

Research Article

Image Processing-Based Spall Object Detection Using Gabor Filter, Texture Analysis, and Adaptive Moment Estimation (Adam) Optimized Logistic Regression Models

Nhat-Duc Hoang ^{1,2}

¹Institute of Research and Development, Duy Tan University, Da Nang 550000, Vietnam

²Faculty of Civil Engineering, Duy Tan University, Da Nang 550000, Vietnam

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

Received 16 May 2020; Revised 5 November 2020; Accepted 12 November 2020; Published 30 November 2020

Academic Editor: Moacir Kriпка

Copyright © 2020 Nhat-Duc Hoang. 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.

This study aims at proposing a computer vision model for automatic recognition of localized spall objects appearing on surfaces of reinforced concrete elements. The new model is an integration of image processing techniques and machine learning approaches. The Gabor filter supported by principal component analysis and *k*-means clustering is used for identifying the region of interest within an image sample. The binary gradient contour, gray level co-occurrence matrix, and color channels' statistical measurements are employed to compute the texture of the extracted region of interest. Based on the computed texture-based features, the logistic regression model trained by the state-of-the-art adaptive moment estimation (Adam) is utilized to establish a decision boundary that delivers predictions on the status of “nonlocalized spall” and “localized spall.” Experimental results demonstrate that the newly developed model is able to achieve good detection accuracy with classification accuracy rate = 85.32%, precision = 0.86, recall = 0.79, negative predictive value = 0.85, and *F1* score = 0.82. Thus, the proposed computer vision model can be helpful to assist decision makers in the task of the periodic survey of structure health condition.

1. Introduction

Public safety is a major concern of civil engineers who design and maintain high-rise buildings. Despite considerable efforts in design and advanced knowledge of building structures, accidents can still happen in the built environment due to excessive usage, structural aging, and inclement weather conditions [1]. Among the hazards occurred in high-rise buildings, falling objects from overhead caused by concrete spalling can be particularly dangerous and have a high potential severity to occupants' health [2]. The effect of concrete debris can be devastating for human lives if it gets broken off from surfaces of exterior wall systems of high-rise buildings [3].

A concrete spall (Figure 1) is regarded as flakes of concrete/mortar broken off from a concrete element (e.g., beam, wall, and ceiling) [4]. Spalling is typically caused by stresses brought about by differential movement of

materials. Most often, spalling in concrete is due to corrosion of steel reinforcement embedded in the structure. To prevent such accidents and to ensure the safety and serviceability of the built environment, periodic visual surveys of structural health condition and proper maintenance activities are very crucial [5].

In developing countries, including Vietnam, manual visual inspection is still the principal method for evaluating structural health conditions. This activity is performed at regular intervals to identify potential damages and guarantee the service/safety requirements of high-rise buildings. Provided the well-trained technicians experienced in structural health assessment, manual visual inspection is able to providing accurate surveying outcomes. Nevertheless, due to the increasing numbers of buildings needed to be inspected periodically and the limited number of experience technicians, timely evaluation of building elements becomes infeasible and inspection deficiencies become a major

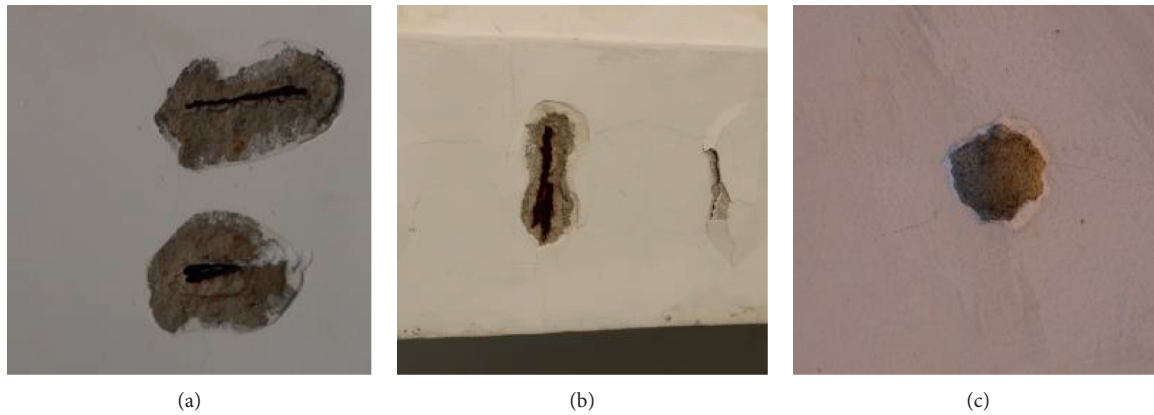


FIGURE 1: Spall objects on (a) ceiling, (b) beam, and (c) wall structures.

concern of property owners. Therefore, there is a practical need to substitute the unproductive manual visual survey with a more effective approach.

Recently, due to the ease of access to low-cost and high-quality visual sensing equipment including digital cameras, computer vision-based models have been increasingly used for automatic structural health monitoring [6]. These advanced approaches have been proved to be viable alternatives to replace the labor-intensive and subjective methods relied on manual survey. With the use of advanced image processing techniques operated on image samples collected from digital cameras, the physical condition of civil structures can be continuously surveyed and reported to maintenance agencies. This evaluation outcome can be effectively used to support the decision-making process regarding maintenance prioritization and funding allocation.

Due to such reasons, a large number of computer vision-based approaches have been proposed to successfully detect various forms of structural defects such as cracking and spalling. Abdel-Qader et al. [7] employs a principle component analysis-based model to recognize cracking defects appeared on bridge surfaces; the principle component analysis is utilized to support data cluster identification with a large database of bridge images. O'Byrne et al. [8] utilizes texture analysis for detecting damages appeared on infrastructural elements; the texture-based image segmentation relies on pixel intensity values and gray level co-occurrence matrix. Subsequently, a support vector machines model is then employed for the data classification task. Lattanzi and Miller [9] rely on the data clustering approach for image segmentation based on the Canny and k -means algorithm; the research finds that the combined algorithms can deliver good accuracy of crack recognition under different environmental circumstances.

As can be seen from the literature, a large number of previous studies have been dedicated to crack detection for concrete structures [10–20]. Only recently, there is an increasing focus on detecting other forms of damage including concrete spalling [21–24]. German et al. [25] constructs a combination of segmentation, template matching, and morphological preprocessing for detecting spall appeared on surfaces of concrete columns. Machine learning models

including support vector machines, Naïve Bayesian classifier, and random forest have been employed to identify concrete defects [8, 26]. A model for localization and quantification of concrete spalling defects based on terrestrial laser scanning has been proposed in [27]. Dawood et al. [21] presented a computer vision-based model for spalling detection used in environment of subway networks.

Hoang [28] relies on a steerable filter and machine learning to recognize wall defects such as cracks and spalls. A concrete spalling detection model for metro tunnel from point cloud that employs a roughness descriptor has been developed by Wu et al. [24]. Hoang [29] presents an image processing approach for automatic detection of concrete spalling using machine learning algorithms and image texture analysis. Nevertheless, this model focused on machine learning-based texture discrimination and was not capable of isolating the entire individual spall object.

Yao et al. [30] establishes a convolutional neural network-based model for concrete bughole detection; a large number (about 10,000) of image examples have been used as a training dataset. Li et al. [31] proposed a model for detecting exposed aggregate appeared on stilling basin slab using the attention U-Net network. Chow et al. [32] employs deep learning of a convolutional autoencoder for anomaly detection of defects existing on concrete structures. A model for recognizing damaged ceiling areas in large-span structures has been proposed by Wang et al. [33]; this model employs a convolutional neural network for pattern recognition. Although deep learning-based models are capable of performing feature extraction phase automatically, these supervised learning models generally demand a large-size training dataset and a meticulous process of data labeling [34–36]. This data labeling process itself can be time-consuming as well as error prone [5]. In addition, the deep learning models also require experience and the trial-and-error process to adjust a significant amount of model's free parameters.

An effort of combining unsupervised learning and machine learning-based data classification has been recently introduced in [37]. The k -means clustering algorithm and machine learning classifier have been integrated to form an automatic approach for estimating stripping of asphalt

coating. The k -means clustering algorithm is utilized to separate pixels with similar values on the surface of aggregates; subsequently, machine learning models are used to categorize the identified clusters into groups of asphalt-coated and uncoated areas.

As pointed out by previous studies, the current challenges faced by computer vision-based concrete damage detection including spall recognition are complex environmental conditions (e.g., noisy background image) [5] and the difficulty of the image labeling process [32]. More efforts should be dedicated to automatically identify the damage's region of interest (ROI) via unsupervised learning methods. Capable machine learning methods with few free parameters should be investigated as viable alternatives to sophisticated models used for data classification. It is because simple and manageable models significantly facilitate the development and application of hybrid computer vision-machine learning approaches for concrete spalling detection.

Based on such motivation, this study proposes and verifies an automated method for recognizing localized spall objects based on an integration of a Gabor filter, k -means clustering, image texture analysis, and logistic regression pattern classification models. The Gabor filter coupled with the principal component analysis and the k -means clustering are used synergistically for automatic identification of ROI on concrete surface. The image texture analysis combines powerful texture discriminators of binary gradient contours, color channels' properties, and the gray level co-occurrence matrix. The logistic regression model trained by the state-of-the-art adaptive moment estimation (Adam) optimizer is employed for data classification.

The subsequent sections of the study are organized as follows: the second section reviews the research methodology. The third section presents the image data collection process. The proposed integrated model used for concrete spall detection is described in the next section, followed by the experimental results and discussion. The final section summarizes the research findings with several concluding remarks.

2. Research Methodology

2.1. Gabor Filter (GF). Image segmentation is the process of separating an image into distinctive regions [38, 39]. The GF is a widely applied approach for segmenting image [40, 41]. This approach is inspired by the multichannel operation of the human visual system used for visual interpretation in real-world circumstances [42–44]. Based on experimentation, it has been shown that the GF resembles simple cells in the Mammalian vision system. Thus, this filter can be a reasonable model of how humans actually recognize and discriminate areas characterized by different texture [45].

The GF consists of two-dimensional Gabor filters which can be described as complex sinusoidal waves modulated by Gaussian envelopes [43]. This filter carries out a localized and oriented frequency analysis of a two-dimensional signal. The GF yields a response that can be mathematically given as follows [45]:

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

where u_0 denotes the frequency of a sinusoidal plane wave along the x axis. σ_x and σ_y represent the space constants of the Gaussian envelope along the x and y axes, respectively. It is noted that the GF with arbitrary orientations can be obtained via a rigid rotation of the x - y coordinate system.

The frequency domain representation of the GF is given by [45]

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

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 worth noticing that it is necessary to specify tuning parameters of the GF including the orientation angles and the radial frequency. Based on the suggestions of Jain and Farrokhnia [45], four values of orientations are often employed: 0° , 45° , 90° , and 135° . Given an image with a width of N_w pixels, the following values of radial frequency u_0 can be considered: $1\sqrt{2}$, $2\sqrt{2}$, $4\sqrt{2}$, ..., $N_w/4\sqrt{2}$.

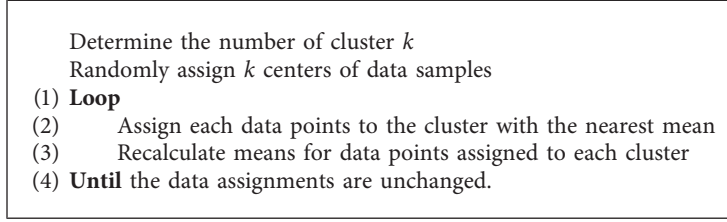
2.2. The K-Means Clustering Algorithm. In this study, the unsupervised machine learning approach of k -means clustering [46] is employed to divide an image into different regions based on the analysis results obtained from the GF. This unsupervised machine learning method is simple yet powerful algorithm for automatic data grouping [47]. Based on such method, image pixels that have the similar properties can be grouped in one cluster. Accordingly, data samples belonging to one cluster feature the smallest degree of variation. The iterative algorithm used to compute the cluster centers is presented in Algorithm 1.

2.3. Image Texture Analysis

2.3.1. Binary Gradient Contours (BGC). The BGC, proposed by Fernández et al. [48], is a group of computationally simple texture descriptors. Given a 3×3 grayscale image patch, these texture descriptors employs a set of eight binary gradients between pairs of pixels all along a closed path around the central pixel [49]. The BGC includes three versions which are single-loop, double-loop, and triple-loop descriptors. Via experimentation, the BGC operator has been found to achieve good texture discrimination outcomes.

A matrix S which is the pixel intensity of an image patch of the size 3×3 is given as follows:

$$S = \begin{bmatrix} I_7 & I_6 & I_5 \\ I_0 & I_c & I_4 \\ I_1 & I_2 & I_3 \end{bmatrix}, \quad (3)$$

ALGORITHM 1: The k -means clustering.

where I_c denotes the central pixel. I_0, I_1, \dots, I_7 are the neighboring pixels.

The schematic representations of BGC with three versions of single, double, and triple loops are presented in Figure 2. In addition, to facilitate the mathematical formulation of these texture descriptors, a square crop $S_{m,n}$ is given by

$$S = \begin{bmatrix} I_{m-1,n-1} & I_{m-1,n} & I_{m-1,n+1} \\ I_{m,n-1} & I_{m,n} & I_{m,n+1} \\ I_{m+1,n-1} & I_{m+1,n} & I_{m+1,n+1} \end{bmatrix}, \quad (4)$$

where $I_{m,n}$ represents the pixel at m^{th} row and n^{th} column.

Accordingly, the formulations of the single, double, and triple-loop versions are given by [48]

(i) Single-loop version:

$$\text{BGC}_1 = \sum_{n=0}^7 \lambda(I_n - I_{(n+1) \bmod 8}) \times 2^n - 1. \quad (5)$$

(ii) Double-loop version:

$$\begin{aligned} \text{BGC}_2 &= 15 \times \sum_{n=0}^3 \lambda(I_{2n} - I_{2(n+1) \bmod 8}) \times 2^n \\ &+ \sum_{n=0}^3 \lambda(I_{2n+1} - I_{(2n+3) \bmod 8}) \times 2^n - 16. \end{aligned} \quad (6)$$

(iii) Triple-loop version:

$$\text{BGC}_3 = \sum_{n=0}^7 \lambda(I_{3n \bmod 8} - I_{3(n+1) \bmod 8}) \times 2^n - 1, \quad (7)$$

$$\text{where } \lambda(x) = \begin{cases} 1, & \text{if } x \geq 0, \\ 0, & \text{if } x < 0. \end{cases}$$

2.3.2. RGB Channels' Properties. Since the color properties of spall and nonspall objects are expected to be dissimilar, this study employs the statistical measurements of three color channels: red (R), green (G), and blue (B) as a means of texture description. Given an image sample I , the first-order histogram $P(I)$ can be computed. Accordingly, the mean (μ_c), standard deviation (σ_c), skewness (S_c), kurtosis (K_c), entropy (E_c), and range (R_c) of the three color channels (R , G , and B) can be calculated [29, 50].

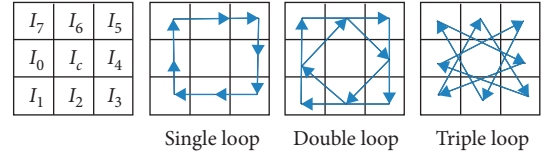


FIGURE 2: The graphical representation of BGC.

2.3.3. Gray Level Co-Occurrence Matrix (GLCM). The GLCM [51, 52] is also an extensively employed approach for characterizing image texture. This approach focuses on capturing the repeated occurrence of certain gray-level patterns [53]. Therefore, indices extracted from a GLCM can be effectively utilized to evaluate the coarseness/fineness of an image region. Let r and θ represent a distance and a rotation relationship between two individual pixels. The GLCM, denoted as P_δ , denotes a probability of the two gray levels of i and j having the relationship specifying by r and θ [54]. Based on the recommendations of Haralick et al. [51], the GLCM can be constructed with $r=1$ and $\theta = 0^\circ, 45^\circ, 90^\circ$, and 135° . Accordingly, for each matrix, four indices of angular second moment (AM), contrast (CO), correlation (CR), and entropy (ET) can be computed as follows [29, 55]:

$$\begin{aligned} \text{AM} &= \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} P_\delta^N(i, j)^2, \\ \text{CO} &= \sum_{k=0}^{N_g-1} k^2 \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} P_\delta^N(i, j), \\ &\quad |i-j|=k \end{aligned} \quad (8)$$

$$\text{CR} = \frac{\sum_{i=1}^{N_g} \sum_{j=1}^{N_g} i \times j \times P_\delta^N(i, j) - \mu_X \mu_Y}{\sigma_X \sigma_Y},$$

$$\text{ET} = - \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} P_\delta^N(i, j) \log(P_\delta^N(i, j)),$$

where N_g denotes the number of gray level values; μ_X, μ_Y, σ_X , and σ_Y are the means and standard deviations of the marginal distribution with respect to $P_\delta^N(i, j)$.

2.4. Logistic Regression Model (LRM). The LRM is a capable method for solving binary classification problems [29, 56–58]. The task at hand is to construct a decision

boundary that categorizes the input data into two distinctive regions. Therefore, given a vector of input data $x_i = x_{i1}, x_{i2}, \dots, x_{iD}$, where D is the number of the features used for classification, the model is able to derive the class output y with either $y = 0$ (for the negative class of nonspall) and $y = 1$ (for the positive class of spall).

The probability of the positive class $h_\theta(x_i)$ derived by a LRM is given by [59]

$$h_\theta(x_i) = h_\theta(x_{i1}, x_{i2}, \dots, x_{iD}) = \frac{1}{1 + \exp(-\eta_i)} \quad (9)$$

$$= \frac{1}{1 + \exp(-\theta^T x_i)},$$

where $\eta_i = \theta_0 + \theta_1 x_{i1} + \theta_2 x_{i2} + \dots + \theta_D x_{iD} = \theta^T x_i$. The vector $\theta = \theta_0, \theta_1, \theta_2, \dots, \theta_D$ is the model parameter.

As a supervised learning approach, a set of training examples needs to be prepared so that the vector θ can be adapted during the model training phase. A LRM can be trained by either minimizing the least square loss function or maximizing the log likelihood function.

The least square loss function is given by [60]

$$\text{Loss}(\theta) = \sum_{i=1}^M (y_i - h_\theta(x_i))^2, \quad (10)$$

where M is the number of training data.

The log likelihood function is described as follows [61, 62]:

$$\text{Loss}(\theta) = \log(L(\theta)) = \sum_{i=1}^M y_i \log(h_\theta(x_i)) + (1 - y_i)(1 - \log(h_\theta(x_i))). \quad (11)$$

A LRM can be trained via the stochastic gradient descent framework [29]. If the least square loss function is used, the update rule for adapting the model parameters is given by [60]

$$\theta_k = \theta_k + Lr \times (y - h_\theta(x_i)) \times h_\theta(x_i)(1 - h_\theta(x_i)) \times x_{i,k}, \quad (12)$$

where Lr denotes the learning rate parameter.

Meanwhile, if the log likelihood function is selected, the update rule used that compute θ is given by [61, 62]

$$\theta_k = \theta_k + Lr \times (y_i - h_\theta(x_i))x_{i,k}. \quad (13)$$

2.5. Adaptive Moment Estimation (Adam) Optimizer. Adam, proposed by [63], is designed as an algorithm for first-order gradient-based optimization of stochastic objective functions. This algorithm is relied on adaptive estimates of lower-order moments. Adam can be considered as an extension of the stochastic gradient descent employed to train machine learning models via an iterative weight updating process [64]. It is noted that the conventional stochastic gradient descent employs a constant learning rate (Lr) for all weight updates. Adam seeks for improving the

model training phase by adaptively fine-tuning the Lr parameter.

Adam harnesses information obtained from the average of the second moments of the gradients. In detail, this optimization algorithm computes an exponential moving average of the gradient and the square gradient. Moreover, a set of parameters (β_1 and β_2) is used to dictate the decay rates of these moving averages [64]. Via experimentation, it can be shown that the advantages of this optimizer include efficient computation, straight forward implementation, no memory requirements, and the capability of dealing with a large number of optimized parameters [63].

In order to implement Adam to optimize a LRM, it is necessary to compute the gradient (g_t). The gradient g_t in the case of using the least square loss function is given by [60]

$$g_t = \frac{\partial \text{Loss}(\theta)}{\partial \theta_k} = 2(y - h_\theta(x_i)) \times h_\theta(x_i) \times (1 - h_\theta(x_i)) \times x_{i,k}. \quad (14)$$

If the log likelihood function is employed, the gradient g_t is given by [22, 61, 62]

$$g_t = \frac{\partial \text{Loss}(\theta)}{\partial (\theta_k)} = (y_i - h_\theta(x_i))x_{i,k}. \quad (15)$$

Accordingly, the Adam procedure (illustrated in Algorithm 2) used for training a LRM can be performed iteratively with the following steps:

- (i) Compute gradient g_t
- (ii) Update the biased first and second raw moment estimates
- (iii) Compute the bias-corrected moment estimates
- (iv) Adapt the optimized parameters

3. Collection of Image Samples

The LRM used in this study belongs to the category of supervised machine learning methods. To train this LRM with the use of the aforementioned Adam optimizer, it is a requisite to prepare a set of training image samples as well as a set of testing image samples to verify the model construction phase. Therefore, this study has carried out field surveys at several high-rise buildings in Danang city (Vietnam) to collect a set of 600 image samples. Among them, 300 samples contain localized spall objects and 300 samples consist of nonlocalized spall objects. Notably, image samples of the two class of nonspall (class label = 0) and spall (class label = 1) have been assigned by a human inspector for the purposes of model training and testing. The Cannon EOS M10 (CMOS 18.0 MP) and Nikon D5100 (CMOS 16.2 MP) have been employed to collect image samples. In addition, the image size has been standardized to be 64×64 to facilitate the computation process. The image set has been collected so that a diverse background (e.g., cracks and stains) can be included. The collected image set is demonstrated in Figure 3.

```

Define step size  $a = 0.001$ 
Define exponential decay rates  $\beta_1 = 0.9$  and  $\beta_2 = 0.9999$ 
Define the objective function  $f(\theta)$ 
Randomly initialize the searched variable  $\theta$ 
Assign  $m_0 = 0$ ,  $v_0 = 0$ , and  $t = 0$ 
(1) While ( $\theta_t$  not converged)
(2)    $t = t + 1$ 
(3)   Compute gradient:  $g_t = \nabla_{\theta} f_t(\theta_{t-1})$ 
(4)   Update biased 1st moment estimate
(5)      $m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t$ 
(6)   Update biased 2nd raw moment estimate
(7)      $v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2$ 
(8)   Calculate bias-corrected first moment estimate
(9)      $\hat{m}_t = m_t / (1 - \beta_1^t)$ 
(10)  Calculate bias-corrected 2nd raw moment estimate
(11)    $\hat{v}_t = v_t / (1 - \beta_2^t)$ 
(12)  Update the searched parameter
(13)    $\theta_t = \theta_{t-1} - \alpha \times (\hat{m}_t / \sqrt{\hat{v}_t} + \epsilon)$ 
(14) End While
(15) Return  $\theta_t$ 

```

ALGORITHM 2: The Adam optimization procedure.

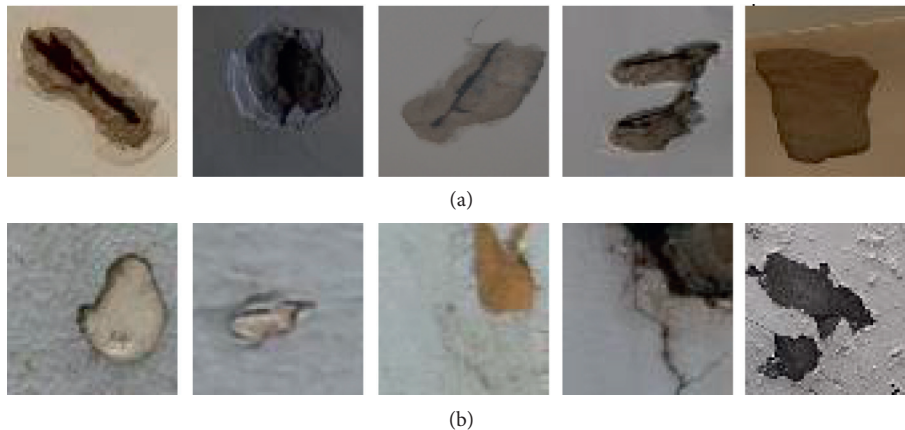


FIGURE 3: Demonstration of the collected image samples: (a) images containing localized spall objects and (b) images containing non-localized spall objects.

4. The Proposed Hybrid Approach of Image Processing and Machine Learning Approach for Automatic Detection of Concrete Spall

This section of the study aims at describing the structure of the proposed hybrid approach of image processing and machine learning used for recognizing localized spall object. The overall structure of the proposed approach is presented in Figure 4. It is noted that the hybrid model used for automatic concrete spall detection has been developed in Visual C#.NET environment (Framework 4.6.2) and implemented with the ASUS FX705GE-EW165T (Core i7 8750H, 8 GB Ram, 256 GB solid-state drive).

The model can be divided into several operational steps:

(i) Automatic ROI identification

(ii) Image texture computation

(iii) Machine learning-based pattern classification

4.1. Automatic Region of Interest (ROI) Identification. To deal with the diverse shapes of localized spall objects, this study relies on the techniques of GF to automatically identify ROIs that contains the potential defects of interest. It is noted that an image sample has been denoised by a median filter with a window size of 4 pixels and converted to a grayscale one. After the GFs with different orientations and radial frequency are computed, the principal component analysis (PCA) is performed to transform the set of GFs and reduce the data dimensionality (Figure 5). The number of the PCA transformed data is selected corresponding to 99% of cumulative variance explained. It is noted that the GF and the

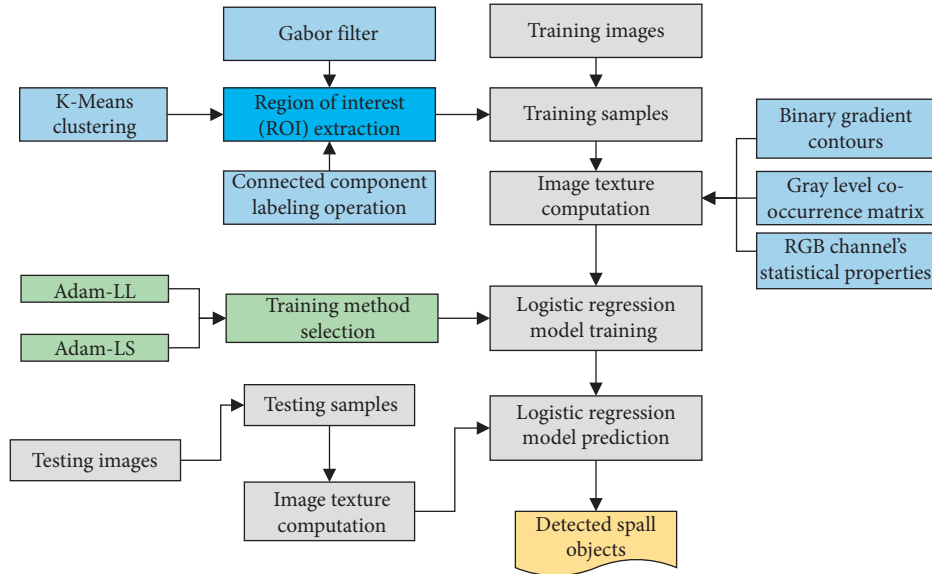


FIGURE 4: The proposed model structure.

PCA operations have been implemented via built-in functions provided by the Accord.NET Framework [65].

Based on the PCA result, the k -means clustering algorithm is used to segment the image sample. Via experimentation, the suitable number of clusters for the collected dataset is found to be 3. Subsequently, the morphological operation of filling and removing small objects are utilized to process the segmented image. Moreover, the operation of background removal is performed to remove redundant objects. In this study, an object within an image sample is considered to be background if its width or height is equal to that of the image sample.

Accordingly, each image cluster or segment is presented as a binary image. The connected component labeling algorithm [66] is then used to analyze the position of the binary-1 pixels and separate them into distinctive component regions. Essentially, all pixels having value binary 1 and are connected to each other are grouped into one object [38]. To remove crack objects, for each grouped pixels obtained from the connected component labeling analysis, an object slenderness index (OSI) is computed as follows:

$$OSI = \text{Max}\left(\frac{L_{OX}}{\mu_{OX}}, \frac{L_{OY}}{\mu_{OY}}\right), \quad (16)$$

where L_{OX} and L_{OY} are the object lengths along the X axis and Y axis, respectively. μ_{OX} and μ_{OY} denote the mean thicknesses of the object along the X axis and Y axis, respectively.

If the OSI of an object is greater than a certain threshold (T_{OSI}), this object is classified as a crack. Via several trial-and-error experiments with the collected image sample, a suitable value for the threshold T_{OSI} is found to be 5. After the ROIs have been identified, operations of image convolution and cropping are employed to isolate the areas of interest within the image sample. The processes of ROI identification and isolation are demonstrated in Figures 6 and 7.

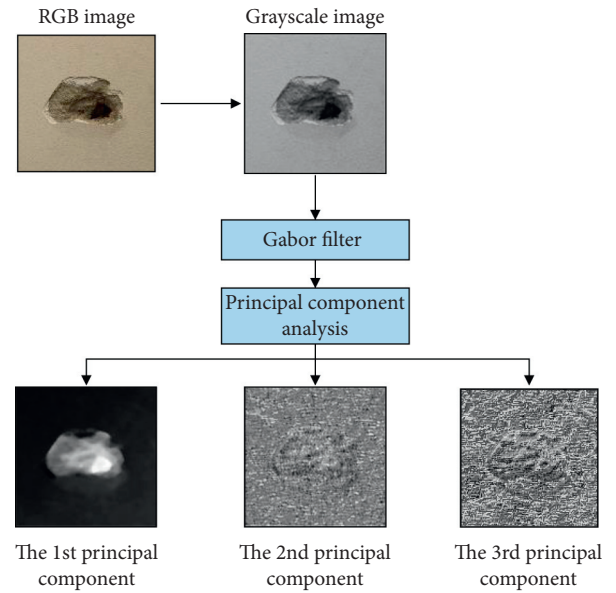


FIGURE 5: Image processed by the Gabor filter.

4.2. Image Texture Computation. Based on ROIs obtained from the previous step, image texture analysis consisting of statistical measurements of BGC, RGB channels, and GLCM is carried out. The BGC texture descriptor includes all of the three variants of single, double, and triple loops. Each of the variants yields a histogram which describes the texture property of an image sample. Accordingly, statistical indices including mean, standard deviation, skewness, kurtosis, and entropy can be computed for each histogram. Therefore, the BGC results in 15 numerical features. As mentioned earlier, the mean (μ_c), standard deviation (σ_c), skewness (S_c), kurtosis (K_c), entropy (E_c), and range (R_c) of the three color channels (R , G , and B) are used to represent the color features of image samples. Thus, there are $6 \times 3 = 18$ additional numerical features. Moreover, properties of the

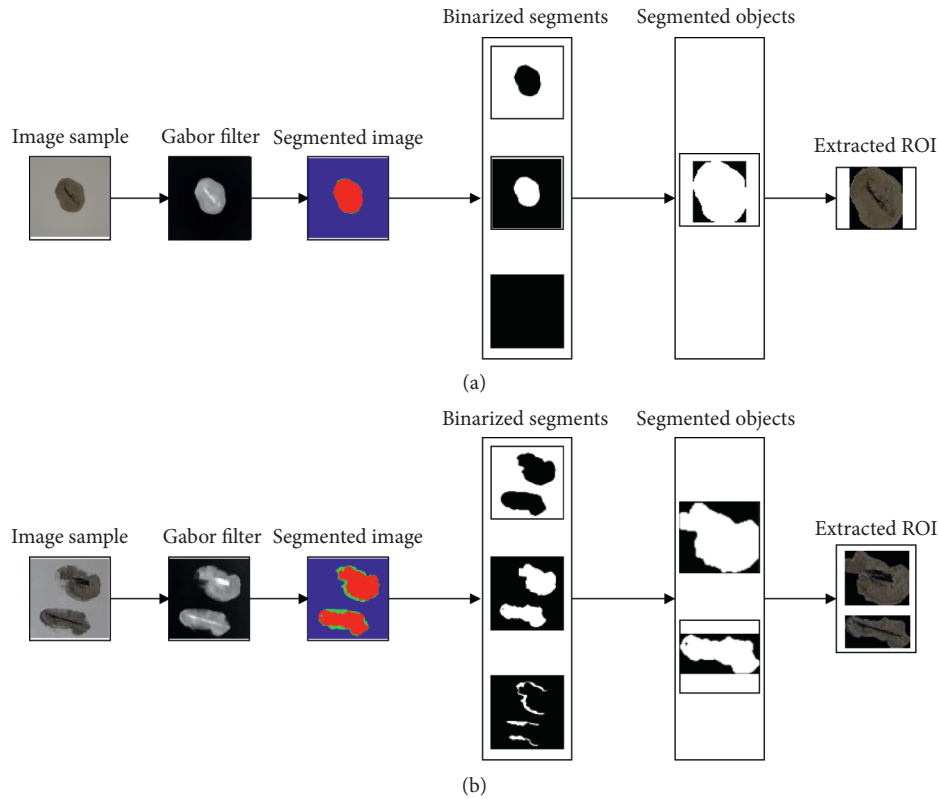


FIGURE 6: ROI extraction for images containing localized spall objects: (a) one object and (b) multiple objects.

GLCM including the four indices of angular second moment (AM), contrast (CO), correlation (CR), and entropy (ET) are used. It is noted that for each image sample, four GLCMs are established. Thus, the GLCM texture descriptor yields $4 \times 4 = 16$ features. In total, there are $15 + 18 + 16 = 49$ numerical features that can be extracted from the image texture computation process.

4.3. Pattern Classification Using LRM Trained by the Adam Optimizer. Using the extracted ROIs and the aforementioned texture descriptors, a dataset with 790 samples and 49 features can be constructed. This dataset contains 465 nonlocalized spall samples and 325 localized spall samples. As stated earlier, the output class is either 0 for the negative class and 1 for the positive class. Moreover, in order to standardize the input features' range, the numerical texture descriptors have been normalized by the Z-score equation as follows:

$$X_{ZN} = \frac{X_o - m_X}{s_X}, \quad (17)$$

where X_o and X_{ZN} represent the original and normalized input data, respectively. m_X and s_X are the mean and the standard deviation of the original input data, respectively.

Based on the aforementioned dataset, the LRM is trained with the Adam optimizer using the least square and log likelihood loss functions. These two LRM is denoted as

Adam-LS and Adam-LL. It is noted that 90% of the collected dataset has been employed to construct the LRM model. Meanwhile, the rest of the dataset is reserved to verify the generalization capability of the model.

5. Research Findings and Discussion

As mentioned earlier, the whole collected dataset is divided into two subsets: a training set (90%) and a testing set (10%). Moreover, to diminish the effect of randomness brought about by data sampling and to assess the generalization capability of the integrated method reliably, the data sampling process has been repeated 20 times. A partitioned datasets used for model training and testing are demonstrated in Table 1. In addition, the LRM trained with the stochastic gradient descent algorithm with the least square and log likelihood loss function are employed as benchmark models. The stochastic gradient descent models coupled with the former and later loss function are denoted as LS-LR and LL-LR, respectively. Furthermore, the two LRMs trained with the Adam optimizer are denoted as Adam-LS-LR and Adam-LL-LR. All of the LRMs have been trained with 1000 iterations.

In addition, the classification accuracy rate (CAR), precision, recall, negative predictive value (NPV), and F1 score are computed to quantify the model predictive accuracy. These performance measurement indices are provided as follows [67]:

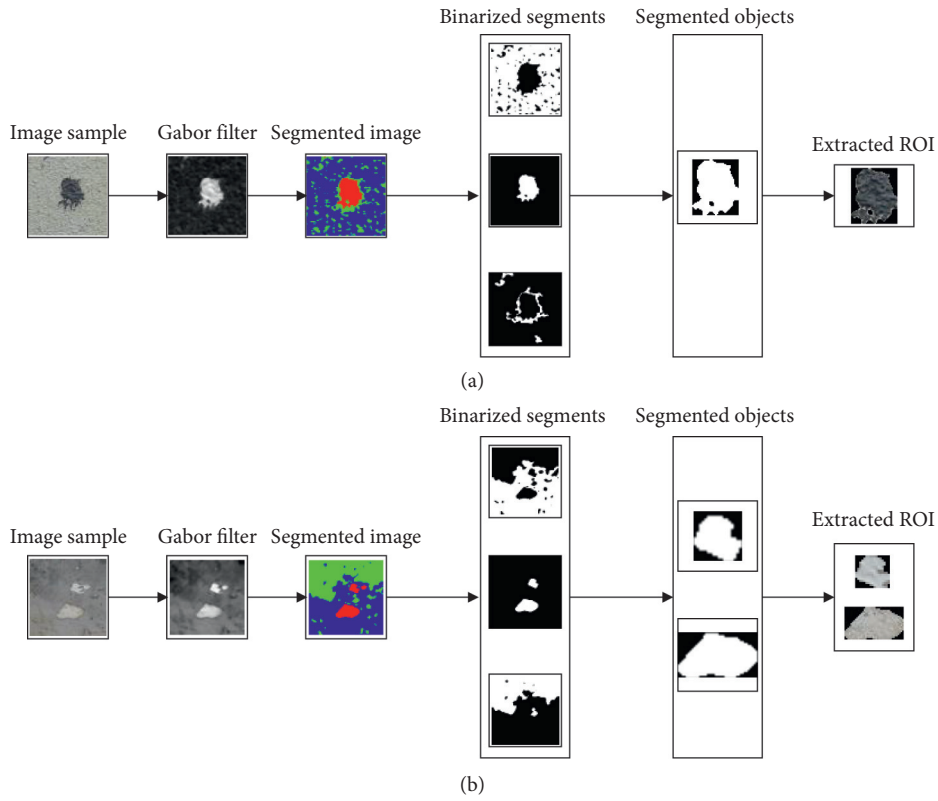


FIGURE 7: ROI extraction for images containing nonlocalized spall objects: (a) one object and (b) multiple objects.

TABLE 1: Demonstration of the collected dataset.

Set	Case	IF1	IF2	IF3	IF4	...	IF46	IF47	IF48	IF49	Class label
Training	1	39.64	37.27	34.53	34.33	...	0.10	745.72	0.00	6.47	0
	2	12.92	11.63	9.60	18.34	...	0.17	497.69	0.00	3.82	0
	3	68.21	65.76	59.38	40.67	...	0.03	770.71	0.00	6.82	0
	4	44.78	45.60	43.86	56.59	...	0.29	1835.96	0.00	4.39	0
	5	65.40	64.84	66.24	60.02	...	0.12	2458.40	0.00	4.12	0

	707	68.85	58.73	45.76	34.57	...	0.01	824.92	0.00	6.77	1
	708	55.85	51.50	46.05	26.86	...	0.01	762.79	0.00	7.48	1
	709	51.27	47.45	42.09	35.80	...	0.05	928.80	0.00	6.93	1
	710	60.32	55.98	49.76	31.67	...	0.01	1340.14	0.00	6.67	1
	711	73.54	65.91	62.20	44.11	...	0.02	1180.24	0.00	7.82	1
Testing	1	56.26	58.54	61.95	25.44	...	0.01	498.38	0.00	7.27	0
	2	93.11	95.22	103.83	41.81	...	0.10	1353.60	0.00	3.32	0
	3	85.93	83.79	93.25	49.92	...	0.07	3975.50	0.00	3.95	0
	4	58.64	59.44	64.39	55.01	...	0.18	968.76	0.00	5.57	0
	5	83.84	84.93	88.35	68.18	...	0.14	493.65	0.00	5.54	0

	75	62.98	47.68	28.07	32.93	...	0.03	244.76	0.00	7.79	1
	76	46.17	44.28	39.69	33.57	...	0.08	467.10	0.00	6.35	1
	77	51.47	46.08	37.80	20.61	...	0.01	370.11	0.00	8.62	1
	78	34.17	30.02	24.89	29.42	...	0.10	413.32	0.00	6.74	1
79	44.89	37.54	27.15	56.16	...	0.32	528.69	0.00	5.10	1	

TABLE 2: Prediction result comparison.

Phase	Indices	Adam-LL-LR		LL-LR		Adam-LS-LR		LS-LR	
		Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.
Training	CAR (%)	85.27	1.76	84.52	3.82	72.52	1.69	71.81	2.52
	TP	246.95	12.48	239.70	26.87	271.85	8.71	265.65	14.59
	TN	359.30	13.68	361.25	14.25	243.75	13.76	244.90	11.57
	FP	46.85	12.77	53.15	26.97	22.50	7.68	28.25	15.63
	FN	57.90	14.13	56.90	14.71	172.90	12.92	172.20	10.72
	Precision	0.84	0.04	0.82	0.09	0.92	0.03	0.90	0.05
	Recall	0.81	0.04	0.81	0.04	0.61	0.02	0.61	0.02
	NPV	0.86	0.03	0.86	0.03	0.58	0.03	0.59	0.03
	F1 score	0.82	0.02	0.81	0.06	0.74	0.01	0.73	0.03
Testing	CAR (%)	85.32	4.64	81.90	4.06	72.03	5.37	70.82	5.92
	TP	26.90	5.19	25.25	3.94	27.60	3.29	28.10	6.03
	TN	40.50	5.68	39.45	4.78	29.30	3.45	27.85	5.17
	FP	4.30	2.26	6.90	2.62	3.05	2.42	3.00	2.02
	FN	7.30	2.76	7.40	2.91	19.05	4.34	20.05	3.73
	Precision	0.86	0.07	0.79	0.08	0.90	0.07	0.90	0.09
	Recall	0.79	0.08	0.78	0.06	0.59	0.07	0.58	0.09
	NPV	0.85	0.06	0.84	0.07	0.61	0.08	0.58	0.07
	F1 score	0.82	0.06	0.78	0.05	0.71	0.06	0.70	0.08

$$\begin{aligned}
\text{CAR} &= \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \times 100\%, \\
\text{Precision} &= \frac{\text{TP}}{\text{TP} + \text{FP}}, \\
\text{Recall} &= \frac{\text{TP}}{\text{TP} + \text{FN}}, \\
\text{NPV} &= \frac{\text{TN}}{\text{TN} + \text{FN}}, \\
\text{F1 score} &= \frac{2\text{TP}}{2\text{TP} + \text{FP} + \text{FN}},
\end{aligned} \tag{18}$$

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

The experimental results obtained from the repetitive data sampling with 20 runs are reported in Table 2. As can be seen from this table, the Adam-LL-LR has achieved the best predictive accuracy in both of the training (CAR = 85.25%, precision = 0.84, recall = 0.81, NPV = 0.86, and F1 score = 0.82) and testing phases (CAR = 85.32%, precision = 0.86, recall = 0.79, NPV = 0.85, and F1 score = 0.82). Since the prediction performances obtained from the training and testing phases of the Adam-LL-LR are relatively similar, it can be shown that this model has not suffered from overfitting. In addition, the LL-LR model is the second best approach (with CAR = 81.90% and F1 score = 0.78), followed by the Adam-LS-LR (with CAR = 72.03% and F1 score = 0.71) and the LS-LR (with CAR = 70.82% and F1 score = 0.70). Herein, the index of the F1 score is emphasized because it presents the harmonic mean of the precision and recall.

The training and testing performances of the employed models are graphically presented in Figures 8 and 9. The boxplot shown in Figure 10 demonstrates the testing

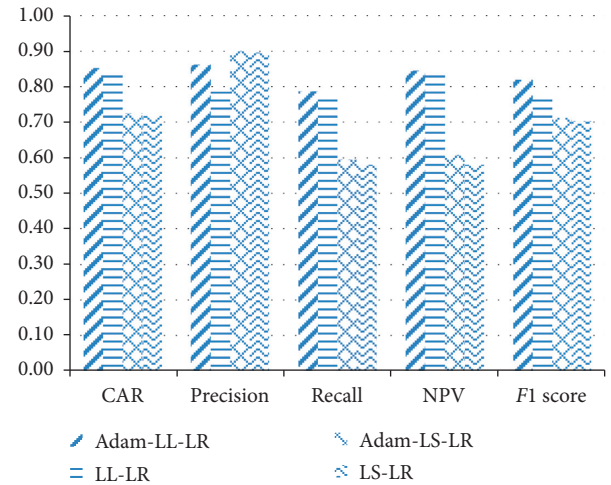


FIGURE 8: Performance measurement indices for the training phase.

performances of LRMs. In addition, to confirm the statistical difference of each pair of the localized spall detection models, the Wilcoxon signed-rank test with a significance level (p value) = 0.05 is employed in this section of the study. The test outcomes are reported in Table 3. Observably, experimental results show that all of the p values are lower than the significance level. Thus, the null hypothesis shows that the performances of the two models under testing are statistically indifferent and can be confidently rejected. This hypothesis test asserts the superiority of the Adam-LL-LR model over other benchmark approaches.

Based on the experimental result, the Adam-LL-LR model is best suited for the collected dataset at hand. The performance of this model is further studied in this section. Illustrations of correctly recognized spall objects yielded by Adam-LL-LR are presented in Figure 11. As can be observed, the model can deliver accurate detection results in the

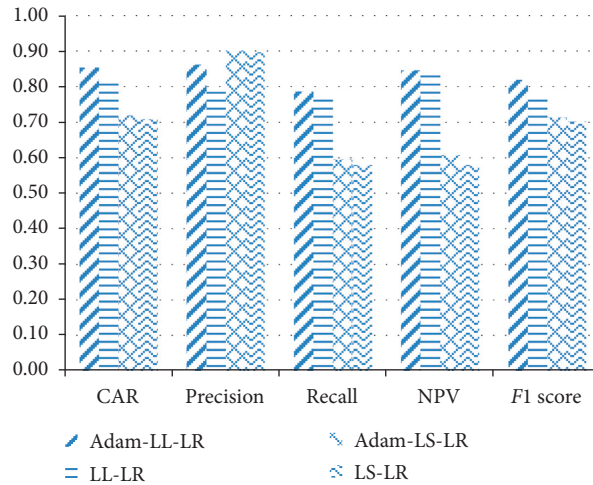


FIGURE 9: Performance measurement indices for the testing phase.

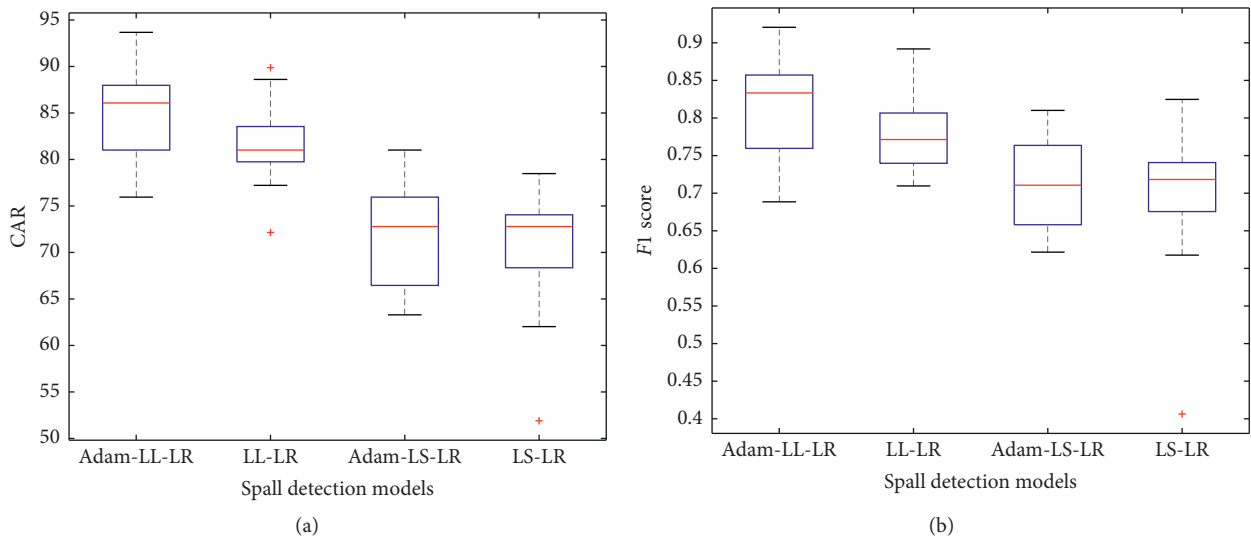


FIGURE 10: Model performances obtained from the repetitive data subsampling process: (a) CAR (%) and (b) F1 score.

TABLE 3: Wilcoxon signed-rank test results.

Indices	Model comparison	Test outcome	<i>p</i> value
CAR (%)	Adam-LL-LR vs. LL-LR	Significant	0.0184
	Adam-LL-LR vs. Adam-LS-LR	Significant	0.0001
	Adam-LL-LR vs. LS-LR	Significant	0.0001
F1 score	Adam-LL-LR vs. LL-LR	Significant	0.0400
	Adam-LL-LR vs. Adam-LS-LR	Significant	0.0004
	Adam-LL-LR vs. LS-LR	Significant	0.0001

presence of a window (Figure 11(a)) and a minor defect on the mortar surface (Figure 11(b)). Notably, the localized spall objects can still be located well in the cases that there are crack objects in the captured scenes (Figures 11(c)–11(e)). Furthermore, Adam-LL-LR has also performed well in the cases that there are multiple spall objects in the image samples (Figures 11(f) and 11(g)). In addition, the proposed

Adam-LL-LR model can be used to quantify the percentage of damaged areas found in image samples; the computation results are demonstrated in Figure 12.

Nevertheless, as shown in Figure 13, the newly developed model has made incorrect detection results in the cases of complex background. As observed in Figure 13(a), an area in the background has the texture property similar to that of the

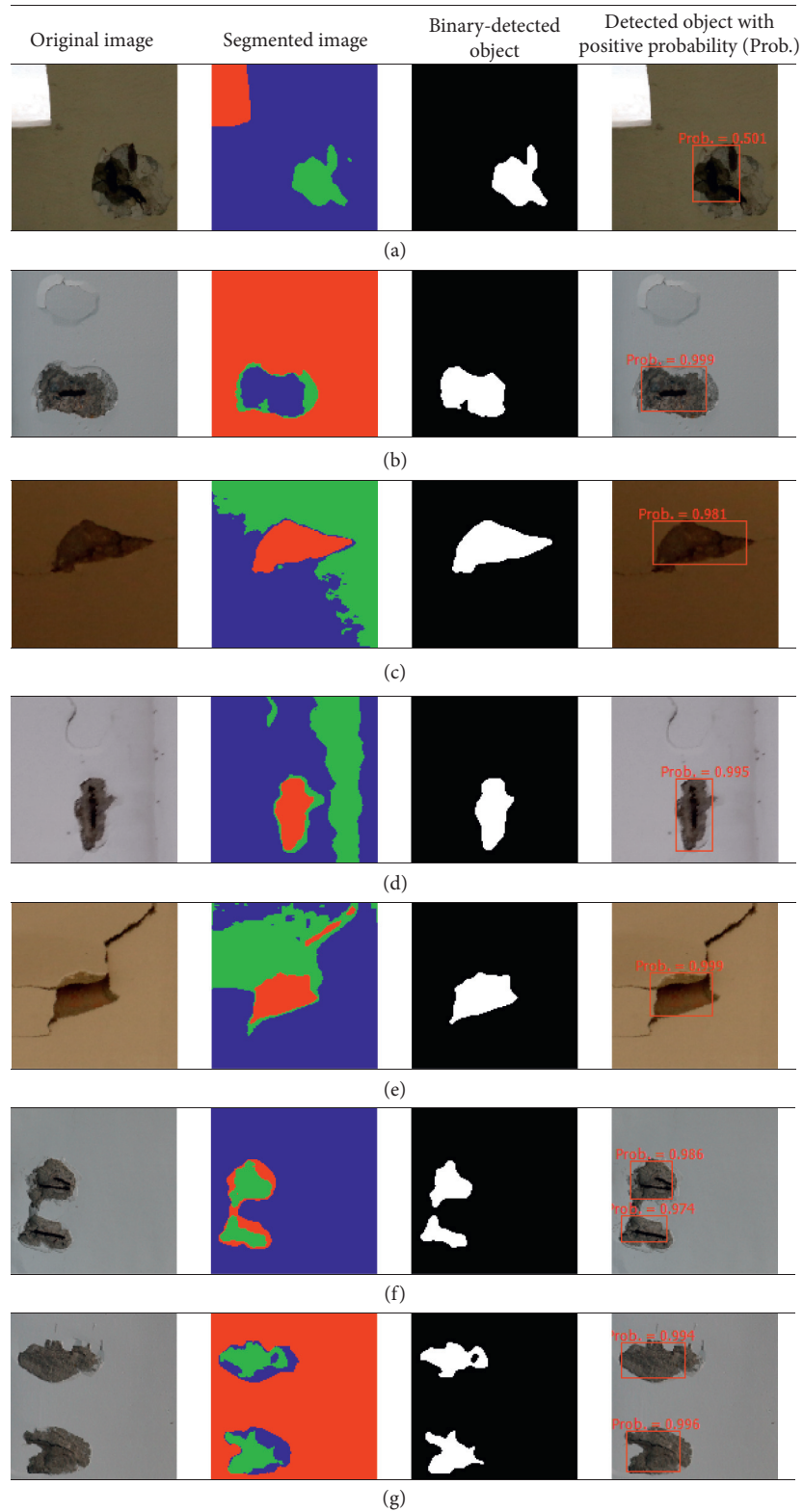


FIGURE 11: Illustrations of correctly classified cases.

Original image	Segmented image	Detected objects	Computation results
			Percentage of damaged area: 8.70%
			Percentage of damaged area of the 1 st object: 4.88%
			Percentage of damaged area of the 2 nd object: 4.84%

FIGURE 12: Demonstration of the calculation of the percentage of damaged areas.

Original image	Segmented image	Binary-detected object	Falsely detected/undetected object
(a)			
			Undetected object
(b)			
			Undetected object
(c)			

FIGURE 13: Illustrations of incorrectly classified cases.

spall object. This can lead to a false-positive detection. Complex background (Figure 13(b)) and irregular lighting conditions (Figure 13(c)) also tend to reduce the model accuracy. These phenomena can lead to false-negative cases.

6. Concluding Remarks

Localized spall is a common defect observed on surfaces of reinforced concrete elements. Accurate detection of this damage is crucial during the phase of the periodic structural health survey. This study has developed and verified a computer vision-based approach for automating the task of

localized spall recognition. The newly developed model is a hybridization of image processing and machine learning approaches. Image processing methods of the GF coupled with *k*-means clustering and morphological analyses are used to automatically identify the ROIs that potentially contain the defect. The BGC, GLCM, and color channels' properties are employed as texture descriptors. Based on the computed image texture, the LRM optimized by the state-of-the-art Adam is used to construct a decision boundary that separate the data samples into two regions of nonlocalized spall and localized spall. Experimental results show that the LRM trained by the Adam optimizer can deliver the most

desired prediction accuracy. Therefore, the proposed integrated model can be a useful tool to assist building maintenance agencies in the task of evaluating structure health condition.

Data Availability

The image dataset used to support the findings of this study has been deposited in the repository of GitHub: https://github.com/NhatDucHoang/LocalizedSpallDetection_AdamLRM.

Conflicts of Interest

The author declares that there are no conflicts of interest.

Acknowledgments

This research was financially supported by Duy Tan University.

References

- [1] M. R. Jahanshahi and S. F. Masri, "Adaptive vision-based crack detection using 3D scene reconstruction for condition assessment of structures," *Automation in Construction*, vol. 22, pp. 567–576, 2012.
- [2] B. A. Soeleman, M. V. A. D. Silva, M. Akkari, and C. Santili, "Accidents due to falls from roof slabs," *Sao Paulo Medical Journal*, vol. 131, no. 3, pp. 153–157, 2013.
- [3] S. M. Verhulst, *Spalling Is Bad, Falling Is Worse*, PiereSearch, Arlington, TX, USA, 2020, <https://piereSearch.com/spalling-is-bad-falling-is-worse/>.
- [4] M. Duckett and G. Pestine, *Concrete Spalling Robson Forensic*, <https://www.robsonforensic.com/articles/concrete-spalling-expert-article/>, 2020.
- [5] 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.
- [6] R. Zaurin, "Structural health monitoring with emphasis on computer vision, damage indices, and statistical analysis," <https://starslibrary.ucf.edu/cgi/viewcontent.cgi?article=4853&context=etdElectronicThesesandDissertations>, University of Central Florida, Orlando, FL, USA, 2009, <https://starslibrary.ucf.edu/cgi/viewcontent.cgi?article=4853&context=etdElectronicThesesandDissertations>.
- [7] I. Abdel-Qader, S. Pashaie-Rad, O. Abudayyeh, and S. Yehia, "PCA-Based algorithm for unsupervised bridge crack detection," *Advances in Engineering Software*, vol. 37, no. 12, pp. 771–778, 2006.
- [8] M. O'Byrne, F. Schoefs, B. Ghosh, and V. Pakrashi, "Texture analysis based damage detection of ageing," *Infrastructural Elements Computer-Aided Civil and Infrastructure Engineering*, vol. 28, pp. 162–177, 2013.
- [9] D. Lattanzi and G. R. Miller, "Robust automated concrete damage detection algorithms for field applications," *Journal of Computing in Civil Engineering*, vol. 28, no. 2, pp. 253–262, 2014.
- [10] D. B. Agyemang and M. Bader, "Surface crack detection using hierarchical convolutional neural network," in *Advances in Computational Intelligence Systems*, pp. 173–186, Springer International Publishing, Cham, Switzerland, 2020.
- [11] 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.
- [12] 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.
- [13] S. Dorafshan, R. J. Thomas, and M. Maguire, "Benchmarking image processing algorithms for unmanned aerial system-assisted crack detection in concrete structures," *Infrastructures*, vol. 4, no. 2, p. 19, 2019.
- [14] 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.
- [15] 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.
- [16] H. Kim, J. Lee, E. Ahn, S. Cho, M. Shin, and S.-H. Sim, "Concrete crack identification using a UAV incorporating hybrid image processing," *Sensors*, vol. 17, no. 9, p. 2052, 2017.
- [17] 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.
- [18] M.-M. Naddaf-Sh, S. Hosseini, J. Zhang, N. A. Brake, and H. Zargarzadeh, "Real-time road crack mapping using an optimized convolutional neural network," *Complexity*, vol. 2019, Article ID 2470735, 17 pages, 2019.
- [19] 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.
- [20] B. Wang, Y. Li, W. Zhao, Z. Zhang, Y. Zhang, and Z. Wang, "Effective crack damage detection using multilayer sparse feature representation and incremental extreme learning machine," *Applied Sciences*, vol. 9, no. 3, p. 614, 2019.
- [21] T. Dawood, Z. Zhu, and T. Zayed, "Machine vision-based model for spalling detection and quantification in subway networks," *Automation in Construction*, vol. 81, pp. 149–160, 2017.
- [22] 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.
- [23] 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.
- [24] H. Wu, X. Ao, Z. Chen, C. Liu, Z. Xu, and P. Yu, "Concrete spalling detection for metro tunnel from point cloud based on roughness," *Descriptor Journal of Sensors*, vol. 2019, Article ID 8574750, 12 pages, 2019.
- [25] S. German, I. Brilakis, and R. DesRoches, "Rapid entropy-based detection and properties measurement of concrete spalling with machine vision for post-earthquake safety assessments," *Advanced Engineering Informatics*, vol. 26, no. 4, pp. 846–858, 2012.

- [26] P. Prasanna, K. J. Dana, N. Gucunski et al., "Automated crack detection on concrete bridges," *IEEE Transactions on Automation Science and Engineering*, vol. 13, no. 2, pp. 591–599, 2016.
- [27] M.-K. Kim, H. Sohn, and C.-C. Chang, "Localization and quantification of concrete spalling defects using terrestrial laser scanning," *Journal of Computing in Civil Engineering*, vol. 29, Article ID 04014086, 2015.
- [28] 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.
- [29] N.-D. Hoang, "Automatic detection of asphalt pavement raveling using image texture based feature extraction and stochastic gradient descent logistic regression," *Automation in Construction*, vol. 105, Article ID 102843, 2019.
- [30] 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.
- [31] Y. Li, X. Li, H. Wang, S. Wang, S. Gu, and H. Zhang, "Exposed aggregate detection of stilling basin slabs using attention U-net network," *KSCE Journal of Civil Engineering*, vol. 24, no. 6, p. 1740, 2020.
- [32] J. K. Dey, Z. Su, J. Wu, P. S. Tan, X. Mao, and Y. H. Wang, "Anomaly detection of defects on concrete structures with the convolutional autoencoder," *Advanced Engineering Informatics*, vol. 45, Article ID 101105, 2020.
- [33] L. Wang, K. i. Kawaguchi, and P. Wang, "Damaged ceiling detection and localization in large-span structures using convolutional neural networks," *Automation in Construction*, vol. 116, Article ID 103230, 2020.
- [34] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning (Adaptive Computation and Machine Learning Series)*, The MIT Press, Cambridge, MA, USA, 2016, ISBN-10: 026203561.
- [35] P. Kim, *MatLab Deep Learning with Machine Learning, Neural Networks and Artificial Intelligence*, Apress, New York, NY, USA, 2017.
- [36] S. Skansi, *Introduction to Deep Learning—from Logical Calculus to Artificial Intelligence*, Springer International Publishing, New York, NY, USA, 2018.
- [37] A. S. Moghaddam, E. R. Azar, Y. Mejias, and H. Bell, "Estimating stripping of asphalt coating using k-means clustering and machine learning-based classification," *Journal of Computing in Civil Engineering*, vol. 34, Article ID 04019044, 2020.
- [38] L. G. Shapiro and G. C. Stockman, *Computer Vision*, Prentice-Hall, Upper Saddle River, NJ, USA, 2001.
- [39] M. Sonka, V. Hlavac, and R. Boyle, *Image Processing, Analysis, and Machine Vision*, Cengage Learning, Printed in the United States of America, Boston, MA, USA, 2013, ISBN-10: 1-133-59360-7.
- [40] S. Molaei and M. E. Shiri Ahmad Abadi, "Maintaining filter structure: a Gabor-based convolutional neural network for image analysis," *Applied Soft Computing*, vol. 88, Article ID 105960, 2020.
- [41] A. Premana, A. P. Wijaya, and M. A. Soeleman, "Image segmentation using Gabor filter and K-means clustering method," in *Proceedings of the 2017 International Seminar on Application for Technology of Information and Communication (iSemantic)*, pp. 95–99, Semarang, Indonesia, October 2017.
- [42] 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, pp. 342–358, 2014.
- [43] K. Hammouda, *Texture Segmentation Using Gabor Filters Technical Report*, University of Waterloo, Waterloo, Canada, 2000, <http://www.pamiuwaterloo.ca/pub/hammouda/sd775-paperpdf>.
- [44] N. C. Kim and H. J. So, "Directional statistical Gabor features for texture classification," *Pattern Recognition Letters*, vol. 112, pp. 18–26, 2018.
- [45] A. K. Jain and F. Farrokhnia, "Unsupervised texture segmentation using Gabor filters," *Pattern Recognition*, vol. 24, no. 12, pp. 1167–1186, 1991.
- [46] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the Fifth Berkeley Sympos Math Statist and Probability (Berkeley, Calif, 1965/66) Vol I: Statistics*, pp. 281–297, Univ California Press, Berkeley, CA, USA, 1967.
- [47] R. Duda, P. Hart, and D. Stock, *Pattern Classification*, John Wiley & Sons, Hoboken, NJ, USA, 2001, ISBN-13: 978-0471056690.
- [48] A. Fernández, M. X. Álvarez, and F. Bianconi, "Image classification with binary gradient contours," *Optics and Lasers in Engineering*, vol. 49, no. 9-10, pp. 1177–1184, 2011.
- [49] F. Alaei, A. Alaei, U. Pal, and M. Blumenstein, "A comparative study of different texture features for document image retrieval," *Expert Systems with Applications*, vol. 121, pp. 97–114, 2019.
- [50] S. Theodoridis and K. Koutroumbas, *Pattern Recognition*, Academic Press, Printed in the United States of America, Cambridge, MA, USA, 2009, ISBN 978-1-59749-272-0.
- [51] R. M. Haralick, K. Shanmugam, and I. H. Dinstein, "Textural features for image classification," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-3, no. 6, pp. 610–621, 1973.
- [52] R. M. Haralick and L. G. Shapiro, *Computer and Robot Vision*, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1992, ISBN 0201569434.
- [53] G. M. Hadjidemetriou, P. A. Vela, and S. E. Christodoulou, "Automated pavement patch detection and quantification using support vector machines," *Journal of Computing in Civil Engineering*, vol. 32, Article ID 04017073, 2018.
- [54] N.-D. Hoang, "Image processing-based pitting corrosion detection using metaheuristic optimized multilevel image thresholding and machine-learning approaches," *Mathematical Problems in Engineering*, vol. 2020, Article ID 6765274, 19 pages, 2020.
- [55] F. Tomita and S. Tsuji, *Computer Analysis of Visual Textures*, Springer Science + Business Media, New York, NY, USA, 1990, ISBN 978-1-4612-8832-9.
- [56] M. Andrejiova, A. Grincova, and D. Marasova, "Failure analysis of rubber composites under dynamic impact loading by logistic regression," *Engineering Failure Analysis*, vol. 84, pp. 311–319, 2018.
- [57] H. C. Chan, C. C. Chang, P. A. Chen, and J. T. Lee, "Using multinomial logistic regression for prediction of soil depth in an area of complex topography in Taiwan," *Catena*, vol. 176, pp. 419–429, 2019.
- [58] W. W. Piegorsch, *Statistical Data Analytics: Foundations for Data Mining, Informatics, and Knowledge Discovery*, John Wiley & Sons, Hoboken, NJ, USA, 2015, ISBN 978-1-118-61965-0.
- [59] A. Agresti, *An Introduction to Categorical Data Analysis*, John Wiley & Sons, Hoboken, NJ, USA, 2019, ISBN 9781119405283.

- [60] S. J. Russell and P. Norvig, *Artificial Intelligence—A Modern Approach*, Pearson Education, Upper Saddle River, NJ, USA, 2010.
- [61] M. Gormley, *Logistic Regression*, <https://www.cs.cmu.edu/~7Emgormley/courses/10701-f16/slides/lecture5.pdf>, 2016.
- [62] A. Ng, *Lecture Notes*, Stanford University, Kunnamangalam, Kerala, 2018, <http://cs229.stanford.edu/notes/cs229-notes1.pdf>.
- [63] 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.
- [64] J. Brownlee, *Gentle Introduction to the Adam Optimization Algorithm for Deep Learning Machine Learning Mastery*, <https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning/>, 2017.
- [65] Accord, “Accord. NET Framework,” 2019, <http://accord-framework.net/>.
- [66] R. M. Haralick, “Some neighborhood operators,” in *Real-Time Parallel Computing: Imaging Analysis*, M. Onoe, K. Preston, and A. Rosenfeld, Eds., Springer, Boston, MA, USA, pp. 11–35, 1981.
- [67] 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.