

## Research Article

# A Novel Approach for Automatic Detection of Concrete Surface Voids Using Image Texture Analysis and History-Based Adaptive Differential Evolution Optimized Support Vector Machine

Nhat-Duc Hoang <sup>1,2</sup> and Quoc-Lam Nguyen<sup>2</sup>

<sup>1</sup>Institute of Research and Development, Duy Tan University, Da Nang 550000, Vietnam

<sup>2</sup>Faculty of Civil Engineering, Duy Tan University, Da Nang 550000, Vietnam

Correspondence should be addressed to Nhat-Duc Hoang; [hoangnhatduc@dtu.edu.vn](mailto:hoangnhatduc@dtu.edu.vn)

Received 30 October 2019; Revised 7 July 2020; Accepted 13 July 2020; Published 28 July 2020

Academic Editor: Qiusong Chen

Copyright © 2020 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.

To inspect the quality of concrete structures, surface voids or bugholes existing on a concrete surface after the casting process needs to be detected. To improve the productivity of the inspection work, this study develops a hybrid intelligence approach that combines image texture analysis, machine learning, and metaheuristic optimization. Image texture computations employ the Gabor filter and gray-level run lengths to characterize the condition of a concrete surface. Based on features of image texture, Support Vector Machines (SVM) establish a decision boundary that separates collected image samples into two categories of no surface void (negative class) and surface void (positive class). Furthermore, to assist the SVM model training phase, the state-of-the-art history-based adaptive differential evolution with linear population size reduction (L-SHADE) is utilized. The hybrid intelligence approach, named as L-SHADE-SVM-SVD, has been developed and compiled in Visual C#.NET framework. Experiments with 1000 image samples show that the L-SHADE-SVM-SVD can obtain a high prediction accuracy of roughly 93%. Therefore, the newly developed model can be a promising alternative for construction inspectors in concrete quality assessment.

## 1. Research Background

The construction industry is widely known as a highly competitive environment within which product quality is a crucial element for a contractor's survival [1]. In addition to the project cost and schedule, quality is a key factor that determines customer satisfaction [2]. Typically, for high-rise concrete buildings, architects and project owners impose strict specifications on the condition of the concrete surface. These requirements often involve the delivery of high-quality concrete surface with minimum presence of surface voids or bugholes [3]. It is because the phenomenon of excessive bugholes is one of the most serious and widely encountered defects on a formed concrete surface [4] and is often a subject of dispute between project owners, architects, and construction contractors [3].

Surface voids (see Figure 1) generally refer to small pits and craters on the concrete surface observed after the

process of formwork removal [5]. These defects are brought about by the migration of an entrapped air bubble to the interface between fresh concrete and formwork [6]. The diameter of these voids typically ranges from several millimeters to 15 millimeters and even 25 millimeters in some cases [4]. For more details on the factors leading to the excessive presence of bugholes, readers are guided to the technical paper compiled by ACI [4]. In addition, bugholes are often distributed scatteredly on the concrete surface [7].

A high density of surface voids can result in several harmful effects on the performance of concrete structures:

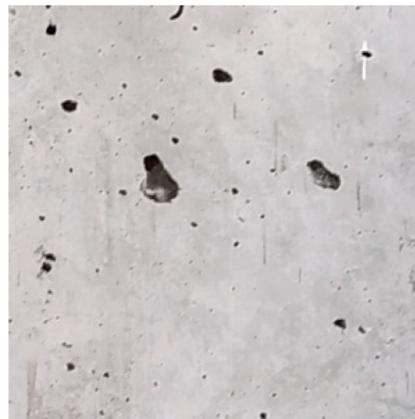
- (i) Bugholes obviously reduce the aesthetics of concrete structures
- (ii) These voids reduce the protective depth of concrete structures and make the reinforcements inside them more vulnerable to corrosion [6]



(a)



(b)



(c)

FIGURE 1: Concrete surface voids.

- (iii) For structures retrofitted by fiber-reinforced plastic (FRP) material, excessive voids reduce the adhesion properties of the FRP material applied to the structure surface [8]
- (iv) Recent works have pointed out that salt accumulated in surface voids can lead to premature degradation of reinforced concrete structures [6, 9]
- (v) Existence of bugholes does increase the cost and time of painting and finishing activities because

additional corrections must be performed to achieve a flat concrete surface [5]

Consequently, project owners and architects demand that the number of surface voids on concrete surfaces should be limited to ensure the aesthetic appearance and durability of concrete structures. Currently, in Vietnam, as well as many other countries, inspection works on uncoated concrete surface are manually performed by human technicians. These procedures rely heavily on domain knowledge and the

experience of inspectors with very limiting assistances of intelligent tools [4, 10–12]. Therefore, the current structure condition assessment process is notoriously time-consuming, laborious, and costly [7, 13–18]. It is due to the sheer volume of surface area needed to be inspected, labeled, and reported. Moreover, as pointed out by Perez et al. [19], the manual procedure also brings safety threat to human inspectors since certain concrete surfaces cannot be reached with ease, especially for those which are located at high/roof levels and narrow space. The current practice is also problematic due to the unavoidable inconsistency in inspecting outcomes. The reason is that the assessment process is dependent of subjective judgment of human technicians [2, 3].

Therefore, project owners are increasingly seeking for fast, effective, and consistent tools to better structure condition assessment [19–24]. The assessment outcomes can also enhance communication between various stakeholders regarding the condition of the buildings. Image processing and machine learning with their fast pace of improvement provide feasible means to achieve such goals. Processed digital images and machine intelligence are capable of automating the concrete surface condition assessment, especially the task of detecting surface voids.

Hence, in the recent years, various state-of-the-art methods relying on these two aforementioned techniques have been proposed to tackle the problem of interest. Zhu and Brilakis [25] put forward an image processing approach employing image segmentation and merging/splitting of pixels to detect air pockets on the concrete surfaces. Santos and Julio [26] presented an approach relying on the digital image processing and laser scanning techniques to analyze the roughness of the substrate surface. Fekri-Ershad and Tajeripour [27] put forward a robust approach based on a one-dimensional local binary pattern for recognizing surface defect. da Silva and Štemberk [28] also employed image processing (i.e., image binarization and morphological analyses) and fuzzy logic to inspect the surface quality of self-consolidating concrete for precast members with a focus on the presence of bugholes. Tajeripour and Fekri-Ershad [29] proposed novel one-dimensional local binary patterns used for recognizing abnormalities in stone textures.

Kwasny et al. [30] investigated the influence of rheology on the quality of surface finish of cement-based mortars; the surface voids existing on concrete were then analyzed and quantified via digital image processing. Sadowski and Mathia [31] pointed out the needs of a more useful method for characterizing properties of a heterogeneous concrete surface; the authors also reviewed various image analyzing tools including image filtering and transformation. The effectiveness of wavelet transform and Gaussian image filtering in detecting surface defects were studied by Goić et al. [32]. Liu and Yang [33] extracted the features of bugholes on a concrete surface via the utilization of the Otsu image binarization method. A texture classification model that incorporates of gray-level run-length matrix and robust illumination normalization techniques has been constructed by Dash and Senapati [34]. Khan et al. [35] relied on ground

penetrating radar to detect water inside the cavities of concrete hollow core slabs.

Yoshitake et al. [36] relied on binary image and color image analyses to detect bugholes distributed on sidewalls and tunnel-lining concrete. Perez et al. [19] utilized the state-of-the-art deep convolutional neural networks (DCNN) for detecting and categorizing building surface defects. A novel instance-level recognition and quantification for concrete surface bughole based on the deep neural network has been recently developed by Wei et al. [18]; this study demonstrates a great capability of machine learning in identifying concrete void surfaces accurately. Nevertheless, the implementation of deep learning models often requires a large number of training samples and a capable computing capability.

As can be seen in the current literature, previous works have mainly relied on image thresholding methods for detecting concrete surface voids. These methods require substantial fine-tuning to adapt to variable characteristics of heterogeneous concrete surface [31]. Due to the complexity of concrete surface background and varying lighting conditions, the accuracy and applicability of image thresholding methods are generally limited. Therefore, image processing techniques should be integrated with advanced machine learning methods to deal with such issues. Hybrid image processing and machine learning tools have demonstrated their outstanding performances in detecting concrete surfaces in previous studies [15, 37–39]. However, too few studies have dedicated to investigating hybrid image processing—machine learning models for the problem of concrete surface void detection. Therefore, the current work is an attempt to fill this gap in the current literature.

In this study, image processing techniques including the Gabor filtering and gray-level run lengths are employed to compute image texture of a concrete surface with and without voids. The texture information is, then, employed by the Support Vector Machines (SVM) to discriminate these two groups of concrete surface. Furthermore, since the training phase of the SVM-based bughole detection model requires a proper setting of the hyperparameters including the penalty coefficient and the kernel function parameter, the history-based adaptive differential evolution with linear population size reduction (L-SHADE) [40, 41] is used to optimize the model training phase. It is because these two hyperparameters strongly influence the learning and predicting performances of the SVM model. If the penalty coefficient and the kernel function parameter are not determined appropriately, the SVM-based surface void detection model cannot deliver the desired due to either overfitting or underfitting phenomena [42, 43].

The subsequent parts of the study are organized in the following order: The second section reviews the research methodology including the employed image processing and computational tools, followed by the third section of the collected image samples; the fourth section presents the proposed hybrid model used for automatic detection of concrete surface voids. The fifth section reports experimental results and concluding remarks of this research are stated in the final section.

## 2. The Employed Image Processing and Computational Intelligence Methods

Since the surface of a concrete structure contains a diverse form of texture (e.g., intact surface, cracks, bugholes, and stains), texture information of an image region needs to be analyzed to support the surface void detection process. Accordingly, a large image is separated into a number of disjoint image samples of  $20 \times 20$  pixels via image cropping operations. Subsequently, numerical features are extracted from these samples to construct the machine learning-based surface void recognition model. In this study, the Gabor filter and gray-level run-length methods are used for feature extraction. A novel method combining the SVM and L-SHADE is employed for data classification.

**2.1. Gabor Filter (GF).** Gabor filters have been widely used for texture segmentation and feature extraction [44, 45]. Due to the capability of optimal joint localization in both spatial and spatial-frequency domains, Gabor filtering is an effective method for recognizing abnormal regions regular textured surfaces [46]. Various successful implementations of this texture discrimination approach have been reported in the literature [45–50]. Basically, a two-dimensional GF is a complex sinusoidal wave modulated by a Gaussian envelope [51]. This filter carries out a localized and oriented frequency analysis of a two-dimensional signal. Mathematically, the response of a GF can be given by the following equation [52]:

$$h(x, y) \exp \left\{ -\frac{1}{2} \left[ \frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2} \right] \right\} \cos(2\pi u_0 x), \quad (1)$$

where  $u_0$  denotes the frequency of a sinusoidal plane wave along the  $x$ -axis.  $\sigma_x$  and  $\sigma_y$  represent the space constants of the Gaussian envelope along the  $x$ - and  $y$ -axis, respectively.

Notably, Gabor filters with different orientations can be attained via a rigid rotation of the  $x$ - $y$  coordinate system [46]. The Fourier transform of the Gabor function described in equation (1) can be expressed as follows [52]:

$$H(u, v) = A \left( \exp \left\{ -\frac{1}{2} \left[ \frac{(u - u_0)^2}{\sigma_u^2} + \frac{v^2}{\sigma_v^2} \right] \right\} + \exp \left\{ -\frac{1}{2} \left[ \frac{(u + u_0)^2}{\sigma_u^2} + \frac{v^2}{\sigma_v^2} \right] \right\} \right), \quad (2)$$

where  $\sigma_u = (1/2)\pi\sigma_x$ ,  $\sigma_v = (1/2)\pi\sigma_y$ , and  $A = 2\pi\sigma_x\sigma_y$ . It is noted that this Fourier transform of the Gabor function determines the amount of each frequency component of the original image that is altered by a GF [50].

Notably, to construct Gabor filters used for texture computation, their tuning parameters including the orientation angles and the radial frequency must be specified. As suggested in the previous work of Jain and Farrokhnia [52],  $0^\circ$ ,  $45^\circ$ ,  $90^\circ$ , and  $135^\circ$  orientations can be used. Moreover, given an image with a width of  $N_w$  pixels and  $N_w$  is a power of 2, the commonly used radial frequency  $u_0$  are as follows:  $1\sqrt{2}$ ,  $2\sqrt{2}$ ,  $4\sqrt{2}$ ,  $\dots$ ,  $(N_w/4)\sqrt{2}$ . Based on the response of the GF operations, statistical measurements can be calculated and employed as features for texture discrimination [53].

**2.2. Gray-Level Run Lengths.** In this study, the task of concrete surface voids is formulated as image texture discrimination. Therefore, information regarding a set of connected image pixels with their distinctive pattern needs to be analyzed. Due to the complex nature of concrete surface, automatic texture discrimination is by no means an easy task. To deal with such challenge, statistical texture analysis models can be employed. Among the statistical models, the gray-level run lengths (GLRL) [54] are very effective to extract information of an image sample based on sizes of homogeneous runs for each gray level [34].

The GLRL was first proposed by Galloway [54] to distill discriminative features from terrain images. This method was, then, applied and improved by various scholars for classifying texture and other tasks in computer vision [34, 55]. This texture computation method relies on the fact that image texture can be considered as a pattern of gray intensity pixel in a particular direction from a reference point [34]. Based on the analysis of second-order statistical information, the GLRL computes the number of gray-level runs which is a collection of linearly adjacent pixels with similar gray intensities.

Given an image sample and a certain direction, a run-length matrix  $p(i, j)$  stores the information regarding the number of times that the sample contains a run length  $j$  of gray level  $i$  [54]. Based on  $p(i, j)$  with different orientations (e.g.  $0^\circ$ ,  $45^\circ$ ,  $90^\circ$ ,  $135^\circ$ ), a variety of texture information can be obtained [56].

Based on a constructed run-length matrix, the Short Run Emphasis (SRE), Long Run Emphasis (LRE), Gray-Level Nonuniformity (GLN), Run-Length Nonuniformity (RLN), and Run Percentage (RP) are computed according to the following equations [54, 57, 58]:

$$\begin{aligned} \text{SRE} &= \frac{1}{N_r} \sum_{i=1}^M \sum_{j=1}^N \frac{p(i, j)}{j^2}, \\ \text{LRE} &= \frac{1}{N_r} \sum_{i=1}^M \sum_{j=1}^N p(i, j) \times j^2, \\ \text{GLN} &= \frac{1}{N_r} \sum_{i=1}^M \left( \sum_{j=1}^N p(i, j) \right)^2, \\ \text{RLN} &= \frac{1}{N_r} \sum_{j=1}^N \left( \sum_{i=1}^M p(i, j) \right)^2, \\ \text{RP} &= \frac{N_r}{N_p}, \end{aligned} \quad (3)$$

where  $M$  and  $N$  are the number of gray levels and the maximum run length.  $N_r$  is the total number of runs, and  $N_p$  denotes the number of pixels in the image.

Furthermore, Chu et al. [55] extended the original GLRL's measurement with the indices of Low Gray-Level Run Emphasis (LGRE) and High Gray-Level Run Emphasis (HGRE):

$$\begin{aligned} \text{LGRE} &= \frac{1}{N_r} \sum_{j=1}^N \sum_{i=1}^M \frac{p(i, j)}{i^2}, \\ \text{HGRE} &= \frac{1}{N_r} \sum_{j=1}^N \sum_{i=1}^M p(i, j) \times i^2. \end{aligned} \quad (4)$$

Additionally, Short Run Low Gray-Level Emphasis (SRLGE), Short Run High Gray-Level Emphasis (SRHGE), Long Run Low Gray-Level Emphasis (LRLGE), and Long Run High Gray-Level Emphasis (LRHGE) are put forward by Dasarathy and Holder [59]:

$$\begin{aligned} \text{SRLGE} &= \frac{1}{N_r} \sum_{j=1}^N \sum_{i=1}^M \frac{p(i, j)}{i^2 \times j^2}, \\ \text{SRHGE} &= \frac{1}{N_r} \sum_{j=1}^N \sum_{i=1}^M \frac{p(i, j) \times i^2}{j^2}, \\ \text{LRLGE} &= \frac{1}{N_r} \sum_{j=1}^N \sum_{i=1}^M \frac{p(i, j) \times j^2}{i^2}, \\ \text{LRHGE} &= \frac{1}{N_r} \sum_{j=1}^N \sum_{i=1}^M p(i, j) \times i^2 \times j^2. \end{aligned} \quad (5)$$

**2.3. The History-Based Adaptive Differential Evolution with Linear Population Size Reduction.** Differential Evolution (DE) [60, 61] is unquestionably a powerful stochastic search for solving numerical optimization. This stochastic search engine relies on a novel integrated mutation-crossover operation to explore and exploit the search space. During the last decade, various enhancements of the standard DE have been put forward to improve its searching performance [62–65].

Among these enhanced DE variants, the history-based adaptive differential evolution with linear population size reduction (L-SHADE) [40, 41] stands out to be a highly successful version with competitive outcomes reported in various comparative studies [66–69]. Therefore, this improved version of the DE is selected in this study to optimize the performance of the machine learning and image processing-based concrete void detection model.

The L-SHADE algorithm, proposed by Tanabe and Fukunaga [41], improves the original DE algorithm via several aspects. First, the mutation scale ( $F$ ) and the crossover probability (CR) are fine-tuned adaptively during the optimization process instead of being fixed. Second, an effective mutation strategy called DE/current-to-pbest/1 is implemented to better explore the search space. Third, a population size shrinking strategy is used to both enhance the convergence rate and reduce computational expense.

The operational flow of the L-SHADE algorithm is tersely presented in Figure 2. Since the L-SHADE inherits the main characteristics of the standard DE, its searching process can also be divided into four steps of population

initialization, mutation, crossover, and selection. In the first step, based on the prespecified searched domain (lower and upper boundaries), the number of decision variable ( $N_{DV}$ ), and an initial number of members ( $N_M$ ), a population of  $N_M$  vectors is randomly created and expected to be distributed evenly throughout the searched space. In the second step, a new candidate solution called a mutated vector is generated via the DE/current-to-pbest/1 strategy. The DE/current-to-pbest/1 strategy used for generating a new candidate solution is given by [40]

$$v_{i,g+1} = x_{i,g} + F_i(x_{r1,g} - x_{r2,g}) + F_i(x_{pbest,g} - x_{i,g}). \quad (6)$$

In the third step, the crossover operation is used to combine the information of the newly created candidate and its parent according to the following manner [60]:

$$u_{j,i,g+1} = \begin{cases} v_{j,i,g+1}, & \text{if } \text{rand}_j \leq \text{Cr or } j = \text{rnb}(i), \\ x_{j,i,g}, & \text{if } \text{rand}_j > \text{Cr and } j \neq \text{rnb}(i). \end{cases} \quad (7)$$

In the last step, a greedy selection which compares the fitness of the candidate solution and its parent is carried out. It is noted that the L-SHADE employs archives of MF and MCR which are vectors of a fixed length  $H$ ; these two archives store the mean values of the mutation scale and the crossover probability. Moreover, the two sets of SF and SCR store all CR and  $F$  values that helped to yield child solutions better than the parents. After each generation, the current population size reduces via the removal of inferior solutions [41].

**2.4. Support Vector Machine.** Support Vector Machines (SVM), constructed on the basis of the statistical learning theory, are a robust method for establishing pattern classification models. Introduced by Vapnik [70], the SVM have gained popularity in the research community via various works which reported their successful implementations [71–73]. It is because this machine learning method features significant advantages including resilience to noisy data via a framework of maximum margin construction and capability of handling nonlinearly separable data by means of kernel tricks. Furthermore, the learning phase of the SVM is bolt down to solving a convex optimization problem; this guarantees a global convergence and avoids being trapped in local optima [74].

The concept of the SVM used for concrete surface void detection is demonstrated in Figure 3. The model deals with nonlinearly separable data by mapping the data from the original input space to a high-dimensional feature space; in such high-dimensional feature space, linear separability can become feasible.

Given a training dataset  $\{x_k, y_k\}_{k=1}^N$  with a numerical feature  $x_k \in R^n$  and corresponding class categories  $y_k \in \{-1, +1\}$ , an SVM model establishes a classification boundary to distinguish data from a positive class +1 (surface void) and a negative class -1 (nonsurface void). It is noted that a numerical feature  $x_k$  is actually texture information extracted from an image sample using the Gabor filter and the GLRL. To establish such classification

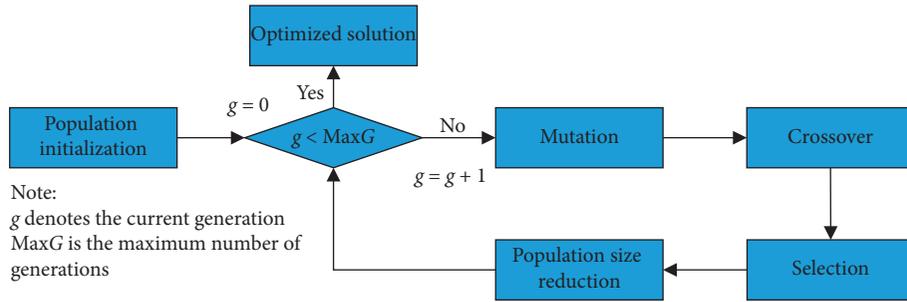


FIGURE 2: The operational flow of the L-SHADE algorithm.

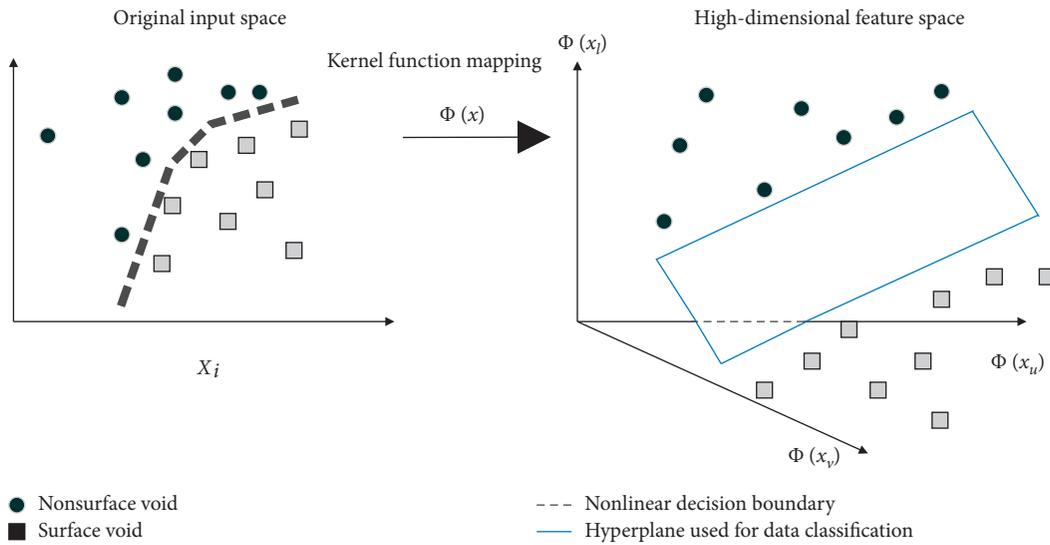


FIGURE 3: Demonstration of an SVM model.

boundary, it is required to solve the following nonlinear programming problem [43]:

$$\text{Minimize } J_p(w, e) = \frac{1}{2}w^T w + c \frac{1}{2} \sum_{k=1}^N e_k^2$$

$$\text{Subjected to } y_k(w^T \varphi(x_k) + b) \geq 1 - e_k, \quad k = 1, \dots, N, \quad e_k \geq 0, \quad (8)$$

where  $w \in R^n$  is a normal vector to the classification hyperplane and  $b \in R$  represents the model bias;  $e_k > 0$  is slack variables;  $c$  represents a penalty constant; and  $\varphi(x)$  denotes the nonlinear data mapping used for dealing with nonlinear separable data.

It is noted that the SVM does not necessitate an explicit expression of the data mapping function  $\varphi(x)$ . The quantity of interest is the product of  $\varphi(x)$  in the input space which is defined as a Kernel function:

$$K(x_k, x_l) = \varphi(x_k)^T \varphi(x_l). \quad (9)$$

The Radial Basis Function Kernel (RBFK) is often employed and it is shown as follows:

$$K(x_k, x_l) = \exp\left(-\frac{\|x_k - x_l\|^2}{2\sigma^2}\right), \quad (10)$$

where  $\sigma$  represents a tuning parameter of the RBFK.

After solving the aforementioned nonlinear programming, the SVM model used for data classification can be tersely presented in the following equation:

$$y(x_l) = \text{sign}\left(\sum_{k=1}^{SV} \alpha_k y_k K(x_k, x_l) + b\right), \quad (11)$$

where  $\alpha_k$  denotes the solution of the dual form of the aforementioned nonlinear programming. SV represents the number of support vectors (the number of  $\alpha_k > 0$ ).

### 3. The Image Samples of the Concrete Surface

To construct the SVM machine learning model used for concrete surface void recognition, the set of image samples capturing the texture of concrete structures must be prepared. This image set includes samples which contain concrete bugholes and samples without such defect. Accordingly, a set of 1000 image samples with assigned ground

truth categories has been manually collected via field trip to several construction sites in Danang city (Vietnam). To guarantee a balanced dataset, the numbers of the negative (without surface voids) and positive (having surface voids) samples are both 500. The categories of image samples have been determined by human inspectors. Herein, the label = -1 means the negative class and the label = 1 denotes a positive class. It is noted that the collected images in this study have been taken by using the Cannon EOS M10 (CMOS 18.0 MP) and Nikon D5100 (CMOS 16.2 MP). To enhance the speed of the texture computation phase and to ensure the consistency of an image region, the image size has been set to be 20x20 pixels. The image samples are illustrated in Figure 4. Additionally, to better cope with the diversity of the concrete surface, the negative class of nonsurface void deliberately includes samples of sound concrete surface, cracks, and stains.

#### 4. The Proposed Hybridization of Image Processing, Machine Learning, and Metaheuristic for Detecting Concrete Surface Voids

The proposed hybridization of image processing, machine learning, and metaheuristic optimization used for detection of concrete surface voids is presented in this section of the study. The integrated model is denoted as L-SHADE-SVM-SVD. The overall model structure is graphically summarized in Figure 5. The L-SHADE-SVM-SVD model is developed in the Visual C#.NET environment (Framework 4.6.2) and run in ASUS FX705GE—EW165T (Core i7 8750H, 8 GB Ram, 256 GB solid-state drive).

The L-SHADE-SVM-SVD operation can be divided into three steps:

- (i) Image texture computation: the step computes texture information of concrete surface obtained from image samples stored in training and testing subsets. The first subset includes 90% of the collected samples and is used for model construction. The second set occupies 10% of the collected samples and is reserved for validating the model predictive capability. Image texture including the Gabor filter and GLRL is computed and used as the numerical feature.
- (ii) The L-SHADE metaheuristic optimization: as mentioned earlier, the model training and pattern classification phases of the SVM require appropriate values of the penalty coefficient ( $c$ ) and the kernel function parameter ( $\sigma$ ). The former hyperparameter dictates how the loss function of the SVM increases due to misclassified data points. The latter hyperparameter affects the smoothness of the decision boundary. Therefore, these hyperparameters strongly influence the learning and predictive performance of the SVM-based bughole detection model. The selection of the penalty coefficient ( $c$ ) and the kernel function parameter ( $\sigma$ ) can be

formulated as an optimization task within which metaheuristic algorithms can be employed. Based on the previous comparative works [41, 67, 68, 75], this study employs the L-SHADE metaheuristic algorithm for conducting the SVR model optimization. This DE variant first generates an initial population of hyperparameters in a random manner. In each generation, the optimization algorithm explores and exploits the search space to gradually guide the population to a better solution representing SVM models with good predictive capability.

- (iii) The SVM-based pattern classification: based on the optimized solution of the model hyperparameters, the SVM model is employed to construct a decision surface that is capable of distinguish concrete surface with and without voids. Notably, the SVM model is constructed via a built-in function supported by the Accord.NET Framework [76].

It is also noted that, to optimize the SVM model performance, a  $K$ -fold cross validation (with  $K=5$ ) is used. Using this cross validation, the whole dataset is separated into 5 mutual exclusive subsets. In each of the five runs, one subset is used for model testing and the other subsets are employed for model training. The average predictive performance is used to quantify the model generalization capability. Accordingly, the following cost function is used by the L-SHADE-SVM-SVD:

$$CF = \frac{\sum_{k=1}^K (FNR_k + FPR_k)}{K}, \quad (12)$$

where  $FNR_k$  and  $FPR_k$  denote the false negative rate (FNR) and the false positive rate (FPR) obtained from  $k$ th run, respectively.

The FNR and FPR indices are computed as follows:

$$\begin{aligned} FNR &= \frac{FN}{FN + TP}, \\ FPR &= \frac{FP}{FP + TN}, \end{aligned} \quad (13)$$

where FN, FP, TP, and TN are false negative, false positive, true positive, and true negative data samples, respectively.

Moreover, it is noted that, to compute the Gabor filter, it is necessary to convert the original image to grayscale ones. Based on the recommendations of Jain and Farrokhnia [52], four values of orientations ( $0^\circ$ ,  $45^\circ$ ,  $90^\circ$ , and  $135^\circ$ ) and four values of radial frequency ( $1\sqrt{2}$ ,  $2\sqrt{2}$ ,  $4\sqrt{2}$ ,  $8\sqrt{2}$ ) have been employed to calculate the Gabor filtering features. In this study, the Gabor filters are implemented with the assistance of built-in functions provided by the Accord.NET Framework [76]. Based on the filtered image, the following statistical indices can be measured to characterize image texture:

- (i) The mean of the Gabor filter response:

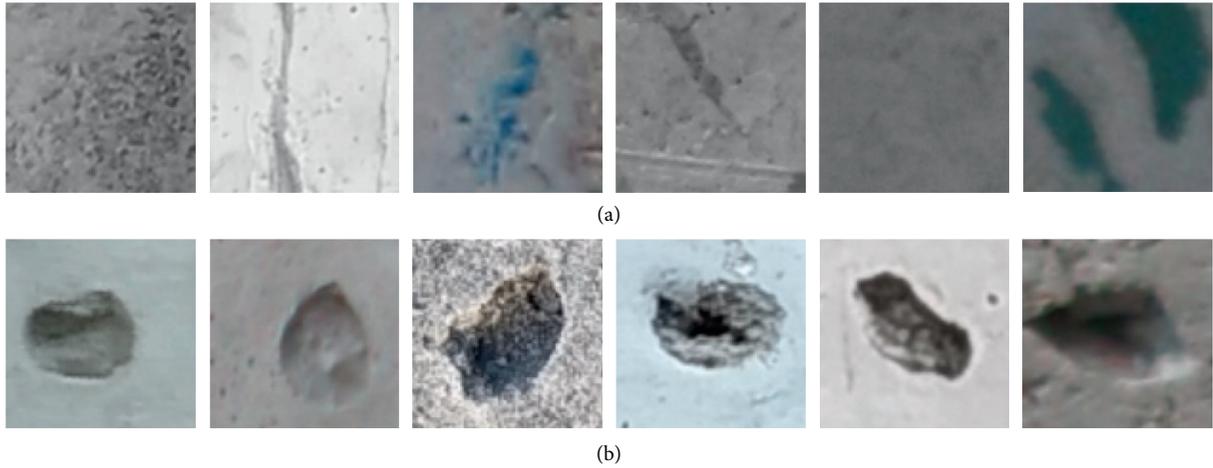


FIGURE 4: The collected image samples: (a) nonsurface void and (b) surface void.

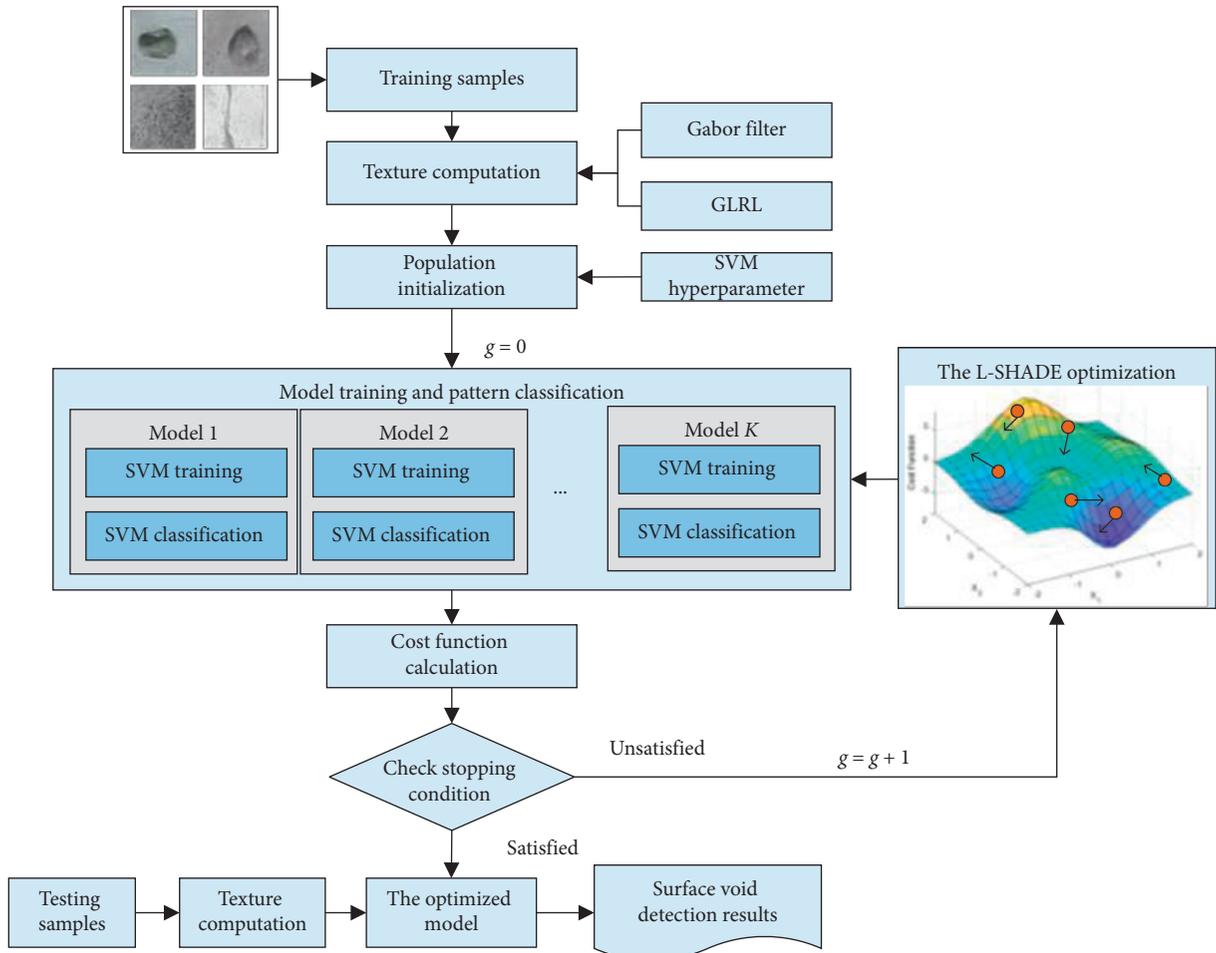


FIGURE 5: The proposed L-SHADE-SVM-SVD used for automatic detection of concrete surface voids.

$$\text{Mean}_{\text{GaborFilter}} = \frac{\sum_{i=0}^{H_{IM}-1} \sum_{j=0}^{W_{IM}-1} \text{GFR}(i, j)}{W_{IM} \times H_{IM}}, \quad (14)$$

where  $H_{IM}$  and  $W_{IM}$  are the width and height of an image sample, respectively.  $\text{GFR}(i, j)$  denotes the Gabor filter response at a pixel  $(i, j)$ .

(ii) The standard deviation of the Gabor filter response:

$$\text{STD}_{\text{GaborFilter}} = \frac{\sum_{i=0}^{H_{\text{IM}}-1} \sum_{j=0}^{W_{\text{IM}}-1} [\text{GFR}(i, j) - \text{Mean}_{\text{GaborFilter}}]^2}{W_{\text{IM}} \times H_{\text{IM}}} \quad (15)$$

$$\text{Skewness}_{\text{GaborFilter}} = \frac{(1/(W_{\text{IM}} \times H_{\text{IM}})) \sum_{i=0}^{H_{\text{IM}}-1} \sum_{j=0}^{W_{\text{IM}}-1} [\text{GFR}(i, j) - \text{Mean}_{\text{GaborFilter}}]^3}{[(1/(W_{\text{IM}} \times H_{\text{IM}} - 1)) \sum_{i=0}^{H_{\text{IM}}-1} \sum_{j=0}^{W_{\text{IM}}-1} \{\text{GFR}(i, j) - \text{Mean}_{\text{GaborFilter}}\}^2]^{3/2}} \quad (16)$$

(iv) The entropy of the Gabor filter response:

$$\text{Entropy}_{\text{GaborFilter}} = - \sum_{i=0}^{\text{NDV}-1} \text{FOH}_{\text{Filter}} \times \log_2 [P_{\text{Filter}}], \quad (17)$$

where  $\text{FOH}_{\text{Filter}}$  represents the first-order histogram of the Gabor filter response. Moreover,  $\text{NDV} = 256$  denotes the number of discrete gray intensity values for an 8 bit grayscale image.

Because 16 Gabor filters are employed for computing image texture and each filter has four statistical indices (the mean, standard deviation, skewness, and entropy), the number of Gabor filtering features is 64. Moreover, since there are four orientations ( $0^\circ$ ,  $45^\circ$ ,  $90^\circ$ , and  $135^\circ$ ) used for GLRL matrix construction and each GLRL matrix has 11 features, the number of GLRL features is 44 [57]. Accordingly, the total number of features used for the SVM pattern classification is  $64 + 44 = 108$ .

Additionally, to facilitate the data classification based on the SVM, the established dataset has been normalized by the Z-score equation given by

$$X_{\text{ZN}} = \frac{X_o - m_X}{s_X}, \quad (18)$$

where  $X_o$  and  $X_{\text{ZN}}$  represent the original and the standardized input feature, respectively.  $m_X$  and  $s_X$  denote the mean and the standard deviation of the original input feature, respectively.

## 5. Experimental Result and Comparison

As mentioned earlier, to train and validate the hybrid model used for concrete surface void detection, the collected dataset has been divided into two packages of training and testing datasets. The training dataset (90% of the original dataset) is used for model construction, and the testing dataset is reserved for evaluating the model performance when predicting novel image samples. Moreover, to diminish the effect of randomness in data sampling and to reliably assess the predictive capability of the newly developed method, the training/testing data sampling has been performed 20 times. In each time of running, 10% of the original dataset is randomly drawn out to form the testing dataset; the rest of the original dataset is used for model

(iii) The skewness of the Gabor filter response [77]:

construction. The datasets used for time of model running are illustrated in Table 1. Herein, the column of sample index expresses the number of data samples employed in the training and testing datasets.

In addition, to quantify the predictive capability of the proposed L-SHADE-SVM-SVD, the classification accuracy rate (CAR), precision, recall, negative predictive value (NPV), and F1 score are computed from the four basic results of TP, TN, FP, and FN. These performance indicators are presented as follows [78]:

$$\begin{aligned} \text{CAR} &= \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \times 100\%, \\ \text{Precision} &= \frac{\text{TP}}{\text{TP} + \text{FP}}, \\ \text{Recall} &= \frac{\text{TP}}{\text{TP} + \text{FN}}, \\ \text{NPV} &= \frac{\text{TN}}{\text{TN} + \text{FN}}, \\ \text{F1 Score} &= \frac{2\text{TP}}{2\text{TP} + \text{FP} + \text{FN}}. \end{aligned} \quad (19)$$

The L-SHADE with an initial population size of 30 and a maximum number of searching generations of 100 was utilized to seek for the most appropriate set of the SVM model's hyperparameters. The searching progress of the L-SHADE metaheuristic is graphically presented in Figure 6. The optimization process yields the penalty parameter ( $c$ ) = 3.712 and the kernel function parameter ( $\sigma$ ) = 6.031. In addition, the statistical outcomes of the L-SHADE-SVM-SVD (mean and standard deviation) using the hyperparameters found by the L-SHADE metaheuristic are reported in Table 2. The averaging values of the CAR, precision, recall, NPV, and F1 score are 92.600%, 0.911, 0.942, 0.940, and 0.926, respectively. In addition, the average runtime of the proposed approach is roughly 13.30 seconds. Since there are 100 data samples in the testing set, the computational time used for classifying one testing data sample is approximately 0.13 seconds.

Furthermore, to demonstrate the predictive ability of the newly developed L-SHADE-SVM-SVD model used for concrete surface void recognition, the model performance has been compared to those of Deep Convolution Neural

TABLE 1: The training and testing datasets.

Datasets	Sample index	Features							Class label
		F1	F2	F3	...	F106	F107	F108	
Training	1	108.115	33.580	-15.236	...	6866.946	11.141	658.625	-1
	2	109.645	33.763	-18.768	...	5026.965	11.007	381.245	-1
	...	...	...	...	...	...	...	...	...
	899	106.063	32.555	-14.066	...	32167.854	10.600	640.701	1
	900	108.305	33.135	-13.449	...	31379.524	10.674	1039.198	1
Testing	1	108.435	32.954	-16.630	...	9425.744	11.802	2016.670	-1
	2	105.565	30.844	-16.665	...	8383.504	11.505	1440.862	-1
	...	...	...	...	...	...	...	...	...
	99	107.228	32.628	-12.087	...	30558.259	10.551	631.624	1
	100	107.223	32.418	-13.006	...	30292.417	10.575	514.372	1

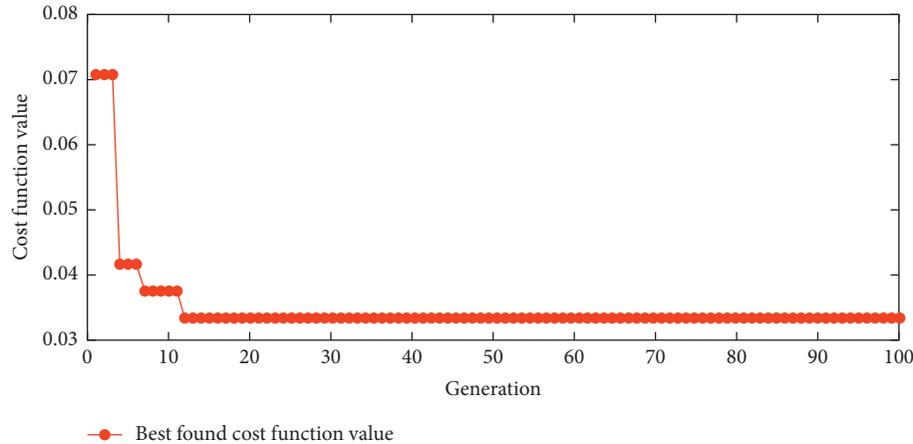


FIGURE 6: The L-SHADE optimization process.

TABLE 2: Prediction performance of the L-SHADE-SVM-SVD.

Phases	Indices	CAR (%)	TP	TN	FP	FN	Precision	Recall	NPV	F1 score
Training	Mean	97.417	437.200	439.550	9.550	13.700	0.979	0.970	0.970	0.974
	Std.	0.222	4.938	5.753	1.605	1.302	0.004	0.003	0.003	0.002
Testing	Mean	92.600	46.200	46.400	4.500	2.900	0.911	0.942	0.940	0.926
	Std.	2.761	4.873	5.305	1.878	2.125	0.037	0.040	0.044	0.027

Network (DCNN) models [79–81] with stochastic gradient descent with momentum (Sgdm) [82], Adaptive Moment Estimation (Adam) [83], and root mean square propagation (Rmsprop) [84]. The three DCNN models are denoted as DCNN-Sgdm, DCNN-Adam, and DCNN-Rmsprop and are implemented via the MATLAB deep learning toolbox [85]. Via a trial-and-error process, appropriate configurations of the DCCN model are selected and shown in Table 3.

Moreover, the minibatch Backpropagation Neural Network (MB-BPNN) [86, 87] is also employed as a benchmark method. The MB-BPNN model has been developed in Visual C#.NET by the authors and trained with the minibatch mode [87, 88]; the batch-size is selected to be 32, and the number of neurons in the hidden layer is set to be  $(2/3)D_X + C_N$ , as suggested by Heaton [89];  $D_X$  and  $C_N$  denote the numbers of features and outputs, respectively. The MB-BPNN model is, then, trained with the sigmoidal

TABLE 3: Configurations of the DCNN models.

CNN layers	Convolution layers		Pooling layers
	Filter number	Filter size	Filter size
1	36	10	2
2	36	8	2
3	36	4	2
4	36	2	2

activation function with the maximum number of epochs = 1000 epochs and the learning rate = 0.01.

The prediction results of the proposed L-SHADE-SVM-SVD model, as well as the benchmark models, are summarized in Table 4 and graphically presented as box plots in Figure 7. As can be seen from the prediction results, the performance of the L-SHADE-SVM-SVD (CAR = 92.600%, Precision = 0.911, Recall = 0.942, NPV = 0.940, and F1 score = 0.926) is better than that of the DCNN-Rmsprop

TABLE 4: Prediction result comparison.

Phase	Indices	L-SHADE-SVM-SVD		DCNN-Rmsprop		DCNN-Adam		DCNN-Sgdm		MB-BPNN	
		Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.
Training	CAR (%)	97.417	0.222	87.922	2.585	89.211	1.937	87.756	6.165	86.763	6.889
	TP	437.200	4.938	369.000	25.984	390.250	17.693	375.650	40.359	333.900	63.570
	TN	439.550	5.753	422.300	20.683	412.650	21.755	414.150	21.866	360.200	9.540
	FP	9.550	1.605	27.700	20.683	37.350	21.755	35.850	21.866	66.100	63.570
	FN	13.700	1.302	81.000	25.984	59.750	17.693	74.350	40.359	39.800	9.540
	Precision	0.979	0.004	0.934	0.044	0.916	0.042	0.912	0.064	0.835	0.159
	Recall	0.970	0.003	0.820	0.058	0.867	0.039	0.835	0.090	0.895	0.012
	NPV	0.970	0.003	0.842	0.041	0.875	0.029	0.851	0.065	0.901	0.024
F1 score	0.974	0.002	0.871	0.030	0.889	0.019	0.870	0.075	0.852	0.132	
Testing	CAR (%)	92.600	2.761	88.350	3.133	86.900	4.204	86.800	6.178	85.700	7.248
	TP	46.200	4.873	41.700	3.246	42.200	3.792	40.900	4.767	40.900	8.130
	TN	46.400	5.305	46.650	2.323	44.700	3.326	45.900	2.674	44.800	2.587
	FP	4.500	1.878	3.350	2.323	5.300	3.326	4.100	2.674	9.100	8.130
	FN	2.900	2.125	8.300	3.246	7.800	3.792	9.100	4.767	5.200	2.587
	Precision	0.911	0.037	0.929	0.045	0.893	0.057	0.909	0.065	0.818	0.163
	Recall	0.942	0.040	0.834	0.065	0.844	0.076	0.818	0.095	0.890	0.042
	NPV	0.940	0.044	0.852	0.049	0.856	0.059	0.839	0.067	0.896	0.052
F1 score	0.926	0.027	0.877	0.036	0.865	0.044	0.859	0.076	0.839	0.137	

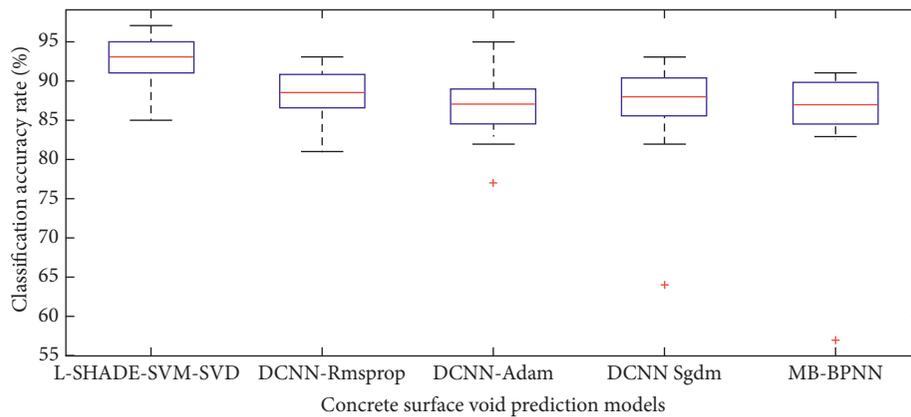


FIGURE 7: Box plots of model performances.

(CAR = 88.350%, Precision = 0.929, Recall = 0.834, NPV = 0.852, and F1 score = 0.877), DCNN-Adam (CAR = 86.900%, Precision = 0.893, Recall = 0.844, NPV = 0.856, and F1 score = 0.865), DCNN-Sgdm (CAR = 86.800%, Precision = 0.909, Recall = 0.818, NPV = 0.839, and F1 score = 0.859), and MB-BPNN (CAR = 85.700%, Precision = 0.818, Recall = 0.890, NPV = 0.896, and F1 score = 0.839).

In addition, the two-sample  $t$ -test [90] is utilized in this study to better confirm the statistical significance of the model predictive capabilities. This statistical test is often employed to inspect the null hypothesis that the model prediction performances of two machine learning models may be drawn from normal distributions with equal means. Herein, the significant level ( $p$  value) of the test is set to be 0.05, and the results of the  $t$ -test are reported in Table 5. As can be observed from this table, the  $p$  values  $< 0.05$  reliably reject the null hypothesis. This fact confirms that the proposed L-SHADE-SVM-SVD is best suited for the task of detecting concrete surface bugholes.

TABLE 5: The  $t$ -test outcomes of pairwise model comparisons.

Model comparison	Test outcome	$p$ value
L-SHADE-SVM-SVD vs. DBNN-Rmsprop	Significant	0.00005
L-SHADE-SVM-SVD vs. DBNN-Adam	Significant	0.00001
L-SHADE-SVM-SVD vs. DBNN-Sgdm	Significant	0.00046
L-SHADE-SVM-SVD vs. MB-BPNN	Significant	0.00030

## 6. Conclusions

Detection of a concrete surface is crucial for inspecting quality of cast-in-place concrete elements. To enhance the productivity and eliminate subjective judgment of concrete quality inspection works, this research proposes an intelligent method that hybridizes state-of-the-art image processing, machine learning, and metaheuristic methods. Image texture is used as the input feature that characterizes the states of a concrete surface. Gabor filter and GLRL-based texture information is computed and employed by the SVM

to construct a decision boundary that divides the input into two categories of no surface void (negative class) and surface void (positive class).

In addition, to optimize the SVM model training phase, the L-SHADE metaheuristic is used. This metaheuristic algorithm autonomously searched for the most appropriate set of the SVM model's hyperparameters including the penalty coefficient and the kernel function parameter. The integrated model, named as L-SHADE-SVM-SVD, has been developed and compiled in the Visual C#.NET framework to ease its implementation. The experimental outcome using 1000 image samples and a repeated data sampling with 20 runs demonstrate that the newly developed L-SHADE-SVM-SVD is able to attain good predictive performances (CAR = 92.600%, Precision = 0.911, Recall = 0.942, NPV = 0.940, and F1 score = 0.926). Thus, the L-SHADE-SVM-SVD can be a helpful tool to assist construction inspectors in assessing concrete surface quality.

Nevertheless, since the current L-SHADE-SVM-SVD model aims at recognizing the status of no surface void and surface void, the task of localizing surface voids on a concrete surface image can be performed in a future study. In addition, other future directions of the current works may include the extension of the collected image dataset to enhance the generalization of the developed machine learning model, investigation of other advanced metaheuristic algorithms to improve the surface void detection performance, and employment of other performance measurements such as model runtime [91].

## Data Availability

The image dataset and the compiled L-SHADE-SVM-SVD program used to support the findings of this study have been deposited in the repository of github (<https://github.com/NhatDucHoang/L-SHADE-SVM-SVD>).

## Conflicts of Interest

The authors confirm that there are no conflicts of interest.

## Acknowledgments

This research was financially supported by Duy Tan University.

## References

- [1] D. W. Halpin and B. A. Senior, *Construction Management*, Wiley, Hoboken, NJ, USA, 4th edition, 2010.
- [2] C. Laofor and V. Peansupap, "Defect detection and quantification system to support subjective visual quality inspection via a digital image processing: a tiling work case study," *Automation in Construction*, vol. 24, pp. 160–174, 2012.
- [3] G. Lemaire, G. Escadeillas, and E. Ringot, "Evaluating concrete surfaces using an image analysis process," *Construction and Building Materials*, vol. 19, no. 8, pp. 604–611, 2005.
- [4] ACI, *Identification and Control of Visible Effects of Consolidation on Formed Concrete Surfaces* American Concrete, ACI, Naples, FL, USA, 2003, <https://www.concrete.org/publications/internationalconcreteabstractsportal/m/details/id/5145>.
- [5] T. Ozkul and I. Kucuk, "Design and optimization of an instrument for measuring bughole rating of concrete surfaces," *Journal of the Franklin Institute*, vol. 348, no. 7, pp. 1377–1392, 2011.
- [6] B. Liu, T. Yang, and Y. Xie, "Factors influencing bugholes on concrete surface analyzed by image processing technology," *Construction and Building Materials*, vol. 153, pp. 897–907, 2017.
- [7] G. Yao, F. Wei, Y. Yang, and Y. Sun, "Deep-learning-based bughole detection for concrete surface image," *Advances in Civil Engineering*, vol. 2019, Article ID 8582963, 12 pages, 2019.
- [8] A. S. Kalayci, B. Yalim, and A. Mirmiran, "Effect of untreated surface disbonds on performance of FRP-retrofitted concrete beams," *Journal of Composites for Construction*, vol. 13, no. 6, pp. 476–485, 2009.
- [9] F. Benito Saorin, I. Miñano Belmonte, C. Parra Costa, C. Rodriguez Lopez, and M. Valcuende Paya, "QSI methods for determining the quality of the surface finish of concrete," *Sustainability*, vol. 10, no. 4, p. 931, 2018.
- [10] ACI, *ACI Manual of Concrete Inspection ACI Committee 311, SP-2(07)*, American Concrete Institute, Detroit, MI, USA, 2007.
- [11] CIB, *Report N. 24: Tolerances on Blemishes of Concrete* Concrete International Board-Commission W29, CIB, Cairo, Egypt, 1973, <https://www.wirbnet.de/daten/iconda/06059000983.pdf>.
- [12] M. S. Thompson, "Blowholes in concrete surface concrete," 1970, [https://www.concreteconstruction.net/products/decorative-concrete-surfaces/blowholes-in-concrete-surfaces\\_o](https://www.concreteconstruction.net/products/decorative-concrete-surfaces/blowholes-in-concrete-surfaces_o).
- [13] M. Gao, X. Wang, S. Zhu, and P. Guan, "Detection and segmentation of cement concrete pavement pothole based on image processing technology," *Mathematical Problems in Engineering*, vol. 2020, Article ID 1360832, 13 pages, 2020.
- [14] N.-D. Hoang, "Image processing-based recognition of wall defects using machine learning approaches and steerable filters," *Computational Intelligence and Neuroscience*, vol. 2018, Article ID 7913952, 18 pages, 2018.
- [15] H. Kim, E. Ahn, M. Shin, and S.-H. Sim, "Crack and noncrack classification from concrete surface images using machine learning," *Structural Health Monitoring*, vol. 18, no. 3, pp. 725–738, 2018.
- [16] S. Li and X. Zhao, "Image-based concrete crack detection using convolutional neural network and exhaustive search technique," *Advances in Civil Engineering*, vol. 2019, Article ID 6520620, 12 pages, 2019.
- [17] T. Qingguo, L. Qijun, B. Ge, and Y. Li, "A methodology framework for retrieval of concrete surface crack's image properties based on hybrid model," *Optik*, vol. 180, pp. 199–214, 2019.
- [18] F. Wei, G. Yao, Y. Yang, and Y. Sun, "Instance-level recognition and quantification for concrete surface bughole based on deep learning," *Automation in Construction*, vol. 107, Article ID 102920, 2019.
- [19] H. Perez, J. H. M. Tah, and A. Mosavi, "Deep learning for detecting building defects using convolutional neural networks," *Sensors*, vol. 19, no. 16, Article ID 3556, 2019.
- [20] P. Asadi, M. Gindy, and M. Alvarez, "A machine learning based approach for automatic rebar detection and quantification of deterioration in concrete bridge deck ground

- penetrating radar B-scan images,” *KSCE Journal of Civil Engineering*, vol. 23, no. 6, pp. 2618–2627, 2019.
- [21] P. C. Fonseca and G. W. Scherer, “An image analysis procedure to quantify the air void system of mortar and concrete,” *Materials and Structures*, vol. 48, no. 10, pp. 3087–3098, 2015.
- [22] Y. Huang, X. He, Q. Wang, and J. Xiao, “Deformation field and crack analyses of concrete using digital image correlation method,” *Frontiers of Structural and Civil Engineering*, vol. 13, no. 5, pp. 1183–1199, 2019.
- [23] W. Song, G. Jia, H. Zhu, D. Jia, and L. Gao, “Automated pavement crack damage detection using deep multiscale convolutional features,” *Journal of Advanced Transportation*, vol. 2020, Article ID 6412562, 11 pages, 2020.
- [24] I. Venanzi, R. Castellani, L. Ierimonti, and F. Ubertini, “An automated procedure for assessing local reliability index and life-cycle cost of alternative girder bridge design solutions,” *Advances in Civil Engineering*, vol. 2019, Article ID 5152031, 17 pages, 2019.
- [25] Z. Zhu and I. Brilakis, “Machine vision-based concrete surface quality assessment,” *Journal of Construction Engineering and Management*, vol. 136, no. 2, pp. 210–218, 2010.
- [26] P. M. D. Santos and E. N. B. S. Julio, “Comparison of methods for texture assessment of concrete surfaces,” *ACI Materials Journal*, vol. 107, no. 5, 2010.
- [27] S. Fekri-Ershad and F. Tajeripour, “A robust approach for surface defect detection based on one dimensional local binary patterns,” *Indian Journal of Science and Technology*, vol. 5, no. 8, pp. 1–7, 2012.
- [28] W. R. L. da Silva and P. Štemberk, “Expert system applied for classifying self-compacting concrete surface finish,” *Advances in Engineering Software*, vol. 64, pp. 47–61, 2013.
- [29] F. Tajeripour and S. Fekri-Ershad, “Developing a novel approach for stone porosity computing using modified local binary patterns and single scale retinex,” *Arabian Journal for Science and Engineering*, vol. 39, no. 2, pp. 875–889, 2014.
- [30] J. Kwasny, M. Sonebi, J. Plasse, and S. Amziane, “Influence of rheology on the quality of surface finish of cement-based mortars,” *Construction and Building Materials*, vol. 89, pp. 102–109, 2015.
- [31] Ł. Sadowski and T. G. Mathia, “Multi-scale metrology of concrete surface morphology: fundamentals and specificity,” *Construction and Building Materials*, vol. 113, pp. 613–621, 2016.
- [32] G. L. Goić, M. Bigerelle, S. Samper, H. Favrelière, and M. Pillet, “Multiscale roughness analysis of engineering surfaces: a comparison of methods for the investigation of functional correlations,” *Mechanical Systems and Signal Processing*, vol. 66–67, pp. 437–457, 2016.
- [33] B. Liu and T. Yang, “Image analysis for detection of bugholes on concrete surface,” *Construction and Building Materials*, vol. 137, pp. 432–440, 2017.
- [34] S. Dash and M. R. Senapati, “Gray level run length matrix based on various illumination normalization techniques for texture classification,” *Evolutionary Intelligence*, 2018.
- [35] M. A. Khan, R. Vehmas, and A. Visa, “Automatic detection of water inside concrete slabs using ground penetrating radar,” in *Proceedings of the 2019 IEEE Radar Conference (Radar-Conf)*, pp. 1–5, Boston, MA, USA, April 2019.
- [36] I. Yoshitake, T. Maeda, and M. Hieda, “Image analysis for the detection and quantification of concrete bugholes in a tunnel lining,” *Case Studies in Construction Materials*, vol. 8, pp. 116–130, 2018.
- [37] G. K. Choudhary and S. Dey, “Crack detection in concrete surfaces using image processing, fuzzy logic, and neural networks,” in *Proceedings of the 2012 IEEE Fifth International Conference on Advanced Computational Intelligence (ICACI)*, pp. 404–411, Nanjing, China, October 2012.
- [38] N.-D. Hoang and Q.-L. Nguyen, “Metaheuristic optimized edge detection for recognition of concrete wall cracks: a comparative study on the performances of Roberts, Prewitt, Canny, and Sobel algorithms,” *Advances in Civil Engineering*, vol. 2018, Article ID 7163580, 16 pages, 2018.
- [39] C. Koch, S. G. Paal, A. Rashidi, Z. Zhu, M. König, and I. Brilakis, “Achievements and challenges in machine vision-based inspection of large concrete structures,” *Advances in Structural Engineering*, vol. 17, no. 3, pp. 303–318, 2014.
- [40] R. Tanabe and A. Fukunaga, “Success-history based parameter adaptation for differential evolution,” in *Proceedings of the 2013 IEEE Congress on Evolutionary Computation*, pp. 71–78, Beijing, China, June 2013.
- [41] R. Tanabe and A. S. Fukunaga, “Improving the search performance of SHADE using linear population size reduction,” in *Proceedings of the 2014 IEEE Congress on Evolutionary Computation (CEC)*, pp. 1658–1665, Beijing, China, July 2014.
- [42] C. Bishop, *Pattern Recognition and Machine Learning*, Springer Science+Business Media, Singapore, 2006.
- [43] N.-D. Hoang, Q.-L. Nguyen, and D. T. Bui, “Image processing-based classification of asphalt pavement cracks using support vector machine optimized by artificial bee colony,” *Journal of Computing in Civil Engineering*, vol. 32, no. 5, Article ID 04018037, 2018.
- [44] D. Dunn and W. E. Higgins, “Optimal Gabor filters for texture segmentation,” *IEEE Transactions on Image Processing*, vol. 4, no. 7, pp. 947–964, 1995.
- [45] N. N. Kachouie, J. Alirezaie, and P. Fieguth, “A hybrid algorithm using discrete cosine transform and Gabor filter bank for texture segmentation,” in *Proceedings of the Canadian Conference on Electrical and Computer Engineering 2004 (IEEE Cat. No. 04CH37513)*, vol. 1803, pp. 1805–1808, Waterloo, Canada, May 2004.
- [46] Z. Eduardo, G. G. B. Jaime, M. Roberto, and L. José, “Road crack detection using visual features extracted by Gabor filters,” *Computer-Aided Civil and Infrastructure Engineering*, vol. 29, no. 5, pp. 342–358, 2014.
- [47] N. C. Kim and H. J. So, “Directional statistical Gabor features for texture classification,” *Pattern Recognition Letters*, vol. 112, pp. 18–26, 2018.
- [48] R. Medina, F. Gayubo, L. M. González-Rodrigo et al., “Automated visual classification of frequent defects in flat steel coils,” *The International Journal of Advanced Manufacturing Technology*, vol. 57, no. 9–12, pp. 1087–1097, 2011.
- [49] S. Zehang, G. Bebis, and R. Miller, “Improving the performance of on-road vehicle detection by combining Gabor and wavelet features,” in *Proceedings of the IEEE 5th International Conference on Intelligent Transportation Systems*, pp. 130–135, San Sebastián, Spain, September 2002.
- [50] S. Zehang, G. Bebis, and R. Miller, “On-road vehicle detection using evolutionary Gabor filter optimization,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 6, no. 2, pp. 125–137, 2005.
- [51] K. Hammouda, “Texture segmentation using Gabor filters,” Technical Report, University of Waterloo, Waterloo, Canada, 2000, <http://www.pamiuwaterloo.ca/pub/hammouda/sd775-paperpdf>.

- [52] A. K. Jain and F. Farrokhnia, "Unsupervised texture segmentation using Gabor filters," *Pattern Recognition*, vol. 24, no. 12, pp. 1167–1186, 1991.
- [53] J. Jo and Z. Jadidi, "A high precision crack classification system using multi-layered image processing and deep belief learning," *Structure and Infrastructure Engineering*, vol. 16, no. 2, pp. 297–305, 2019.
- [54] M. M. Galloway, "Texture analysis using gray level run lengths," *Computer Graphics and Image Processing*, vol. 4, no. 2, pp. 172–179, 1975.
- [55] A. Chu, C. M. Sehgal, and J. F. Greenleaf, "Use of gray value distribution of run lengths for texture analysis," *Pattern Recognition Letters*, vol. 11, no. 6, pp. 415–419, 1990.
- [56] N.-D. Hoang and V.-D. Tran, "Image processing-based detection of pipe corrosion using texture analysis and metaheuristic-optimized machine learning approach," *Computational Intelligence and Neuroscience*, vol. 2019, Article ID 8097213, 13 pages, 2019.
- [57] N.-D. Hoang, Q.-L. Nguyen, and X.-L. Tran, "Automatic detection of concrete spalling using piecewise linear stochastic gradient descent logistic regression and image texture analysis," *Complexity*, vol. 2019, Article ID 5910625, 14 pages, 2019.
- [58] T. Xiaoou, "Texture information in run-length matrices," *IEEE Transactions on Image Processing*, vol. 7, no. 11, pp. 1602–1609, 1998.
- [59] B. V. Dasarathy and E. B. Holder, "Image characterizations based on joint gray level-run length distributions," *Pattern Recognition Letters*, vol. 12, no. 8, pp. 497–502, 1991.
- [60] K. Price, R. M. Storn, and J. A. Lampinen, *Differential Evolution—A Practical Approach to Global Optimization*, Springer-Verlag, Berlin, Germany, 2005.
- [61] R. Storn and K. Price, "Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, 1997.
- [62] S. Das, S. S. Mullick, and P. N. Suganthan, "Recent advances in differential evolution—an updated survey," *Swarm and Evolutionary Computation*, vol. 27, pp. 1–30, 2016.
- [63] A. P. Piotrowski, "Review of differential evolution population size," *Swarm and Evolutionary Computation*, vol. 32, pp. 1–24, 2017.
- [64] X. Shen, D. Zou, X. Zhang, Q. Zhang, and P. Xiao, "A phase-based adaptive differential evolution algorithm for the economic load dispatch considering valve-point effects and transmission losses," *Mathematical Problems in Engineering*, vol. 2018, Article ID 4585403, 24 pages, 2018.
- [65] A. Viktorin, R. Senkerik, M. Pluhacek, T. Kadavy, and A. Zamuda, "Distance based parameter adaptation for success-history based differential evolution," *Swarm and Evolutionary Computation*, vol. 50, Article ID 100462, 2019.
- [66] P. Biswas, P. Suganthan, and G. Amaratunga, "Optimal power flow solutions using algorithm success history based adaptive differential evolution with linear population reduction," in *Proceedings of the 2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 249–254, Miyazaki, Japan, October 2018.
- [67] A. P. Piotrowski, "L-SHADE optimization algorithms with population-wide inertia," *Information Sciences*, vol. 468, pp. 117–141, 2018.
- [68] A. P. Piotrowski and J. J. Napiorkowski, "Step-by-step improvement of JADE and SHADE-based algorithms: success or failure?" *Swarm and Evolutionary Computation*, vol. 43, pp. 88–108, 2018.
- [69] A. Zamuda, "Adaptive constraint handling and success history differential evolution for CEC 2017 constrained real-parameter optimization," in *Proceedings of the 2017 IEEE Congress on Evolutionary Computation (CEC)*, pp. 2443–2450, June 2017.
- [70] V. N. Vapnik, *Statistical Learning Theory*, John Wiley & Sons, Inc., Hoboken, NJ, USA, 1998.
- [71] S. Deng, X. Wang, Y. Zhu, F. Lv, and J. Wang, "Hybrid grey wolf optimization algorithm based support vector machine for groutability prediction of fractured rock," *Mass Journal of Computing in Civil Engineering*, vol. 33, no. 2, Article ID 04018065, 2019.
- [72] B. T. Pham, A. Jaafari, I. Prakash, and D. T. Bui, "A novel hybrid intelligent model of support vector machines and the MultiBoost ensemble for landslide susceptibility modeling," *Bulletin of Engineering Geology and the Environment*, vol. 78, no. 4, pp. 2865–2886, 2018.
- [73] H. Wei, M. Wang, B. Song, X. Wang, and D. Chen, "Study on the magnitude of reservoir-triggered earthquake based on support vector machines," *Complexity*, vol. 2018, Article ID 2830690, 10 pages, 2018.
- [74] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*, Springer, Berlin, Germany, 2011.
- [75] S. Guo, J. S. Tsai, C. Yang, and P. Hsu, "A self-optimization approach for L-SHADE incorporated with eigenvector-based crossover and successful-parent-selecting framework on CEC 2015 benchmark set," in *Proceedings of the 2015 IEEE Congress on Evolutionary Computation (CEC)*, pp. 1003–1010, Sendai, Japan, May 2015.
- [76] Accord, *Accord.NET Framework*, <http://accord-framework.net/>, 2019.
- [77] D. N. Joanes and C. A. Gill, "Comparing measures of sample skewness and kurtosis," *Journal of the Royal Statistical Society: Series D (The Statistician)*, vol. 47, no. 1, pp. 183–189, 1998.
- [78] D. Tien Bui, N.-D. Hoang, H. Nguyen, and X.-L. Tran, "Spatial prediction of shallow landslide using bat algorithm optimized machine learning approach: a case study in Lang Son Province, Vietnam," *Advanced Engineering Informatics*, vol. 42, Article ID 100978, 2019.
- [79] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning (Adaptive Computation and Machine Learning Series)*, The MIT Press, Cambridge, MA, USA, 2016.
- [80] P. Kim, *MatLab Deep Learning with Machine Learning, Neural Networks and Artificial Intelligence*, Apress, New York, NY, USA, 2017.
- [81] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
- [82] N. Qian, "On the momentum term in gradient descent learning algorithms," *Neural Networks*, vol. 12, no. 1, pp. 145–151, 1999.
- [83] D. P. Kingma and J. Ba, "Adam: a method for stochastic optimization," in *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*, San Diego, CA, USA, May 2015.
- [84] T. Tieleman and G. Hinton, "Lecture 6.5—RMSProp COURSERA: neural networks for machine learning," Technical Report, University of Toronto, Toronto, Canada, 2012.
- [85] Matwork, "Deep learning toolbox," 2018, <https://www.mathworks.com/help/deeplearning/index.html>.
- [86] M. T. Hagan, H. B. Demuth, M. H. Beale, and O. D. Jesús, *Neural Network Design*, University of Colorado Bookstore, Boulder, CO, USA, 2nd edition, 2014.

- [87] G. Montavon, G. Orr, and K.-R. Müller, “Neural networks: tricks of the trade,” in *Lecture Notes in Computer Science*, Springer-Verlag, Berlin, Germany, 2012.
- [88] S. Skansi, *Introduction to Deep Learning—from Logical Calculus to Artificial Intelligence*, Springer International Publishing, New York, NY, USA, 2018.
- [89] J. Heaton, *Artificial Intelligence for Humans, Volume 3 Deep Learning and Neural Networks*, Heaton Research, Inc., St. Louis, MO, USA, 2015.
- [90] G. W. Snedecor and W. G. Cochran, *Statistical Methods*, Iowa State University Press, Iowa City, IA, USA, 8th edition, 1989.
- [91] S. Fekri-Ershad and F. Tajeripour, “Impulse-noise resistant color-texture classification approach using hybrid color local binary patterns and Kullback-Leibler divergence,” *The Computer Journal*, vol. 60, no. 11, pp. 1633–1648, 2017.