



# Skeleton-based noise removal algorithm for binary concrete crack image segmentation<sup>☆</sup>

Hamish Dow<sup>a,\*</sup>, Marcus Perry<sup>a</sup>, Jack McAlorum<sup>a</sup>, Sanjeetha Pennada<sup>a</sup>, Gordon Dobie<sup>b</sup>

<sup>a</sup> Department of Civil and Environmental Engineering, University of Strathclyde, Glasgow, G1 1XJ, UK

<sup>b</sup> Department of Electronic and Electrical Engineering, University of Strathclyde, Glasgow, G1 1XJ, UK

## ARTICLE INFO

### Keywords:

Concrete defect detection  
Automated inspection  
Denoising  
Binary classification  
Image segmentation  
Skeletalisation  
Connected components  
Morphological reconstruction

## ABSTRACT

Image processing methods for automated concrete crack detection are often challenged by binary noise. Noise removal methods decrease the false positive pixels of crack detection results, often at the cost of a reduction in true positives. This paper proposes a novel method for binary noise removal and segmentation of noisy concrete crack images. The method applies an area threshold before reducing the pixel groups in the image to a skeleton. Each skeleton is connected to its nearest neighbour before the remaining short skeletons in the image are removed using a length threshold. A morphological reconstruction follows to remove all elements in the original noisy image that do not intersect with the skeleton. Finally, pixel groups in close proximity to the endpoints of the pixel groups in the resulting image are reinstated. Testing was conducted on a dataset of noisy binary crack images; the proposed method (Skele-Marker) obtained recall, precision, intersection over union, and F1 score results of 77%, 91%, 72%, and 84%, respectively. Skele-Marker was compared to other methods found in literature and was found to outperform other methods in terms of precision, intersection over union and F1 score. The proposed method is used to make crack detection results more reliable, supporting the ever-growing demand for automated inspections of concrete structures.

## 1. Introduction

Traditional human visual inspections of structures are unsafe and produce inconsistent results [1,2]. To improve safety and efficiency, efforts have been made to automate both the physical inspection of the structure [3] and the detection and segmentation of any damage [4]. Techniques such as conventional image processing (so called “white-box” techniques) and less transparent artificial neural network approaches (black-box techniques) are replacing human analysis effort [5]. White-box techniques have the advantage of being computationally inexpensive, traceable, transparent and they do not require a large dataset for training. Despite the success of black-box techniques in concrete crack detection and classification, there is still a place for white-box techniques and their development. Hybrid or “grey-box” approaches that utilise both techniques can perform better at pixel-level segmentation tasks [6].

Various white-box methods can be used to segment an image of a crack into a binary cracked and uncracked pixel image. Algorithms broadly fall into the categories of edge-based (e.g. Canny edge detector or Wavelet transform), and threshold-based (e.g. Otsu thresholding) [7]. Despite their effectiveness, these methods also detect many

false positives in the form of noise; this reduces the accuracy of crack detection results.

In this work, we demonstrate a novel white-box approach dubbed “Skele-Marker”, for enhancing noise removal and segmentation. This improved method can increase the accuracy and positive prediction rate of crack detection, and could allow for the use of images that would have otherwise been disregarded for having excessive binary noise. Following a brief review of noise removal methods in Section 2, the Skele-Marker method is outlined in Section 3. We subsequently curate a concrete data set of over 2,000 images, and use this to: (i) optimise Skele-Marker’s parameters; and (ii) benchmark its performance against other noise removal techniques from the literature (see Sections 4 and 5). To our knowledge, this is the first direct comparison of binary noise removal methods in concrete crack images.

## 2. Review of noise removal methods

Table 1 summarises the common causes of noise in binarised concrete crack images. Other surface defects such as markings, spalling and delamination are considered as noise as they are false positives in terms

<sup>☆</sup> This work was funded by the Scottish Funding Council, UK, and the University of Strathclyde, UK’s Advanced Nuclear Research Centre.

\* Corresponding author.

E-mail address: [hamish.dow@strath.ac.uk](mailto:hamish.dow@strath.ac.uk) (H. Dow).

**Table 1**  
Factors that produce noise in binary crack images.

Noise type	Example causes
Environmental	<ul style="list-style-type: none"> <li>• Non-uniform and dark lighting conditions.</li> <li>• Rough and uneven concrete surface.</li> <li>• Air voids on concrete surface.</li> <li>• Other surface defects (e.g. markings, spalling and delamination).</li> </ul>
Hardware	<ul style="list-style-type: none"> <li>• High sensor sensitivity to light, commonly referred to as GAIN or ISO (International Standards Organization).</li> <li>• Non-uniform lighting (e.g. camera flash).</li> </ul>
Post-processing	<ul style="list-style-type: none"> <li>• Image contrast enhancement.</li> <li>• Image compression (e.g. JPEG, PNG).</li> </ul>

of crack detection. Other detection methods (white-box or black-box) should be employed to detect these defect types.

Noise in binary images can be minimised with modifications to the hardware and post-processing techniques used in image acquisition. However, remaining noise and noise resulting from the environment relies on algorithms for removal.

White-box algorithms for noise removal methods typically employ:

- blurring;
- morphological operations;
- connected component analysis;
- gravitational attraction;
- area thresholding; and
- gap connection.

These methods and their drawbacks are briefly reviewed in the following subsections.

### 2.1. Blurring

The typical approach to noise removal in white-box techniques is applying a global Gaussian or median blur to a grayscale image [8]; often prior to a Laplacian filter, which smooths out the pixels that would result in noise before any segmentation occurs. However, these techniques are destructive to edges in images, which could distort or even remove cracks in our application.

### 2.2. Morphological operations

Opening morphological operations are also commonly used [9–11]. This erodes the pixel groups to remove noise before dilating to return remaining pixel groups (cracks) to their original size. Similar to blurring, this method heavily distorts the crack.

### 2.3. Connected component analysis

Another strategy for binary noise removal is to apply a connected components size threshold, where groups of positive pixels smaller than the fixed threshold size are removed. Authors in Dorafshan et al. [12] proposed a method where the area threshold value adapts to the image; in this approach, the area threshold value is taken as the standard deviation of all areas of connected components in the image. While the area threshold method is effective at removing small noise-pixel groups, there are significant drawbacks: (i) large noise-pixel groups remain in the image; and (ii) small pixel groups that are true positives are removed.

### 2.4. Gravitational attraction

Sorncharean and Phiphobmongkol [13], proposed a unique technique based on Newton's law of gravitational attraction. The

gravitational attraction of each positive-pixel group to every other positive-pixel group is found. If a positive-pixel group has high variability in gravitational attraction, it is assumed to be a crack; with low variability, it is defined as noise. While novel, the performance of this method is unclear; testing was only conducted on two images, and other researchers have not yet adopted the technique.

### 2.5. Area thresholding and gap connection

Another technique is to apply a large area threshold so that all noise and smaller elements of a crack are removed. A "gap connection" algorithm is then used to reinstate the removed crack pixels. The method for connecting gaps varies across crack detection literature. Authors in Huang and Zhang [14] developed a method that connects gaps depending on the geometry of the two lines to be connected. Authors in Song et al. [15] proposed a method that creates two images with different length thresholding values to connect gaps in the crack. In Song et al. [15], stated performance is good; however, testing was conducted on a very small dataset, and the methodology has gaps that make its reproduction challenging. Yet another technique, called "Region-growing crack connection" [16], follows a similar geometry-based approach but there is no evaluation of the method's performance.

In general, gap connection methods are effective, but suffer from many limitations.

- Elements may be incorrectly connected on cracks with branches or other complex geometry.
- The gap connection pixels do not consider crack width.
- The gap connections will not follow the crack line if the initial noise removal is too strong. If the noise removal is too weak, the gap connections will incorrectly connect many elements together.
- A gap connection will not replace small pixels at the end of the crack line.

### 2.6. Overview of noise removal methods for crack detection

Acceptable errors and tolerances are highly application dependent. For example, in the detection of large road cracks, the errors resulting from blurring, area thresholding and gap connection are acceptable, as cracks below these tolerances are inconsequential. However, for concrete structures, such as water-retaining structures which require a crack detection limit of 0.1 mm, it is critical that reducing noise pixels does not reduce the image's true positive pixels.

## 3. The Skele-Marker method

Unlike other noise removal approaches, the Skele-Marker algorithm described in this paper does not:

- remove small groups of pixels on the crack line that could easily be mistaken for noise;
- connect components together that were unconnected in the imported noisy image.

The Skele-Marker algorithm is summarised in Fig. 1 and described in detail within the following subsections. Example image outputs at each step of the process are illustrated in Table 2.

### 3.1. Image pre-processing

For this study, the noisy input images were obtained using contrast enhanced crack images and a Laplacian filter, as described in

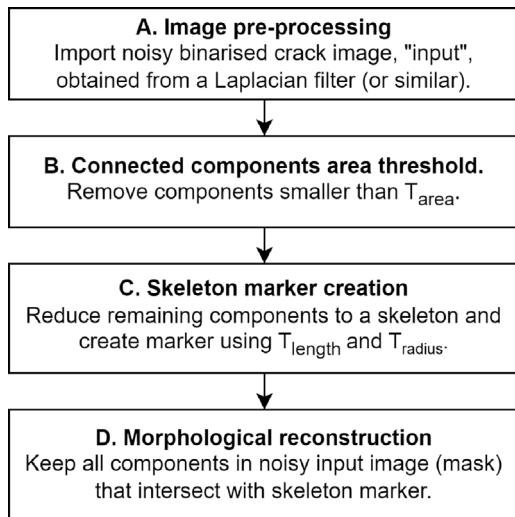


Fig. 1. Flowchart of Skele-Marker algorithm.

Section 4.1. However, the method will work with any binary crack image.

The input image is loaded in uint8 format with pixels  $P(x,y)$  of values 0 and 255. A pixel intensity of 0 (black) indicates uncracked pixels  $U$ , where the binary value,  $V$  is 0. A pixel intensity of 255 (white) indicates cracked pixels  $C$ , where the binary value,  $V$ , is 1 i.e.:

$$U \in P(x, y), \quad \text{where } P(x, y) = 0, \quad (1)$$

$$C \in P(x, y), \quad \text{where } P(x, y) = 255, \quad (2)$$

$$V = \frac{P(x, y)}{255}. \quad (3)$$

It follows that the number of pixels in the image is:

$$\text{count}(C) + \text{count}(U) = \text{count}(P(x, y)). \quad (4)$$

### 3.2. Connected components area threshold

Connected components analysis using eight-neighbour connectivity yields  $N$  connected groups of pixels,  $G_i = G_1, \dots, G_N$ , each with area  $A_i = A_1, \dots, A_N$ . Pixel groups with an area less than an area threshold  $T_{\text{area}}$  are defined as “uncracked regions”, and all pixels in that group are assigned a binary value of 0, i.e.:

$$V = 1 \text{ for all } P(x, y) \in G_i \quad \text{if } A_i > T_{\text{area}}, \quad (5)$$

$$V = 0 \text{ for all } P(x, y) \in G_i \quad \text{if } A_i < T_{\text{area}}. \quad (6)$$

### 3.3. Skeleton marker creation

The remaining connected components are reduced to 1 pixel thin using the Zhang–Suen thinning algorithm. This is an iterative process that creates a skeleton of the crack by removing pixels with fewer than eight neighbours [17].

Similar to Huang and Zhang [14], to account for small breaks in the crack, the endpoints of every connected component skeleton are detected, and each endpoint is connected to its corresponding nearest endpoint. An endpoint will not be connected if no other endpoints are available or an endpoint lies at the edge of an image. It can now be assumed that the larger skeletons in the image are crack lines and smaller skeletons are the result of noise. A length threshold of  $T_{\text{length}}$  pixels is applied, with skeletons shorter than this threshold erased from the image to remove the smaller lines.

Table 2  
Image processing steps of the Skele-Marker method.

Step	Image
Import noisy binary image.	
Apply connected components area threshold $T_{\text{area}}$ .	
Create skeleton and connect each point to its corresponding nearest end point.	
Apply skeleton length threshold $T_{\text{length}}$ and reinstate all pixels within radius $T_{\text{radius}}$ of all end points.	
Use a morphological reconstruction to keep all connected components from the original image that intersect with marker image.	

The endpoints of any remaining skeleton lines are found, and all pixels within a radius,  $T_{\text{radius}}$ , of these endpoints are reinstated back into the marker image; this ensures that pixel groups close to the endpoint of cracks are retained during the morphological reconstruction.

### 3.4. Morphological reconstruction

A morphological reconstruction is implemented to keep all pixel groups in the original imported image that intersect with the skeleton derived in Section 3.3; this process is illustrated in Fig. 2. MATLAB's `imreconstruct` function is used with a radius of 1, to create a 3 by 3 structuring kernel. The function requires two images as an input, "mask", the original noisy image, and "marker", the skeleton image. The kernel is used to iteratively expand the marker image by performing a morphological dilation. After each iteration, the expanded image is combined with the mask using a bitwise AND operation. This operation ensures that the expanded image only includes pixels that are also present in the mask image. The function continues to iterate until the expanded image no longer changes. Once this condition is met, the final iteration image is returned.

### 4. Parameter optimisation and benchmarking method

Finding the optimum parameters of  $T_{\text{area}}$ ,  $T_{\text{length}}$ ,  $T_{\text{radius}}$  for the Skele-Marker method and benchmarking this method against other white-box techniques requires:

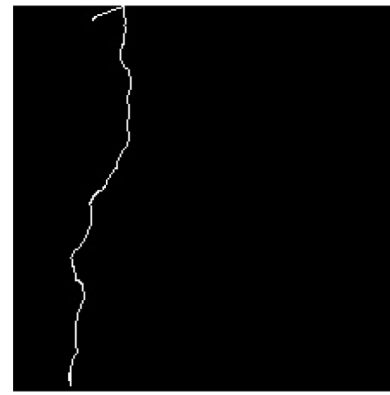
- acquisition and preparation of a dataset;
- ground truth definition of cracks at pixel-level;
- labelling of images as cracked or uncracked; and
- definition of sensitivity analyses and performance metrics.

#### 4.1. Image acquisition and crack detection

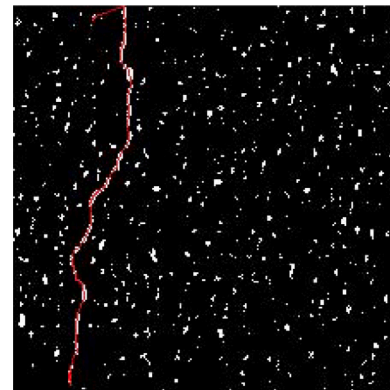
The dataset consists of ten images captured from two large damaged concrete slab samples, each with a total surface area of  $1 \text{ m}^2$ . As shown in Fig. 3a to increase noise, the concrete surfaces have varying textures, colours and imperfections. The widths of the cracks on these samples were measured manually using a magnified ruler; crack width measurements ranged from 0.07 mm to 0.3 mm. Images of the slabs were captured using a 1" sensor FLIR BFS-U3-200S6C-C machine vision camera. The camera's SONY IMX183 sensor was paired with an 8 mm focal length lens. To ensure sharpness in the image, the lens aperture was set to  $f8$ , and all images were captured with a lens-surface distance of 250 mm. This provides a camera field of view of approximately  $402 \times 269 \text{ mm}$ . Distortion corrections are applied to the camera using intrinsic and extrinsic parameters found through a checkerboard calibration; these corrections ensure that the captured images have minimal distortion and measurements are true to real-world measurements. After distortion corrections, the captured images have a resolution of  $5429 \times 3458 \text{ pixels}$ .

Due to the circle of confusion, lenses perform better at the centre of an image. In the captured dataset, blurring was present at the left and right edges of the image. As such, the image 3:2 aspect ratio was cropped to an aspect ratio of 1:1 to maintain sharpness. To encourage noise, image contrast was increased. The images were then converted to grayscale and convolved with a  $3 \times 3$  Laplacian kernel to extract the edges. To allow noise (and detail) to pass through into the binary image, no Gaussian filter was applied prior to the Laplacian kernel convolution. To convert the Laplacian image to binary, a global threshold  $T_1$  was applied, where  $T_1$  was defined using the pixel intensities of the Laplacian image. The equation used by Dorafshan et al. [12] was chosen, where  $T_1$  is found using the average of the pixel intensities,  $\mu_E$ , and the standard deviation of the pixel intensities  $\sigma_E$  (see Eq. (7)).

$$T_1 = \mu_E + 3\sigma_E \quad (7)$$



(a)



(b)

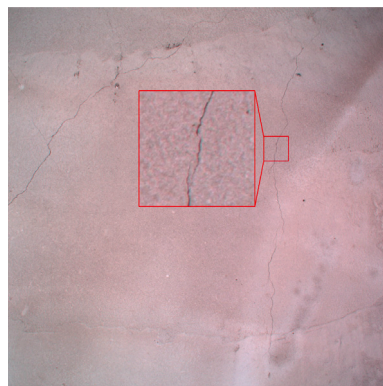


(c)

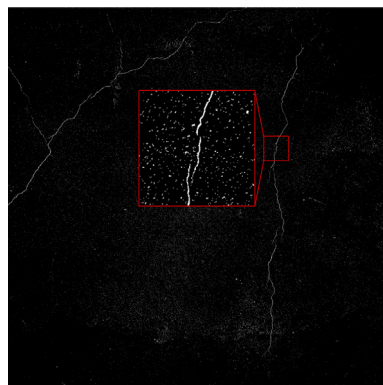
Fig. 2. Morphological reconstruction. (a) marker, (b) overlay of marker on original noisy image (mask), (c) output of morphological reconstruction.

#### 4.2. Ground truth definition

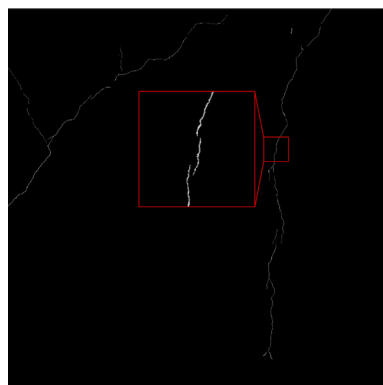
Cracks from the original images were manually identified and digitally traced by hand. The tracing outline was then masked with a white layer and placed onto a black background resulting in a ground truth; this process was then repeated for each dataset image. The user-defined ground truths will not be 100% accurate, but as tracing is conducted by hand on a pixel-by-pixel level, it is a robust estimate of the true crack size and location. Fig. 3 shows the captured image, Laplacian output and ground truth for one of the dataset images.



(a)



(b)



(c)

Fig. 3. Dataset image. (a) original image, (b) Laplacian output of image, (c) ground truth of cracked pixels. The red box indicates a  $224 \times 224$  pixels block used for image segmentation in Section 4.3.

#### 4.3. Image labelling

The Laplacian and ground truth images were split up into blocks with resolutions of  $224 \times 224$  pixels. The ground truth blocks were then manually labelled as cracked and uncracked; these labels were then applied to the corresponding Laplacian image blocks. The final dataset is 2,083 images, each of size  $224 \times 224$  pixels. 412 of these images show cracked faces, and 1,671 showed uncracked faces.

This dataset was split into training and testing at a ratio of 17%–83%. The training dataset is used to explore optimal parameters values for the Skele-Marker method during sensitivity study, outlined in Section 4.5. The testing dataset is used to benchmark the Skele-Marker method against other noise removal methods as described

Table 3

Performance metrics for a classifier.

Name	Description	Equation
True positive rate (recall) (TPR)	The estimated probability that an actual positive will test positive.	$\frac{TP}{TP + FN}$
True negative rate (TNR)	The estimated probability that an actual negative will test negative.	$\frac{TN}{TN + FP}$
Accuracy (ACC)	The fraction of classifications, both positive and negative, that were true.	$\frac{TP + TN}{TP + FP + TN + FN}$
Positive predictive value (precision) (PPV)	The estimated probability that a positive prediction is a true positive.	$\frac{TP}{TP + FP}$
Negative predictive value (NPV)	The estimated probability that a negative prediction is a true negative.	$\frac{TN}{TN + FN}$
Intersection over Union (IoU)	The ratio of the intersected area of predictions and ground truth to the combined area of predictions and ground truth.	$\frac{TP}{TP + FP + FN}$
F1 score (F1)	The weighted average of recall and precision.	$\frac{2TP}{2TP + FN + FP}$

in Section 4.6. To break any possible correlations between datasets, the training and testing datasets were formed from different concrete surfaces.

#### 4.4. Performance metrics

When an image is compared to its respective ground truth, the true positives (TP), false negatives (FN), false positives (FP), and true negatives (TN) are found. These values allow the performance metrics described in Table 3 to be calculated. All metrics range from 0 to 1, with higher values indicating better performance.

The dataset's class imbalance is the result of considerably more uncracked pixels,  $U$ , than cracked pixels,  $C$ ; this is expected as thin cracks only account for a small area of the image. Regardless of crack detection performance, all methods will have a high number of TN. As such, the metrics of TNR, ACC, and NPV, which are all highly sensitive to TN, will not show significant differences between the different methods. However, TPR, PPV and IoU are not dependent on TN values; they use TP values which makes them a more appropriate crack detection metric, as results will be low if detection performance is low.

A high recall (TPR) indicates that an algorithm does not miss many cracks, a high precision (PPV) indicates that the algorithm does not classify many  $U$  pixels as  $C$ . Both metrics are sensitive to the number of TP pixels in an image, which is affected by crack size and shape. IoU will be higher when  $C$  pixels are correctly classified as being within the ground truth.

#### 4.5. Sensitivity analysis of Skele-Marker method

A sensitivity analysis was conducted to obtain the optimal parameters for the Skele-Marker method's variables:  $T_{\text{area}}$ ,  $T_{\text{length}}$  and  $T_{\text{radius}}$ . For each cracked  $224 \times 224$  block of the training dataset, the Skele-Marker method was repeatedly applied with varying parameters. Each output image was then compared to the corresponding ground truth to allow TP, FN, FP and TN to be found.

After an initial trial and error approach, the parameter variations shown in the testing matrix (Table 4) were selected.  $T_{\text{area}}$  and  $T_{\text{radius}}$

**Table 4**  
Parameter variations for Skele-Marker method.

	$T_{\text{area}}$	$T_{\text{length}}$	$T_{\text{radius}}$
Var1	4	25	4
Var2	6	30	6
Var3	8	35	8
Var4	10	40	10
Var5	12	45	12
Var6	14	50	14
Var7		55	16
Var8		60	18
Var9		65	20
Var10			22
Var11			24

have a small step size of 2 pixels as the results were found to be more sensitive to these parameters.

For each parameter variation, the TP, FN, FP and FN results were aggregated over all training examples. These aggregations were then used to calculate the precision, recall and F1 score for each parameter variation across the whole training dataset.

#### 4.6. Benchmarking of Skele-Marker method

The Skele-Marker method, using parameter values for  $T_{\text{area}}$ ,  $T_{\text{length}}$  and  $T_{\text{radius}}$  found in the sensitivity analysis, was compared against:

- Dorafshan et al. connected components size threshold (Dorafshan) [12].
- Sorncharean & Phiphobmongkol gravitational attraction (Sorncharean) [13].
- Huang & Zhang gap connection (Huang) [14].

Similar to the sensitivity analysis in Section 4.5, the results of each method were then compared to their corresponding ground truth, allowing the performance metrics in Table 3 to be calculated for each noise removal method.

For uncracked images, ground truths are set to  $V = 0$  for all pixels, meaning there are no true positives or false negatives. As such, only true negatives and false positives were calculated for these images.

## 5. Results and discussion

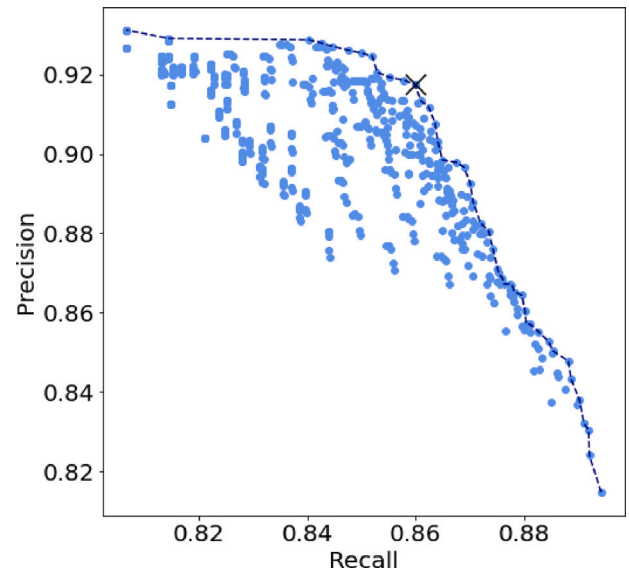
### 5.1. Sensitivity analysis

Fig. 4 shows the precision recall curve for the training dataset plotted with a Pareto front; this illustrates the extreme values in the results with desirable high precision and recall.

The parameter variation with the highest F1 score (89%), shown as a cross in Fig. 4, falls on the Pareto front of the precision recall curve. This parameter variation has precision and recall values of 92% and 86%, respectively. For the purposes of this study, the parameter variation with the highest F1 score was chosen. If higher recall was preferred, a value with greater recall and lower precision that lies on the Pareto front could be selected.

Fig. 5 shows a typical data series of the general trend between F1 score and each parameter of the Skele-Marker method.

$T_{\text{radius}}$  and  $T_{\text{length}}$  show a quadratic dependence on F1, and  $T_{\text{area}}$  shows a fourth-order polynomial dependence. As such, multivariate non-linear regression can be used to find a relationship between F1 and the parameters  $T_{\text{length}}$ ,  $T_{\text{radius}}$  and  $T_{\text{area}}$ . The full dataset and its non-linear fits are shown in Fig. 6 for selected parameter values. The data points in Fig. 6a were found by iteratively adjusting  $T_{\text{length}}$  and  $T_{\text{radius}}$ , with  $T_{\text{area}}$  fixed at 6 pixels. The data points in Fig. 6b were found by iteratively adjusting  $T_{\text{length}}$  and  $T_{\text{area}}$ , with  $T_{\text{radius}}$  fixed at 8 pixels.



**Fig. 4.** Precision recall curve with Pareto front shown as a dashed line. The parameter variation with the highest F1 score is shown as a cross.

**Table 5**  
Eq. (8) coefficients and respective 95% confidence intervals.

Coefficient	Value	Lower bound	Upper bound
$a_1$	8.0E-4	5.2E-4	10.1E-4
$a_2$	-5.8E-5	-7.3E-5	-4.3E-5
$b_1$	1.6E-3	1.5E-3	1.7E-3
$b_2$	-1.5E-5	-1.6E-5	-1.4E-5
$c_1$	1.4E-2	1.2E-2	1.5E-2
$c_2$	-1.5E-3	-1.7E-3	-1.3E-3
$c_3$	6.4E-5	5.6E-5	7.3E-5
$c_4$	-9.3E-7	-10.9E-7	-7.7E-7
D	0.795	0.791	0.800

The surface fits were found using multivariate non-linear regression on the data points. Eq. (8) can be used to represent how the parameters  $T_{\text{length}}$ ,  $T_{\text{radius}}$  and  $T_{\text{area}}$  affect F1 score.

$$F1 = a_1 T_{\text{radius}} + a_2 T_{\text{radius}}^2 + b_1 T_{\text{length}} + b_2 T_{\text{length}}^2 + c_1 T_{\text{area}} + c_2 T_{\text{area}}^2 + c_3 T_{\text{area}}^3 + c_4 T_{\text{area}}^4 + D \quad (8)$$

Using multivariate non-linear regression, the coefficients in Eq. (8) were populated using the Levenberg–Marquardt fitting method. Eq. (9) shows the residual equation chosen to be minimised. The resulting coefficients and their respective 95% confidence intervals are described in Table 5.

$$obj = (F1 - F1_m)^2 \quad (9)$$

The populated equation was then used with a range of  $T_{\text{length}}$ ,  $T_{\text{radius}}$  and  $T_{\text{area}}$  to generate F1 score values for the surface plots in Fig. 6.

The non-linear multivariate regression results in Fig. 6 show how the F1 score of the Skele-Marker algorithm varies with its input parameters. The algorithm is very sensitive to  $T_{\text{area}}$ , where too small or too large a value will have a detrimental effect on results. These results were used to find that the maximum achievable F1 score occurs at a value of  $T_{\text{length}} = 55$  pixels,  $T_{\text{radius}} = 8$  pixels and  $T_{\text{area}} = 6$  pixels. As such, these values have been selected for the method benchmarking.

### 5.2. Noise removal method comparison

Fig. 7 shows the performance of all noise removal methods on the testing dataset. An example output for the same binarised input image

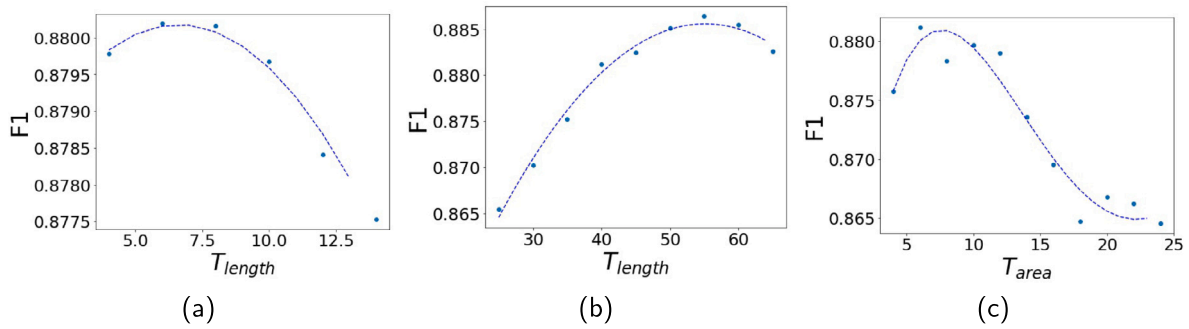


Fig. 5. Typical results of F1 score vs (a)  $T_{radius}$  (b)  $T_{length}$  and (c)  $T_{area}$ . The dashed line represents the curve fit for the current data series.

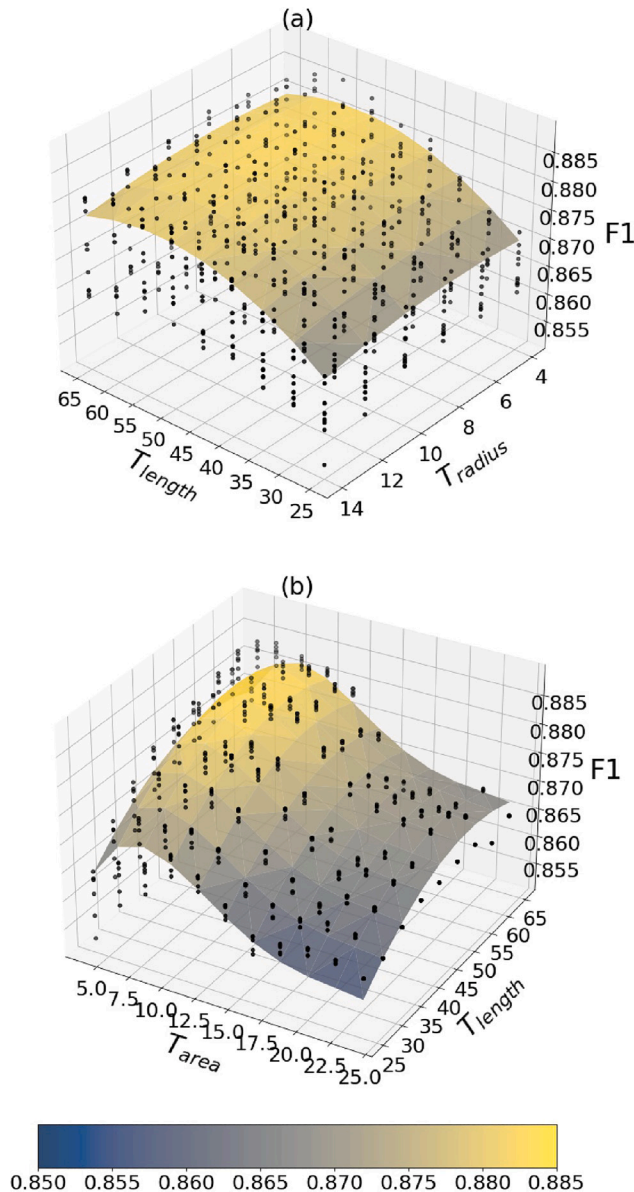


Fig. 6. Surface plots of  $T_{length}$  vs (a)  $T_{radius}$ , with  $T_{area}$  fixed at 6 pixels and vs (b)  $T_{area}$ , with  $T_{radius}$  fixed at 8 pixels.

is shown for each method in Fig. 8. The results for TPR, PPV, IoU and F1 vary greatly between the noise removal methods showing that there are performance differences between methods.

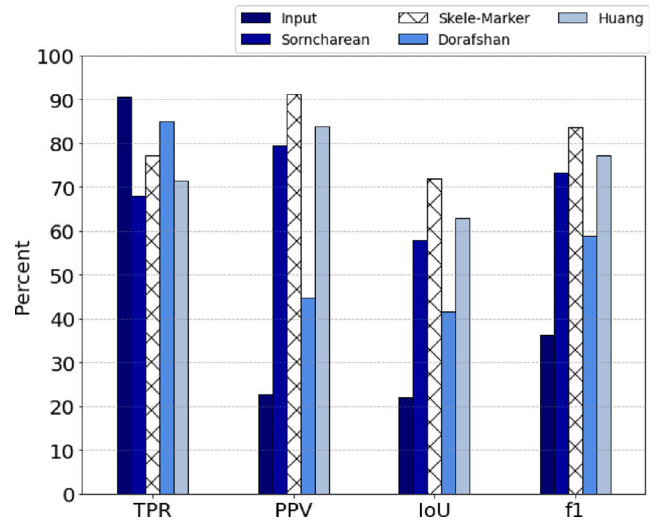


Fig. 7. Performance metrics of all noise removal methods across the testing dataset. Abbreviations are described in Table 3.

The recall (TPR) of the noisy Laplacian image (labelled “input” in 7) is 90%. It is not 100% because the application of the Laplacian filter sacrifices detail to acquire a binarised image. This value of 90% is therefore the maximum TPR that the noise removal methods Dorafshan, Sorncharean and Skele-Marker can achieve as they remove noise from the images. As the Huang method adds pixels into the image, it could technically produce a higher TPR than the noisy Laplacian image but it has not done so. The Dorafshan noise removal method obtained the highest TPR at 85%, with the Skele-Marker method achieving 77%. Due to a high number of false positives created by the noise, the precision (PPV) of the input binary image was extremely low at 22%. As such, all noise removal methods increased precision as the number of false positives was reduced. The same observation can be made for IoU, where Skele-Marker obtained the highest value at 72%. The Dorafshan method had the lowest precision of all noise removal methods at 45%. The Skele-Marker method had the greatest precision and F1 score at 91% and 84%, respectively. The effects of the high precision of the Skele-Marker method can be seen in Fig. 8, with minimal false positives in comparison to other methods.

The outputs of the Skele-Marker algorithm when tested on various noisy concrete crack images are shown in Appendix. These figures highlight the algorithm’s ability to de-noise images with varying levels of noise severity while maintaining TP pixels.

The false positives of each noise removal method on the uncracked dataset were calculated as a percentage of all false positives in the input binary image prior to noise removal. Subtracting these values from 100 yielded the percentage of noise removed. Fig. 9 shows these results.

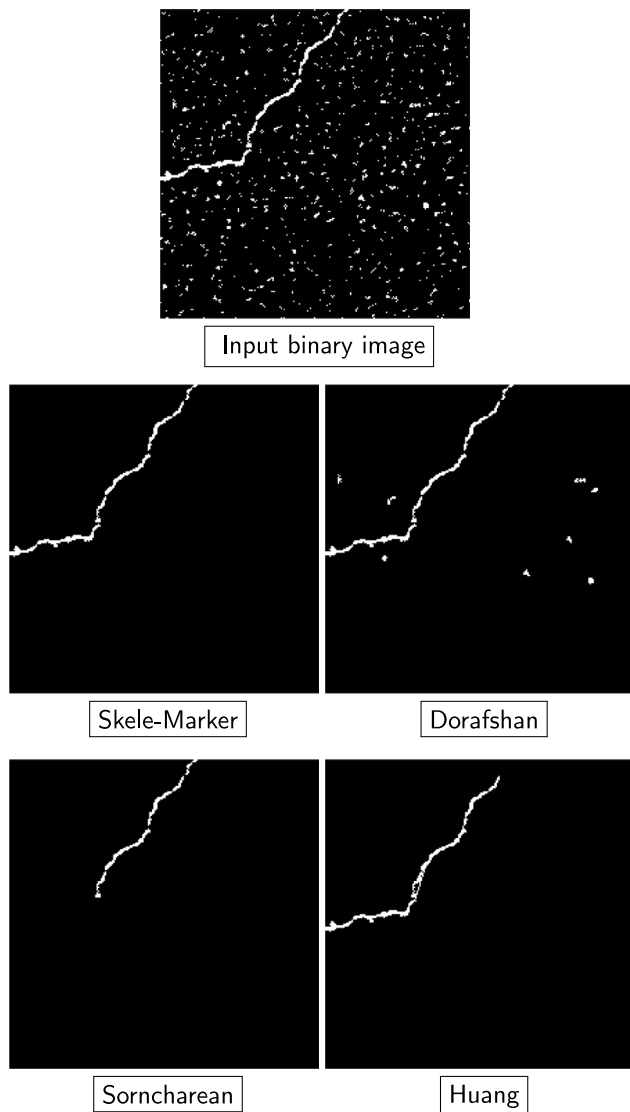


Fig. 8. Output of noise removal methods on a 224 × 224 dataset image.

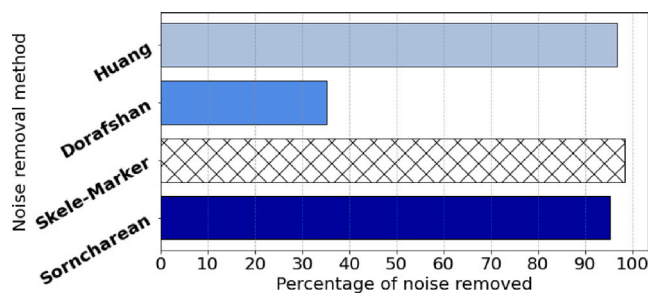


Fig. 9. False positives remaining after noise removal on negative crack dataset. Results are shown as a percentage of false positives in the input binary image.

The trend of noise removal performance on uncracked datasets matches the findings from cracked datasets. Fig. 9 shows that the Skele-Marker method removed 98.5% of the noise in the uncracked testing dataset. In comparison, the Dorafshan method removed 35.3% of noise. The Huang and Sorncharean methods achieved 96.8% and 95.3%, respectively. These results are high compared to Dorafshan but still fall short of the Skele-Marker method.

The Skele-Marker method has outperformed the other noise removal methods on all testing metrics with the exception of recall (TPR) on the cracked testing dataset, where the Dorafshan method scored highest. Dorafshan’s high recall performance is due to the method removing fewer pixels than other methods, with many false positives remaining, as shown by its low precision. The sensitivity analysis results in Section 5.1 highlights that the recall and precision trade-off of the Skele-Marker method can be adjusted by varying the method’s parameters; this allows higher recall (at the cost of precision) to be achieved if preferred.

## 6. Conclusions and future work

This paper proposed a novel noise removal method for binary images of concrete cracks. The method, dubbed Skele-Marker, thins a noisy crack image to a skeleton, before using a morphological reconstruction to remove all elements in the noisy image that do not intersect with the skeleton. Training and testing datasets were formed using concrete slabs with cracks of widths ranging from 0.07–0.3 mm, image contrast was increased to create more noise in the binarised images.

A sensitivity analysis was conducted on the training dataset to refine the method and tune its parameters. It was found that the algorithms parameters  $T_{radius}$ ,  $T_{length}$  and  $T_{area}$  should be set to 8, 55 and 6 pixels, respectively. These parameters were then used to test the method on a separate cracked dataset and compare its performance to three other noise removal methods in crack detection literature. The Skele-Marker method outperformed the other methods, achieving high recall, precision, intersection over union and F1 score. Testing was also conducted on the uncracked dataset, which again, saw the Skele-Marker method outperform other methods.

This method will make results from pixel-level segmentation of concrete cracks more reliable and robust, improving methods for automated inspections of concrete structures.

The constants calculated from the sensitivity analysis are likely only suitable for cracks of widths that match the dataset; future works should investigate how these parameters vary on different crack sizes and how the method could be adapted to other types of concrete defects such as spalling, corrosion and exposed rebar. Further comparison with other noise removal methods could also be made. The Skele-Marker method should be tested with more challenging images that contain additional noise-inducing elements; for example, a dataset of concrete crack images captured from real-world structures and not a laboratory setting.

### CRedit authorship contribution statement

**Hamish Dow:** Conceptualization, Methodology, Software. **Jack McAlorum:** Data curation, Writing – original draft.

### Declaration of competing interest

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

### Data availability

The data that has been used is confidential.

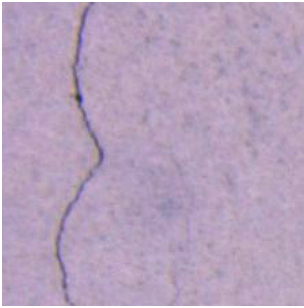
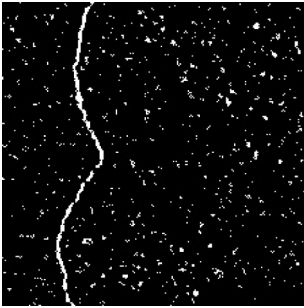
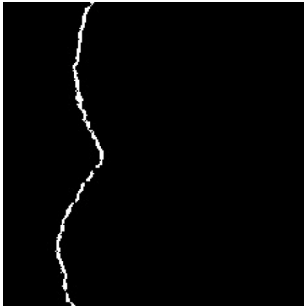
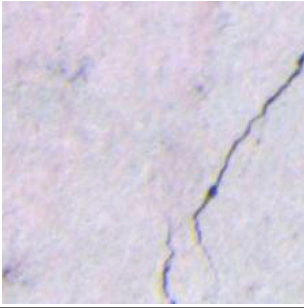
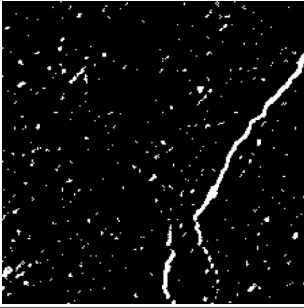


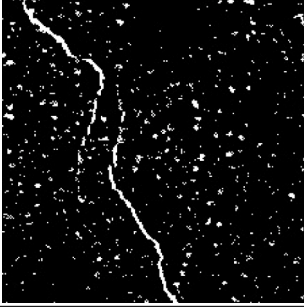
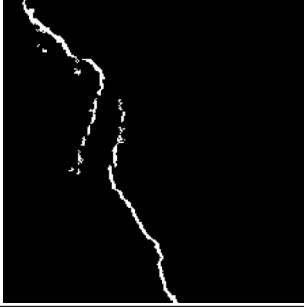
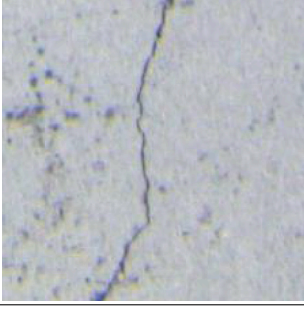
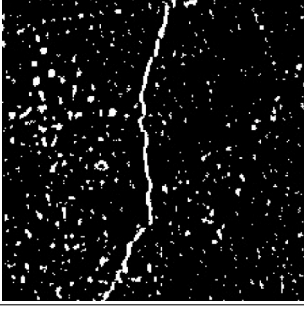
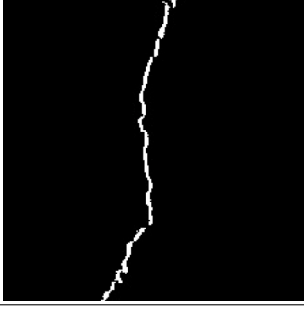
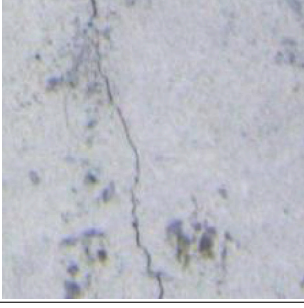
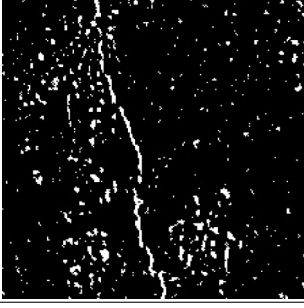
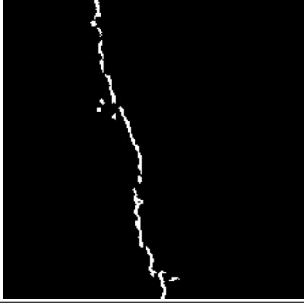
## Appendix

### A.1. Sample images

Table A.1 shows the outputs of the Skele-Marker algorithm on binary images with noise levels ranging from low to very high.



**Table A.1**  
Inputs and outputs of Skele-Marker algorithm.

Noise severity	Image area	Binarised crack	Skele-Marker output
Low			
Moderate			
Moderate			
High			
Very High			

## References

- [1] B. Chan, H. Guan, J. Jo, M. Blumenstein, Towards UAV-based bridge inspection systems: a review and an application perspective, *Struct. Monit. Maint.* 2 (3) (2015) 283–300, <http://dx.doi.org/10.12989/smm.2015.2.3.283>, Publisher: Techno-Press.
- [2] S. Chen, D.F. Laefer, E. Mangina, S.M.I. Zolanvari, J. Byrne, UAV bridge inspection through evaluated 3D reconstructions, *J. Bridge Eng.* 24 (4) (2019) 05019001, [http://dx.doi.org/10.1061/\(ASCE\)BE.1943-5592.0001343](http://dx.doi.org/10.1061/(ASCE)BE.1943-5592.0001343), Publisher: American Society of Civil Engineers.
- [3] J.K. Chow, K.-f. Liu, P.S. Tan, Z. Su, J. Wu, Z. Li, Y.-H. Wang, Automated defect inspection of concrete structures, *Autom. Constr.* 132 (2021) 103959, <http://dx.doi.org/10.1016/j.autcon.2021.103959>.
- [4] L. Ali, S. Harous, N. Zaki, W. Khan, F. Alnajjar, H.A. Jassmi, Performance evaluation of different algorithms for crack detection in concrete structures, in: 2021 2nd International Conference on Computation, Automation and Knowledge Management, ICCAKM, 2021, pp. 53–58, <http://dx.doi.org/10.1109/ICCAKM50778.2021.9357717>.
- [5] Y.-A. Hsieh, Y.J. Tsai, Machine learning for crack detection: Review and model performance comparison, *J. Comput. Civ. Eng.* 34 (5) (2020) 04020038, [http://dx.doi.org/10.1061/\(ASCE\)CP.1943-5487.0000918](http://dx.doi.org/10.1061/(ASCE)CP.1943-5487.0000918), Publisher: American Society of Civil Engineers.
- [6] 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, <http://dx.doi.org/10.1016/j.conbuildmat.2018.08.011>.
- [7] N. Kheradmandi, V. Mehranfar, A critical review and comparative study on image segmentation-based techniques for pavement crack detection, *Constr. Build. Mater.* 321 (2022) 126162, <http://dx.doi.org/10.1016/j.conbuildmat.2021.126162>.
- [8] B. Lei, N. Wang, P. Xu, G. Song, New crack detection method for bridge inspection using UAV incorporating image processing, *J. Aerosp. Eng.* 31 (5) (2018) 04018058, [http://dx.doi.org/10.1061/\(ASCE\)AS.1943-5525.0000879](http://dx.doi.org/10.1061/(ASCE)AS.1943-5525.0000879), Publisher: American Society of Civil Engineers.
- [9] Z. Fan, H. Lin, C. Li, J. Su, S. Bruno, G. Loprencipe, Use of parallel ResNet for high-performance pavement crack detection and measurement, *Sustainability* 14 (3) (2022) 1825, <http://dx.doi.org/10.3390/su14031825>, Number: 3 Publisher: Multidisciplinary Digital Publishing Institute.
- [10] S. Lyasheva, V. Tregubov, M. Shleymovich, Detection and recognition of pavement cracks based on computer vision technology, in: 2019 International Conference on Industrial Engineering, Applications and Manufacturing, ICIEAM, 2019, pp. 1–5, <http://dx.doi.org/10.1109/ICIEAM.2019.8742778>.
- [11] S. Dorafshan, M. Maguire, X. Qi, Automatic surface crack detection in concrete structures using OTSU thresholding and morphological operations, 2016, p. 152.
- [12] S. Dorafshan, R.J. Thomas, M. Maguire, Benchmarking image processing algorithms for unmanned aerial system-assisted crack detection in concrete structures, *Infrastructures* 4 (2) (2019) 19, <http://dx.doi.org/10.3390/infrastructures4020019>, Number: 2 Publisher: Multidisciplinary Digital Publishing Institute.
- [13] S. Sorncharean, S. Phiphobmongkol, Noise removal in crack detection algorithm on asphalt surface images, in: Proceedings of the Fourth International Conference on Computer Vision Theory and Applications, SciTePress - Science and Technology Publications, Lisboa, Portugal, 2009, pp. 269–272, <http://dx.doi.org/10.5220/0001797902690272>, URL: <http://www.scitepress.org/DigitalLibrary/Link.aspx?doi=10.5220/0001797902690272>.
- [14] W. Huang, N. Zhang, A novel road crack detection and identification method using digital image processing techniques, in: 2012 7th International Conference on Computing and Convergence Technology, ICCCT, 2012, pp. 397–400.
- [15] H. Song, W. Wang, F. Wang, L. Wu, Z. Wang, Pavement crack detection by ridge detection on fractional calculus and dual-thresholds, *Int. J. Multimed. Ubiquitous Eng.* 10 (4) (2015) 19–30.
- [16] J. Li, N. Wang, Y. Liu, Y. Yang, A study of crack detection algorithm, in: 2015 Fifth International Conference on Instrumentation and Measurement, Computer, Communication and Control, IMCCC, 2015, pp. 1184–1187, <http://dx.doi.org/10.1109/IMCCC.2015.254>.
- [17] T.Y. Zhang, C.Y. Suen, A fast parallel algorithm for thinning digital patterns, *Commun. ACM* 27 (3) (1984) 236–239.