166. https://doi.org/10.12989/sss.2022.29.1.153

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

Liu, Z.Y., Xie, C.I., Li, J.J., Sang, Y. and Wu, S. (2020), "Canny Edge Detection Algorithm Based on Improved Sequential Statistical Filter", *Proceedings of the 39th Chinese Control Conference* (*CCC*), Shenyang, China, July, pp. 3245-3251.

Long, J., Shelhamer, E. and Darrell, T. (2015), "Fully convolutional networks for semantic segmentation", *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Boston, MA, USA, June, pp. 3431-3440.

Maguire, M., Dorafshan, S. and Thomas, R.J. (2018), "SDNET2018: A concrete crack image dataset for machine learning applications", Utah State University, UT, USA. https://doi.org/10.15142/T3TD19

Maini, R. and Aggarwal, H (2009), "Study and comparison of various image edge detection techniques", *Int. J. Image Porcess.* (*IJIP*), **3**(1), 1-11.

MATLAB (2021), Version 9.10.0.1649659 (R2021a), The MathWorks Inc., Natick, MA, USA.

Milan, S., Vaclav, H. and Roger, B. (1998), "Image processing, analysis, and machine", Vision 2nd Ed Brooks Cole.

Moore, M., Phares, B.M., Graybeal, B., Rolander, D., Washer, G. and Wiss, J. (2001), "Reliability of visual inspection for highway bridges, volume I", Reliability of visual inspection for highway bridges; Technical Report #FHWA-RD-01-020, Federal Highway Administration, Washington DC, USA.

Narazaki, Y., Hoskere, V., Hoang, T.A., Fujino, Y., Sakurai, A. and Spencer Jr, B.F. (2020), "Vision-based automated bridge component recognition with high-level scene consistency", *Comput.-Aided Civil Infrastr. Eng.*, **35**(5), 465-482. https://doi.org/10.1111/mice.12505

Nishikawa, T., Yoshida, J., Sugiyama, T. and Fujino, Y. (2012), "Concrete crack detection by multiple sequential image filtering", *Comput.-Aided Civil Infrastr. Eng.*, **27**(1), 29-47. https://doi.org/10.1111/j.1467-8667.2011.00716.x

Otsu, N (1979), "A threshold selection method from gray-level histograms", *IEEE Transact. Syst. Man Cybernet.*, **9**(1), 62-66.

Parker, J.R. (1991), "Gray level thresholding in badly illuminated images", *IEEE Transact. Pattern Anal. Mach. Intell.*, **13**(08), 813-819. https://doi.org/10.1109/34.85672

Qi, Y., Yuan, C., Kong, Q., Xiong, B. and Li, P. (2021), "A deep learning-based vision enhancement method for UAV assisted visual inspection of concrete cracks", *Smart Struct. Syst., Int. J.*, **27**(6), 1031-1040. https://doi.org/10.12989/sss.2021.27.6.1031

Qiang, S., Guoying, L., Jingqi, M. and Hongmei, Z. (2016), "An edge-detection method based on adaptive canny algorithm and iterative segmentation threshold", *Proceedings of the 2nd International Conference on Control Science and Systems Engineering* (*ICCSSE*), Singapore, July, pp. 64-67. https://doi.org/10.1109/CCSSE.2016.7784354

Shahrokhinasab, E., Hosseinzadeh, N., Monirabbasi, A. and Torkaman, S (2020), "Performance of image-based crack detection systems in concrete structures", *J. Soft Comput. Civil Eng.*, **4**(1), 127-139.

Sobel, I. and Feldman, G. (1968), "A 3x3 isotropic gradient operator for image processing", a talk at the Stanford Artificial Project in, pp. 271-272.

Souza, D. and Menegalli, F.C. (2011), "Image analysis: Statistical

study of particle size distribution and shape characterization", *Powder Technol*., **214**(1), 57-63. https://doi.org/10.1016/j.powtec.2011.07.035
Spencer Jr, B.F., Hoskere, V. and Narazaki, Y. (2019), "Advances in computer vision-based civil infrastructure inspection and monitoring", *Eng*., **5**(2), 199-222. https://doi.org/10.1016/j.eng.2018.11.030
Talab, A.M.A., Huang, Z., Xi, F. and HaiMing, L. (2016), "Detection crack in image using Otsu method and multiple filtering in image processing techniques", *Optik*, **127**(3), 1030-1033. https://doi.org/10.1016/j.ijleo.2015.09.147
Wang, P. and Huang, H. (2010), "Comparison analysis on present image-based crack detection methods in concrete structures." *Proceedings of the 3rd International Congress on Image and Signal Processing*, Yantai, China, October, Vol. 5, pp. 2530-2533.
Wang, G. and Xiang, J. (2021), "Railway sleeper crack recognition based on edge detection and CNN", *Smart Struct. Syst., Int. J.*, **28**(6), 779-789. https://doi.org/10.12989/sss.2021.28.6.779
Wang, Y., Zhang, J.Y., Liu, J.X., Zhang, Y., Chen, Z.P., Li, C.G., He, K. and Yan, R.B. (2019), "Research on crack detection algorithm of the concrete bridge based on image processing", *Procedia Comput. Sci*., **154**, 610-616. https://doi.org/10.1016/j.procs.2019.06.096
Wang, L., Spencer Jr., B.F., Li, J. and Hu, P. (2021), "A fast image-stitching algorithm for characterization of cracks in large-scale structures", *Smart Struct. Syst., Int. J.*, **27**(4), 593-605. https://doi.org/10.12989/sss.2021.27.4.593
Xie, S. and Tu, Z. (2015), "Holistically-nested edge detection", *Proceedings of the IEEE International Conference on Computer Vision*, Santiago, Chile, December, pp. 1395-1403.
Xuan, L. and Hong, Z. (2017), "An improved canny edge detection algorithm", *Proceedings of the 8th IEEE International Conference on Software Engineering and Service Science* (*ICSESS*), Tianjin, China, August, pp. 275-278.
Ye, X.W., Jin, T. and Yun, C.B. (2019), "A review on deep learning-based structural health monitoring of civil infrastructures", *Smart Struct. Syst., Int. J.*, **24**(5), 567-585. https://doi.org/10.12989/sss.2019.24.5.567
Zheng, Z., Zha, B., Yuan, H., Xuchen, Y., Gao, Y. and Zhang, H. (2020), "Adaptive edge detection algorithm based on improved grey prediction model", *IEEE Access*, **8**, 102165-102176. https://doi.org/10.1109/ACCESS.2020.2999071
Zhu, Z., German, S. and Brilakis, I. (2011), "Visual retrieval of concrete crack properties for automated post-earthquake structural safety evaluation", *Automat. Constr*., **20**(7), 874-883. https://doi.org/10.1016/j.autcon.2011.03.004

HJ

Appendix. Used of Deep-learning algorithm for edge detecting step

Among various CNN-based edge detection algorithms, the HED algorithm is chosen in this appendix due to its simplicity. The HED algorithm uses a pre-trained VGG-16 Net model such that input image features can be learned. Thus, the HED is beneficial over other deep-learning methods in that the scale-space edge fields are autonomously learned and hierarchically connected. The framework implemented in this section is built on top of FCN (Long *et al*. 2015) and DCN (Lee *et al*. 2015), and is currently available from *Caffe* library (Jia *et al*. 2014). Note that mean values of RGB that are subtracted from an image that was used for training the model are 104.007, 116.669, and 122.679, respectively. Fig. A1(a) shows the results of HED for Fig. 2(f), which is then applied to the proposed crack classification algorithm: Because the HED is less susceptible to the size of the input image, the original size and the images are used. As can be seen, the algorithm well identifies the crack shapes. These detected cracks are then converted into binary edges by applying the Canny algorithm (see Fig. A1(b)). The accuracy of this edge detection is highly related to the HED precision, and thus, clear edges are obtained, which slightly outperforms that from the filter-based approach. Then the results from the Categorization process are summarized in Fig. A1(c). As can be seen, the image has been well classified as multiple cracks, with radial zone detected. Note that vaguely detected cracks at the lower part of Fig. 2(f) became thicker when HED is used, resulting in a better performance. Thus, the results of this study reveal that the proposed classification algorithm is flexible that can be incorporated with any crack edge detection algorithm to examine whether the image is crazing-type, single crack, or multiple cracks with severe zones.

(a) Edge detection (b) Binarized edge (c) Classification resultsr

Fig. A1 Deep-learning based classification