

Research Article

Metaheuristic Optimized Edge Detection for Recognition of Concrete Wall Cracks: A Comparative Study on the Performances of Roberts, Prewitt, Canny, and Sobel Algorithms

Nhat-Duc Hoang¹ and Quoc-Lam Nguyen²

¹Faculty of Civil Engineering, Institute of Research and Development, Duy Tan University, P809-03 Quang Trung, Da Nang, Vietnam

²Faculty of Civil Engineering, Duy Tan University, R. 202–No. 03 Quang Trung, Da Nang 550000, Vietnam

Correspondence should be addressed to Nhat-Duc Hoang; hoangnhatduc@dtu.edu.vn

Received 8 August 2018; Revised 11 October 2018; Accepted 28 October 2018; Published 12 November 2018

Academic Editor: Giuseppe Quaranta

Copyright © 2018 Nhat-Duc Hoang and Quoc-Lam Nguyen. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Crack detection is a crucial task in the periodic survey of high-rise buildings and infrastructure. Manual survey is notorious for low productivity. This study is aimed at establishing an image processing-based method for detecting cracks on concrete wall surfaces in an automatic manner. The Roberts, Prewitt, Canny, and Sobel algorithms are employed as the edge detection methods for revealing the crack textures appearing in concrete walls. The median filtering and object cleaning operations are used to enhance the image and facilitate the crack recognition outcome. Since the edge detectors, the median filter, and the object cleaning operation all require the appropriate selection of tuning parameters, this study relies on the differential flower pollination algorithm as a metaheuristic to optimize the image processing-based crack detection model. Experimental results point out that the newly constructed approach that employs the Prewitt algorithm can achieve a good prediction outcome with classification accuracy rate = 89.95% and area under the curve = 0.90. Therefore, the proposed metaheuristic optimized image processing approach can be a promising alternative for automatic recognition of cracks on the concrete wall surface.

1. Introduction

Large vertical concrete structures are widely observed in high-rise buildings, retaining structures, and infrastructure. Due to the combined effects of aging, inclement climate conditions, thermal expansion/contraction, and human activities, the health of concrete structures is reducing over time [1]. Thus, ensuring the acceptable level of integrity of these structures is a crucial part in maintenance tasks. Notably, the periodic condition survey by means of visual inspection is the commonly used method to obtain valuable information on the current state of the structure. It is because visual changes in structures directly indicate structural health conditions [2].

A large number of previous works have particularly focused on detecting cracks in concrete structures [3–5]. The reason is that cracks are a major concern when considering the

safety, durability, and serviceability of structures [1, 6]. Identification of cracks is an important step in structure maintenance, which facilitates the effort in reducing their harmful effect. Moreover, timely information about cracks is crucial to prevent potential catastrophic events due to structural failures.

Despite the critical roles of reinforced concrete walls in various structures (e.g., high-rise buildings, retaining walls, dams, and tunnels), human-based visual inspection is still a common method especially in developing countries. This fact is due to the limited access to sophisticated but costly equipment used for machine-based periodic surveys of structure conditions. Needless to say, manual inspection is not only slow in progress but also has low consistency due to quantitative evaluation [1]. The assessment outcomes of human inspectors are largely determined by their level of training and experience. Notably, the observation of

inspectors is also hindered by irregular lighting conditions and elevations [7]. These facts significantly reduce the accuracy of the manual crack detection. Moreover, the phase of data processing and compiling is also very time-consuming especially for structures with a large surface area of concrete walls such as skyscrapers and dams. Kim et al. [8] summarized that manual visual inspection is ineffective in terms of cost, safety, and evaluation accuracy.

Due to those reasons, developing automatic methods for detecting concrete wall cracks has drawn attentions of many scholars [9–11]. Recently, various image processing techniques have been employed to boost the performance of concrete surface crack detection in both aspects of accuracy and productivity. In these methods, these techniques are aimed at recognizing cracks presented on a two-dimensional image, followed by further analysis of crack properties [12, 13].

Abdel-Qader et al. [14] employed edge-detection approaches including Sobel, Canny, and fast Haar transform for detecting cracks appearing in bridges. A crack-measuring system relying on multitemporal images with the application of the first derivative of a Gaussian filter was proposed by Chen et al. [15]. Chen and Hutchinson [16] utilized the level set method and morphological image processing techniques to recognize and analyze cracks in the laboratory environment. An image processing technique that uses various morphological operations including background brightness adjustment, binarisation, and shape analysis has been put forward by Lee et al. [6] to improve the crack detection performance. Adhikari et al. [17] established an edge detection-based method for measuring crack width of the bridge concrete structure. Dorafshan [18] relied on three commonly employed edge detectors including the Sobel, Roberts, and Gaussian high-pass filter methods for detecting cracks on surfaces of bridge decks; this study points out that the Sobel edge detector has achieved the most accurate crack detection outcome with an accuracy rate of 92%.

Recently, Yang et al. [5] investigated the performance of a novel approach that employs image matching based on the optical flow and subpixel to analyze slight concrete surface displacements; this approach has shown good performance in detecting thin cracks. An image processing-based model for detection of surface cracks in building structures using an improved Otsu method for image thresholding has been proposed by Hoang [12]. Nguyen et al. [3] proposed a multiphase technique for analyzing images that use a B-spline level set model and a Savitzky–Golay filter.

Besides edge detection and image filtering approaches, machine learning has also been successfully employed in concrete surface crack detection [8–10]. Particularly, convolutional neural network (CNN) models have drawn attention of many scholars in constructing automatic crack recognition models [2, 4, 19, 20].

Recently, Dorafshan et al. [21] have compared the capability of six edge detectors (the Roberts, Prewitt, Sobel, Laplacian of Gaussian, Butterworth, and Gaussian algorithms) and CNN for crack detection in concrete structures. The research finding is that CNN has outperformed those edge detectors by a large margin. The main advantage of

CNN is that the feature extraction and pattern classification can be performed in an integrated and autonomous manner. However, CNNs often require a large number of training data samples and a considerable computational expense. Moreover, the performance of the edge detectors can be potentially optimized and improved with the use of metaheuristic algorithms.

Since automatic concrete wall crack detection is a challenging task due to arbitrary forms of cracks, inconsistent lighting condition, and disturbing noise patterns (e.g., spallings, stains, and holes) [8], investigating other alternative tools can be useful in both academic and practicing aspects. Moreover, despite the fact that edge detection algorithms have been employed for concrete surface crack detection, few studies have been dedicated in investigating the performance of these algorithms integrated with metaheuristics. It is noted that the implementation of edge recognition models demands a proper setting of tuning parameters. The fine tuning of these parameters potentially leads to improvements in crack detection accuracy. Therefore, there is a need to supplement the body of knowledge by conducting studies with the focus on the hybridization of edge detection and metaheuristic algorithms.

This study is aimed at filling the aforementioned gap in the literature by constructing novel edge detection-based methods for concrete wall crack recognition. The new method relies on a median filter for noise removal, four edge detection algorithms (Roberts, Prewitt, Sobel, and Canny), and a morphological operation for filtering out unwanted background objects in digital images. The median filter, the four edge detection algorithms, and the employed morphological operation all necessitate appropriate settings of their tuning parameters. As being observed in the literature, the selection of parameters of image processing models can be formulated as optimization problems [22–24]. Hence, our study relies on the differential flower pollination, as a metaheuristic algorithm, to optimize the crack detection model by means of identifying an appropriate set of model hyperparameters. Furthermore, a dataset including 1080 image samples has been collected to validate the new construction approach.

The rest of the paper is organized as follows: the second part briefly reviews the research methodology, followed by the third part that describes the image acquisition process; the proposed metaheuristic optimized edge detection model is presented in the fourth section; the experimental results will be reported in the fifth section, followed by several concluding remarks of the study being stated in the final section.

2. Research Methodology

2.1. Median Filter (MF). MF is an effective approach to smooth the digital image. This filter helps to remove the unwanted noise existing in the concrete wall background such as small stains or spallings. In image processing, MF is highly preferable since this image smoothing method can preserve edges which are potentially cracks. For a fixed window size and in the aspect of edge preservation, Arias-

Castro and Donoho [25] experimentally showed that the result of MF can be better than that of Gaussian blur.

For each pixel in an image under analysis for crack detection, MF replaces each pixel with the median of its neighboring pixels [26]. The number of the neighboring pixels depends on the window size which is a parameter of MF. The common choice of the window size is definitely context dependent and usually requires a trial-and-error process to determine the appropriate setting. MF can help suppress unwanted details on the image background. However, a too large value of window size can lead to a significant loss of information regarding the cracking objects. In this study, the window size is considered to be a hyperparameter of the integrated crack detection model, and it is optimized by the employed metaheuristic algorithm.

2.2. The Employed Edge Detection Algorithms. Generally, edge detectors are mathematical methods that are able to recognize points in an image at which the gray-level intensity expresses discontinuities [27]. The point locations at which the gray-level intensity varies sharply can be grouped into segments called edges. Edge detection algorithms are highly suitable for identifying cracks in concrete walls. It is because crack pixels are highly associated with pixel locations having discontinued gray-level intensity. Therefore, this study employs four edge detection approaches (the Roberts, Prewitt, Sobel, and Canny algorithms) for automatic crack recognition. The descriptions of these edge detection approaches are briefly presented in this section of the study.

2.2.1. Roberts Edge Detection Method. The Roberts method, first described by Roberts [28], is a simple and fast algorithm for calculating the spatial gradient measurement on a digital image. This algorithm quickly reveals locations featuring high spatial frequency which often correspond to crack objects. To implement the Roberts method, the image is first converted from the RGB format into the grayscale format. After being processed by this algorithm, pixel values at each location of the output image are the approximated absolute magnitude of the spatial gradient of the original grayscale image at the location [29].

The following filters R_x and R_y are applied to the whole image separately:

$$\begin{aligned} R_x &= \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}, \\ R_y &= \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}. \end{aligned} \quad (1)$$

Subsequently, these two filters are combined to calculate the absolute magnitude of the gradient as follows:

$$R_G = \sqrt{(R_x \cdot N_{IR})^2 + (R_y \cdot N_{IR})^2}, \quad (2)$$

where R_G is the absolute magnitude of the gradient, the symbol \cdot denotes a dot product of two matrices, and N_{IR} denotes an image neighborhood with the size of 2×2 pixels [29].

The Roberts edge detector requires a hyperparameter T_R which is a threshold parameter. If the gradient values of pixels in the image are smaller than the threshold value, they are replaced by these threshold values. Thus, an image with detected edges can be obtained from the gradient with the use of a threshold value T_R .

Figure 1 illustrates edge detection results using the Roberts method for an image without cracks and an image with cracks. It is clearly seen that the quality of the crack detection outcome strongly depends on the selection of the parameter T_R .

2.2.2. Prewitt Edge Detection Method. The Prewitt edge detection [30] also relies on two filters to estimate the derivatives of each location within an image. Similar to the Roberts algorithm, the Prewitt algorithm also requires a parameter T_P which serves as a threshold value for determining edges. Figure 2 provides examples of edge detection results using the Prewitt method for an image without cracks and an image with cracks.

The following filters to compute the approximation to the derivatives are used to highlight edge pixels:

$$\begin{aligned} P_x &= \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}, \\ P_y &= \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}. \end{aligned} \quad (3)$$

The two aforementioned filters are combined to compute the total gradient as follows:

$$P_P = \sqrt{(P_x \cdot N_{IP})^2 + (P_y \cdot N_{IP})^2}, \quad (4)$$

where N_{IP} denotes an image neighborhood with the size of 3×3 pixels [29].

2.2.3. Sobel Edge Detection Method. As proposed in the previous work in [31], the Sobel edge detection is a widely employed method in image processing [32]. Notably, this edge detector highlights edges by first smoothing the image before calculating the derivatives. The filter h_x is employed for smoothing the image before computing the partial derivative in the x direction:

$$h_x = \begin{bmatrix} 1 & 1 & 1 \\ 2 & 2 & 2 \\ 1 & 1 & 1 \end{bmatrix}. \quad (5)$$

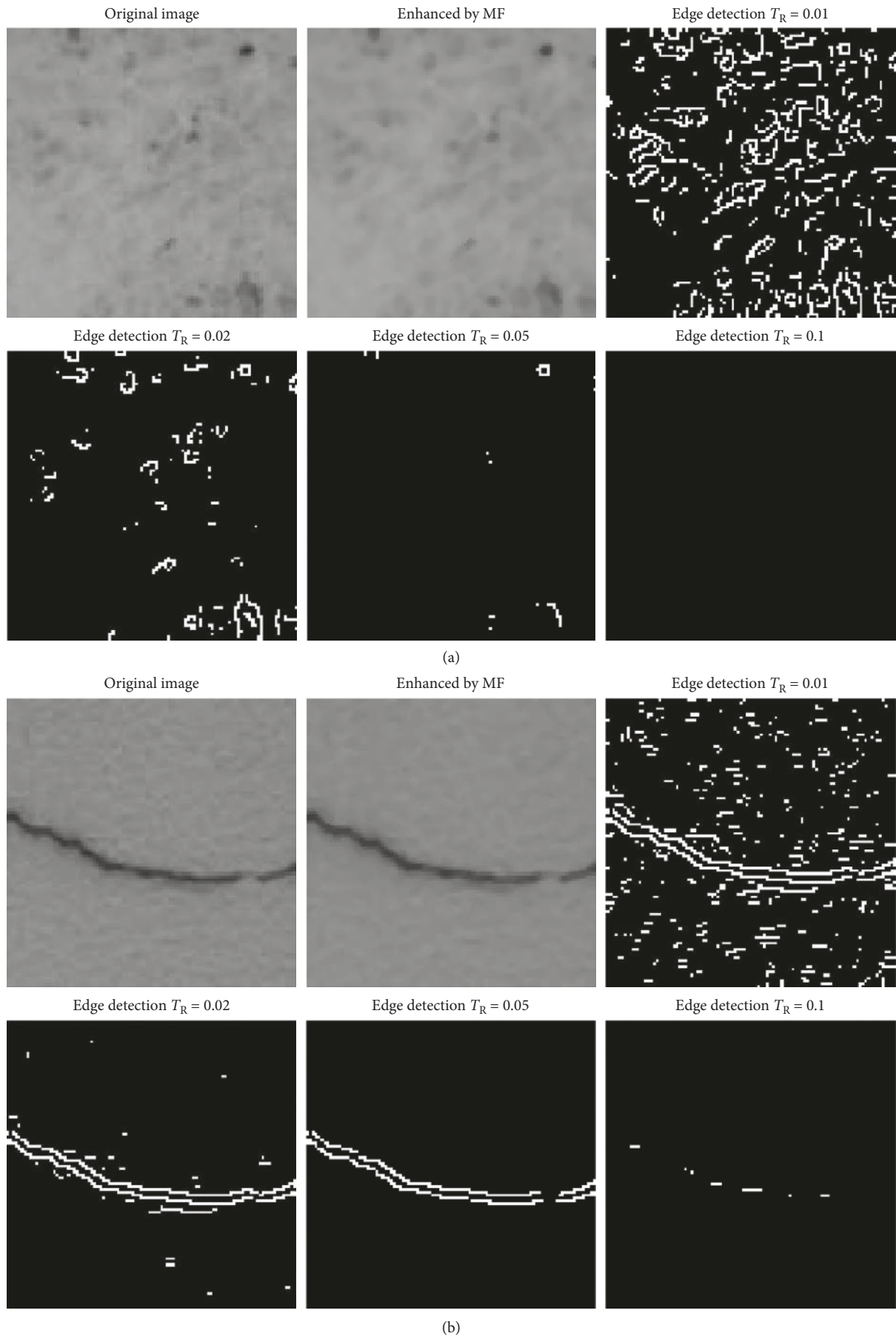


FIGURE 1: Edge detection result using the Roberts method: (a) an image without cracks and (b) an image with cracks.

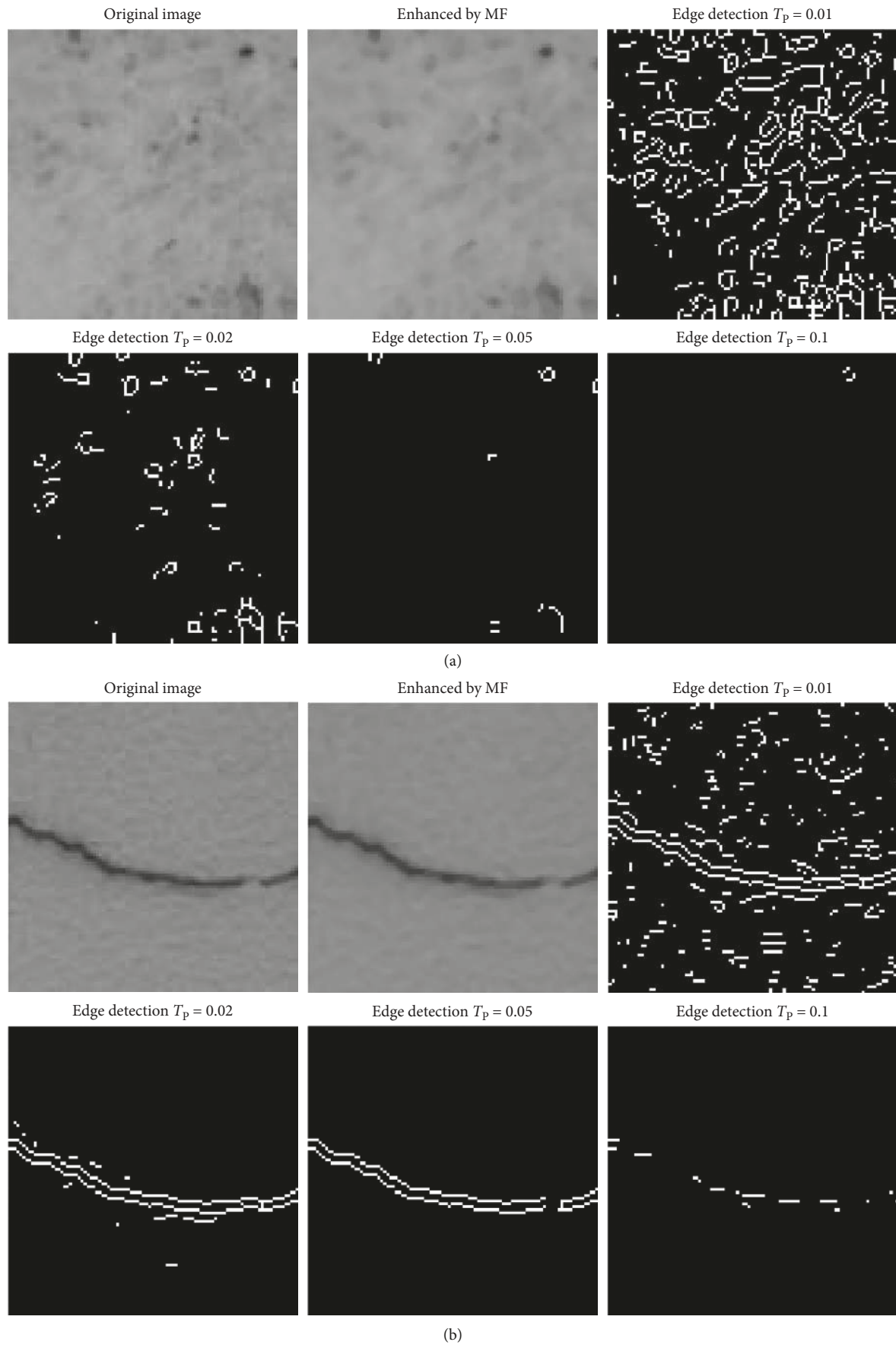


FIGURE 2: Edge detection result using the Prewitt method: (a) an image without cracks and (b) an image with cracks.

Because the filters used for derivative approximation and image smoothing are both linear, they can be integrated into the following form for the x direction [29]:

$$h_{\text{Sobel},x} = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}, \quad (6)$$

where $h_{\text{Sobel},x}$ is the filter employed for approximating the derivative for the x direction.

In the same manner, the filter that computes the partial derivative in the y direction is shown as follows [29]:

$$h_{\text{Sobel},y} = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}, \quad (7)$$

where $h_{\text{Sobel},y}$ is the filter used for approximating the derivative for the y direction.

Accordingly, the final outcome of the gradient approximations can be combined to give the gradient magnitude via the following formula:

$$h_{\text{Sobel}} = \sqrt{(h_{\text{Sobel},x} \cdot N_{\text{IS}})^2 + (h_{\text{Sobel},y} \cdot N_{\text{IS}})^2}, \quad (8)$$

where h_{Sobel} is the combined result of the gradient approximations and N_{IS} denotes an image neighborhood with the size of 3×3 pixels [29].

Notably, a threshold value T_s determines the presentation of the output image with all edges being displayed (Figure 3). If the Sobel gradient values of pixels are lesser than the threshold value T_s , they are substituted by the threshold value [33].

2.2.4. Canny Edge Detection Method. Canny [34] introduced a multistep algorithm for edge recognition. A Gaussian convolution is first applied to the image. Accordingly, a 2D first-derivative operator is computed to reveal locations of the image featuring intensity discontinuities. At the first step, the employed Gaussian filter is presented as follows:

$$g(m, n) = G_\sigma(m, n) * f(m, n), \quad (9)$$

where

$$G_\sigma = \frac{1}{\sqrt{2\pi}\sigma^2} \exp\left(-\frac{m^2 + n^2}{2\sigma^2}\right). \quad (10)$$

Moreover, G_σ is a Gaussian function with the variance of σ^2 . Herein, the value of σ is chosen to be the default value $\sqrt{2}$ as suggested by the MATLAB Image Processing Toolbox [35]. The symbol $*$ denotes the convolution operator; m and n are the indices used to specify the location of a pixel within an image. $f(m, n)$ represents an image neighborhood at the pixel coordinate of (m, n) .

Accordingly, the gradient of $g(m, n)$ using a certain gradient operator (e.g., Sobel) is calculated using the following equation:

$$g_{m,n}(m, n) = \sqrt{(g_m(m, n) \cdot N_{\text{IC}})^2 + (g_n(m, n) \cdot N_{\text{IC}})^2}, \quad (11)$$

where N_{IC} represents an image neighborhood and $g_{m,n}(m, n)$ denotes the combined result of the estimated gradients.

In the next step, nonmaximum suppression is performed to thin out the edges [34]. Moreover, it is noted that the Canny method relies on two parameters of T_1 and T_2 to carry out a double thresholding strategy (Figure 4). Edge pixels that are stronger than the upper threshold T_2 are determined as strong edges. Those that are weaker than the lower threshold T_1 are suppressed. Moreover, edge pixels that range between T_1 and T_2 are determined to be weak edges. Finally, all the edge pixels not connected to strong edge pixels and belonging to the weak edge group are all suppressed.

2.3. The Differential Flower Pollination (DFP) Algorithm for Model Parameter Optimization. The determination of the hyperparameters of image processing approaches can be a challenging problem [36]. The first reason is the landscape feature of the cost function to be minimized can be complicated and may contain a large number of local minima. Another reason is that the search space of the parameters is continuous; thus, there is an infinite number of possible solutions. Accordingly, metaheuristic algorithms are highly suitable for dealing with such a circumstance.

DFP [37] is a global optimizer that inherits the advantages from the two popular metaheuristic algorithms of the differential evolution (DE) and the flower pollination algorithm (FPA). DFP employs the Lévy-flight-based global explorative search of FPA [38] and the explorative local search based on mutation-crossover operators of DE. As experimentally demonstrated in the previous work [37], DFP can help optimize the model parameters with satisfactory outcomes.

The general picture of the DFP optimization algorithm is demonstrated in Figure 5. In the first iteration ($g_{\text{DFP}} = 1$), all population members with PopSize individuals are randomly initiated within the feasible search domain. During the evolutionary process, each member's position is altered through either the FPA-based global pollination operator or the DE-based local pollination operator. Moreover, based on suggestions of previous works [37, 39], a selection probability $P = 0.8$ is used to determine the frequencies of the global and local pollination phases.

The process of Lévy-flight-based global pollination can be expressed in the following equation:

$$X_i^{\text{trial}} = X_i^{g_{\text{DFP}}} + L \cdot (X_i^{g_{\text{DFP}}} - X_{\text{best}}), \quad (12)$$

where g is the index of the current generation and X_i^{trial} is a newly generated trial individual.

The process of DE-based local pollination is divided into two steps and described in the following way:

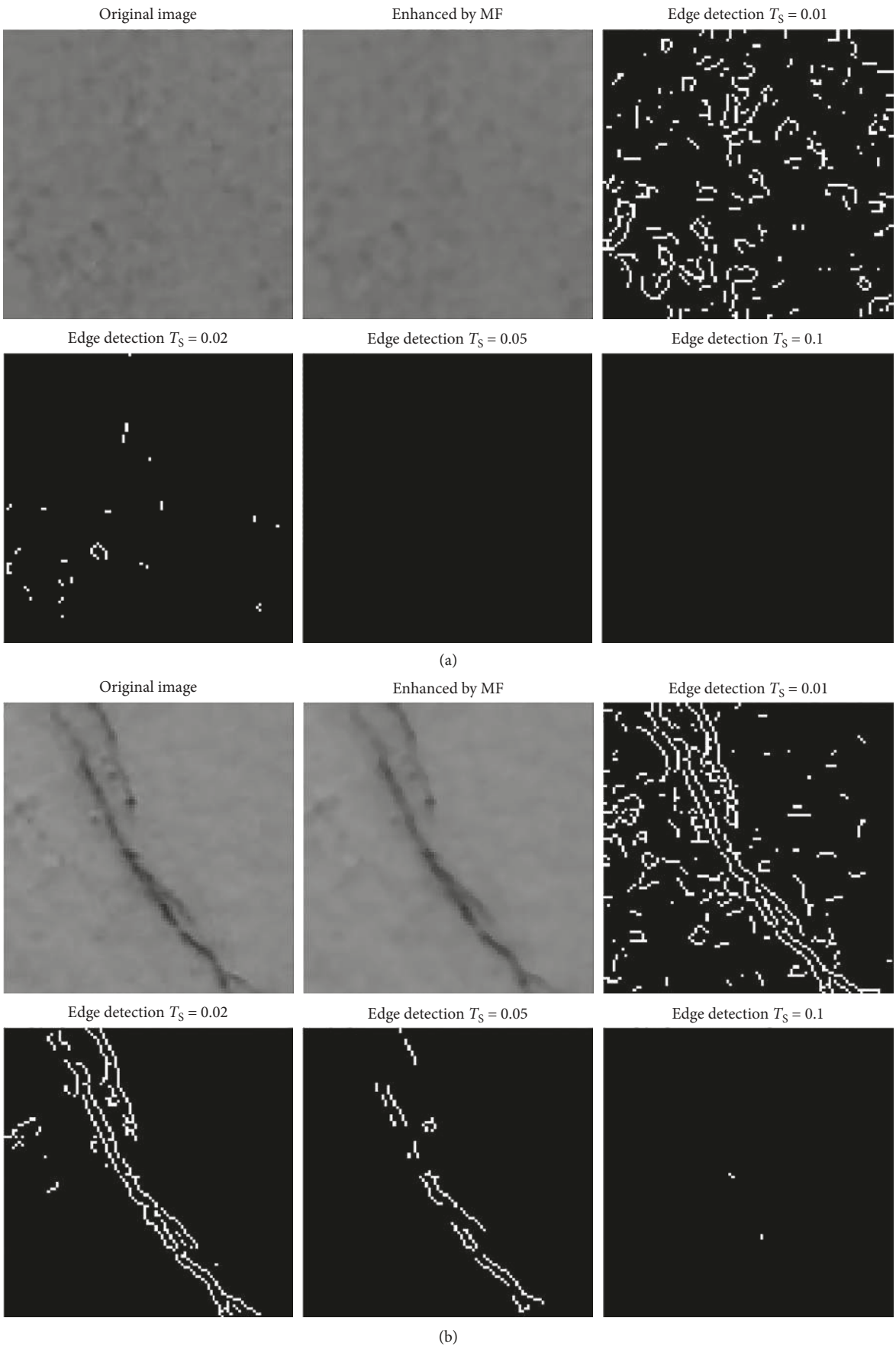


FIGURE 3: Edge detection outcome employing the Sobel method: (a) an image without cracks and (b) an image with cracks.

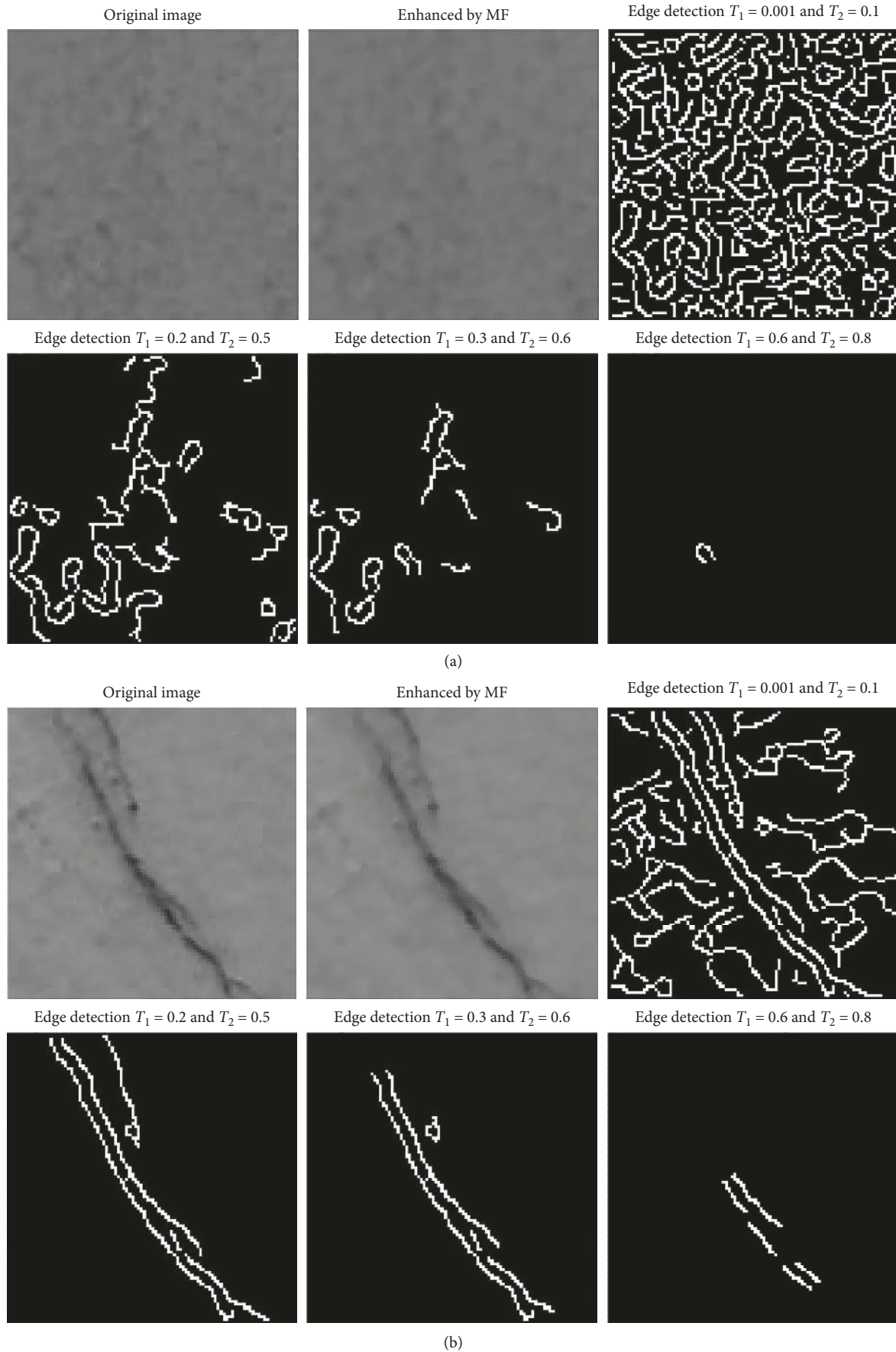


FIGURE 4: Edge detection outcome employing the Canny method: (a) an image without cracks and (b) an image with cracks.

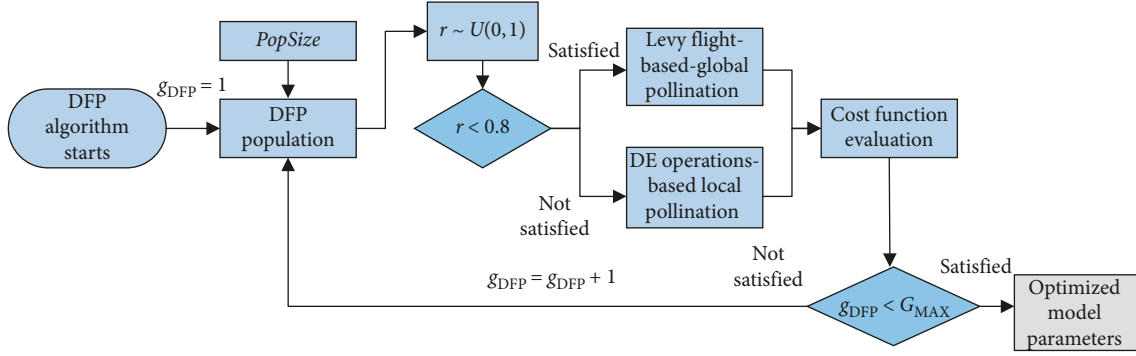


FIGURE 5: The DFP optimization algorithm. Note: G_{MAX} denotes the maximum number of generations. $U(0, 1)$ is a uniform number generator. r denotes a random number generated from $U(0,1)$. g_{DFP} is the index of the current generation.

$$\begin{aligned} \text{generate a mutated flower: } X_{i,g_{DFP}}^{\text{mutated}} \\ = x_{r_1,g_{DFP}} + F \cdot (x_{r_2,g_{DFP}} - x_{r_3,g_{DFP}}), \end{aligned} \quad (13)$$

where r_1 , r_2 , and r_3 represent three random indices and F denotes a mutation scale factor:

$$\begin{aligned} \text{generate a crossed flower: } X_{j,i,g_{DFP}+1}^{\text{crossed}} \\ = \begin{cases} X_{j,i}^{\text{mutated}}, & \text{if } \text{rand}_j \leq \text{Cr} \text{ or } j = \text{rnb}(i), \\ X_{j,i,g_{DFP}}, & \text{if } \text{rand}_j > \text{Cr} \text{ and } j \neq \text{rnb}(i), \end{cases} \end{aligned} \quad (14)$$

where Cr represents the crossover probability which is often chosen to be 0.8 [40]. Moreover, as recommend by Hoang et al. [37], F is drawn from a Gaussian distribution with mean = 0.5 and standard deviation = 0.15.

3. Acquisition of Concrete Wall Images

To construct the image processing-based model for detecting cracks appearing on concrete wall surfaces, this study has collected images from several buildings in the Da Nang city (Vietnam). The camera is positioned at a distance of about 1 meter from the concrete walls. It is noted that image samples with their ground truth status of crack or noncrack have been categorized by inspectors. To alleviate the computation cost, image size is fixed to be 50×50 pixels. Thus, image cropping operation is performed to create image samples. In addition, since one pixel represents an area of roughly $3.0 \times 3.0 \text{ mm}^2$, the surface area covered by one image sample is approximately $150 \times 150 \text{ mm}^2$. For each class label of concrete wall condition, 540 image samples have been collected. Hence, the image dataset includes 1080 samples. The collected dataset is demonstrated in Figure 6. It is noted that all the images have been captured by the camera of Asus ZenFone 4 Max Pro (16 MP resolution and F2.0 aperture lens).

4. The Proposed Metaheuristic Optimized Image Processing Model for Concrete Wall Crack Detection

This section describes the proposed method for automatic detection of cracks on the concrete wall surface. Since the

model is a combination of metaheuristic and edge detection algorithms used for concrete wall crack recognition, it is named the “Metaheuristic Optimized Edge Detection model for concrete wall Crack Recognition” (MO-EDCR). The MO-EDCR model (Figure 7) consists of three basic steps:

- (1) Noise suppression by means of MF
- (2) Edge identification using the algorithms of the Roberts, Prewitt, Sobel, and Canny detectors
- (3) Morphological operations for removing unwanted small objects

As can be seen from Figure 7, to construct the automatic crack recognition model, there are three groups of hyper-parameters needed to be determined: the window size parameter (W) used in the step of median filtering, the edge detectors’ thresholds (T_E), and a threshold parameter (Th_m) used in the morphological operation for removing small objects. It is noted that the tuning parameters of the edge detection algorithms are T_R for the Roberts method, T_P for the Prewitt method, T_S for the Sobel method, and T_1 and T_2 for the Canny method. The ranges of the window size parameter (W), the edge detectors’ thresholds (T_E), and a threshold parameter (Th_m) are $[2 \ 10]$, $[0 \ 1]$, and $[0, 0.1]$, respectively.

It is noted that the size of an image sample is 50×50 pixels. Thus, if elements of W are greater than 10 which surpasses 20% of the image size, the processed image sample is too blurred, and this leads to a significant loss in the image detail. Thus, the maximum value of elements of W is selected to be 10. The range of T_E is chosen to comply with the feasible range of thresholding parameters used in edge detectors which are specified by the MATLAB Image Processing Toolbox [35]. In the case of Th_m , since the removed objects are often dirt or noisy points, their sizes are comparatively smaller than those of crack objects. Based on several trial-and-error experiments, the range of Th_m of $[0, 0.1]$ is found to be appropriate for the collected image samples. Therefore, this range of Th_m is selected to ease the optimization process of DFP.

When the images with recognized edges are obtained, the objects, demonstrated by pixels having the intensity 1, are identified and isolated with the use of the MATLAB

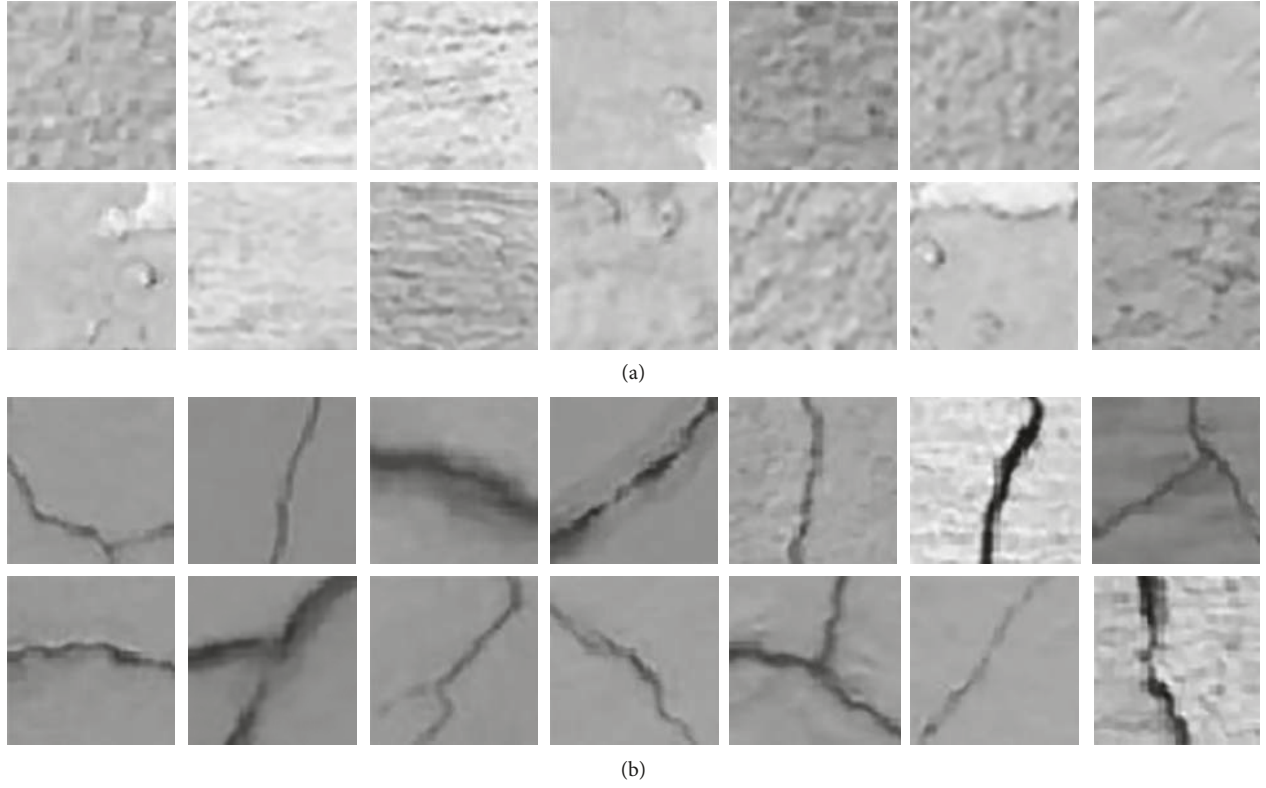


FIGURE 6: Samples of the image dataset: (a) noncrack images and (b) crack images.

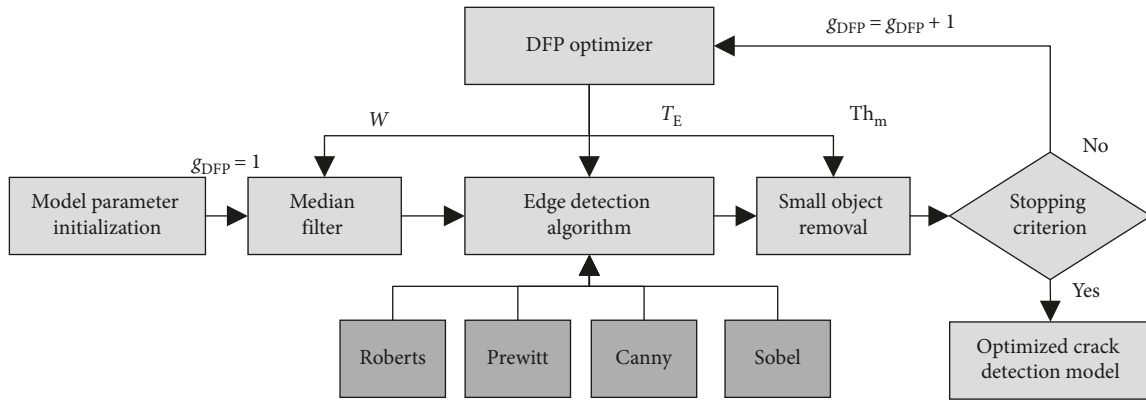


FIGURE 7: The proposed model for recognition of cracks on the concrete wall surface (MO-EDCR). Note: W denotes the window size parameter. T_E is the edge detection threshold. Th_m is a threshold parameter used in the morphological operation for small object removal.

Image Processing Toolbox [35]. Objects that have the size smaller than a certain threshold are removed from the digital image because they are often unwanted noncrack ones. The minimum size (MS) of an object is computed as follows:

$$MS = PN \times Th_m, \quad (15)$$

where PN is the total number of pixels of an image. Herein, $PN = 50 \times 50 = 2500$. As mentioned earlier, Th_m is a threshold parameter employed for removing small objects; this parameter is also automatically determined by DFP.

Moreover, the following cost function (CF) is used to evaluate the quality of a set of model parameters:

$$CF = \frac{FPR + FNR}{2}, \quad (16)$$

where FPR and FNR are the false-positive rate and the false-negative rate, respectively. The equations used to compute FPR and FNR will be presented in the next section. Thus, if the aforementioned CF is minimized, then it is possible to obtain a model that features low values of both FPR and FNR.

5. Experimental Results

5.1. Performance Evaluation. Since the detection of cracks in concrete walls is formulated as a two-class categorization problem, the measurement indices used for quantifying the classification model can be employed. The first commonly used index is the classification accuracy rate (CAR). The higher the CAR value, the better the model performance. As mentioned earlier, ground truth labels of crack or noncrack for each image sample are assigned by human inspectors. Since this study serves as a preliminary survey to collect the current state of buildings, the ground truth labels are determined at an image level.

In addition, the true-positive rate TPR (the percentage of positive instances correctly classified), the false-positive rate FPR (the percentage of negative instances misclassified), the false-negative rate FNR (the percentage of positive instances misclassified), and the true-negative rate TNR (the percentage of negative instances correctly classified) are also widely employed. These four rates are computed as follows:

$$\begin{aligned} \text{TPR} &= \frac{\text{TP}}{\text{TP} + \text{FN}}, \\ \text{FPR} &= \frac{\text{FP}}{\text{FP} + \text{TN}}, \\ \text{FNR} &= \frac{\text{FN}}{\text{TP} + \text{FN}}, \\ \text{TNR} &= \frac{\text{TN}}{\text{TN} + \text{FP}}, \end{aligned} \quad (17)$$

where TP, TN, FP, and FN denote the true-positive, true-negative, false-positive, and false-negative values, respectively.

Furthermore, based on the outcomes of TP, FP, and FN, the precision or positive predictive value (PPV), negative predictive value (NPV), recall, and *F1* score can be computed for measuring predictive capability. These indices are calculated as follows [41, 42]:

$$\begin{aligned} \text{precision} &= \frac{\text{TP}}{\text{TP} + \text{FP}}, \\ \text{NPV} &= \frac{\text{TN}}{\text{TN} + \text{FN}}, \\ \text{recall} &= \frac{\text{TP}}{\text{TP} + \text{FN}}, \\ \text{F1 Score} &= \frac{2\text{TP}}{2\text{TP} + \text{FP} + \text{FN}}. \end{aligned} \quad (18)$$

5.2. Model Prediction Result and Performance Comparison. Before the model construction phase, the dataset consisting of 1080 image samples has been divided into two sets: the training set which accounts for 70% of the data and the

testing set which consists of 30% of the data. The first set is used for model construction, and the second set is employed for demonstrating predictive performance of the proposed model. During the model construction phase, DFP, as a metaheuristic algorithm, optimizes the tuning parameters of the image processing model. The most appropriate values of tuning parameters are then employed to establish the proposed MO-EDCR.

Optimization results of MO-EDCR which employed the Roberts, Prewitt, Sobel, and Canny algorithms are illustrated in Figure 8. The maximum number of iterations (G_{MAX}) of the DFP-based search engine is set to be 100 iterations. As observed from this figure, the DFP metaheuristic algorithm can help the image processing-based crack detection model to quickly converge to a good solution of model parameters. The optimized parameters of the proposed MO-EDCR for each edge detector are reported in Table 1. As can be seen from this table, the window sizes used in MF are 5×5 for the Roberts, Prewitt, and Sobel algorithms. However, the most appropriate window size in the case of the Canny algorithm is 3×3 . In addition, the best threshold values of the Robert, Prewitt, and Sobel edge detectors are 0.0160, 0.0213, and 0.0220, respectively. Since the Canny algorithm requires two threshold values, these two values are found to be 0.3144 and 0.9058 by DFP.

The most suitable values of Th_m used in the morphological operation phase of the Roberts, Prewitt, Sobel, and Canny algorithms are 0.0137, 0.0081, 0.0082, and 0.0196, respectively. It is interesting to see that the employment of the Prewitt and Sobel methods has resulted in a quite similar value of Th_m . Meanwhile, the Roberts and Canny approaches require a comparatively higher value of Th_m . Thus, it can be seen that the values of Th_m for different edge detectors may not be similar. This phenomenon can be explained by the fact that the edges obtained from different algorithms have different thicknesses. Moreover, the window size parameter employed by the Canny algorithm ($W = [3, 3]$), which is automatically identified by DFP, is lower than those of other algorithms ($W = [5, 5]$) because the Gaussian filter has been used in the Canny algorithm to partially smooth the image sample.

To reliably assess the model performance, the model construction and verification phases have been repeated 20 times to obtain the measurement indices of CAR, AUC, TPR, FPR, FNR, TNR, precision, and recall. It is noted that the training and testing datasets in an individual run are different and randomly sampled from the original dataset.

Table 2 provides the detailed outcomes of the predictive performances of the MO-EDCR that employs the Roberts algorithm, the Prewitt algorithm, the Sobel algorithm, and the Canny algorithm. In this table, the mean, standard deviation (Std), and coefficient of variation (COV) of the prediction performance of each crack detection model are reported. It is noted that COV is the ratio of the standard deviation to the mean of prediction performance.

The MO-EDCR model using these four edge detectors is denoted as DFP-Roberts, DFP-Prewitt, DFP-Sobel, and DFP-Canny, respectively. Observably, the DFP-Prewitt has attained the best prediction performance with

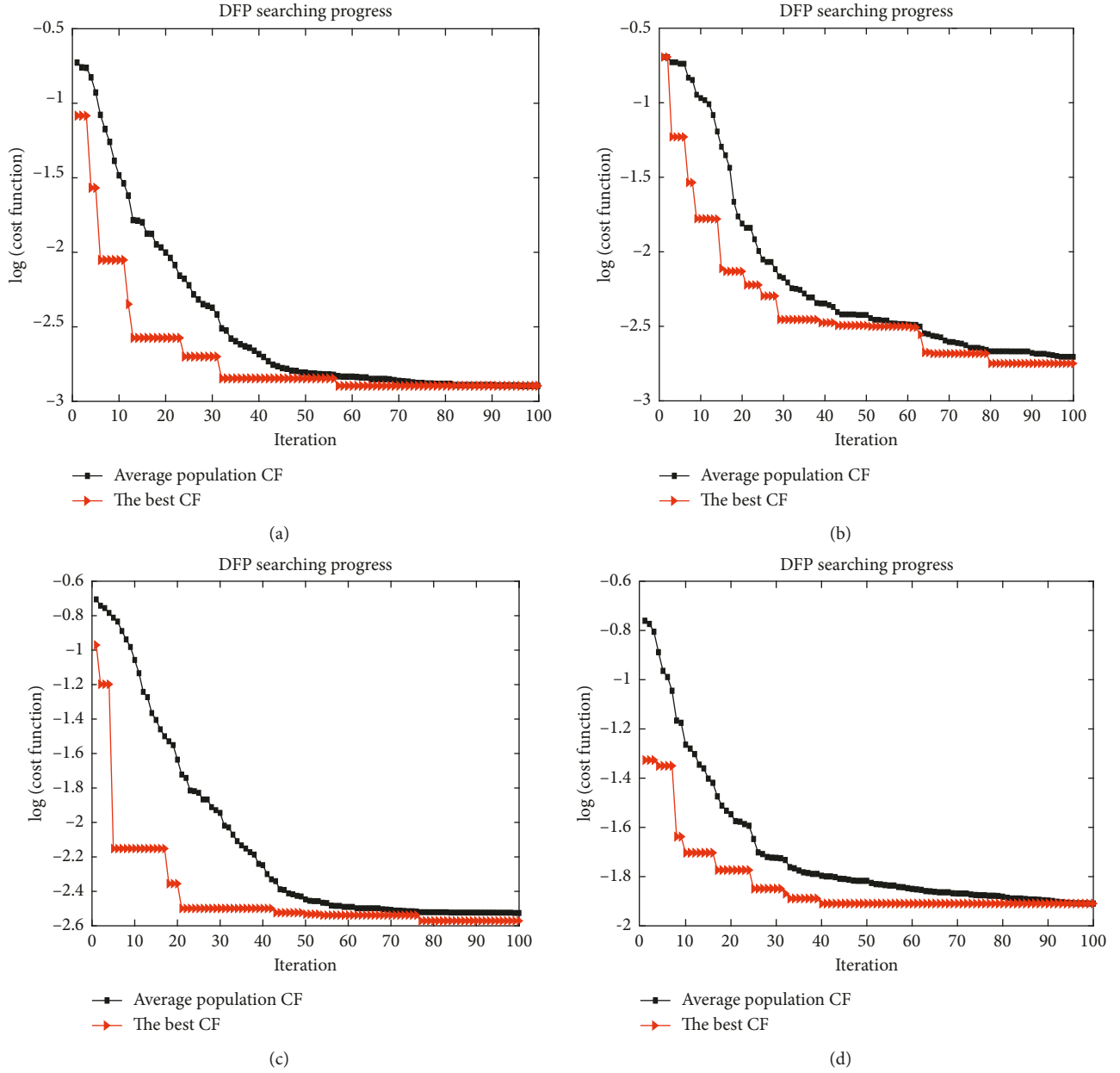


FIGURE 8: Optimizing result of MO-EDCR that employs (a) the Roberts algorithm, (b) the Prewitt algorithm, (c) the Sobel algorithm, and (d) the Canny algorithm.

TABLE 1: The optimized model parameters.

Edge detector	Optimized model parameter
Roberts	$T_R = 0.0160$, $W = [5.0000, 5.0000]$, and $Th_m = 0.0137$
Prewitt	$T_P = 0.0213$, $W = [5.0000, 5.0000]$, and $Th_m = 0.0081$
Sobel	$T_S = 0.0220$, $W = [5.0000, 5.0000]$, and $Th_m = 0.0082$
Canny	$T_1 = 0.3144$, $T_2 = 0.9058$, $W = [3.0000, 3.0000]$, and $Th_m = 0.0196$

CAR = 89.954%, AUC = 0.900, precision = 0.910, and recall = 0.890, followed by DFP-Roberts (CAR = 89.630%, AUC = 0.896, precision = 0.930, and recall = 0.857), DFP-Sobel

(CAR = 89.475%, AUC = 0.895, precision = 0.925, and recall = 0.858), and DFP-Canny (CAR = 83.411%, AUC = 0.834, precision = 0.808, and recall = 0.880).

DFP-Prewitt also attains the highest NPV = 0.881, followed by DFP-Roberts (0.878). The NPV of DFP-Sobel (0.866) is relatively close to that of DFP-Canny (0.868). The best prediction outcome in terms of F1 score is achieved by DFP-Roberts (0.903). The F1 score of DFP-Prewitt (0.895) is equal to that of DFP-Sobel. DFP-Canny attains the lowest F1 score = 0.833.

Although the Roberts algorithm alone may not be as effective in edge detection as other algorithms, the DFP-Roberts has obtained the prediction performance which surpasses other models in several performance measurement

TABLE 2: Classification result comparison.

Prediction result	DFP-Prewitt			DFP-Roberts			DFP-Sobel			DFP-Canny		
	Mean	Std.	COV	Mean	Std.	COV	Mean	Std.	COV	Mean	Std.	COV
CAR	89.954	1.033	0.011	89.630	1.307	0.015	89.475	1.069	0.012	83.411	1.666	0.020
AUC	0.900	0.010	0.012	0.896	0.013	0.014	0.895	0.011	0.012	0.834	0.017	0.020
TPR	0.890	0.022	0.024	0.857	0.015	0.018	0.858	0.019	0.022	0.880	0.028	0.031
FPR	0.091	0.017	0.181	0.065	0.019	0.289	0.069	0.016	0.239	0.213	0.027	0.126
FNR	0.110	0.022	0.197	0.143	0.015	0.106	0.142	0.019	0.132	0.120	0.028	0.231
TNR	0.909	0.017	0.018	0.935	0.019	0.020	0.931	0.016	0.018	0.787	0.027	0.034
Precision	0.910	0.014	0.016	0.930	0.017	0.019	0.925	0.018	0.019	0.808	0.020	0.024
Recall	0.890	0.022	0.024	0.857	0.015	0.018	0.858	0.019	0.022	0.880	0.028	0.031
NPV	0.881	0.020	0.022	0.879	0.018	0.020	0.866	0.018	0.021	0.868	0.018	0.021
F1 score	0.895	0.017	0.019	0.903	0.016	0.017	0.895	0.014	0.016	0.833	0.017	0.021

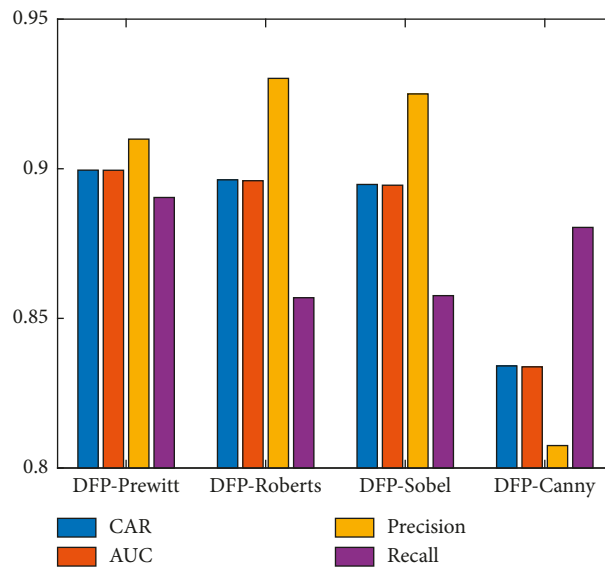


FIGURE 9: Model performance comparison.

indices. This result can be due to the integration of the Roberts algorithm with the DFP metaheuristic approach, median filter, and morphological operations used for image cleaning. The latter three methods of computational intelligence and image processing techniques have assisted the Roberts algorithm by alleviating its weakness. Therefore, the integrated model of DFP-Roberts has obtained comparatively good crack classification performance for the currently collected image dataset.

The recall and TPR values of DFP-Canny (0.880) are close to those of DFP-Roberts (0.890); however, other indices of DFP-Canny are inferior to those of the other three edge detectors. The results of MO-EDCR using the four edge detectors are graphically shown in Figures 9 and 10. Figure 10 presents the statistical characteristics of classification performance of the four crack detection models obtained from the repeated subsampling process with 20 runs. The model performances are graphically described by the four box plots. It is noted that the bottom and top of each box plot are the first and third quartiles of data, respectively; the red band within the box denotes the median [43].

Based on the result comparison and with the consideration that AUC is the main measurement index, it can be concluded that the MO-EDCR model that employs the Prewitt algorithm is best suited for the image dataset collected at hand. Figure 11 provides examples of crack detection results performed by MO-EDCR using the Prewitt edge detector. Moreover, the processing time of the MO-EDCR model employing the four edge detection algorithms, which is obtained through 20 repeated model runs, is reported in Table 3.

6. Conclusion

This study has constructed an automatic approach for the periodic survey of concrete wall structures. The new approach is aimed at quickly and accurately identifying cracks on the concrete wall surface by analyzing the images captured by digital cameras. The model, named MO-EDCR, consists of three main steps: MF-based noise suppression, edge detection, and cleaning of small objects. Each of these three steps requires hyperparameters to be appropriately set.

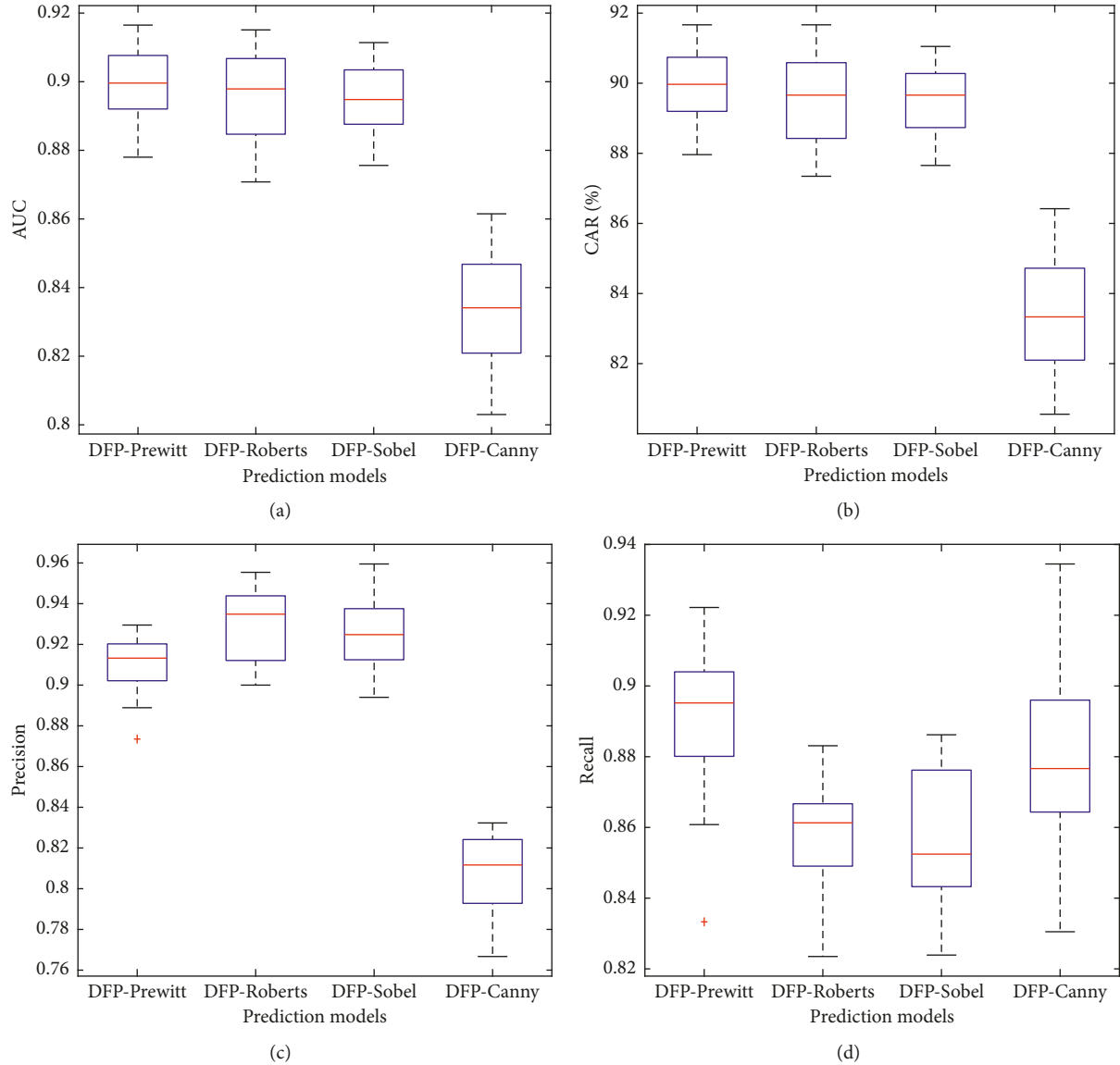


FIGURE 10: Model performance comparison in terms of (a) AUC, (b) CAR, (c) precision, and (d) recall.

The hyperparameter of the first step is the window size of MF.

The free parameter of the second step is the thresholding values of the four edge detection methods (the Roberts, Prewitt, Sobel, and Canny algorithms). Therefore, DFP is used in this research to automatically identify the proper setting for those hyperparameters of MO-EDCR. Experimental results point out that MO-EDCR with the employment of the Prewitt method can help attain the most desired prediction outcome with CAR = 89.954% and AUC = 0.900. With CAR close to 90% and AUC of 0.9, the newly constructed MO-EDCR is a promising alternative to assist maintenance agencies in the tasks of periodic surveys of buildings and infrastructure.

Future developments of the current work may include the investigation of other advanced edge detectors and metaheuristic algorithms to meliorate the prediction

accuracy. Moreover, comparative works that benchmark the performances of metaheuristic optimized edge detectors and deep neural networks can also be promising research directions in the field of image-based crack recognition.

As mentioned earlier, at this current stage of the study, the ground truth labels of samples are considered at an image level. Accordingly, the classification result is considered to be a true positive if the edge detection model can detect a part of a crack object. This is a limitation of the current model since it is beneficial for building maintenance agencies to be capable of detecting the whole crack object. Therefore, in a future work, a more sophisticated image processing model with the capability of recognizing cracks at a pixel level can be constructed to perform deeper analyses on the properties of cracks on the building surface.

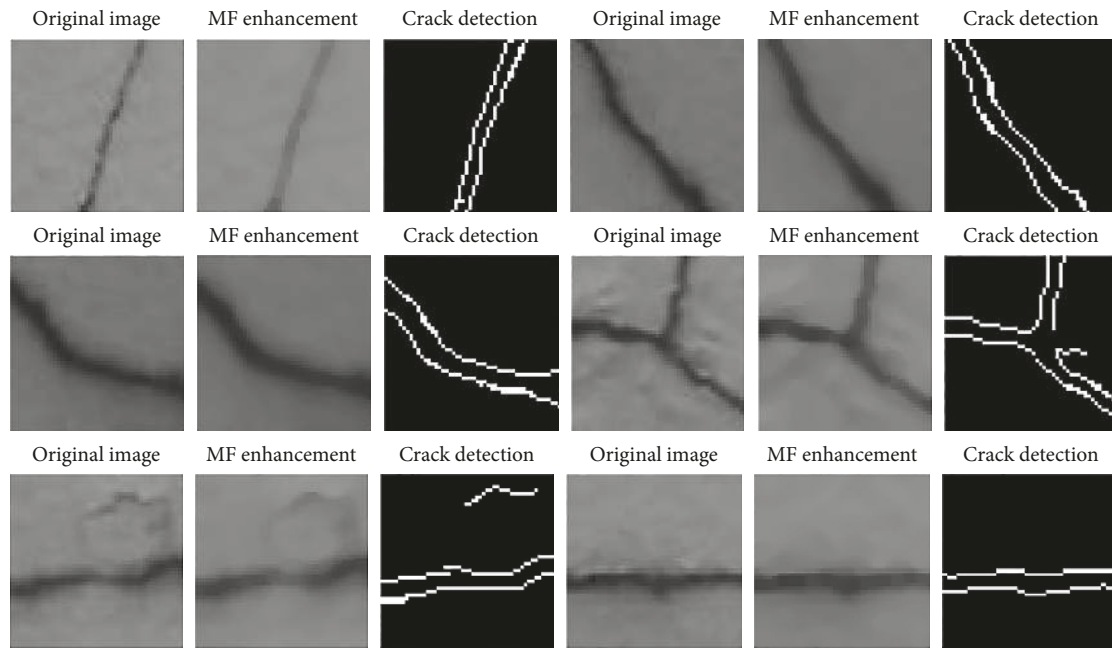


FIGURE 11: Illustrations of crack detection results of MO-EDCR.

TABLE 3: Processing time of the models.

	DFP-Prewitt	DFP-Roberts	DFP-Sobel	DFP-Canny
Processing time (s)	74.69	78.13	85.94	79.69

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this research work.

References

- [1] W. Zhang, Z. Zhang, D. Qi, and Y. Liu, "Automatic crack detection and classification method for subway tunnel safety monitoring," *Sensors*, vol. 14, no. 10, pp. 19307–19328, 2014.
- [2] Y. J. Cha, W. Choi, G. Suh, S. Mahmoudkhani, and O. Büyüköztürk, "Autonomous structural visual inspection using region-based deep learning for detecting multiple damage types," *Computer-Aided Civil and Infrastructure Engineering*, vol. 33, no. 9, pp. 731–747, 2018.
- [3] H. N. Nguyen, T. Y. Nguyen, and D. L. Pham, "Automatic measurement of concrete crack width in 2D multiple-phase images for building safety evaluation," in *Proceedings of Intelligent Information and Database Systems*, pp. 638–648, Springer International Publishing, Cham, Switzerland, February 2018.
- [4] W. R. L. da Silva and D. S. de Lucena, "Concrete cracks detection based on deep learning image classification," *Proceedings*, vol. 2, no. 8, p. 489, 2018.
- [5] Y. S. Yang, C. L. Wu, T. T. C. Hsu, H. C. Yang, H. J. Lu, and C. C. Chang, "Image analysis method for crack distribution and width estimation for reinforced concrete structures," *Automation in Construction*, vol. 91, pp. 120–132, 2018.
- [6] B. Y. Lee, Y. Y. Kim, S. T. Yi, and J. K. Kim, "Automated image processing technique for detecting and analysing concrete surface cracks," *Structure and Infrastructure Engineering*, vol. 9, no. 6, pp. 567–577, 2013.
- [7] S. Dorafshan, R. J. Thomas, and M. Maguire, "Fatigue crack detection using unmanned aerial systems in fracture critical inspection of steel bridges," *Journal of Bridge Engineering*, vol. 23, no. 10, article 04018078, 2018.
- [8] H. Kim, E. Ahn, M. Shin, and S. H. Sim, "Crack and noncrack classification from concrete surface images using machine learning," *Structural Health Monitoring*, article 1475921718768747, 2018.
- [9] C. Koch, K. Georgieva, V. Kasireddy, B. Akinci, and P. Fieguth, "A review on computer vision based defect detection and condition assessment of concrete and asphalt civil infrastructure," *Advanced Engineering Informatics*, vol. 29, no. 2, pp. 196–210, 2015.
- [10] A. Mohan and S. Poobal, "Crack detection using image processing: a critical review and analysis," *Alexandria Engineering Journal*, vol. 57, no. 2, pp. 787–798, 2018.
- [11] C. M. Yeum and S. J. Dyke, "Vision-based automated crack detection for bridge inspection," *Computer-Aided Civil and Infrastructure Engineering*, vol. 30, no. 10, pp. 759–770, 2015.
- [12] N. D. Hoang, "Detection of surface crack in building structures using image processing technique with an improved Otsu method for image thresholding," *Advances in Civil Engineering*, vol. 2018, Article ID 3924120, 10 pages, 2018.
- [13] T. Nishikawa, J. Yoshida, T. Sugiyama, and Y. Fujino, "Concrete crack detection by multiple sequential image filtering," *Computer-Aided Civil and Infrastructure Engineering*, vol. 27, no. 1, pp. 29–47, 2012.
- [14] I. Abdel-Qader, O. Abudayyeh, and M. E. Kelly, "Analysis of edge-detection techniques for crack identification in bridges,"

- Journal of Computing in Civil Engineering*, vol. 17, no. 4, pp. 255–263, 2003.
- [15] L. C. Chen, Y. C. Shao, H. H. Jan, C. W. Huang, and Y. M. Tien, “Measuring system for cracks in concrete using multitemporal images,” *Journal of Surveying Engineering*, vol. 132, no. 2, pp. 77–82, 2006.
 - [16] Z. Chen and T. C. Hutchinson, “Image-based framework for concrete surface crack monitoring and quantification,” *Advances in Civil Engineering*, vol. 2010, Article ID 215295, 18 pages, 2010.
 - [17] R. S. Adhikari, O. Moselhi, and A. Bagchi, “Image-based retrieval of concrete crack properties for bridge inspection,” *Automation in Construction*, vol. 39, pp. 180–194, 2014.
 - [18] S. Dorafshan, “Comparing automated image-based crack detection techniques in spatial and frequency domains,” in *Proceedings of the 26th ASNT Research Symposium*, pp. 34–42, Jacksonville, FL, USA, August 2017.
 - [19] Y. J. Cha and W. Choi, “Vision-based concrete crack detection using a convolutional neural network,” in *Dynamics of Civil Structures*, J. Caicedo and S. Pakzad, Eds., vol. 2 of Conference Proceedings of the Society for Experimental Mechanics Series, pp. 71–73, Springer International Publishing, Cham, Switzerland, 2017.
 - [20] Y. J. Cha, W. Choi, and O. Büyüköztürk, “Deep learning-based crack damage detection using convolutional neural networks,” *Computer-Aided Civil and Infrastructure Engineering*, vol. 32, no. 5, pp. 361–378, 2017.
 - [21] S. Dorafshan, R. J. Thomas, and M. Maguire, “Comparison of deep convolutional neural networks and edge detectors for image-based crack detection in concrete,” *Construction and Building Materials*, vol. 186, pp. 1031–1045, 2018.
 - [22] A. Banharsakun, “Hybrid ABC-ANN for pavement surface distress detection and classification,” *International Journal of Machine Learning and Cybernetics*, vol. 8, no. 2, pp. 699–710, 2017.
 - [23] L. Li, L. Sun, J. Guo, J. Qi, B. Xu, and S. Li, “Modified discrete grey wolf optimizer algorithm for multilevel IMAGE thresholding,” *Computational Intelligence and Neuroscience*, vol. 2017, Article ID 3295769, 16 pages, 2017.
 - [24] O. P. Verma and A. S. Parihar, “An optimal fuzzy system for edge detection in color images using bacterial foraging algorithm,” *IEEE Transactions on Fuzzy Systems*, vol. 25, no. 1, pp. 114–127, 2017.
 - [25] E. Arias-Castro and D. L. Donoho, “Does median filtering truly preserve edges better than linear filtering?,” *Annals of Statistics*, vol. 37, no. 3, pp. 1172–1206, 2009.
 - [26] M. Sonka, V. Hlavac, and R. Boyle, *Image processing, Analysis, and Machine Vision*, Cengage Learning, Boston, MA, USA, 2013, ISBN-10: 1-133-59360-7.
 - [27] D. Sundararajan, *Digital Image Processing—A Signal Processing and Algorithmic Approach*, Springer Nature, Singapore, 2017.
 - [28] L. Roberts, “Machine perception of 3-D solids,” Ph.D. thesis, Massachusetts Institute of Technology, Department of Electrical Engineering, Cambridge, MA, USA, 1965.
 - [29] R. C. Gonzalez, R. E. Woods, and S. L. Eddins, *Digital Image Processing Using Matlab*, Gatesmark Publishing, Knoxville, TN, USA, 2009.
 - [30] J. M. S. Prewitt, “Object enhancement and extraction,” in *Picture Processing and Psychopictorics*, B. Lipkin and A. Rosenfeld, Eds., pp. 75–149, Academic Press, New York, NY, USA, 1970.
 - [31] I. Sobel, “Neighborhood coding of binary images for fast contour following and general binary array processing,” *Computer Graphics and Image Processing*, vol. 8, no. 1, pp. 127–135, 1978.
 - [32] N. D. Hoang, Q. L. Nguyen, and V. D. Tran, “Automatic recognition of asphalt pavement cracks using metaheuristic optimized edge detection algorithms and convolution neural network,” *Automation in Construction*, vol. 94, pp. 203–213, 2018.
 - [33] T. Bose, *Digital Signal and Image Processing*, Wiley, Hoboken, NJ, USA, 2003.
 - [34] J. Canny, “A computational approach to edge detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, no. 6, pp. 679–698, 1986.
 - [35] Mathworks, *Image Processing Toolbox User’s Guide*, MathWorks, Inc., Natick, MA, USA, 2017, https://www.mathworks.com/help/pdf_doc/images/images_tb.pdf.
 - [36] D. P. Acharjya and V. Santhi, *Bio-Inspired Computing for Image and Video Processing*, Chapman and Hall/CRC, Boca Raton, FL, USA, 2018.
 - [37] N. D. Hoang, D. Tien Bui, and K. W. Liao, “Groutability estimation of grouting processes with cement grouts using differential flower pollination optimized support vector machine,” *Applied Soft Computing*, vol. 45, pp. 173–186, 2016.
 - [38] X. S. Yang, *Nature-Inspired Optimization Algorithms*, Elsevier, Amsterdam, Netherlands, 2014.
 - [39] X. S. Yang, “Flower pollination algorithm for global optimization,” in *Unconventional Computation and Natural Computation*, H. Berlin, Ed., pp. 240–249, Springer, Heidelberg, Germany, 2012.
 - [40] K. Price, R. M. Storn, and J. A. Lampinen, *Differential Evolution—A Practical Approach to Global Optimization*, Springer-Verlag, Berlin, Germany, 2005.
 - [41] D. M. W. Powers, “Evaluation: from precision, recall and F-measure to ROC, informedness, markedness & correlation,” *Journal of Machine Learning Technologies*, vol. 2, no. 1, pp. 37–63, 2011.
 - [42] D. T. Bui, H. Shahabi, A. Shirzadi et al., “A novel integrated approach of relevance vector machine optimized by imperialist competitive algorithm for spatial modeling of shallow landslides,” *Remote Sensing*, vol. 10, no. 10, p. 1538, 2018.
 - [43] N. D. Hoang and D. T. Bui, “Predicting earthquake-induced soil liquefaction based on a hybridization of kernel Fisher discriminant analysis and a least squares support vector machine: a multi-dataset study,” *Bulletin of Engineering Geology and the Environment*, vol. 77, no. 1, pp. 191–204, 2018.

