Concrete Spalling Severity Classification Using Image Texture Analysis and a Novel Jellyfish Search Optimized Machine Learning Approach

Nhat-Duc Hoang,1,2 Thanh-Canh Huynh,1,2 and Van-Duc Tran2,3

1Institute of Research and Development, Duy Tan University, Da Nang 550000, Vietnam
2Faculty of Civil Engineering, Duy Tan University, Da Nang 550000, Vietnam
3International School, Duy Tan University, Da Nang 550000, Vietnam

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

Received 17 May 2021; Revised 13 September 2021; Accepted 7 November 2021; Published 10 December 2021

Copyright © 2021 Nhat-Duc Hoang et al. 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.

During the phase of building survey, spalling and its severity should be detected as earlier as possible to provide timely information on structural health to building maintenance agency. Correct detection of spall severity can significantly help decision makers develop effective maintenance schedule and prioritize their financial resources better. This study aims at developing a computer vision-based method for automatic classification of concrete spalling severity. Based on input image of concrete surface, the method is capable of distinguishing between a minor spalling in which the depth of the broken-off material is less than the concrete cover layer and a deep spalling in which the reinforcing steel bars have been revealed. To characterize concrete surface condition, image texture descriptors of statistical measurement of color channels, gray-level run length, and center-symmetric local binary pattern are used. Based on these texture-based features, the support vector machine classifier optimized by the jellyfish search metaheuristic is put forward to construct a decision boundary that partitions the input data into two classes of shallow spalling and deep spalling. A dataset consisting of 300 image samples has been collected to train and verify the proposed computer vision method. Experimental results supported by the Wilcoxon signed-rank test point out that the newly developed method is highly suitable for concrete spall severity classification with accuracy rate $\approx 93.33\%$, $F_1$ score $\approx 0.93$, and area under the receiver operating characteristic curve $= 0.97$.

1. Introduction

Spalling is a notable defect widely encountered in surface of reinforced concrete structures (refer to Figure 1). The appearance of spalling significantly deteriorates the integrity and durability of reinforced concrete elements. This defect can be caused by severe servicing environment and loads. More importantly, the appearance of spalls may indicate more serious damages in the internal structure of reinforced concrete elements, e.g., corrosion of steel reinforcement.

Spalling should be detected as earlier as possible due to several reasons. First, spall objects badly affect the aesthetics of building structures and therefore bring about discomfort for occupants. Second, if the layer of concrete cover is removed due to spalling, reinforcing steel bars are exposed to the environment and this fact expedites the corrosion of the steel bars (as shown in Figure 1(b)). Subsequently, the area and the depth of spall objects increase over time. Third, spalling appeared in ceilings, cladding structures, or concrete beams is particularly hazardous for occupants. The materials broken off from spalled areas can cause significant injuries and even loss of human lives.

As a consequence, periodic visual inspection is necessary to detect and evaluate the severity of spalling defects. Most importantly, deep spalling in which the layer of concrete cover has completely broken off and steel reinforcement has been exposed should be detected timely and requires urgent remedy. In Vietnam as well as in other developing countries, visual inspection performed by human technicians and manual visual data processing are the main approaches for
spall detection. Although these approaches can help to detect and evaluate the severity of this distress accurately, they are also notoriously known to be labor and time-consuming. With a large surface area of concrete structure, timely inspection and fast visual data processing are virtually impossible for a limited number of inspectors.

Therefore, maintaining good serviceability via periodic visual inspection and evaluation is crucial to keep building environment operational and protect occupants’ health. In recent years, due to the availability of low-cost digital cameras as well as a rapid improvement of image processing techniques, computer vision-based structural health monitoring systems have been increasingly used to enhance the productivity of periodic building survey [1–3]. These systems have been demonstrated to be viable tools for building defect detections. They are capable of not only producing acceptable detection accuracy but also guaranteeing consistency in assessment outcomes. The computer vision-based methods yield objective evaluation results; they are not affected by subjective judgments in data processing performed by humans.

Due to the aforementioned advantages, various automated and data-driven methods used for concrete spalling detection have been constructed in the literature. German et al. [4] constructed an automated model for detecting spalled regions on the surfaces of concrete columns based on a local entropy-based thresholding algorithm, a global adaptive thresholding algorithm, and morphological operations; the model is tested with concrete columns during a postearthquake investigation. Dawood et al. [5] proposed a computer vision-based approach for spalling detection and quantification in subway networks; this study employs various image processing techniques including image thresholding, histogram equalization, and filtering in an attempt to detect the quantify the severity of spall objects. This computer vision-based method is validated with a set of 75 image samples and attains an accuracy rate of 89.3%.

Hoang [6] relied on a steerable filter used for feature extraction and machine learning-based data classification to recognize wall defects including concrete spalling. The method of roughness descriptor based on Hough transformation and similarity analysis is described in Wu et al. [7]; this approach is utilized for recognizing concrete spalling occurring in metro tunnel surface. A model that integrates image processing techniques of texture analysis and machine learning has been proposed in Hoang et al. [8]; a piecewise linear stochastic gradient descent logistic regression has been used to categorized images of concrete surface into two classes of “nonspall” and “spall.”

Abdelkader et al. [9] harnessed the capability of particle swarm optimization metaheuristic coupled with the Tsallis entropy function and discrete wavelet transform to automate the detection of spalling area. Hoang [10] developed an image processing-based spall object detection method relying on Gabor filter for region of interest extraction, texture analysis methods for characterizing feature of concrete surface, and logistic regression models used for data classification; this integrated approach can effective locate the spall objects but is not capable of classifying spall severity.


Recently, deep learning methods have also been applied to tackle the problem of interest. The main advantage of the deep learning models is that the feature extraction phase can be performed automatically [13, 14]. Through various convolutional and pooling operations, useful features such as edges, shapes, texture, and so on can be revealed by the machine and used for the subsequent pattern recognition tasks in a fully connected layer [15]. Wei et al. [16] proposed deep learning-based recognition and quantification for concrete surface bughole; the employed artificial intelligent

**Figure 1:** Appearances of spall in reinforced concrete surface: (a) shallow spall and (b) deep spall.
method is convolutional neural network (CNN); the main research finding is that the CNN-based model can effectively replace the traditional detection methods carried out by manual inspection.

Another deep learning-based concrete surface void detection method has been put forward in [17]; this method is trained by small-sized image of 28 × 28 pixels, and its performance outperformed conventional image processing techniques of the Laplacian of Gaussian algorithm and the Otsu method. A CNN-based method used for detecting building defects has been developed in [18]; this method is capable of automatically detecting and localizing key building defects such as mould, deterioration, and stain.

Although CNN-based methods are generally capable tools for detecting spalling and other defects in concrete surface, the deep learning approaches typically demand a large volume of image datasets in order to construct reliable classifiers [13, 19]. This fact requires a great effort in visual data collection and a meticulous data labeling process. In addition, successful implementation of deep learning models also necessitates experience and the trial-and-error process to adjust a significant number of model tuning parameters.

In general, based on recent reviewing works performed by Koch et al. [20]; Feng and Feng [21]; Dong and Catbas [22]; and Yadhunath et al. [23], there is an increasing trend of applying image processing and machine learning for automatically detecting concrete surface distresses including spall. Therefore, investigations of other image processing tools and machine learning frameworks are helpful to provide a broader view on the possibility and capability of computer vision methods in dealing with the task at hand. It is also noted that although various models for spall object detection have been put forward and verified, few studies have constructed spall severity classification models based on two-dimensional digital images. Such models can be immensely helpful for the decision maker and building maintenance agencies to schedule their maintenance and prioritize their budgets spent on treatment of building elements effectively.

In addition, although machine learning methods have been extensively used in computer vision-based structural health monitoring [3, 12, 24–26], hybrid approaches that combine the strengths of machine learning and metaheuristic algorithms are rarely investigated in this field especially for concrete spall recognition. Metaheuristic algorithms can be used to optimize the learning phase of machine learning models and therefore help to achieve better predictive performances [27–33].

Accordingly, the current study aims at contributing to the body of knowledge by constructing a hybrid machine learning and metaheuristic approach used for computer vision-based concrete spall severity recognition. The employed machine learning method is support vector machine [34] because SVM has been proven to be a highly capable tool for pattern recognition especially for nonlinear and multivariate datasets [35–40]. To optimize the performance of SVM, a novel and recently proposed metaheuristic approach of jellyfish search is utilized.

The jellyfish search metaheuristic algorithm is employed to identify the most suitable tuning parameters of the SVM model that yields the desired predictive performance on reinforced concrete spall severity recognition. SVM is used in this study to recognize concrete surface subject to the defects of shallow spall and deep spall. Herein, the first class represents spall objects with its depth smaller than the concrete cover; the latter class contains spall objects having their embedded reinforcement exposed.

Moreover, since the areas of the aforementioned classes have different surfacing properties such as coarseness/finesse, image texture descriptors of statistical measurements of color channels [41], gray-level run length [42, 43], and center-symmetric local binary pattern [44] are used to characterize the surface properties of concrete used for spall severity classification. These texture descriptors have been selected by this study due to their ease of implementation, fast computation, and good discriminative capability [8, 45–50]. In addition, as demonstrated in previous studies [25, 51, 52], the combination of image’s color properties and texture is able to bring about good image classification accuracy.

In summary, the main contribution of the current study to the body of knowledge can be stated as follows:

(i) This study proposes and verifies a computer vision-based method that is capable of categorizing concrete spall severity. This approach can significantly boost the productivity and effectiveness of the periodic survey on the structural health of concrete elements.

(ii) The proposed approach is a hybridization of JSO metaheuristic and SVM. The JSO algorithm is used to optimize the SVM training phase automatically.

(iii) The integration of various texture descriptors, which include statistical measurements of color channels, gray-level run length, and center-symmetric local binary pattern, aims at describing the surface feature of concrete surface effectively.

(iv) The computer vision-based method is trained and optimized automatically with minimum human intervention and effort on parameter fine-tuning.

The subsequent sections of the study are organized as follows. Section 2 reviews the research methodology. The next section describes the structure of the proposed computer vision-based approach employed for spall severity classification. Experimental results are reported in Section 4. Concluding remarks and main research findings are summarized in the last section of the article.

2. Research Methodology

This section of the article presents the research methodology of the current study. The research methodology includes four main sections: image acquisition, image texture computation, model optimization, and model construction. The overall research methodology is depicted in Figure 2. The subsequent parts of this section review the image texture descriptors used for feature extraction, the machine learning, and the metaheuristic algorithm employed for model optimization.
2.1. The Employed Image Texture Descriptors. It is observable that surfacing properties of concrete with different categories of spalling severity can be used for pattern classification. Therefore, this study relies on the statistical measurement of image pixel intensity [41], the gray-level run length [42, 43], and the center-symmetric local binary pattern for concrete spall severity classification [44].

2.1.1. Statistical Measurement of Image Pixel Intensity. This study relies on 2-dimensional RGB image samples to recognize concrete spall severity. One image sample has three color channels of red (R), green (G), and blue (B) and is commonly represented by three separated matrices, each of which contains information of pixel intensity in one color channel. To extract the statistical
measurements of image pixel intensity of an image sample \( I \), it is necessary to establish the first-order histogram \( P(I) \) describing the statistical distribution of pixels’ gray level. Using \( P(I) \), the metrics of mean (\( \mu_c \)), standard deviation (\( \sigma_c \)), skewness (\( \delta_c \)), kurtosis (\( \eta_c \)), entropy (\( \rho_c \)), and range (\( \Delta_c \)) are computed for a color channel \( c = [R, G, B] \). Since each color channel yields six statistical measurements, the total number of features describing the pixel intensity distribution of one image sample with three color channels is \( 6 \times 3 = 18 \).

The indices of mean (\( \mu_c \)), standard deviation (\( \sigma_c \)), skewness (\( \delta_c \)), kurtosis (\( \eta_c \)), entropy (\( \rho_c \)), and range (\( \Delta_c \)) are obtained in the following equations [51]:

\[
\begin{align*}
\mu_c &= \frac{\sum_{i=0}^{N_l-1} I_{i,c} \times P_c(I)}{\sum_{i=0}^{N_l-1} P_c(I)}, \\
\sigma_c &= \sqrt{\frac{\sum_{i=0}^{N_l-1} (I_{i,c} - \mu_c)^2 \times P_c(I)}}{\sum_{i=0}^{N_l-1} P_c(I)}, \\
\delta_c &= \frac{\sum_{i=0}^{N_l-1} (I_{i,c} - \mu_c)^3 \times P_c(I)}{\sigma_c^3}, \\
\eta_c &= \frac{\sum_{i=0}^{N_l-1} (I_{i,c} - \mu_c)^4 \times P_c(I)}{\sigma_c^4}, \\
\rho_c &= -\sum_{i=0}^{N_l-1} P_c(I) \times \log_2(P_c(I)), \\
\Delta_c &= \text{Max}(I_c) - \text{Min}(I_c),
\end{align*}
\]

where \( N_l = 256 \) represents the number of discrete intensity values, \( c \) is the index of color channels (R, G, or B), and \( P(I) \) denotes the first-order histogram of an image.

2.1.2. Gray-Level Run Length (GLRL). GLRL, proposed in [42], is a powerful method for extracting statistical properties of spatial distribution of gray levels. This method utilizes higher-order statistics that analyze the joint distribution of multiple pixels [48]. First, GLRL matrices are computed from a gray-scale image. Subsequently, the occurrence of runs of pixels in a given direction is inspected and statistically quantified. GLRL is useful for characterizing the coarseness or fineness of image region due to the observation that coarse textures are presented by a large number of neighboring pixels featuring the same gray intensity. On the contrary, a small number of neighboring pixels with similar gray-level intensity are observed in fine textures. Given an image of interest, the GLRL constructs a run-length matrix as the number of runs that stems from a location \((i, j)\) of the image in a certain direction [47]. Commonly, for one image sample, four GLRL matrices are computed for the horizontal direction, vertical directions, and two diagonal directions [53].

Let \( p(i, j) \) denote a run-length matrix, and the short run emphasis (SRE), long run emphasis (LRE), gray-level non-uniformity (GLN), run length nonuniformity (RLN), and run percentage (RP) are calculated as follows [19, 42]:

\[
\begin{align*}
\text{SRE} &= \frac{1}{N_r} \sum_{i=1}^{N} \sum_{j=1}^{M} \frac{p(i, j)}{j^2}, \\
\text{LRE} &= \frac{1}{N_r} \sum_{i=1}^{N} \sum_{j=1}^{M} p(i, j) \times j^2, \\
\text{GLN} &= \frac{1}{N_r} \sum_{i=1}^{N} \left( \sum_{j=1}^{M} p(i, j) \right)^2, \\
\text{RLN} &= \frac{1}{N_r} \sum_{i=1}^{N} \left( \sum_{j=1}^{M} p(i, j) \right)^2, \\
\text{RP} &= \frac{N_r}{P_r},
\end{align*}
\]

where \( M \) and \( N \) denote the number of gray levels and the maximum run length, \( N_r \) represents the total number of runs and \( N_p \) is the number of pixels, and \( i \) and \( j \) denote the coordinates of a pixel within an image sample.

In addition to the aforementioned indices, Chu et al. [54] proposed the low gray-level run emphasis (LGRE) and high gray-level run emphasis (HGRE) described as follows:

\[
\begin{align*}
\text{LGRE} &= \frac{1}{N_r} \sum_{i=1}^{N} \sum_{j=1}^{M} p(i, j) \times i^2, \\
\text{HGRE} &= \frac{1}{N_r} \sum_{i=1}^{N} \sum_{j=1}^{M} p(i, j) \times j^2.
\end{align*}
\]

Dasarathy and Holder [55] put forward additional indices extracted from GLRL matrices. These indices are the 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); their equations are given by

\[
\begin{align*}
\text{SRLGE} &= \frac{1}{N_r} \sum_{i=1}^{N} \sum_{j=1}^{M} p(i, j) \times i^2 \times j^2, \\
\text{SRHGE} &= \frac{1}{N_r} \sum_{i=1}^{N} \sum_{j=1}^{M} p(i, j) \times i^2 \times j^2, \\
\text{LRHGE} &= \frac{1}{N_r} \sum_{i=1}^{N} \sum_{j=1}^{M} p(i, j) \times i^2 \times j^2.
\end{align*}
\]
2.1.3. Center-Symmetric Local Binary Pattern (CS-LBP).

CS-LBP, proposed in [44], is a modified version of the standard local binary pattern (LBP) [56, 57]. CS-LBP inherits the capability of LBP in describing the texture of an interest region via the distribution of its local structures as well as the intolerance against illumination changes. Both CS-LBP and LBP are widely recognized as simple yet effective texture descriptors [50, 58]. Nevertheless, one major drawback of the original LBP is that it yields a long histogram and therefore produces a large number (i.e., 256) of features to be learnt. A large number of data dimensions usually impose a significant challenge for machine learning model which relies on the data to construct classifiers of interest [59, 60]. Furthermore, the standard texture descriptor is not robust in describing flat image regions [44, 58].

To improve the performance of LBP, CS-LBP is devised by proposing a new scheme of pairwise pixel comparison as shown in Figure 3. Given a patch of ×3 pixels, the CS-LBP compares center-symmetric pairs of pixels in the neighborhood to yield different binary patterns. The function $\Delta$ is employed to inspect the significance of the gray intensity differences of 2 pixels.

The center-symmetric pairs of pixels are compared to characterize the local structure of image texture. Therefore, the total number of extracted features is only 16 instead of 256 as required by LBP. In addition, to improve the robustness on flat image region, a thresholding value $T$ is used to determine the significance of the gray-level differences between two pixels of interest. The thresholding value $T$ is commonly set to be 3 as suggested in [61]. Accordingly, the formula used to compute the CS-LBP descriptor is given by

$$\Delta (x) = \begin{cases} 1, & \text{if } x > T, \\ 0, & \text{otherwise}, \end{cases}$$

(5)

where $T$ denotes a thresholding value employed to inspect the significance of the gray intensity differences of 2 pixels.

The formula used to compute the CS-LBP descriptor is given by

$$F_{\text{CS-LBP}}(i, k) = \sum_{i=0}^{(N/2)-1} \Delta(p_i - p_{i+(N/2)}) \times 2^k,$$

(6)

where $i$ and $k$ denote the coordination of a pixel within an image sample and $N=8$ which is the number of neighboring pixels.

2.2. Support Vector Machine Classification (SVC).

The pattern recognition method of SVC was first proposed in [34]. This method is a highly effective tool suitable for dealing with classification tasks in high-dimensional space. In this study, SVC is used to categorize the input image data into two class labels of deep and shallow concrete spalling. Let $D = \{(x, y) | x \in S \text{ and } y = f(x)\}$ denote a training dataset. Herein, the input feature refers to numerical data extracted by the aforementioned texture descriptors of the statistical measurement of color channels, GLRL, and CS-LBP. Using SVC, an approximated function $\hat{f}(x): X \rightarrow \{-1, +1\}$ can be established with the label $-1$ which means “shallow spall” and the label $+1$ which corresponds to “deep spall.”

To cope with nonlinear separable datasets, SVC relies on kernel functions to construct a mapping from the original input space to a high-dimensional feature space within which linear separation of datasets is feasible. The data mapping and the construction of a hyperplane used for data separation are demonstrated in Figure 4. To establish such hyperplane, the following nonlinear optimization problem must be solved:

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

subjected to $y_k (w^T \varphi(x_k) + b) \geq 1 - e_k, \quad k = 1, \ldots, N, e_k \geq 0,$

(7)

where $w \in \mathbb{R}^n$ and $b \in \mathbb{R}$ are the parameters of the hyperplane, $e$ denotes the vector of slack variables, and $C$ and $\varphi(x)$ represent the penalty coefficient and the nonlinear data mapping function, respectively.

In the formulation of a SVC model, the explicit form of the mapping function is not required. Instead of that, the dot product of $\varphi(x)$ denoting a kernel function $K(x_k, x_l)$ can be obtained. For nonlinear pattern recognition, the kernel function of choice is the radial basis kernel function (RBKF) [62]; its formula is given by

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

(8)

where $\sigma$ denotes a hyperparameter of the kernel function.

2.3. Jellyfish Search (JS) Metaheuristic.

As can be shown in the previous section, the establishment of a SVC model used for spall severity classification requires a suitable determination of the penalty coefficient $C$ and the kernel function-based data mapping which is reflected in the tuning parameter $\sigma$ of the RBKF. The penalty coefficient $C$ indicates the amount of penalty suffered by misclassified data samples during the model training phase; the tuning parameter $\sigma$ of the RBKF controls the locality of the kernel function which influences the generalization of a SVC model [63].

It is noted that the task of searching for those hyperparameters can be considered as a global optimization problem [28, 32, 64–71]. Moreover, since $C$ and $\sigma$ are searched in continuous space, the number of parameter combinations is infinitely large. This fact makes an exhaustive search for the best
hyperparameters infeasible. Therefore, this study employs on the JS metaheuristic to tackle such optimization problem.

The JS metaheuristic, proposed in [72], is a nature-inspired algorithm highly suitable for solving global optimization problems. This metaheuristic is motivated by the behaviors of jellyfish in the ocean. Herein, each searching agent is modeled as a jellyfish. The movements of searching agents in an artificial ocean which is the search space of interest mimic their actual movements in the real-world ocean which are governed by the ocean current, the motion within a swarm, and a time control mechanism for motion mode changing.

Chou and Truong [72] proposed three idealized rules to formulate the JS optimization algorithm. The first rule is that the jellyfish may either follow the ocean current or change their locations within a swarm and there is a time control function that governs their switching of motion type. The second rule is that the jellyfish alter their location in order to search for better food source. The third rule is that the fitness of a location (reflected in the value of the cost function) as well as the jellyfish at this location is proportionate to the amount of food.

After a swarm of jellyfish is randomly generated, the searching agents start to explore and exploit the artificial ocean to search for better food source. The first type of jellyfish movement is following the ocean current. Herein, the direction of the current is expressed as follows:

$$T = X_{\text{Best}} - \beta \times \text{rand} \times \frac{\sum_{i=1}^{N_J} X_{ij}}{N_J}$$

where $T$ denotes the direction of the ocean current, $X_{\text{Best}}$ is the location of the current best jellyfish, $N_J$ is the number of jellyfish, $\beta = 3$ is the scaling factor, and rand denotes a uniform random number within $[0, 1]$. 

Accordingly, the location of a jellyfish is updated via

$$X_i(t + 1) = X_i(t) + \text{rand} \times T$$

Inside a swarm, jellyfish demonstrate both passive and active motions [73, 74]. Initially, when a swarm has just been established, the jellyfish tend to exhibit passive motion. Subsequently, the jellyfish have the tendency to follow active motion. The passive motion is mathematically formulated as follows:

$$X_i(t + 1) = X_i(t) + \gamma \times \text{rand} \times (UB - LB),$$

where $\gamma = 0.1$ is a motion coefficient and LB and UB are the lower and upper boundaries of the search variables.

The active motion of jellyfish is determined by the quantity of food stored in a randomly selected location. Generally, jellyfish approach a better food source in a swarm. The location of an individual within a swarm is iteratively revised as follows:

$$X_i(t + 1) = X_i(t) + \text{rand} \times DJ,$$

$$DJ = \begin{cases} X_i(t) - X_i(t), & \text{if } f(X_i) \geq f(X_j), \\ X_i(t) - X_j(t), & \text{if } f(X_i) < f(X_j), \end{cases}$$

where DJ denotes the direction of a jellyfish, $X_i$ is the target jellyfish, $X_j$ is a randomly selected jellyfish within the swarm, and $f$ denotes the cost function of the problem of interest.

Furthermore, to govern the movement of jellyfish between following the ocean current and moving inside the swarm, the time control mechanism including a time control function $c(t)$ and a constant $C_0 = 0.5$ is employed. The time control function yields a random value ranging from 0 to 1. If the value of $c(t)$ surpasses $C_0$, the jellyfish attach to the ocean current. On the contrary, the jellyfish move within a swarm. The time control function is mathematically described as follows:

$$c(t) = \left(1 - \frac{t}{T_{\text{Max}}}ight) \times (2 \times \text{rand} - 1),$$

where $T_{\text{Max}}$ denotes the maximum number of searching iterations.
objective, this study has carried out field surveys in Danang city (Vietnam) to collect image samples of reinforced concrete surface. This image set includes two categories of shallow spalling and deep spalling. The first category consists of spalling objects in which the depth of spalling is smaller than the concrete cover layer. The second category includes spalling objects in which reinforcing bars have been exposed to the outside environment.

The total number of collected image samples is 300; the number of data in each category is 150 to ensure a balanced classification problem. The collected image samples are illustrated in Figure 5. It is noted that the image samples have been captured by the 18-megapixel resolution Canon EOS M10 and the 16.2-megapixel resolution Nikon D5100. The labels of the image data have been assigned by human inspectors.

3. The Computer Vision-Based Jellyfish Search Optimized Support Vector Classification (JSO-SVC) for Concrete Spalling Severity Classification

This section of the article aims at describing the overall structure of the proposed computer vision-based approach used for automatic classification of concrete spalling severity. The overall structure of the newly developed approach consists of three modules: (i) image texture computation, (ii) JS-based model optimization, and (iii) SVC-based spalling severity categorization based on input image samples. Figure 6 demonstrates integrated modules of the proposed model named as JSO-SVC. The proposed method for automatic classification of concrete spalling severity is an incorporation of image texture descriptions, supervised machine learning-based pattern recognition, and stochastic search-based model optimization.

The image texture description methods of statistical measurements of pixel intensity, GLRL, and CS-LBP are used to extract texture-based features from the collected digital images. The SVC pattern recognizer assisted by the JS stochastic search is employed to establish a class boundary that divided the input feature space into two categories of “shallow spall” and “deep spall.” The role of the JS stochastic search is to optimize the parameter setting of the SVC model. It is noted that the texture computation module has been developed by the authors in Microsoft Visual Studio with Visual C# .NET. Furthermore, the SVC model optimized by the JS algorithm is coded in MATLAB environment with the help of the Statistics and Machine Learning Toolbox [75] and the source code of JS which can be accessed at [76]. The optimized computer vision-based model which relies on the module of texture computation and the JS-SVC model has been coded and compiled in Visual C# .NET Framework 4.7.2 and the built-in functions provided by the Accord.NET Framework [77].

To characterize the properties of concrete surface, this study relies on texture description methods of statistical measurements of pixel intensity, GLRL, and CS-LBP. The first texture descriptor measures statistical indices of the three color channels (red, green, and blue). For each channel, six indices of mean, standard deviation, skewness, kurtosis, entropy, and range are computed. Therefore, the first descriptor produces $3 \times 6 = 18$ features. Moreover, since one objective of the study is to detect the appearance of reinforcing bars within an image sample, the occurrence of runs of pixels in a given direction can be useful. Thus, it is beneficial to employ the GLRL approach in the feature extraction phase. Four GLRL matrices with orientations of $0^\circ$, $45^\circ$, $90^\circ$, and $135^\circ$ are computed, each of which yields 11 statistical measurements. Accordingly, the GLRL descriptor produces $4 \times 11 = 44$ features. Finally, the CS-LBP texture description method is computed to characterize the local pattern of image regions. It is noted that to compute the CS-LBP, the number of neighboring pixels around a central pixel is 8. In other words, the radius of this texture descriptor is 1 pixel. As mentioned earlier, the CS-LBP yields 16 texture-based features. Accordingly, the total number of texture-based features used for spall severity classification is $18 + 44 + 16 = 78$. The texture computation processes for the two class labels of interest area are demonstrated in Figure 7.

To train and validate the proposed JSO-SVC model, the collected dataset has been randomly partitioned into a training set (90%) and a testing set (10%). The training set is used for model construction phase; the testing set is reserved for inspecting the model predictive capability when predicting unseen data sample. In addition, prior to the model training phase, the Z-score normalization is commonly employed to preprocess the extracted features [78]. Accordingly, all of the extracted features are approximately centered at 0 and have a unit standard deviation. The Z-score equation is given by

$$X_Z = \frac{X_D - M_X}{\text{STD}_X},$$

(14)

where $X_Z$ and $X_D$ denote the normalized and the original input data, respectively, and $M_X$ and STD$_X$ represent the mean value and the standard deviation of the original input data, respectively.

In addition, the jellyfish stochastic search with 20 jellyfish is used to assist the SVC training phase. It is noted that the number of optimization iterations of the JS metaheuristic is 100. The JS algorithm’s parameters including the scaling factor ($\beta$), the motion coefficient ($\gamma$), and the parameter $C_0$ of the time control function are set to be 3, 0.1, and 0.5 according to the suggestions of Chou and Truong [72].

This stochastic search engine optimizes the model selection of the SVC model used for spall severity classification via an appropriate setting of the model hyperparameters. Through operations based on ocean current following and motions within a swarm, a population of jellyfish gradually explores and exploits an artificial ocean and identifies a good set of the penalty coefficient and the RBKF parameter. Herein, the lower and upper boundaries of the searched variables are [0.1, 0.01] and [1000, 1000], respectively. Furthermore, to effectively optimize the machine learning model, a 5-fold cross-validation-based objective function has been employed. This objective function of the JSO-SVC is given by [19]
where FNR_k and FPR_k denote false negative rate (FNR) and false positive rate (FPR) computed in the kth data fold, respectively.

The FNR and FPR metrics are given by
\[
\begin{align*}
FNR & = \frac{FN}{FN + TP}, \\
FPR & = \frac{FP}{FP + TN},
\end{align*}
\]

where FN, FP, TP, and TN refer to the false negative, false positive, true positive, and true negative data samples, respectively.

4. Experimental Results and Discussion

As mentioned earlier, the JS-SVC model has been coded and compiled in Visual C# .NET Framework 4.7.2. Moreover, experiments with the compiled computer program have been performed on the ASUS FX705GE-EW165T (Core i7 8750H and 8 GB Ram) platform. The JS metaheuristic is used to fine-tune the SVC-based spall severity classification approach. After 100 iterations, the JS metaheuristic has located the best values of the search variables as follows: the penalty coefficient = 867.6788 and the RBKF parameter = 58.6156. The best-found cost function value is 1.0696. The optimization process of the jellyfish swarm is demonstrated in Figure 8.

As described in the previous section, the collected dataset which includes 300 data samples has been randomly separated into a training set (90%) and a testing set (10%). Moreover, to reliably evaluate the predictive performance of the proposed JSO-SVC, this study has repeated the model training and testing processes with 20 independent runs. The statistical measurements obtained from these 20 independent runs are employed to quantify the model predictive capability in the task of concrete spalling severity recognition. This repeated model evaluation aims at reducing the variation caused by the randomness in the data separation process.

In addition, to demonstrate the JSO-SVC predictive performance, the random forest classification (RFC) model [79] and convolutional neural network (CNN) models [80] have been employed as benchmark approaches. The RFC and CNN are selected for result comparison in this study because these two machine learning approaches have been successfully applied in various works related to computer vision-based or nondestructive testing-based structural health monitoring/diagnosis [14, 26, 81–88].

The RFC has been constructed with the MATLAB’s Statistics and Machine Learning Toolbox [75]. Adaptive moment estimation (Adam) [89] and root mean square propagation (RMSprop) [90] are the two state-of-the-art approaches for training the deep neural network. The CNN models trained by Adam and RMSprop are denoted as CNN-Adam and CNN-RMSprop, respectively. These two models are constructed with the help of the MATLAB deep learning toolbox [91]. The model structures of the benchmark methods have been identified via several trial-and-error experiments with the collected dataset. The number of classification trees used in the random forest ensemble is 50. In addition, the two CNN models have been trained with 3000 epochs and the batch size of 8. The employed CNN models have been trained with the learning rate parameter = 0.001; moreover, L2 regularization with the regularization coefficient of 0.0001 has been employed to mitigate model overfitting [91]. To implement the deep neural computing models, the size of the input images has been standardized to be 32 × 32 pixels. The model structure of the employed CNN models is shown in Table 1.

In addition, to appraise the prediction capability of the proposed JSO-SVC and the employed benchmark approaches, a set of performance measurement metrics is employed in this section. Since the problem of spall severity has been modeled as a two-class classification problem, the indices of classification accuracy rate (CAR), precision, recall, negative predictive value (NPV), F1 score, and area under the receiver operating characteristic curve (AUC) [92, 93] are employed to quantify the classification model.
Figure 6: The proposed computer vision method for automatic concrete spall severity classification.
Figure 7: Demonstration of the feature extraction process: (a) a sample of the “shallow spall” class and (b) a sample of the “deep spall” class.

Figure 8: The JS-based model optimization progress.
The performance of the receiver operating characteristic curve and computation of the AUC, readers are guided to the previous work of van Erkel and Pattynama [94]. The detailed calculations of CAR, precision, recall, NPV, and F1 score are given by [92, 95]

\[
\text{CAR} = \frac{N_C}{N_A} \times 100\%,
\]

\[
\text{precision} = \frac{TP}{TP + FP},
\]

\[
\text{recall} = \frac{TP}{TP + FN},
\]

\[
\text{NPV} = \frac{TN}{TN + FN},
\]

\[
F1 \text{ score} = \frac{2TP}{2TP + FP + FN},
\]

where \(N_C\) and \(N_A\) denote the numbers of correctly predicted data and the total number of data, respectively. As mentioned earlier, FN, FP, TP, and TN represent the false negative, false positive, true positive, and true negative data samples, respectively.

The prediction performances of the proposed JSO-SVC and other benchmark methods are shown in Table 2 which reports the mean and standard deviation (Std) of the employed performance measurement metrics. Observable from this table, the proposed hybridization of JS and SVC has attained the most accurate classification of concrete spalling severity with \(\text{CAR} = 93.33\%\), precision \(= 0.932\), recall \(= 0.936\), NPV \(= 0.963\), and F1 score \(= 0.933\). The model construction phase of the JSO-SVC requires a computational time of 1067.4 s. The computational time of the proposed approach is roughly 3.3 s.

\[
\text{CAR} = \frac{N_C}{N_A} \times 100\%,
\]

where \(N_C\) and \(N_A\) denote the numbers of correctly predicted data and the total number of data, respectively. As mentioned earlier, FN, FP, TP, and TN represent the false negative, false positive, true positive, and true negative data samples, respectively.

The RFC is the second best method with \(\text{CAR} = 87.50\%\), precision \(= 0.871\), recall \(= 0.890\), NPV \(= 0.892\), and F1 score \(= 0.877\), followed by CNN-Adam (CAR \(= 81.50\%\), precision \(= 0.877\), recall \(= 0.750\), NPV \(= 0.788\), and F1 score \(= 0.799\) and CNN-RMSprop (CAR \(= 79.16\%\), precision \(= 0.809\), recall \(= 0.777\), NPV \(= 0.794\), and F1 score \(= 0.785\)). With \(\text{CAR} > 90\%\) and F1 score \(> 0.9\), it can be seen that the predictive result of the JSO-SVC is highly accurate. The performance of the decision tree ensemble of RFC with \(\text{CAR} = 87.5\%\) and F1 score \(= 0.877\) is fairly accurate and acceptable. Meanwhile, with CAR values of around 80% and F1 score approaching 0.8, the performance of the CNN models is clearly inferior to the machine learning approaches of JSO-SVC and RFC. The boxplots demonstrating the statistical distributions of the models' performance in terms of CAR and F1 score obtained from 20 independent runs are provided in Figures 9 and 10.

Besides the aforementioned metrics, ROC curves and AUC are also important indicators of classification performance. A ROC curve is a graph depicting the performance of a model when classification threshold varies. The horizontal axis of the graph is the false positive rate and the vertical axis of the graph is the true positive rate. The ROC curves of the proposed JSO-SVC and other benchmark models are provided in Figures 11–14. From those curves, the AUC values can be computed. AUC measures the two-dimensional area beneath the ROC curves. This indicator depicts an aggregate evaluation of the model performance with all possible values of the classification threshold. AUC varies between 0 and 1 with 0 indicating a useless classifier and 1 demonstrating a perfect classifier. Observed from the experimental outcomes, the JSO-SVC has also attained the highest AUC of 0.969, followed by RFC (AUC = 0.944), CNN-Adam (AUC = 0.896), and CNN-RMSprop (AUC = 0.855). The boxplot of the AUC results is illustrated in Figure 15.

In addition, to reliably assert the superiority of the newly developed JSO-SVC model used for concrete spalling severity classification, this study has employed the Wilcoxon signed-rank test [96] with the significant level (\(p\) value) = 0.05. The Wilcoxon signed-rank test is a widely employed nonparametric statistical hypothesis test used for model performance comparison [97]. One significant advantage of this statistical hypothesis test is that it does not require the assumption of normally distributed data [65]. Therefore, the Wilcoxon signed-rank test offers robust statistical power and is likely to yield statistical significant outcome.

The important performance measurement metrics of CAR, F1 score, and AUC are subject to this nonparametric hypothesis test. Herein, the null hypothesis is that the means of the prediction performances of two models are actually equal. The Wilcoxon signed-rank test outcomes are reported in Tables 3–5. Observably, with \(p\) values \(<0.05\), the null hypothesis can be rejected and the superiority of the proposed hybrid method can be firmly stated.

The experimental results have shown the superiority of the JSO-SVC over deep neural computing approaches of CNN models. It can be seen that although the CNN models have been demonstrated to be powerful methods in various computer vision tasks, their performance largely depends on the size of the training samples [48]. The main advantages of CNN lie in its capability of automatic feature representation via convolutional operators and its hierarchical architecture for learning high-level features from raw data. However, both of these advantages can only be realized with a sufficiently large number of image samples with correct ground truth labeling. As stated in [98], when the number of training samples is insufficient, the performance of deep learning models can be inferior to those of hand-crafted features-based prediction approaches.

For the case in which there are a limited number of data samples, the CNN models have difficulties in properly fine-tuning their internal structures with a huge number of parameters needed to be specified in various hidden layers.

---

**Table 1: The structures of the employed CNN model.**

<table>
<thead>
<tr>
<th>CNN layers</th>
<th>Convolutional layers</th>
<th>Pooling layers</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Number of filters</td>
<td>Filter size</td>
</tr>
<tr>
<td>1</td>
<td>32</td>
<td>8</td>
</tr>
<tr>
<td>2</td>
<td>32</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>16</td>
<td>2</td>
</tr>
</tbody>
</table>

**Table 2: Performance comparison**

<table>
<thead>
<tr>
<th>Model</th>
<th>CAR</th>
<th>Precision</th>
<th>Recall</th>
<th>NPV</th>
<th>F1 score</th>
</tr>
</thead>
<tbody>
<tr>
<td>RFC</td>
<td>87.5</td>
<td>0.877</td>
<td>0.936</td>
<td>0.963</td>
<td>0.933</td>
</tr>
<tr>
<td>CNN-Adam</td>
<td>81.5</td>
<td>0.877</td>
<td>0.750</td>
<td>0.788</td>
<td>0.799</td>
</tr>
<tr>
<td>CNN-RMSprop</td>
<td>79.1</td>
<td>0.809</td>
<td>0.777</td>
<td>0.794</td>
<td>0.785</td>
</tr>
</tbody>
</table>
Table 2: Experimental result comparison.

<table>
<thead>
<tr>
<th>Phase</th>
<th>Metrics</th>
<th>JSO-SVC</th>
<th>RFC</th>
<th>CNN-Adam</th>
<th>CNN-RMSprop</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mean</td>
<td>Std</td>
<td>Mean</td>
<td>Std</td>
<td>Mean</td>
</tr>
<tr>
<td>Training</td>
<td>CAR (%)</td>
<td>95.926</td>
<td>0.574</td>
<td>98.500</td>
<td>0.441</td>
</tr>
<tr>
<td></td>
<td>Precision</td>
<td>0.950</td>
<td>0.008</td>
<td>0.978</td>
<td>0.005</td>
</tr>
<tr>
<td></td>
<td>Recall</td>
<td>0.969</td>
<td>0.007</td>
<td>0.993</td>
<td>0.006</td>
</tr>
<tr>
<td></td>
<td>F1 score</td>
<td>0.960</td>
<td>0.006</td>
<td>0.985</td>
<td>0.004</td>
</tr>
<tr>
<td></td>
<td>AUC</td>
<td>0.983</td>
<td>0.002</td>
<td>0.998</td>
<td>0.001</td>
</tr>
<tr>
<td>Testing</td>
<td>CAR (%)</td>
<td>93.333</td>
<td>3.801</td>
<td>87.500</td>
<td>4.941</td>
</tr>
<tr>
<td></td>
<td>Precision</td>
<td>0.932</td>
<td>0.048</td>
<td>0.871</td>
<td>0.070</td>
</tr>
<tr>
<td></td>
<td>Recall</td>
<td>0.936</td>
<td>0.062</td>
<td>0.890</td>
<td>0.069</td>
</tr>
<tr>
<td></td>
<td>F1 score</td>
<td>0.933</td>
<td>0.042</td>
<td>0.877</td>
<td>0.047</td>
</tr>
<tr>
<td></td>
<td>AUC</td>
<td>0.969</td>
<td>0.020</td>
<td>0.944</td>
<td>0.046</td>
</tr>
</tbody>
</table>

Figure 9: Boxplot of CAR values obtained from the employed machine learning models.

Figure 10: Boxplot of F1 score values obtained from the employed machine learning models.
Therefore, with a dataset of 300 image samples, the hybrid machine learning method of JSO-SVC with texture-based feature extraction is capable of outperforming the CNN methods. The model optimization process via gradient descent algorithms employed by CNN encounters certain difficulty in identifying the fittest set of network parameters because the number of trained parameters greatly outnumbers the data size. This fact is partly reflected in the stability of the predictive outcomes of the deep learning models.

To quantify the stability of the model prediction, this study has employed the coefficient of variation (COV) [100]. This index is defined as the ratio of the standard deviation to the mean. Generally speaking, a small COV value indicates a small variation of prediction result and is associated with a reliable model. The COV indices of the proposed model, RFC, and the two CNN models are reported in Figure 16. Considering the metrics of CAR, F1 score, and AUC, the COV of JSO-SVC (with \(\text{COV}_{\text{CAR}} = 4.07\%\), \(\text{COV}_{\text{F1-score}} = 4.54\%\), and \(\text{COV}_{\text{AUC}} = 2.04\%\)) is significantly lower than that of the CNN-Adam (\(\text{COV}_{\text{CAR}} = 6.96\%\), \(\text{COV}_{\text{F1-score}} = 8.35\%\), and \(\text{COV}_{\text{AUC}} = 6.65\%\)) and CNN-RMSprop (\(\text{COV}_{\text{CAR}} = 10.48\%\), \(\text{COV}_{\text{F1-score}} = 12.28\%\), and \(\text{COV}_{\text{AUC}} = 9.36\%\)).

On the other hand, the proposed approach based on the SVC is the pattern recognizer that lends itself to learning with small or medium-size datasets because the SVC focuses on sparseness property when building a classification model.

![Figure 11: ROCs of JSO-SVC: (a) training phase and (b) testing phase.](image)

![Figure 12: ROCs of RFC: (a) training phase and (b) testing phase.](image)
from data. Put differently, the final SVC model only resorts to a small subset of the dataset to construct the classification boundary. The data points contained in such small subset are called support vectors, and they are highly relevant and informative for carrying out the task of concrete spall severity classification. This is a significant advantage of the JSO-SVC because a sparse concrete spall severity classification model is less likely to suffer from data overfitting. This point is clearly demonstrated via the learning performance (CAR = 95.926%) and testing performance (93.333%) of the JSO-SVC. The accuracy rates of the proposed approaches in the training and testing phases are relatively close to each other.

The classification model based on the integration of JSO and SVC also features a high degree of learning stability due to its sparseness property because the sparse model is capable of mitigating the effect of noisy samples within the collected dataset. Moreover, the SVC model construction boils down to solving a quadratic programming problem which can guarantee a learning convergence to a global optimal solution. This feature of the JSO-SVC also facilitates the reliability and stability of the spall severity recognition performance. The aforementioned analysis on COV has revealed these facts. The COV of JSO-SVC, which is less than 5%, is comparatively lower than that of other benchmark models.

Figure 13: ROCs of CNN-Adam: (a) training phase and (b) testing phase.

Figure 14: ROCs of CNN-RMSprop: (a) training phase and (b) testing phase.
However, one disadvantage of the proposed spall severity recognition model is that the optimization process required determining an optimal set of parameters of the SVC can be costly. The reason is that the SVC-based model training and prediction phases operate inside the cost function computing phase of the utilized JSO. Another limitation of the JSO-SVC-based spall severity classifier is that it has not been equipped with advanced feature selection. Such drawbacks ought to be addressed in future extensions of the current work.
5. Concluding Remarks

This study has proposed and verified a computer vision-based approach for automatic classification of concrete spalling severity. The proposed approach is an integration of image texture analysis methods, metaheuristic optimization, and machine learning-based pattern recognition. The texture descriptors of statistical measurement of color channels, GLRL, and CS-LBP are used to characterize images of concrete surface with respect to color, gray pixel run length, and local structure. With such extracted features, the SVC machine learning optimized by JS metaheuristic is employed to construct a decision boundary that separates the input data into two classes of deep spalling and shallow spalling. A dataset including 300 image samples has been collected to train the proposed computer vision method. Experimental results point out that the integrated model can help to attain the most desired spall severity classification with CAR = 93.333%, precision = 0.932, recall = 0.936, NPV = 0.963, F1 score = 0.933, and AUC = 0.969. These results are significantly better than those of the benchmark methods including RFC and CNN models. Therefore, the newly developed JSO-SVC can be a potential tool to assist building maintenance agencies in the task of periodic structural health survey. Further improvements of the current approach may include the following:

(i) The utilization of the hybrid model to detect other concrete surface defects such as crack, buhole, algal colonization, and so on.
(ii) The employment of other sophisticated texture descriptors for representing characteristics of concrete surface and better dealing with noise in the surface background.
(iii) Increasing the size of the current dataset to molloriate the applicability of the current method.
(iv) Investigating the possibility of combing handcrafted texture-based features with deep learning models used for concrete spalling severity classification.
(v) Employing advanced techniques of metaheuristic-based model optimization and feature selection to enhance the performance of the spall severity recognition task.

Data Availability

The dataset used to support the findings of this study has been deposited in the repository of GitHub at https://github.com/NDHoangDTU/ConcreteSpallSeverity_JSO_SVC. The first 78 columns of the data are texture-based features extracted from image samples. The last column is the label of the data instances with 0 = “shallow spalling” and 1 = “deep spalling.”

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was funded by Vietnam National Foundation for Science and Technology Development (NAFOSTED) under grant no. 107.01-2019.332.

References


[59] C. M. Bishop, *Pattern Recognition and Machine Learning* (Information Science and Statistics, Springer, /Heidelberg, Germany, 2011, https://www.google.com/search?q=1C1GCEB_enIN980IN980eq=Berlin&stick=H4sIAAAAAAAACAAAAAOngVuLQz9UMDQ1K1zEyuaUWpSm7rCyAUzQwPhUAA&xv=2&xv=2&xv=j8UEk7jio2zVkoL0AhV1zDgGHm7DWhsOxmoA2COECGMQAw April 6, 2011).


[66] H. Luo and S. G. Paal, “Metaheuristic least squares support vector machine-based lateral strength modelling of


