

Research Article

Automatic Impervious Surface Area Detection Using Image Texture Analysis and Neural Computing Models with Advanced Optimizers

Nhat-Duc Hoang ^{1,2}

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

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

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

Received 18 June 2020; Revised 18 December 2020; Accepted 29 January 2021; Published 16 February 2021

Academic Editor: Akbar S. Namin

Copyright © 2021 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.

Up-to-date information regarding impervious surface is valuable for urban planning and management. The objective of this study is to develop neural computing models used for automatic impervious surface area detection at a regional scale. To achieve this task, advanced optimizers of adaptive moment estimation (Adam), a variation of Adam called Adamax, Nesterov-accelerated adaptive moment estimation (Nadam), Adam with decoupled weight decay (AdamW), and a new exponential moving average variant (AMSGrad) are used to train the artificial neural network models employed for impervious surface detection. These advanced optimizers are benchmarked with the conventional gradient descent with momentum (GDM). Remotely sensed images collected from Sentinel-2 satellite for the study area of Da Nang city (Vietnam) are used to construct and verify the proposed approach. Moreover, texture descriptors including statistical measurements of color channels and binary gradient contour are employed to extract useful features for the neural computing model-based pattern recognition. Experimental result supported by statistical test points out that the Nadam optimizer-based neural computing model has achieved the most desired predictive accuracy for the data collected in the studied region with classification accuracy rate of 97.331%, precision = 0.961, recall = 0.984, negative predictive value = 0.985, and F1 score = 0.972. Therefore, the model developed in this study can be a helpful tool for decision-makers in the task of urban land-use planning and management.

1. Introduction

Urban impervious surface, developed by anthropogenic activities, is one of the most crucial land cover forms. The impenetrable surface areas consist of buildings, roads, parking lots, sidewalks, pavements, and many others. These surfaces prevent the absorption of water into the soil. Previous works have pointed out the impact of impervious surface areas on water quality and the frequency/intensity of downstream runoff [1–5]. Therefore, they have been identified as a key indicator used in evaluating urbanization influences on surrounding natural environment and ecosystem [6].

Due to such reasons, up-to-date information regarding impervious surface is of paramount importance for supporting urban land management/planning, detection of

unplanned built-up areas, study of regional land-use pattern, and ecosystem monitoring [5, 7–10]. In developing countries including Vietnam, the conventional approach for obtaining such information is field survey. Nevertheless, this approach is very time-consuming and requires considerable effort in data collection, processing, and storing. Therefore, a quick and cost-effective method for substituting this conventional approach is a practical need for municipal land-use managers.

In recent years, remote sensing technology and processing of satellite images have been increasingly applied to tackle various challenging problems in a wide span of domains including agriculture [11–13], natural hazard prevention [14–17], civil engineering [18–20], and environmental engineering [21–23].

Following this trend of research, scholars and practitioners have increasingly relied on remote sensing and geographic information system (GIS) technologies to improve the productivity and accuracy of the impervious surface detection task [24, 25]. These technologies have been proven to be viable tools for surveying urban landscapes which are rapidly changing and providing timely information regarding urban growth [26–29]. Based on remotely sensed images, statistical and machine learning models can be constructed for automatic impervious surface extraction [8].

Lo [30] developed a computer-based model for analyzing remote sensing data obtained from Landsat image; this model only relied on spectral information of image pixels to derive land form categories. Zha et al. [29] performed built-up areas mapping with the utilization of normalized difference vegetation index and normalized difference built-up index; the proposed model analyzed Landsat Thematic Mapper images and achieved an accuracy of 92.6%. A multivariate statistical analysis approach has been put forward in [28] for characterizing urban growth; this approach could reduce the modeling error to less than 10%. Yang et al. [5] employed a combination of Landsat ETM+ and high-resolution imagery to construct a decision tree-based impervious surface mapping. Multilayer perceptron neural network and support vector machine have been used in [6] to classify image samples obtained from Landsat-5 TM Imager.

Zhang et al. [31] integrated spectral information and multivariate texture to extract numerical features from remotely sensed image; the one-class support vector machine is then used for pattern classification. Zhang et al. [32] investigated the capability of random forest approach for impervious surface estimation with a combined utilization of synthetic aperture radar and optical remote sensing images. A backpropagation neural network has been constructed by Patel and Mukherjee [33] to extract the impervious features using Landsat Thematic Mapper data. Son et al. [34] introduced an impervious surface fraction algorithm (ISFA) for automatic impervious surface extraction; this algorithm is applied with Landsat data and attains an accuracy of 92.8%. Gupta et al. [7] compared the performances of supervised maximum likelihood algorithms, index-based classification, and neural classification and points out that the neural classification model achieves the most desired outcome.

It can be seen from the literature that most of the previous works have employed the medium resolution open-source image dataset such as Landsat Thematic Mapper to extract impervious surface [35]. Because of the complex texture of urban landscape, these coarse resolution images feature certain limitations on impervious surface mapping. Xu et al. [36] investigated the use of the 10 m resolution Sentinel-2A dataset for impervious area extraction and pointed out the superiority of high-resolution data over the conventional 30 m resolution Landsat dataset. Misra et al. [8] attempted to employ high-resolution image obtained from Sentinel-2 to improve the quality of impervious surface detection result; the authors rely on spectral angle mapper,

support vector machine, and neural network to carry out pattern recognition task.

Among the machine learning approaches employed in remote sensing and GIS field, neural computing models have been extensively employed and remain effective tools for recognizing patterns in remotely sensed images [37–44]. It is because neural computational models with their capability of universal function approximator are capable of learning and recognizing complex patterns [45]. Nevertheless, the employed neural computing approaches have mainly relied on the conventional gradient descent for model training [7, 8, 37, 46]. Although this conventional training method can help to attain acceptable results in many application cases, it also suffers from slow convergence rate and trapping in local optimal [47]. These facts definitely reduce the generalization and accuracy of prediction models constructed by neural computational approaches. Therefore, there is a pressing need to investigate and apply advanced training algorithms to mitigate the disadvantages of the conventional gradient descent.

In recent years, various advanced gradient-based optimization algorithms have been proposed and used for training neural computing models. However, few research works have investigated these state-of-the-art algorithms in constructing neural computing models used for remote sensing-based impervious surface detection. Therefore, this study is an attempt to fill this gap in the current literature. The advanced optimizers of adaptive moment estimation (Adam) [48], a variation of Adam called Adamax [48], Nesterov-accelerated adaptive moment estimation (Nadam) [49], Adam with decoupled weight decay (AdamW) [50], and a new exponential moving average variant (AMSGrad) [51] are employed for automatic impervious surface extraction.

In addition, Da Nang city (Vietnam) has been selected as the study area. Image texture analysis technique including statistical measurements of color channels [52] and binary gradient contour [53] are used to extract useful features from remotely sensed images obtained from Sentinel-2 satellite. The extracted features are then employed by neural computing models for automatic impervious surface detection in the study area. Therefore, one major contribution of the current study is to establish an advanced hybridization of machine learning and image processing used for constructing an impervious surface map for the study area of Da Nang city.

The subsequent sections of the article are organized as follows: the research methodology is reviewed in the second section. The proposed neural computing model trained by the aforementioned advanced optimizers used for impervious surface area detection is presented in the next section, followed by the fourth section which reports experimental results. Several concluding remarks on the current study are stated in the final section.

2. Research Methodology

2.1. General Description of the Study Area. Da Nang city is located in the Central Vietnam (refer to Figure 1). Its latitude is between $15^{\circ}15'20''\text{N}$ and $16^{\circ}14'10''\text{N}$; its longitude is from



FIGURE 1: The study area.

107°18'30"E to 108°20'00"E [54]. It is a port city located on the coast of the East Sea. In 2015, Da Nang had a population of 1,046,876 and an area of 1,285.4 km² [55]. This city is divided into 8 districts: 6 urban districts and 2 rural districts [56].

Da Nang is ranked as the fourth largest city by population in Vietnam and serves as an economic base in the service and industrial sectors in Central Vietnam. Due to such reasons, the population of this city is rising rapidly from approximately 673,000 in 1997 to about 1 million in 2014 [57]. This population growth leads to a significant urban expansion. Therefore, Da Nang city is selected as the study area in this article.

2.2. The Image Data Used. The image data obtained from the Sentinel-2 on March 13, 2020, is used in this study to perform impervious surface extraction. The bands of 4 (red), 3 (green), and 2 (blue) with spatial resolution of 10 m are selected to compose the image of the study area. The size of each image file (i.e., bands 2, 3, and 4) is 235,484 KB. A full-scene map of Da Nang city (5559 × 3444 pixels) is presented in Figure 2. It is noted that these Sentinel-2's bands have been opened in Sentinel Application Platform (SNAP) software package [58]. The original Sentinel-2's bands obtained from USGS [59] are

converted to TIF format using the geometric operation of resampling supported by the SNAP software package. For more details of the SNAP software documentation, readers are guided to articles provided in [60]. Moreover, the used map projection of the obtained images is Universal Transverse Mercator (UTM) within Zone 48N–Datum World Geodetic System (WGS) 84.

Based on the original composed image, the contrast enhancement technique of histogram equalization (refer to Figure 3) is employed to create a better image for subsequent analysis. The purpose of histogram equalization is to construct an image with equally distributed brightness levels [61]. This image processing technique meliorates the global contrast of the original image and highlights the image texture. In addition, to facilitate the process of impervious surface detection, pixels at mountainous regions covered by cloud and large beach areas are cast out by masking operation [62]. In addition, the normalized difference vegetation index (NDVI) [63] is computed to remove large water bodies from the study area. Via experimentation with the collected image, pixels belonging to large water bodies are associated with negative NDVI values and can be effectively excluded. The NDVI computation requires the band 4 (red) and band 8 (near-infrared band); it is obtained via the following equation:

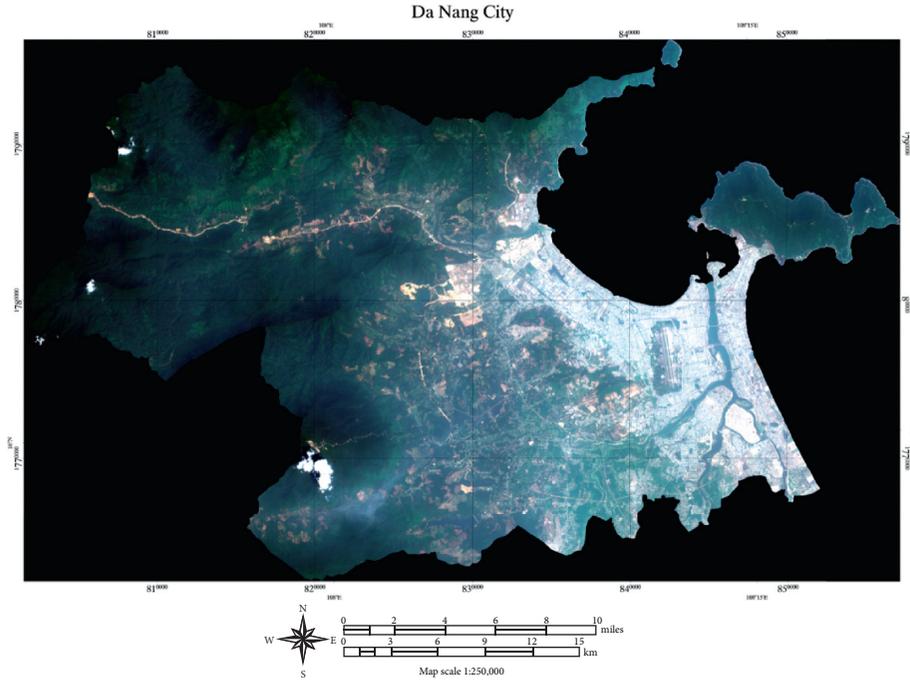


FIGURE 2: Color composite of Sentinel-2 bands 4 (red), 3 (green), and 2 (blue) of the study area (Da Nang city, Vietnam).



FIGURE 3: The map of the study area enhanced by histogram equalization.

$$\text{NDVI} = \frac{\text{NIR} - B4}{\text{NIR} + B4} \quad (1)$$

where NIR and B4 denote a near-infrared band and band 4, respectively.

2.3. Image Texture Analysis

2.3.1. Statistical Measurements of Image Bands. For the purpose of impervious surface detection, the statistical measurements of bands 4 (red), 3 (green), and 2 (blue) are

employed in this study. It is noted that this research performs impervious surface detection for each image patch of 10×10 pixels. Thus, to derive statistical measurements, the first-order histogram of an image patch S denoted as $P(I)$ is computed as follows [52, 64]:

$$P_b(I) = \frac{N_{I,b}}{\text{PN}}, \quad (2)$$

where b denotes a band index, $N_{I,b}$ is the number of pixels having the value of I , and PN denotes the number of pixels within an image patch.

Accordingly, the mean (μ_b), standard deviation (σ_b), skewness (S_b), kurtosis (K_b), entropy (E_b), and range (R_b) are computed as follows [52, 64]:

$$\begin{aligned}\mu_b &= \sum_{i=0}^{NL-1} I_{i,b} \times P_b(I), \\ \sigma_b &= \sqrt{\sum_{i=0}^{NL-1} (I_{i,b} - \mu_b)^2 \times P_b(I)}, \\ S_b &= \frac{\sum_{i=0}^{NL-1} (I_{i,b} - \mu_b)^3 \times P_b(I)}{\sigma_b^3}, \\ K_b &= \frac{\sum_{i=0}^{NL-1} (I_{i,b} - \mu_b)^4 \times P_b(I)}{\sigma_b^4}, \\ E_b &= - \sum_{i=0}^{NL-1} P_b(I) \times \log_2(P_b(I)), \\ R_b &= \text{Max}(I_b) - \text{Min}(I_b),\end{aligned}\quad (3)$$

where $I_{i,b} = 0, 1, 2, \dots, 255$. For 8-bit image, $NL = 256$ represents the number of discrete intensity values.

2.3.2. Binary Gradient Contour (BGC) for Texture Discrimination. Texture is a crucial tool for visual perception in computer vision. Since image patches of impervious surface and pervious surface can have different properties of coarseness, roughness, directionality, contrast, and regularity, using texture analysis can be helpful to delineate them. This study employs the BGC technique [53] for the task of texture discrimination. The BGC combines the analyses of local structures and occurrences to derive texture features. Notably, this method has the advantages of low computational expense and invariant to monotonic illumination changes [65]. It also possesses good discriminative powers demonstrated in previous studies [53, 66].

Essentially, the BGC carries out a pairwise comparison of adjacent pixels located in one or more closed paths along the periphery of a neighborhood of the size 3×3 pixels [65]. Fernandez et al. [53] put forward three versions of BGC which are the single-loop (BGC₁), double-loop (BGC₂), and triple-loop (BGC₃) descriptors. To ease the description of these descriptors, a square image patch $S_{m,n}$ is denoted as follows:

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

where $I_{m,n}$ denotes the central pixel.

Based on the definition of S , the single-, double-, and triple-loop BGC formulas are expressed as follows [53]:

$$\begin{aligned}\text{BGC}_1 &= \sum_{n=0}^7 \lambda(I_n - I_{(n+1)\text{mod}8}) \times 2^n - 1, \\ \text{BGC}_2 &= 15 \times \sum_{n=0}^3 \lambda(I_{2n\text{mod}8} - I_{2(n+1)\text{mod}8}) \times 2^n \\ &\quad + \sum_{n=0}^3 \lambda(I_{2n+1} - I_{(2n+3)\text{mod}8}) \times 2^n - 16, \\ \text{BGC}_3 &= \sum_{n=0}^7 \lambda(I_{3n\text{mod}8} - I_{3(n+1)\text{mod}8}) \times 2^n - 1,\end{aligned}\quad (5)$$

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

2.4. Artificial Neural Network (ANN) for Pattern Classification.

A general structure of an ANN for two-class pattern recognition tasks is presented in Figure 4. This model typically contains an input layer, a hidden layer, and an output layer [67]. In this study, the input layer receives signals in the form of texture information. The hidden layer consists of M neurons which process the texture information to yield the input of the softmax activation function. The usual activation function used by the neurons in the hidden layer is the log-sigmoid function. The softmax activation function converts its input signals to class probabilities within the range of 0 and 1. It is noted that interactions among neurons are expressed by connection weights. These sophisticated interactions permit the overall neural computing model to learn and infer complex mapping relationships [68].

The knowledge learnt by a neural computing model is stored in matrices of connection weights. Herein, W_{L0L1} denotes the matrix of connection weights between the input and hidden layer. W_{L1L2} represents that between the hidden and the softmax layer. Let $X \in R^D$ denote the matrix of input feature. The computation process of an ANN model can be compactly defined as follows:

$$f(X) = \delta[b_1 + W_{L1L2} \times (f_A(b_0 + W_{L0L1} \times X))], \quad (6)$$

where b_0 and b_1 denote two bias vectors of the input and hidden layers, respectively, f_A represents the activation function, and δ represents the softmax activation function.

The softmax activation function is given by

$$\delta(z) = \frac{\exp(z_i)}{\sum_{i=0}^{CN-1} \exp(z_i)}, \quad (7)$$

where $CN = 2$ denotes the number of output classes.

2.5. The Employed Network Training Methods

2.5.1. The Network's Cost Function. To construct a neural computing model used for impervious surface detection, its model parameters must be identified. Herein, given a set of training data samples, the network parameters including the two matrices of W_{L0L1} and W_{L1L2} can be adapted via the

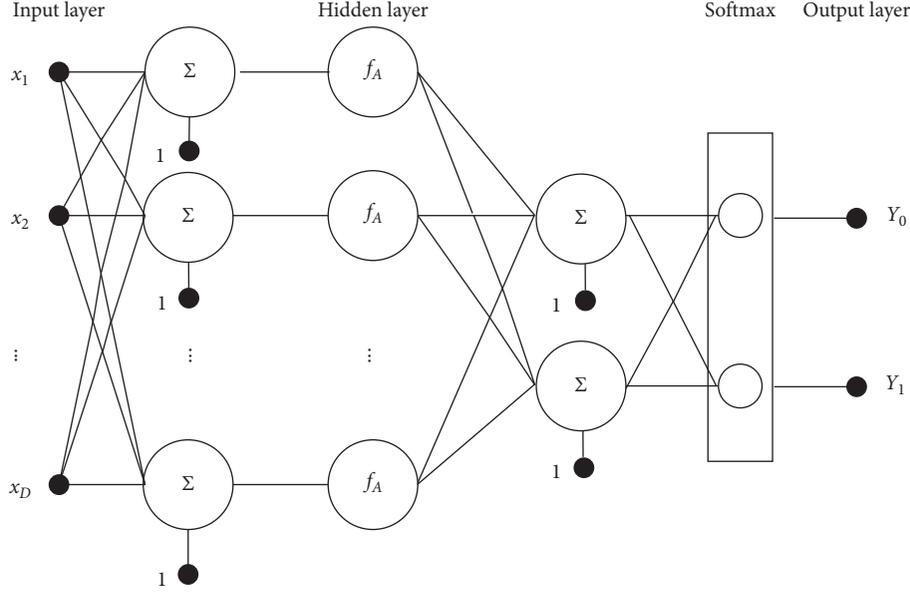


FIGURE 4: General structure of the employed ANN model used for two-class pattern recognition.

framework of error backpropagation [69, 70] with mini-batch mode [71]. For the task of data classification, the cross entropy cost function is often used as the objective function (E) for training a neural computing model [72]. The cross-entropy function is given by

$$E = -\frac{1}{N_d} \sum_{n=1}^{N_d} T \ln(Y) + (1 - T) \ln(1 - Y), \quad (8)$$

where N_d denotes the number of data samples; T and Y represent the actual and predicted class labels, respectively.

2.5.2. The Network's Optimizers

Gradient Descent with Momentum (GDM). The conventional method of gradient descent with momentum (GDM) is widely employed for training neural networks and is used as the benchmark method in this study. Via the GDM, the weights of a neural computing model are adapted as follows:

$$w_{t+1} = w_t - \alpha_L \times \frac{dE}{dw_t} + \lambda_M \times w_t, \quad (9)$$

where w_t and w_{t+1} are the previous and updated network weights, E represents the objective function, and α_L and λ_M are the learning rate and the momentum term, respectively. *Adaptive Moment Estimation (Adam)*. The Adam, introduced by Yoshua and Yann [48], can be considered as a general algorithm for first-order gradient-based optimization of stochastic objective functions. One notable advantage of this optimizer is that it is capable of adaptively fine-tuning the learning rate parameter during the training process. The Adam relies on information obtained from the average of the second moments of the gradients. This optimizer also utilizes an exponentially decaying average of past gradients. In

addition, this optimizer requires an initial setting of three hyperparameters: the step size α and the two exponential decay rates ($\beta_1 = 0.9$ and $\beta_2 = 0.9999$). When the gradient of model parameters is computed, the optimized parameters of a neural computing model are adapted via [48]

$$w_t = w_{t-1} - \alpha \times \frac{\hat{m}_t}{\sqrt{\hat{v}_t + \varepsilon}}, \quad (10)$$

where \hat{m}_t and \hat{v}_t denote the bias-corrected first moment estimate and the bias-corrected second raw moment estimate, respectively. *Adamax*. The Adamax [48] is a variant of the Adam in which the update rule for model weights is to scale their gradients inversely proportional to a L^p norm of their current and previous gradients. The neural network's weights are updated as follows:

$$w_t = w_{t-1} - \frac{\alpha}{1 - \beta_1^t} \times \frac{m_t}{\max(\beta_2 u_{t-1}, |g_t|)}, \quad (11)$$

where $u_t = 0$ at $t = 0$; u_t represents the biased second raw moment estimate. *Nesterov-Accelerated Adaptive Moment Estimation (Nadam)*. The Nadam optimizer, described in [49], attempts to incorporate Nesterov-accelerated adaptive moment estimation into the Adam. The major advantage of this integrated approach is that the employed adaptive moment estimation helps to perform highly accurate step in the gradient direction via updates of model parameters with the momentum step before the computation of the gradient [73]. The update rule of the Nadam is stated as follows [49, 73]:

$$w_t = w_{t-1} - \alpha \times \frac{\bar{m}_t}{\sqrt{\bar{v}_t + \varepsilon}}, \quad (12)$$

where

$$\begin{aligned}\bar{m}_t &= (1 - \beta_{1,t})\hat{g}_t + \beta_{1,t+1}\hat{m}_t, \\ \hat{m}_t &= \frac{m_t}{1 - \prod_{i=1}^{t+1}\beta_{1i}}, \\ \hat{g}_t &= \frac{g_t}{1 - \prod_{i=1}^{t+1}\beta_{1i}}.\end{aligned}\quad (13)$$

Adam with Decoupled Weight Decay (AdamW). The AdamW [50] optimizer integrates weight decay into the original Adam. The weight decay is a widely used approach for regularizing the network weights. It is because large weights may lead to an overfitted model. Accordingly, the update rule of the AdamW algorithm is given by

$$w_t = w_{t-1} - \alpha \times \left(\frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \varepsilon} + \lambda w_{t-1} \right), \quad (14)$$

where λ denotes a hyperparameter. *A New Exponential Moving Average Variant (AMSGrad)*. The AMSGrad optimizer [51] attempts to improve the convergence of the Adam optimizer by the employment of long-term memory of past optima. To avoid poor convergence and trapping in local optima, Reddi et al. [51] argues that the maximum of past squared gradients v_t should be used for parameter update instead of the exponential average employed by the Adam optimizer. The following equation is used to update the neural network's parameter:

$$w_t = w_{t-1} - \alpha \times \frac{m_t}{\sqrt{\hat{v}_t} + \varepsilon}, \quad (15)$$

where \hat{v}_t denotes the updated bias-corrected 2nd raw moment estimate.

3. The Proposed Neural Computing Model with Advanced Optimizers for Automatic Impervious Surface Detection

This section of the article presents the general description of the proposed neural computational method employed for automatic impervious surface detection. The proposed model is an integration of image texture analysis, neural network-based pattern recognition, and advanced optimizers used for neural network training. An overview of the data processing and the training phase of the proposed neural computing model used for impervious surface detection is demonstrated in Algorithm 1. The general structure of the newly developed model is presented in Figure 5. It is noted that the proposed neural computational model used for impervious surface detection has been developed in Visual C#.NET environment (Framework 4.6.2) and performed with the ASUS FX705GE-EW165T (Core i7 8750H, 8 GB Ram, 256 GB solid-state drive).

The model operation can be divided into four steps:

- (i) Data preprocessing
- (ii) Image data sampling

(iii) Image texture computation

(iv) Neural computing model training and prediction

3.1. Data Preprocessing. In this step, the original Sentinel-2's bands are opened in the SNAP software package and converted to TIFF format. The image process technique of histogram equalization is employed to enhance the contrast of the original image. As mentioned earlier, the NDVI is also calculated using the obtained bands to cast out large water bodies from the study area.

3.2. Image Data Sampling. To establish the neural network model for automatic impervious surface area detection, it is required to prepare a training dataset with assigned ground truth labels. This study has sampled pervious and impervious areas within the map of the study area (refer to Figure 6). Each sample with the size of 100×100 pixels is used to create nonoverlapped image patches with the size of 10×10 pixels. In total, there are 3,000 image patches that are generated from image samples. To ensure a balanced dataset, the numbers of the negative (pervious surface) and positive (impervious surface) samples are both 1,500. Based on these image patches, the image texture computation methods can be carried out to extract useful feature for the pattern recognition phase.

3.3. Image Texture Computation. Using image samples generated from the previous step, the texture analysis methods using statistical measurements of color channels and the BGC can be performed. The texture computation process converts image samples of the negative (pervious surface) and the positive (impervious surface) classes into numerical features. These numerical features are subsequently used for the task of pattern recognition performed by the neural computing models. The statistical measurements of the three color channels include the mean, standard deviation, skewness, kurtosis, entropy, and range indices. Since the number of the employed bands obtained from the Sentinel-2 is 3, there are $6 \times 3 = 18$ features attained from statistical measurements of color channels (refer to Figure 7).

In addition, the BGC is performed with the three versions of single-loop (BGC₁), double-loop (BGC₂), and triple-loop (BGC₃). Each of them produces a histogram which describes the texture information of image samples. This study computes the measurements of mean, standard deviation, skewness, kurtosis, and entropy from each histogram. Hence, the BGC texture descriptors yield $5 \times 3 = 15$ features (refer to Figure 8).

Thus, the total number of features extracted from the used texture descriptors is 33. Moreover, to facilitate the data classification based on the employed neural computing model, the texture-based features have been preprocessed by the Z-score data normalization. The Z-score equation is given by

```

//Retrieve remote sensing data and data preprocessing
Obtain spectral bands of Sentinel-2 for the study area
Open spectral bands in Sentinel Application Platform (SNAP)
Image enhancement via histogram equalization
Normalized difference vegetation index (NDVI) computation
Removal of large water bodies using the computed NDVI
//Generate dataset for the two classes of impervious and pervious surfaces
//Image data sampling
Create image dataset PosSet for impervious class
Create image dataset NegSet for pervious class
//Image texture computation
For each image in PosSet
    Image texture computation using the following:
        (i) Statistical measurements of bands
        (ii) BGC
End for
For each image in NegSet
    Image texture computation using the following:
        (i) Statistical measurements of bands
        (ii) BGC
End for
Construct numerical dataset D containing texture features and class labels
//Neural computing model training and evaluation
Set the total number of model evaluation times  $RN=20$ 
Establish a set of optimizers  $SO = \{GDM, Adam, AdaMax, Nadam, AdamW, AMSGrad\}$ 
For  $r=0$  to  $RN-1$ 
    Randomly extract 70% of the dataset to form a training dataset
    Select an optimizer from  $SO$ 
    Perform model training
    Evaluating the model training performance
End for
    
```

ALGORITHM 1: The data processing and the training phase of the proposed neural computing model used for impervious surface detection.

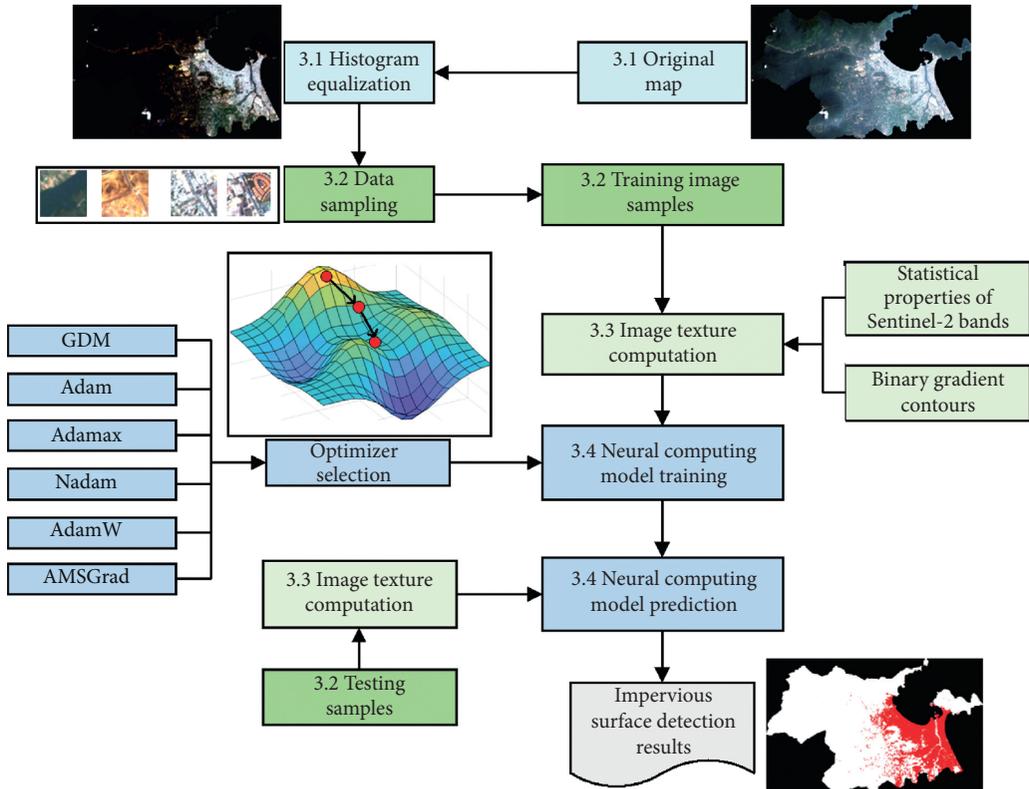


FIGURE 5: The proposed neural computing model for impervious surface detection.

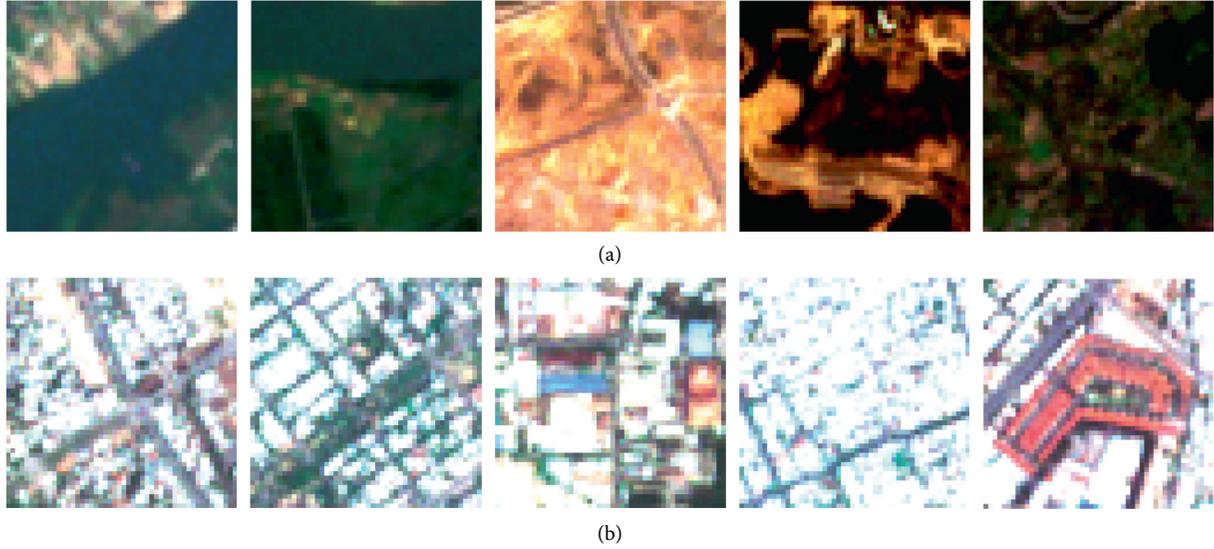


FIGURE 6: Demonstration of the collected image samples: (a) pervious class and (b) impervious class.

Image	Feature (F) computed by statistical measurements of color channels					
	F1	F2	F3	F4	F5	F6
	234.15	239.49	249.13	32.24	29.93	16.64
	F7	F8	F9	F10	F11	F12
	-1.77	-2.15	-3.22	5.43	6.85	13.35
	F13	F14	F15	F16	F17	F18
	3.25	2.27	0.98	165.00	143.00	121.00

(a)

Image	Feature (F) computed by statistical measurements of color channels					
	F1	F2	F3	F4	F5	F6
	23.21	34.07	18.10	13.77	16.14	12.82
	F7	F8	F9	F10	F11	F12
	1.42	0.03	0.66	13.83	3.22	2.70
	F13	F14	F15	F16	F17	F18
	2.86	3.10	1.86	172.00	128.00	74.00

(b)

FIGURE 7: Illustration of features computed by statistical measurements of color channels: (a) an impervious surface and (b) a pervious surface.

$$X_{ZN} = \frac{X_o - m_X}{s_X}, \quad (16)$$

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

3.4. Neural Computing Model Training and Prediction. As stated earlier, a dataset including 3,000 instances and 33 features has been prepared to train and verify the neural computing approach used for impervious surface detection. Each instance of the dataset has the class label of either pervious (denoted as 0) or impervious (denoted as 1). Each

data record contains texture characteristic of an image region within the map of the study area. As mentioned earlier, the statistical measurements of color channels and statistical measurements of the three BGC variants (BGC_1 , BGC_2 , and BGC_3) are used as texture descriptors.

The neural computing model is used to generalize a decision boundary that can distinguish data instances of the two categories of pervious and impervious surfaces. Accordingly, the original dataset has been randomly split into two mutual exclusive sets: a training set (70%) and a testing set (30%). The first set is used for model construction. The latter set is used to evaluate the model's predictive capability. It is noted that the neural computing models in this study are trained with the mini-batch mode [74]. Accordingly, the training data are split

Image	Feature (F) computed by the BGC texture descriptor				
	F19	F20	F21	F22	F23
	1,513.94	4,015.05	-0.34	0.11	40.53
	F24	F25	F26	F27	F28
	1,506.90	3,996.13	-0.34	0.11	39.26
	F29	F30	F31	F32	F33
	1,500.43	3,979.17	-0.34	0.11	40.47

(a)

Image	Feature (F) computed by the BGC texture descriptor				
	F19	F20	F21	F22	F23
	1,482.51	3,931.85	-0.34	0.11	40.88
	F24	F25	F26	F27	F28
	1,493.42	3,960.45	-0.34	0.11	39.02
	F29	F30	F31	F32	F33
	1,483.41	3,934.08	-0.34	0.11	40.60

(b)

FIGURE 8: Illustration of features computed by the BGC texture descriptor: (a) an impervious surface and (b) a pervious surface.

TABLE 1: Training performance.

Models	Statistics	Performance measurement indices								
		CAR (%)	TP	TN	FP	FN	Precision	Recall	NPV	F1 score
GDM	Mean	94.210	999.800	978.600	50.650	70.950	0.952	0.939	0.933	0.944
	Std	3.613	28.037	82.229	31.922	86.497	0.030	0.060	0.081	0.030
Adam	Mean	97.771	1,008.450	1,044.750	36.850	9.950	0.965	0.990	0.991	0.977
	Std	0.348	11.859	12.825	7.164	3.186	0.007	0.003	0.003	0.004
Adamax	Mean	97.340	1,008.500	1,035.650	42.700	13.150	0.959	0.987	0.987	0.973
	Std	0.392	18.629	15.415	5.780	3.991	0.006	0.004	0.004	0.004
Nadam	Mean	97.967	1,019.050	1,038.250	36.250	6.450	0.966	0.994	0.994	0.979
	Std	0.367	14.116	13.634	6.898	2.636	0.006	0.003	0.003	0.004
AdamW	Mean	97.793	1,014.100	1,039.550	40.050	6.300	0.962	0.994	0.994	0.978
	Std	0.310	12.992	12.698	6.289	3.480	0.006	0.003	0.003	0.003
AMSGrad	Mean	98.117	1,016.450	1,044.000	29.250	10.300	0.972	0.990	0.990	0.981
	Std	0.465	13.959	14.601	7.175	5.178	0.007	0.005	0.005	0.005

into small batches and these batches are used to calculate the model error and the gradients of the neural computing models' parameters.

The training process of the neural computing model aims at adapting the two matrices of W_{LOL1} and W_{LIL2} that specify the model structure. The size of the first matrix, which is the connection weight between the input and the hidden layer, is $M \times (N_I + 1)$ matrix where M and N_I represent the number of neurons in the hidden layer and the number of input features, respectively. Herein, $N_I = 33$ which is equal to the number of features extracted from the employed texture descriptors.

It is noted that the selection of the tuning parameters of the neural computing model in this study is based on recommendation of previous works and experimental trials using the collected dataset. Based on the suggestion of Heaton [75] and Tien Bui et al. [44], the number of neurons in the hidden layer in this study is set to be $(2/3)N_I + N_O$, where $N_O = 2$ denotes the number of the output classes. In addition, the log-sigmoid is

chosen as the activation function since it is commonly used for constructing shallow neural network models used for pattern classification [76, 77]. The softmax activation function is employed in the final layer to yield class probabilities within the range of 0 and 1 [67, 78]. Moreover, the number of training epochs is also required to be set appropriately. It is worth noticing that this tuning parameter may strongly affect the training outcome. An insufficient number of epochs can result in an underfitted model. Meanwhile, an excessive number of epochs may lead to an overfitted model. In this study, via several trial-and-error experiments with the collected dataset, the suitable number of training epoch is found to be 100.

When the number of neurons in the hidden layer is determined, the size of the matrices that contain connection weights can be specified. Herein, the size of the W_{LIL2} , which stores connection weights between the hidden and output layer is $N_O \times (M + 1)$. Thus, the total number of variables needed to be identified by the employed optimizers is $N_R \times N_I + N_O \times N_R + 2$. In this study, the optimizers of GDM, Adam, Adamax, Nadam,

TABLE 2: Testing performance.

Models	Statistics	Performance measurement indices								
		CAR (%)	TP	TN	FP	FN	Precision	Recall	NPV	F1 score
GDM	Mean	93.389	424.700	415.800	24.850	34.650	0.944	0.930	0.922	0.935
	Std	3.428	23.046	41.248	15.203	35.373	0.035	0.055	0.081	0.027
Adam	Mean	97.050	435.550	437.900	19.150	7.400	0.958	0.983	0.983	0.970
	Std	0.522	11.182	11.388	4.542	3.216	0.010	0.007	0.007	0.005
Adamax	Mean	96.572	427.800	441.350	21.000	9.850	0.953	0.977	0.978	0.965
	Std	0.728	17.189	16.487	6.025	2.903	0.013	0.007	0.006	0.008
Nadam	Mean	97.311	427.250	448.550	17.450	6.750	0.961	0.984	0.985	0.972
	Std	0.420	12.980	11.923	4.444	3.064	0.010	0.007	0.006	0.005
AdamW	Mean	97.028	424.600	448.650	21.250	5.500	0.952	0.987	0.988	0.969
	Std	0.520	11.061	11.897	4.898	2.110	0.011	0.005	0.005	0.005
AMSGrad	Mean	96.556	435.800	433.200	18.500	12.500	0.959	0.972	0.972	0.966
	Std	0.840	12.659	14.020	4.955	5.005	0.011	0.011	0.011	0.008

AdamW, and AMSGrad are used to search for the most appropriate values of the two matrices of W_{L0L1} and W_{L1L2} .

4. Experimental Results

As stated earlier, to train and verify the neural computing model used for impervious surface area detection, the extracted dataset has been divided into two sets of training (70%) and testing (30%) datasets. In addition, to alleviate the undesired effect of randomness on data sampling and to accurately assess the generalization capability of the newly developed model, the training/testing data sampling processes have been performed 20 times. In each time, 30% of the dataset, which corresponds to 900 instances, is randomly drawn out to form the testing dataset. The rest of the dataset including 2100 instances is used for model training.

Based on the model configuration in the previous section, the employed neural computing model is an artificial neural network consisting of 33 neurons in the hidden layer. The log-sigmoid is used as the activation function in the hidden layer. In the output layer, the softmax function is utilized to derive the probability of the two class labels of impervious surface and pervious surface.

In addition, to evaluate the prediction results of the employed neural computing models, classification accuracy rate (CAR), precision, recall, negative predictive value (NPV), and F1 score are calculated as follows [79]:

$$\begin{aligned}
 \text{CAR} &= \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \times 100\%, \\
 \text{precision} &= \frac{\text{TP}}{\text{TP} + \text{FP}}, \\
 \text{recall} &= \frac{\text{TP}}{\text{TP} + \text{FN}}, \\
 \text{NPV} &= \frac{\text{TN}}{\text{TN} + \text{FN}}, \\
 \text{F1 score} &= \frac{2\text{TP}}{2\text{TP} + \text{FP} + \text{FN}},
 \end{aligned} \tag{17}$$

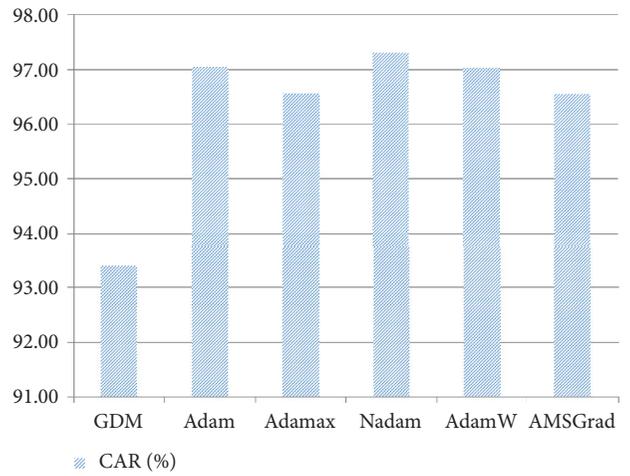


FIGURE 9: Model comparison with CAR (%).

where TP, TN, FP, and FN represent true-positive, true-negative, false-positive, and false-negative instances, respectively.

The outcomes of the artificial neural network models optimized by the used optimizers obtained from the training and testing phases are reported in Tables 1 and 2. It is observable that the neural computing models optimized by the Nadam have achieved the most desired performance with CAR = 97.331%, precision = 0.961, recall = 0.984, NPV = 0.985, and F1 score = 0.972. As can be seen from Tables 1 and 2, the prediction performances obtained from the training phase (CAR = 97.967%) and testing phase (CAR = 97.311%) of the Nadam-based model are relatively close to each other. This fact indicates that the Nadam-optimized neural computing model used for impervious surface detection does not suffer from overfitting issue. The experimental results also demonstrate that the selected number of training epochs is reasonable and help to prevent both overfitting and underfitting. The Adam optimizer is the second best approach (CAR = 97.050%), followed by the AdamW (CAR = 97.028%), Adamax (CAR = 96.572%), AMSGrad (CAR = 96.556%), and GDM (CAR = 93.389%). The model result comparison is also graphically presented by Figures 9 and 10.

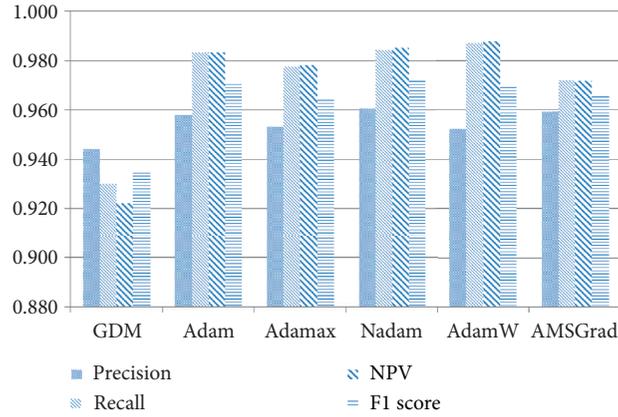


FIGURE 10: Model comparison with the indices of precision, recall, NPV, and F1 score.

Models	GDM	Adam	Adamax	Nadam	AdamW	AMSGrad
GDM	x	0.00009	0.00009	0.00009	0.00009	0.00016
Adam	0.00009	x	0.00685	0.16618	0.98260	0.06688
Adamax	0.00009	0.00685	x	0.00063	0.07101	0.91076
Nadam	0.00009	0.16618	0.00063	x	0.09263	0.00426
AdamW	0.00009	0.98260	0.07101	0.09263	x	0.08328
AMSGrad	0.00016	0.06688	0.91076	0.00426	0.08328	x

(a)

Models	GDM	Adam	Adamax	Nadam	AdamW	AMSGrad
GDM	x	--	--	--	--	--
Adam	++	x	++	-	+	+
Adamax	++	--	x	--	-	+
Nadam	++	+	++	x	+	++
AdamW	++	-	+	-	x	+
AMSGrad	++	-	-	--	-	x

(b)

FIGURE 11: Results of the Wilcoxon signed-rank test: (a) results of p values and (b) test outcomes. *Note.* The symbols ++, +, --, and - denote a significant win, a win, a significant loss, and a loss.

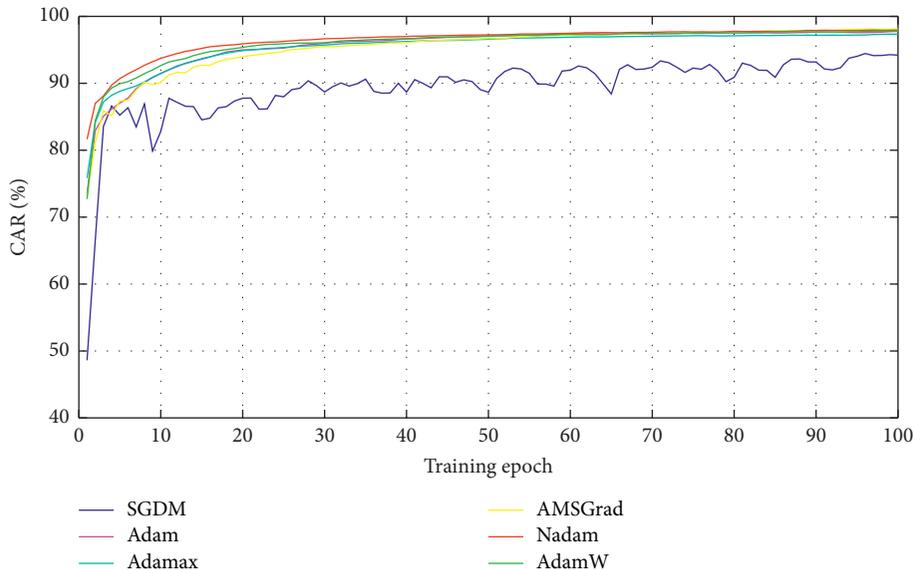


FIGURE 12: Training progress comparison.

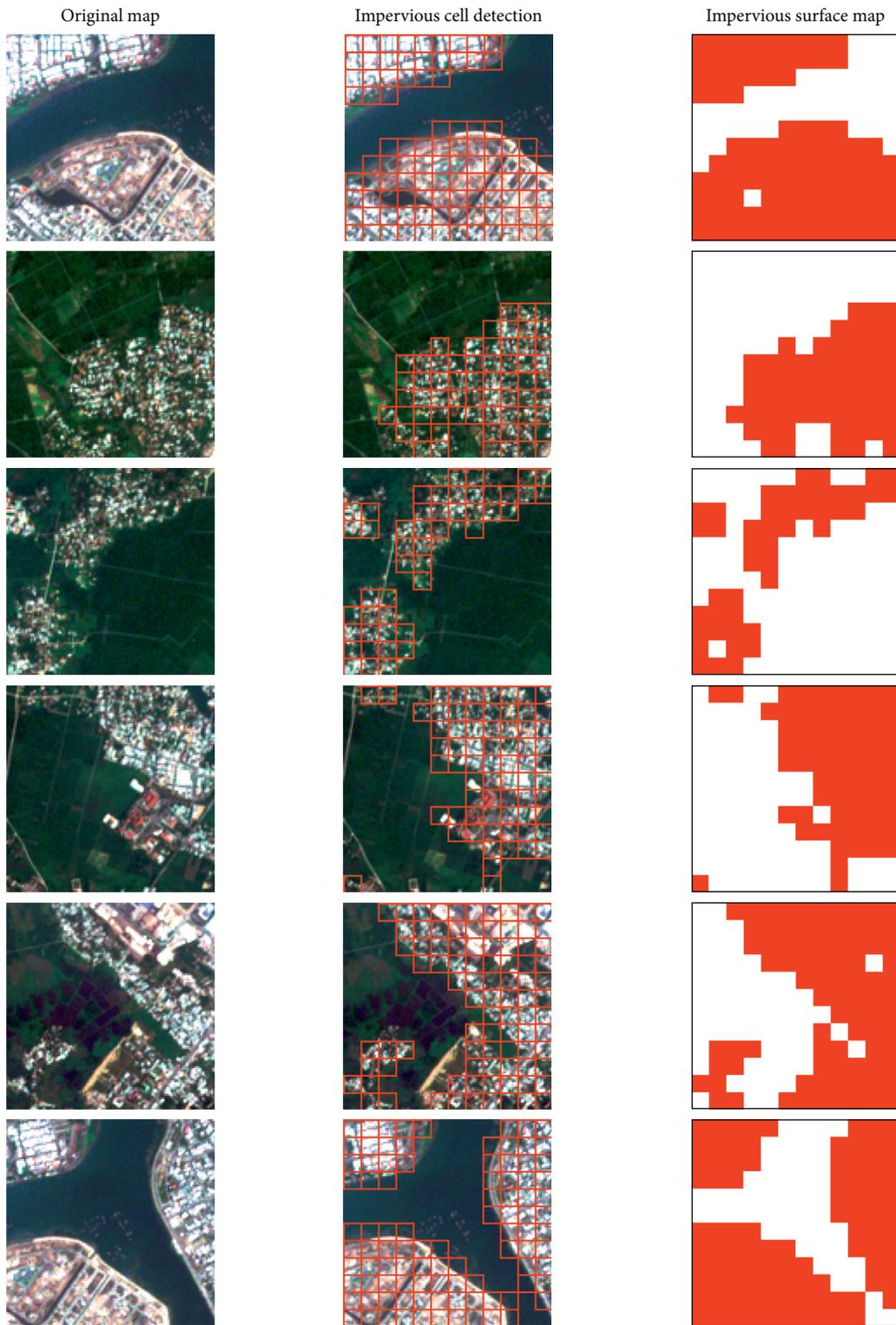


FIGURE 13: Demonstrations of the model classification outcomes with small-scale maps. *Note.* A red cell and a white cell denote impervious and pervious areas, respectively.

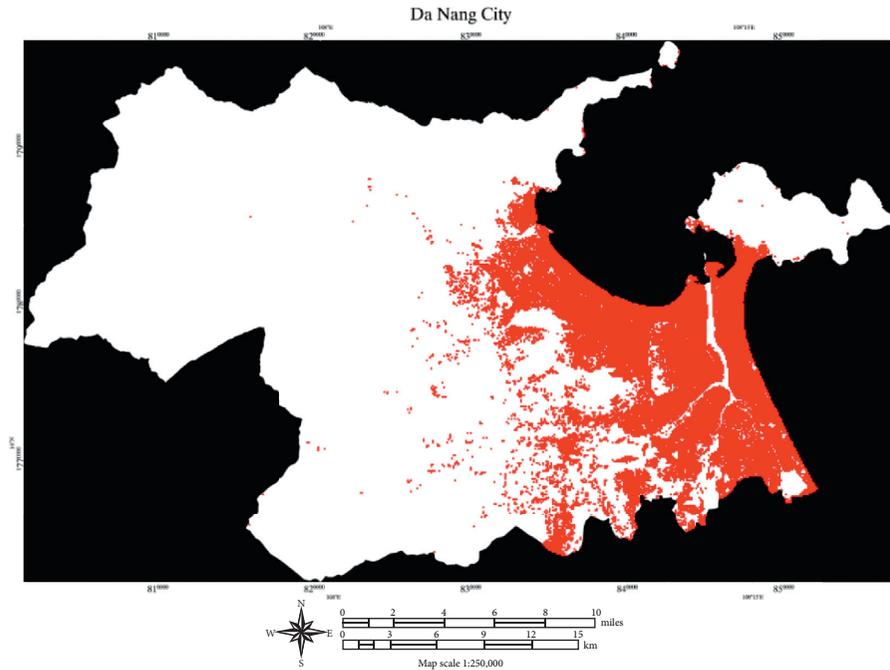


FIGURE 14: The map illustrates the impervious surface area for Da Nang city area. *Note.* A red cell and a white cell denote an impervious and pervious area, respectively.

Furthermore, to confirm the statistical difference of each pair of the neural computing models used for impervious surface detection, the Wilcoxon signed-rank test with significance level (p value) = 0.05 is used. The test results are provided in Figure 11. Observed from the test outcomes, all of the advanced optimizers including Adam, Adamax, Nadam, AdamW, and AMSGrad significantly outperformed the conventional GDM. The Nadam as the best approach achieves three significant wins and two wins. Notably, the benchmark method of GDM gets five significant losses. The average convergence records of all the employed optimizers are also provided in Figure 12.

The experimental results have demonstrated the superiority of the Nadam optimizer in constructing the neural computing model-based impervious surface detection for the study area. The outstanding performance of the Nadam algorithm can be explained by the fact that this advanced optimizer is a combination of the powerful Adam and Nesterov-accelerated gradient (NAG) approaches. The Adam optimizer has a significant advantage of computing adaptive learning rates for each parameter of the neural computing model [73]. Moreover, since Nesterov momentum provides a correction factor to the standard method of momentum, the NAG often results in good training performance [74]. The Nadam algorithm harnesses the advantages of the Adam and NAG approaches. Therefore, this optimizer has achieved the most desired performance for the collected dataset.

Since the Nadam-optimized neural computing model, denoted as Nadam-NCM, has achieved an outstanding accuracy of 97.311%, this model can be employed to accomplish the objective of impervious surface mapping in a reliable manner. Figure 13 demonstrates the application of the Nadam-

based model in detecting impervious surface for small-scaled maps. The impervious surface map of the study area is provided in Figure 14. Based on the classification result, the impervious surface areas account for roughly 18.25% of the study area.

5. Concluding Remarks

Up-to-date information regarding the impervious surface areas is crucial for the task of land-use planning, monitoring, and management. This study investigates the employment of neural computing models trained by the advanced optimizers used for automatic impervious surface area detection. The conventional GDM algorithm and the advanced optimizers of the Adam, Adamax, Nadam, AdamW, and AMSGrad are employed to train the neural computing models used for the pattern recognition task of interest. Experimental results supported by the Wilcoxon signed-rank test points out that the Nadam-optimized neural computing model has achieved the most desired predictive accuracy with CAR = 97.311%. Therefore, this model can potentially serve as an effective tool for extracting built-up impervious surfaces at regional scale. Future extensions of the current work may include the following:

- (i) The application of the Nadam-optimized neural computing model in impervious surface extraction for other study areas
- (ii) Investigation of capabilities of other advanced optimizers in training neural network models
- (iii) Exploring the effect of different neural network structures (e.g., different activation functions and number of neurons in the hidden layer) on the

- accuracy of the impervious surface detection problem
- (iv) Incorporating state-of-the-art regularization techniques (e.g., dropout regularization) into the training process of the neural computing models
 - (v) Investigating the possibility of using metaheuristic algorithms to meliorate the model training performance
 - (vi) The employment of other advanced texture descriptors for improving the classification accuracy rate
 - (vii) Incorporation of statistical and metaheuristic-based feature selection methods into the current model to further enhance the prediction accuracy
 - (viii) Investigation of capabilities of other advanced machine learning models (e.g., deep learning and Markov models) for impervious surface extraction
 - (ix) Developing intelligent models for predicting time series of remotely sensed impervious surface data with other advanced neural computing models including recurrent neural network and long short-term memory.

Data Availability

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

Conflicts of Interest

The author confirms that there are no conflicts of interest regarding the publication of this article.

Acknowledgments

This research was financially supported by Duy Tan University.

References

- [1] J. Dams, J. Dujardin, R. Reggers, I. Bashir, F. Canters, and O. Batelaan, "Mapping impervious surface change from remote sensing for hydrological modeling," *Journal of Hydrology*, vol. 485, pp. 84–95, 2013.
- [2] A. Ebrahimian, J. S. Gulliver, and B. N. Wilson, "Effective impervious area for runoff in urban watersheds," *Hydrological Processes*, vol. 30, no. 20, pp. 3717–3729, 2016.
- [3] C. Li, M. Liu, Y. Hu, M. Zong, M. Zhao, and M. Todd Walter, "Characteristics of impervious surface and its effect on direct runoff: a case study in a rapidly urbanized area," *Water Supply*, vol. 19, pp. 1885–1891, 2019.
- [4] S. H. Li, L. Hong, B. X. Jin, J. S. Zhou, and S. Y. Peng, "Spatiotemporal patterns of impervious surface area and water quality response in the fuxian lake watershed," *Journal of Environmental and Public Health*, vol. 2020, Article ID 4749765, 12 pages, 2020b.
- [5] L. Yang, C. Huang, C. G. Homer, B. K. Wylie, and M. J. Coan, "An approach for mapping large-area impervious surfaces: synergistic use of Landsat-7 ETM+ and high spatial resolution imagery," *Canadian Journal of Remote Sensing*, vol. 29, no. 2, pp. 230–240, 2003.
- [6] Z. Sun, H. Guo, X. Li, L. Lu, and X. Du, "Estimating urban impervious surfaces from Landsat-5 TM imagery using multilayer perceptron neural network and support vector machine," *Journal of Applied Remote Sensing*, vol. 5, Article ID 053501, 2011.
- [7] S. Gupta, S. Islam, and M. M. Hasan, "Analysis of impervious land-cover expansion using remote sensing and GIS: a case study of Sylhet sadar upazila," *Applied Geography*, vol. 98, pp. 156–165, 2018.
- [8] M. Misra, D. Kumar, and S. Shekhar, "Assessing machine learning based supervised classifiers for built-up impervious surface area extraction from sentinel-2 images," *Urban Forestry & Urban Greening*, vol. 53, Article ID 126714, 2020.
- [9] H. S. Sudhira, T. V. Ramachandra, and K. S. Jagadish, "Urban sprawl: metrics, dynamics and modelling using GIS," *International Journal of Applied Earth Observation and Geo-information*, vol. 5, pp. 29–39, 2004.
- [10] H. Xu, D. Lin, and F. Tang, "The impact of impervious surface development on land surface temperature in a subtropical city: Xiamen, China," *International Journal of Climatology*, vol. 33, no. 8, pp. 1873–1883, 2013.
- [11] Y. Huang, Z.-X. Chen, T. Yu, X.-Z. Huang, and X.-F. Gu, "Agricultural remote sensing big data: management and applications," *Journal of Integrative Agriculture*, vol. 17, no. 9, pp. 1915–1931, 2018.
- [12] J. Jung, M. Maeda, A. Chang, M. Bhandari, A. Ashapure, and J. Landivar-Bowles, "The potential of remote sensing and artificial intelligence as tools to improve the resilience of agriculture production systems," *Current Opinion in Biotechnology*, vol. 70, pp. 15–22, 2021.
- [13] L. H. Nguyen, J. Zhu, Z. Lin et al., "Spatial-Temporal multi-task learning for within-field cotton yield prediction," in *Advances in Knowledge Discovery and Data Mining*, pp. 343–354, Springer International Publishing, Cham, Switzerland, 2019.
- [14] C. du Plessis, G. van Zijl, J. Van Tol, and A. Manyevere, "Machine learning digital soil mapping to inform gully erosion mitigation measures in the Eastern Cape, South Africa," *Geoderma*, vol. 368, Article ID 114287, 2020.
- [15] K. Khosravi, B. T. Pham, K. Chapi et al., "A comparative assessment of decision trees algorithms for flash flood susceptibility modeling at Haraz watershed, northern Iran," *Science of the Total Environment*, vol. 627, pp. 744–755, 2018.
- [16] A. Merghadi, "Machine learning methods for landslide susceptibility studies: a comparative overview of algorithm performance," *Earth-science Reviews*, vol. 207, Article ID 103225, 2020.
- [17] P.-T. T. Ngo, T. D. Pham, V.-H. Nhu et al., "A novel hybrid quantum-PSO and credal decision tree ensemble for tropical cyclone induced flash flood susceptibility mapping with geospatial data," *Journal of Hydrology*, Article ID 125682, 2020, In press.
- [18] I. Dochev, P. Gorzalka, V. Weiler et al., "Calculating urban heat demands: an analysis of two modelling approaches and remote sensing for input data and validation," *Energy and Buildings*, vol. 226, Article ID 110378, 2020.
- [19] F. Huang, Y. Yu, and T. Feng, "Automatic building change image quality assessment in high resolution remote sensing based on deep learning," *Journal of Visual Communication and Image Representation*, vol. 63, Article ID 102585, 2019.
- [20] M. Li, A. Stein, and K. M. de Beurs, "A Bayesian characterization of urban land use configurations from VHR remote

- sensing images,” *International Journal of Applied Earth Observation and Geoinformation*, vol. 92, Article ID 102175, 2020a.
- [21] C.-Y. Chen, H. W. Chen, C.-T. Sun, Y. H. Chuang, K. L. P. Nguyen, and Y. T. Lin, “Impact assessment of river dust on regional air quality through integrated remote sensing and air quality modeling,” *Science of the Total Environment*, vol. 755, Article ID 142621, 2021.
- [22] H. Lamphar, “Spatio-temporal association of light pollution and urban sprawl using remote sensing imagery and GIS: a simple method based in Otsu’s algorithm,” *Journal of Quantitative Spectroscopy and Radiative Transfer*, vol. 251, Article ID 107060, 2020.
- [23] K. Shi, J. Shen, L. Wang, M. Ma, and Y. Cui, “A multiscale analysis of the effect of urban expansion on PM_{2.5} concentrations in China: Evidence from multisource remote sensing and statistical data,” *Building and Environment*, vol. 174, Article ID 106778, 2020.
- [24] S. Hooper and R. E. Kennedy, “A spatial ensemble approach for broad-area mapping of land surface properties,” *Remote Sensing of Environment*, vol. 210, pp. 473–489, 2018.
- [25] S. Pok, B. Matsushita, and T. Fukushima, “An easily implemented method to estimate impervious surface area on a large scale from MODIS time-series and improved DMSP-OLS nighttime light data,” *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 133, pp. 104–115, 2017.
- [26] V. Demir and O. Kisi, “Flood hazard mapping by using geographic information system and hydraulic model: mert river, samsun, Turkey,” *Advances in Meteorology*, vol. 2016, Article ID 4891015, 9 pages, 2016.
- [27] X. Li, Y. Zhou, Z. Zhu, L. Liang, B. Yu, and W. Cao, “Mapping annual urban dynamics (1985–2015) using time series of Landsat data,” *Remote Sensing of Environment*, vol. 216, pp. 674–683, 2018.
- [28] X. Yang and Z. Liu, “Use of satellite-derived landscape imperviousness index to characterize urban spatial growth,” *Computers, Environment and Urban Systems*, vol. 29, no. 5, pp. 524–540, 2005.
- [29] Y. Zha, J. Gao, and S. Ni, “Use of normalized difference built-up index in automatically mapping urban areas from TM imagery,” *International Journal of Remote Sensing*, vol. 24, no. 3, pp. 583–594, 2003.
- [30] C. P. Lo, “Land use mapping of Hong Kong from Landsat images,” *An Evaluation International Journal of Remote Sensing*, vol. 2, pp. 231–252, 1981.
- [31] J. Zhang, P. Li, and J. Wang, “Urban built-up area extraction from Landsat TM/ETM+ images using spectral information and multivariate texture,” *Remote Sensing*, vol. 6, no. 8, pp. 7339–7359, 2014a.
- [32] Y. Zhang, H. Zhang, and H. Lin, “Improving the impervious surface estimation with combined use of optical and SAR remote sensing images,” *Remote Sensing of Environment*, vol. 141, pp. 155–167, 2014b.
- [33] N. Patel and R. Mukherjee, “Extraction of impervious features from spectral indices using artificial neural network,” *Arabian Journal of Geosciences*, vol. 8, no. 6, pp. 3729–3741, 2015.
- [34] N.-T. Son, C.-F. Chen, C.-R. Chen, and S.-H. Chiang, “Mapping urban growth of the capital city of Honduras from Landsat data using the impervious surface fraction algorithm,” *Geocarto International*, vol. 31, no. 3, pp. 328–341, 2016.
- [35] P. Jian, L. Yanxu, S. Hong, X. Pan, H. Xiaoxu, and W. Yanglin, “Using impervious surfaces to detect urban expansion in Beijing of China in 2000s,” *Chinese Geographical Science*, vol. 26, no. 2, pp. 229–243, 2016.
- [36] R. Xu, J. Liu, and J. Xu, “Extraction of high-precision urban impervious surfaces from sentinel-2 multispectral imagery via modified linear spectral mixture analysis,” *Sensors*, vol. 18, no. 9, p. 2873, 2018.
- [37] R. V. d. L. Silva and J. M. V. Grzybowski, “Artificial neural network ensembles applied to the mapping of landslide susceptibility,” *Catena*, vol. 184, Article ID 104240, 2020.
- [38] R. Costache, Q. Bao Pham, E. Corodescu-Roşca et al., “Using GIS, remote sensing, and machine learning to highlight the correlation between the land-use/land-cover changes and flash-flood potential,” *Remote Sensing*, vol. 12, no. 9, p. 1422, 2020.
- [39] S. Gopal, “Artificial neural networks in geospatial analysis,” in *International Encyclopedia of Geography: People, the Earth, Environment and Technology*, pp. 1–7, Wiley, Hoboken, NJ, USA, 2016.
- [40] N.-D. Hoang and D. Tien Bui, “GIS-based landslide spatial modeling using batch-training back-propagation artificial neural network: a study of model parameters,” in *Advances and Applications in Geospatial Technology and Earth Resources*, pp. 239–254, Springer International Publishing, Cham, Switzerland, 2018.
- [41] B. Kalantar, B. Pradhan, S. A. Naghibi, A. Motevalli, and S. Mansor, “Assessment of the effects of training data selection on the landslide susceptibility mapping: a comparison between support vector machine (SVM), logistic regression (LR) and artificial neural networks (ANN),” *Geomatics, Natural Hazards and Risk*, vol. 9, no. 1, pp. 49–69, 2018.
- [42] B. T. Pham, D. Tien Bui, I. Prakash, and M. B. Dholakia, “Hybrid integration of multilayer perceptron neural networks and machine learning ensembles for landslide susceptibility assessment at Himalayan area (India) using GIS,” *Catena*, vol. 149, pp. 52–63, 2017.
- [43] C. Polykretis and C. Chalkias, “Comparison and evaluation of landslide susceptibility maps obtained from weight of evidence, logistic regression, and artificial neural network models,” *Natural Hazards*, vol. 93, no. 1, p. 249, 2018.
- [44] D. Tien Bui, H. V. Le, and N.-D. Hoang, “GIS-based spatial prediction of tropical forest fire danger using a new hybrid machine learning method,” *Ecological Informatics*, vol. 48, pp. 104–116, 2018.
- [45] K. Hornik, “Approximation capabilities of multilayer feed-forward networks,” *Neural Networks*, vol. 4, pp. 251–257, 1991.
- [46] C. Zhou, K. Yin, Y. Cao et al., “Landslide susceptibility modeling applying machine learning methods: a case study from Longju in the Three Gorges Reservoir area, China,” *Computers & Geosciences*, vol. 112, pp. 23–37, 2018.
- [47] F. Han, J. Jiang, Q.-H. Ling, and B.-Y. Su, “A survey on metaheuristic optimization for random single-hidden layer feedforward neural network,” *Neurocomputing*, vol. 335, pp. 261–273, 2019.
- [48] B. Yoshua and L. Yann, “A Method for Stochastic Optimization,” in *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*, San Diego, CA, USA, May 2015.
- [49] Dozat T. (2016) Incorporating Nesterov Momentum into Adam ICLR Workshop, (1), 2013–2016.
- [50] I. Loshchilov and F. Hutter, *Decoupled Weight Decay Regularization*, 2017, <https://arxiv.org/abs/1711.05101>.
- [51] S. J. Reddi, S. Kale, and S. Kumar, *On the Convergence of Adam and beyond*, 2019, <https://arxiv.org/abs/1904.09237>.

- [52] S. Theodoridis and K. Koutroumbas, *Pattern Recognition*, Academic Press, Cambridge, MA, USA, 2009.
- [53] A. Fernández, M. X. Álvarez, and F. Bianconi, “Image classification with binary gradient contours,” *Optics and Lasers in Engineering*, vol. 49, pp. 1177–1184, 2011.
- [54] MPI, *Da Nang City Ministry of Planning and Investment*, MIP, Winnipeg, Canada, 2020, <http://www.mpi.gov.vn/Pages/tinhthanhchitiet.aspx?idTinhThanh=41>.
- [55] N. Dang, T. T. V. Le, V. C. M. Tran, and M. Yasuhiro, “Scenario analysis on operation efficiency for waste collection and transport: a case study in da Nang city,” *Vietnam Journal of Environmental and Social Sciences*, vol. 5, no. 1, 2017.
- [56] T. T. Phan Hoang and T. Kato, “Measuring the effect of environmental education for sustainable development at elementary schools: a case study in Da Nang city, Vietnam,” *Sustainable Environment Research*, vol. 26, no. 6, pp. 274–286, 2016.
- [57] D. Rockwood and D. Q. Tran, “Urban immigrant worker housing research and design for Da Nang, Viet Nam,” *Sustainable Cities and Society*, vol. 26, pp. 108–118, 2016.
- [58] ESA, *Sentinel Application Platform (SNAP) the European Space Agency*, ESA, Paris, France, 2020, <http://stepesaint/main/toolboxes/snap/>.
- [59] (2020) Earth Explorer US Department of the Interior, <https://earthexplorer.usgs.gov/>.
- [60] ESA, *Documentation European Space Agency*, ESA, Paris, France, 2020, <http://stepesaint/main/doc/tutorials/>.
- [61] M. Sonka, V. Hlavac, and R. Boyle, *Image processing, Analysis, and Machine Vision*, Cengage Learning, Boston, MA, USA, 2013.
- [62] T. Lillesand, R. W. Kiefer, and J. Chipman, *Remote Sensing and Image Interpretation*, John Wiley & Sons, Hoboken, NJ, USA, 2015.
- [63] J. W. Rouse, R. H. Haas, J. A. Schell, and D. W. Deering, “Monitoring vegetation systems in the great plains with ERTS,” in *Conference Proceedings of the Third ERTS Symposium, NASA SP-351*, pp. 309–317, Washington, DC, USA, 1973.
- [64] N.-D. Hoang, “Image processing based automatic recognition of asphalt pavement patch using a metaheuristic optimized machine learning approach,” *Advanced Engineering Informatics*, vol. 40, pp. 110–120, 2019.
- [65] A. Humeau-Heurtier, “Texture feature extraction methods,” *A Survey IEEE Access*, vol. 7, pp. 8975–9000, 2019.
- [66] I. El Khadiri, M. Kas, Y. El Merabet, Y. Ruichek, and R. Touahni, “Repulsive-and-attractive local binary gradient contours: New and efficient feature descriptors for texture classification,” *Information Sciences*, vol. 467, pp. 634–653, 2018.
- [67] C. C. Aggarwal, *Neural Networks and Deep Learning*, Springer, Berlin, Germany, 2018.
- [68] S. O. Haykin, *Neural Networks and Learning Machines*, Pearson, London, UK, 2008.
- [69] T. Hegazy, P. Fazio, and O. Moselhi, “Developing practical neural network applications using back-propagation,” *Computer-Aided Civil and Infrastructure Engineering*, vol. 9, pp. 145–159, 1994.
- [70] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning representations by back-propagating errors,” *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.
- [71] Y. A. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller, G. B. O. Grégoire Montavon and Klaus-Robert Müller, Ed., “Efficient BackProp,” in *Lecture Notes in Computer Science Vol 7700 2012 (Neural Networks Tricks of the Trade)*, Springer Berlin Heidelberg, Berlin, Germany, 2002.
- [72] G. Hinton, N. Srivastava, and K. Swersky, *Neural Networks for Machine Learning Lecture Notes*, University of Toronto, Toronto, Canada, 2012, <http://www.cs.toronto.edu/~hinton/coursera/lecture4/lec4.pdf>.
- [73] S. Ruder, “An overview of gradient descent optimization algorithms,” 2017, <https://arxiv.org/abs/1609.04747>.
- [74] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning (Adaptive Computation and Machine Learning Series)*, The MIT Press, Cambridge, MA, USA, 2016.
- [75] J. Heaton, *Introduction to Neural Networks for C#*, Heaton Research, Inc., Washington, DC, USA, 2008.
- [76] Apress, *MatLab Deep Learning with Machine Learning, Neural Networks and Artificial Intelligence*, Apress, New York, NY, USA, 2017.
- [77] P.-T. Ngo, N.-D. Hoang, B. Pradhan et al., “A novel hybrid swarm optimized multilayer neural network for spatial prediction of flash floods in tropical areas using sentinel-1 SAR imagery and geospatial data,” *Sensors*, vol. 18, no. 11, p. 3704, 2018.
- [78] J. Heaton, *Artificial Intelligence for Humans, Volume 3 Deep Learning and Neural Networks*, Heaton Research, Inc., Washington, DC, USA, 2015.
- [79] 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.