This is the accepted version of the publication Mohammed Abdelkader E, Moselhi O, Marzouk M, Zayed T., Hybrid Elman Neural Network and an Invasive Weed Optimization Method for Bridge Defect Recognition, Transportation Research Record (2021;2675(3)) pp. 167-199. Copyright © 2020 National Academy of Sciences: Transportation Research Board. DOI: 10.1177/0361198120967943.

Hybrid Elman Neural Network and Invasive Weed Optimization Method for Bridge Defects Recognition

Eslam Mohammed Abdelkader

1- Department of Building, Civil, and Environmental Engineering

Concordia University, Montreal, QC, Canada

2- Structural Engineering Department, Faculty of Engineering

Cairo University, Egypt

Email: eslam_ahmed1990@hotmail.com

Corresponding author

Osama Moselhi

Department of Building, Civil, and Environmental Engineering

Concordia University, Montreal, QC, Canada

Email: O_Moselhi@encs.concordia.ca

Mohamed Marzouk

Structural Engineering Department, Faculty of Engineering

Cairo University, Egypt

Email: mm_marzouk@yahoo.com

Tarek Zayed

Department of Building and Real Estate

The Hong Kong Polytechnic University, Hung Hom, Hong Kong

Email: tarek.zayed@polyu.edu.hk

Abstract

Existing bridges are aging and deteriorating; raising concerns for public safety and preservation of these valuable assets. Furthermore, large number of bridges exists in transportation networks; simultaneously maintenance budgets are being squeezed. These states of affairs necessitate the development of a computer vision-based method in an attempt to alleviate shortcomings of visual inspection-based methods. In this context, the present study proposes a three-tier method designated for the automated detection and recognition of bridge defects. In the first tier, singular value decomposition (SVD) is adopted for the sake of formulating the feature vector set through mapping most dominant spatial domain features in images. The second tier encompasses a hybridization of Elman neural network (ENN) and invasive weed optimization algorithm (IWO) to enhance the prediction performance of the Elman neural network. This is accomplished through designing a variable optimization mechanism that aims at searching for the optimum exploration-exploitation trade-off in the neural network. The third tier involves its validation through comparisons against a set of conventional machine learning and deep learning models capitalizing on performance prediction and statistical significance tests. A computerized platform was programmed in C#.net to facilitate its implementation by the users. It was found that the developed method outperformed other prediction models achieving overall accuracy, F-measure, Kappa coefficient, balanced accuracy, Matthews's correlation coefficient and area under curve of 0.955, 0.955, 0.914, 0.965, 0.937 and 0.904, respectively as per cross validation. It is expected that the developed method can improve the decision-making process in bridge management systems.

Keywords: Bridges, computer vision, singular value decomposition, Elman recurrent neural network, invasive weed optimization, C#.net

1. INTRODUCTION

Bridges are regarded as one of the core elements of the infrastructure systems. Meanwhile, they are vulnerable to severe deterioration agents such as freeze-thaw cycles, excessive distress loads due to the traffic overload, sulfates, alkali-silica reaction (ASR), poor construction practices, etc. As per the Canadian infrastructure report card, 26% of the bridges are either “Fair”, “Poor” or “Very Poor” [1]. Moreover, one-third of Canada’s bridges were reported to have structural or functional deficiencies with short remaining service life, whereas 20 million light vehicles, 750,000 trucks, and 15,000 public transits use the Canadian bridges annually (National Research Council Canada, 2013). The average age of the bridges is 24.5 years in 2007 compared to a mean service life of 43.3 years. Therefore, 57% of the estimated service life has already been consumed [2]. In addition to that, the backlog of bridge maintenance, rehabilitation and replacement is estimated to be equal to $10 billion. The continuous increase in the backlog results in a significant deterioration in the condition of the bridge elements [3].

In view of the above, it is very decisive to evaluate the condition of the bridge decks in order to preserve them within safe condition and to ensure the public safety. As such, maintenance-related interventions should be condition-driven in order to maintain the structural health, reliability and durability of bridge decks within allocated budget limitations. The current routine inspections are usually carried out by inspectors to evaluate the condition of the bridges based on the assessment of surface defects. These condition ratings are error-prone because the visual inspection-based methods are highly dependent on the skills and experience of inspectors, which created wide variations among the evaluations of the inspectors. Therefore, the subjectivity, uncertainty and safety risks of the visual inspection process in addition to its labor extensive and time-consuming nature lead in turn to imprecise judgments.

Recently, the use of computer vision-based methods became a trend to evaluate surface defects in concrete structures. The development of automated detection and evaluation methods enable the transportation agencies to overcome the drawbacks of the visual inspection-based methods, which are considered as the most common practice to monitor the condition of the bridge decks. Computer vision-based methods aid in obtaining more objective and precise defects evaluation. Additionally, they are characterized by being time and cost efficient [4-6]. Detection of defects is not the only required task but also their recognition and evaluation of their magnitude of severities. In this regard, each surface defect is assessed using a set of descriptors and different types of defects require different types of maintenance. Ground truth annotated dataset is essential for the detection and recognition of surface and subsurface defects. For instance, establishing a labelled dataset for non-destructive tests is usually a more difficult task than visual images [7]. One of the approaches to establish a ground truth is to take number of cores at different locations [8]. Another approach is to generate an agreement map based on group of inspection techniques including infrared thermography, hammer sounding and limited physical sampling [9]. In view of this situation, the development of automated detection and recognition method is vital as an initial stage to objectively evaluate the condition of the bridge elements. As such, the main objectives of the present study are as follows:

1- Develop a self-adaptive hybrid $\mathrm{SVD-ENN-IWO}$ method designated for the automated detection and recognition of surface defects
2- Validate the proposed method through comparison with some widely-recognized deep learning and machine learning models.

2. LITERATURE REVIEW

Several previous machine learning-based methods were developed for the automated and accurate detection and recognition of surface defects. Yao et al. [10] presented a bridge crack detection and classification model based on a climbing root using a set of image processing techniques. Wiener filtering method was applied to remove the motion blur of the acquired images. Then, the wavelet transform was employed to minimize the texture effects of the crack area and finally, support vector machine (SVM) was implemented to classify the cracks and evaluate their severity levels. Tong et al. [11] presented a new method for image-based crack detection to facilitate the automatic bridge inspection process. Gaussian filter was used to remove the noise and enhance the image quality. Morphological operations are used to ensure the connection between the crack segments. The objective of the model was to decide whether the binary images represent a crack or not based on some criteria such as circularity of the region, aspect ratio, perimeter and area. The proposed model achieved an accuracy of 93% and it outperformed some other methods such as Fujita method, canny edge detection method and Sobel edge detection method.

Moon and Kim [12] proposed an automatic system for crack detection using some image processing techniques to enable the inspectors to perform the crack monitoring task effectively. The irregular illumination present in the images was removed using improved subtraction method by applying a median filter and then gray image is subtracted from the enhanced image. Then a Gaussian low-pass filter was utilized to connect small gaps and to adjust the distortion in the crack shape. The tuning parameters such as threshold value, median filter size, Gaussian filter size, standard deviation of the Gaussian filter, were determined via a set of organized experiments using signal to noise ratio metric. Finally, a back-propagation artificial neural network is designed to binary classify whether the concrete images contain cracks or not. The model was capable to detect crack images by 90% and non-crack images by 92%, whereas the number of hidden layers was defined based on experience. Xuejun and Yan [13] developed a bridge crack detection system using video frame processing. The classification of bridge cracks was performed via deep belief network (DBN). The proposed model was capable of achieving a classification accuracy of 94%, 93% and 90% for the transverse cracks, longitudinal cracks, and network cracks. DBN outperformed some conventional utilized classification methods such as SVM and back propagation neural network.

Chen et al. [14] proposed a method for the detection of concrete cracks using Otsu algorithm. In the developed model, Gaussian filter was applied to remove noise from background. Morphological operations were applied to remove noise from the segmented image while maintaining the shape features of the cracking. It was highlighted that the developed model could efficiently generate a binary image of cracks. Dawood et al. [15] proposed a machine vision-based method for the evaluation of spalling in subway network. In it, Gaussian filter was employed to remove the noise and enhance the image details. The developed method utilized a histogram-based thresholding that encompassed the selection of optimum threshold value according to trial and error technique. Artificial neural network model was then developed to automatically evaluate spalling area. It was inferred that the developed model could assist asset managers in establishing reliable and timely intervention plans.

Bu et al. [16] introduced a model that integrates both wavelet features and support vector machines to detect bridge cracks automatically in images. They compared between three feature extraction methods which are: Daubechies Wavelet features, Gabor filter and Zernike moments. They concluded that Daubechies Wavelet features provided the best performance followed by Gabor filter and finally Zernike moments. They also highlighted that support vector machine achieved an accuracy of 93% in normal images, 90% in complex images, and 92% in overall images. Cen et al. [17] utilized convolutional neural network to detect the presence of cracks in reinforced concrete bridges. The images were captured using unmanned aerial vehicle (UAV) such that images of size 48×48 were used as an input to train the model. They experimented different sizes of filter window, whereas they concluded that the filter window size 48×2 achieved the highest accuracy. The proposed detection method was capable of achieving 93.12% prediction accuracy using 2304 crack images and 5368 non-crack images.

Li et al. [18] introduced a two-stage crack detection method based on convolutional neural network. The first stage involved feeding a small patch centering each pixel into the predictor to compute the probability that a pixel belong to a cracked area. In the second stage, a bigger patch elicited from the first confidence map is fed into the second predictor to obtain a second confidence map. Finally, the two confidence maps are combined to generate a final confidence map, which is used to map whether or not a certain pixel belong to cracked regions. The introduced method outperformed the canny edge detector method and STRUM (Spatially tuned robust multi-feature) method as per accuracy, precision and sensitivity. Yeboah et al. [19] developed an approach for the automatic detection and classification of bridge cracks using a robotic system. They utilized radon transform and directional projection variance for feature extraction. Finally, adaBoosted RVM (Relevance Vector Machines) was utilized to classify the cracks into no-cracks, simplex cracks and complex cracks. The proposed method outperformed a set of classifiers such as adaboost, RVM, back propagation artificial neural network and prior mathematical modelling.

Kruachottikul et al. [20] utilized a pre-trained deep convolutional neural network called "Halcon" for the detection of defects in bridge sub structure surface. They established a binary classification model that enabled to determine whether the images encompassed defects or not. They highlighted that the developed transfer learning-based model achieved a total accuracy of 89.3% which could improve the inspection process conducted by the departments of highways. Xu et al. [21] developed a convolutional neural network-based model for the purpose of detection of cracks in bridges. In the developed model, the feature extraction was envisioned on the Atrous spatial pyramid pooling to map the multi-scale context. Additionally, depthwise separable convolution was applied after the convolutional layer to minimize the computational effort and number of parameters of the deep learning model. They deduced that the developed model outperformed a set of pre-trained networks by achieving accuracy, precision, sensitivity, specificity and F-measure of 96.37%, 78.11%, 100%, 95.83% and 0.8771, respectively.

Kim et al. [22] introduced a region-based convolutional neural network model coupled with transfer learning for identification of cracks in an aging concrete bridge. In this model, the deep neural network model was pre-trained using the Cifar-10 dataset. Furthermore, a dataset of 384 images was utilized for training and testing purposes. It was concluded that this model achieved a relative error of 1-2% in the quantification of cracks. Cha et al. [23] employed convolutional neural network for the detection of concrete cracks. The architecture of the deep convolutional neural network (DCNN) was composed of four convolutional layers, two pooling layers, one rectified linear unit layer and one softmax layer. The developed model was validated through comparisons against Canny and Sobel edge detection methods. It outperformed them providing training accuracy and testing accuracy of 98.22% and 97.42%, respectively.

Wang et al. [24] utilized an integration of AlexNet and VGG11 pre-trained deep convolutional neural networks for bridge crack identification. The feature maps created by the two networks are concentrated in series to be sent to a softmax classifier for categorization of cracks. It was found that the developed model yielded improvements in the prediction accuracies by 0.32% and 0.41% with respect to AlexNet and VGG11 networks, respectively. Słonski [25] compared the performances of four different architectures of deep convolutional neural networks in the automated detection of concrete surface cracks. This comprised small convolutional network with and without data augmentation, pre-trained VGG16 with data augmentation alongside VGG16 with a combination of data augmentation and fine-tuning. It was reported that the VGG16 coupled data augmentation and fine-tuning provided the highest classification performance achieving training and validation accuracies of 95% and 93%, respectively.

Dorafshan et al. [26] investigated the implementation of two modes of deep convolutional neural network in concrete crack detection. In the first mode, the AlexNet architecture was fully-trained from scratch capitalizing on the dataset captured using small unmanned aerial systems. The second mode encompassed a transfer learning-based network of same topology that was pre-trained using ImageNet dataset. The performances of the deep neural networks were evaluated stepping on three datasets. It was reported that the transfer learning-based network had higher training accuracy than the fully trained network. Furthermore, it achieved higher validation accuracies for the three datasets by values ranging from 5.3% to 10%. Dung and Anh [27] presented a crack detection method capitalizing on deep fully convolutional neural network. The main pillar of the fully convolutional neural network was VGG16 that was pre-trained using the ImageNet dataset. In this regard, the proposed encoder incorporated all the convolutional and pooling layers of the VGG16 except the fully connected and softmax layers. VGG16 was selected over other pre-trained networks including ResNet and InceptionV3 since it provided better performance in crack image classification. The encoder-decoder fully convolutional neural network was then trained end to end based on crack-labeled images dataset. It was highlighted that the developed segmentation method achieved average precision and maximum F1-score of 89.3% for the testing dataset.

In the light of the previous studies, it can be inferred that current practices of visual inspection provide subjective, inconsistent and inaccurate evaluation of the condition of the bridge decks. It can be also noticed that most of the conducted studies were primarily concerned with defects' detection alongside detection and evaluation of cracks. There is a lack of investigation of other surface defects including scaling and spalling, whereas "Defect" and "No Defect" classes or "Crack" and "No Crack" classes are not sufficient to evaluate the severities of bridge defects. In this context, this may create incomprehensive and unreliable condition assessment models that can substantially influence the maintenance prioritization and planning models in the different managerial levels.

Another shortcoming can be observed is that some of the reported models require conducting segmentation as a pre-processing to build the automated detection model of surface defects [10-

14]. In this context, the segmented image is mainly utilized as an input to the machine learning model. This increases the net computational time and complexity as a result of the increase in number of pre-processing stages. Furthermore, these segmentation models are primarily histogram-based, clustering-based, region-based or edge detection-based. In this context, the previously- mentioned segmentation models are highly variant and sensitive to low contrast and non-uniformly illuminated images due to the multimodality of intensity histograms. This may induce many error points and significant degradation in defects' extraction because the prediction accuracy of the machine learning model is becoming highly dependent on the defects' segmentation algorithm [28-30]. In this context, it is more practical and efficient to rely on the gray-level images to design the classification model of surface defects.

Another point to consider is that some models counted on feeding the whole input image directly to build the machine learning model. In this regard, high computational cost and resources are consumed per epoch during the training process endeavoring to explore the multi-dimensional space [15]. In similar context, some previous models utilize a set of simple defects' descriptors such as area, eccentricity, ratio of major axis to minor axis. These models suffer from the absence of advanced feature extraction algorithm [10-12]. A feature extraction algorithm is necessitated in the case of presence of complex and noisy texture of images of bridge deck, which are mainly characterized by weak signals of defects patterns, in homogeneity of defects and the diversity of defects [31-33]. The presence of advanced and efficient feature extraction algorithm is essential in distilling the useful features in the images [34-35]. Their absence may undermine the discrimination and learning capacity of the machine learning, and cause its failure to distinguish the defects from background elicited from its failure to delineate the important features in the input images [36-37]. In this context, more attention should be dedicated to the implementation of feature extraction algorithm in an attempt to improve the learning capacity of the machine learning model.

Some previous models counted on artificial neural network models to model the surface defects [12, 15]. In this regard, gradient descent algorithm is considered as one of the most commonly utilized for their training. It is based on finding the partial derivative of the error function to update the weights of the connections between neurons. The training process based on the gradient descent often gets trapped in a local minima or premature convergence and sometimes causes over-fitting and under-fitting problems especially in the case of presence of multilayer neural network [38-40]. The multi-layer neural network is normally linked with large search space, multi-local minima points, non-differential function and complex multi-dimensional curve [41-44]. Furthermore, in some cases, the global minimum is hidden between the local minima. Thus, the gradient descent algorithm can end up oscillating between the local minima [40].

Deep learning has been adopted in the recent years to analyze and evaluate the surface defects in the different assets. However, there are shortcomings encountered in the adoption of deep convolutional neural networks. Deep learning requires huge training dataset to capture the features and build the relationships between the set of independent variables and the dependent variables, which sometimes can be difficult to find and tedious to create. This sometimes implicate over-fitting triggered by the presence of small dataset, which may drastically influence the recognition capacity of the machine learning models [45-46]. Another disadvantage of deep learning is that sometime it induces detection latency as a result of the high processing demands [47].

Another shortcoming of the deep learning models is the presence of wide range of hyper parameters that substantially influence the performance of the deep learning models. This includes the number of filters, stride size, padding size and kernel sizes to create the feature map, number of convolutional layers, type of transfer function, type of pooling operation, number of fully connected layers, number of neurons, and weights of the connections between neurons. There are infinite numbers of possible solutions that needs exhaustive search to be carried out, which causes the manual tuning of the hyper parameters to be a very challenging and tedious task [48-50]. Also some of the reported models adopted classical machine learning models for the detection of surface defects [12, 15-16, 19]. However, these models are not fully automated and require manual tuning of their hyper parameters. In this context, the absence of automated systematic method to define the optimum parameters of the deep convolutional and machine learning models can lead to its entrapment in local minima which yields inferior solutions triggered by the long computational time and poor convergence [40, 51-53]. Hence, this necessitates the development of a self-adaptive method that can autonomously tune its parameters based on the available dataset with minimum human intervention.

Some previous studies relied on transfer learning-based deep neural networks to detect the surface defects. In this context, pre-trained source deep convolutional neural networks coupled with transfer learning mechanism, are utilized for defects' recognition rather than training the deep neural network from scratch. These models are vulnerable to negative transfer, which allude to the situations where the transfer of information learned from the source domain has a detrimental implication on the prediction of the target domain. The absence of sufficient degree of similarity between the features of the source domain and target domain undermines the learning performance of the target errand [54-55]. In view of the above, the present study proposes a self-adaptive hybrid $\mathrm{SVD}-\mathrm{ENN}-\mathrm{IWO}$ designated for the automated detection and recognition of surface defects. This is expected to enhance the prediction accuracies of defects detection and recognition alongside facilitating the implementation of autonomous bridge inspection platform, which eventually aids in establishing efficient condition assessment models and reliable maintenance prioritization plans.

3. PROPOSED METHOD

The ultimate objective of the present study is to design a self-adaptive three-tier method. It is envisioned on integration of singular value decomposition, Elman recurrent neural network and invasive weed optimization algorithm to automatically detect and classify the bridge defects in reinforced concrete bridges. The bridge defects detection is a binary classification model to detect whether or not the images contain defects. The bridge defects recognition model aims at identifying if the defected images contain cracking or spalling or scaling. In the present study, the images are manually labelled using visual inspection [56-58]. The framework of the proposed $\mathrm{SVD}-\mathrm{ENN}-\mathrm{IWO}$ method is depicted in Figure 1. The proposed method houses three main modules which are: feature extraction, hybrid parameter-structural learning and performance evaluation, whereas the output of the first module is the feature vector set while the output of the second module is the classification scheme. Finally, the output of the third module is designated for evaluating the recognition capacity of the developed method capitalizing on a set of performance prediction and statistical significance comparisons.

For the first module, the first step is to convert the true-color image RGB to the grayscale image, whereas the intensity values of the gray-scale image vary from 0 to 255. For the RGB image, R stands for red, G stands for green, and B stands for blue. The gray-scale images can improve the image processing while preserving important features of the defect. Then, the images are standardized to size 100×100 in order to facilitate the further processing stages. The proposed method utilizes Wiener filer as a frequency domain filter to remove the maximum noise from the degraded image while maintaining the significant features in the image. The proposed method adopts un-supervised SVD to capture the underlying essential features in the images by eliminating the insignificant features and reducing the computational complexity of the data, which leads to lesser computational time and more accurate analysis. SVD is utilized to compute the singular values of the images which are extracted in the form of feature vectors. This feature vector set is then used as an input to feed the Elman recurrent neural network. There are different types of feature selection algorithms including principal component analysis, singular value decomposition, non-negative matrix factorization, latent semantic analysis and locality preserving projections. In this regard, singular value decomposition is preferred over other feature extraction algorithms because it proved its efficiency in dealing with wide range of engineering application including forecasting weekly solar radiation [59], streamflow forecasting [60] and acoustic event classification [61]. Additionally, it is characterized by its low computational complexity [62-63]. It is also worth mentioning that singular value decomposition demonstrated superior dimensionality reduction accuracy against principal component analysis according to a set of performance evaluation tests [64-65].

The second module is the hybrid parameter-structural learning, whereas the proposed method utilizes invasive weed optimization to enhance the training process of the Elman neural network by addressing the exploration-exploitation trade-off dilemma. Invasive weed optimization algorithm is deployed for both parametric and structural learning, i.e., to automatically optimize the hyper parameters of Elman neural network including the weights alongside its best possible architecture. The Elman neural network is trained by designing a variable-length single-objective optimization problem which encompasses a fitness function of minimization of misclassification error. The steps of the invasive weed optimization algorithm are repeated until satisfying the convergence criteria, i.e., reaching maximum number of iterations. The optimum transfer functions, number of hidden and context layers, number of hidden and context neurons, and weights and bias terms establish the optimized Elman neural network, which is appended and utilized to simulate the instances of testing dataset. Invasive weed optimization algorithm is selected because it demonstrated its higher search capacity in solving diverse and sophisticated engineering problems such as optimal resource operation [66], optimization of energy supply systems [67] and prediction of compression index of limited-treated expansive clays [68]. In addition to this, invasive weed optimization algorithm outperformed a set of common and efficient meta-heuristics including non-dominated sorting genetic algorithm II, particle swarm optimization algorithm, artificial immune system and artificial bee colony [69-71].

The third module is carried out for the purpose of validating the recognition accuracy of the developed method capitalizing on two folds of comparison namely, performance prediction and statistical significance tests. The comparative analysis is conducted against a set of conventional machine learning models and deep learning models reported for their higher accuracies. The well-performing machine learning models encompass discriminant analysis (DA), K-nearest neighbors (KNN), random forest (RF), support vector machines and decision tree (DT), back propagation artificial neural network (ANN) and Elman neural network. More details about the afore-mentioned classifiers can be found in Rathi and Palani [72], Yang et al. [73], Feng et al. [74], Ahmad et al. [75] and El-Zahab et al. [76]. The deep learning models involve a deep convolutional neural network trained from scratch (CONVNET) alongside a group of different pre-trained deep neural network architectures, namely AlexNet, VGG16, VGG19 and CaffeNet. The prediction models are analyzed using both split validation and K-fold cross validation based on F-measure, Kappa coefficient, balanced accuracy (BACC), Matthews's correlation coefficient (MCC) and area under curve (AUC). In the present study, 10-fold cross validation is adopted to guarantee the training and testing of the whole dataset, which truncates the risk of encountering over-learning or over-fitting by the prediction models.

The second fold of comparison comprises a set of statistical significance tests. In this context, Shapiro-Wilk is at first employed to analyze the normality of accuracies of the different folds at significance level of 0.05. Parametric or non-parametric testing is then carried out capitalizing on the normality assessment of observed data. A set of box plots are created for the purpose of graphical analysis of the robustness of prediction models with respect to a certain performance indicator. The recognition accuracies of the prediction models are analyzed using a dataset constructed by the authors denoted as dataset I in addition to the bridge deck images existing in the public benchmark dataset SDNET2018 [77], which is denoted as dataset II. This is carried out for the sake of conducting a further analysis of the robustness of the proposed method in dealing with different sizes of datasets. Average ranking (AR) method is eventually utilized for the sake of establishing a unified assessment of the performances of prediction models across the different datasets. More details about the computational procedures of the average and standard deviation of rankings utilized in the AR method can be adopted from Yu et al. [78]. The previous models are automated using a computerized platform that encompasses an integration of visual C#.net and Matlab programming languages.

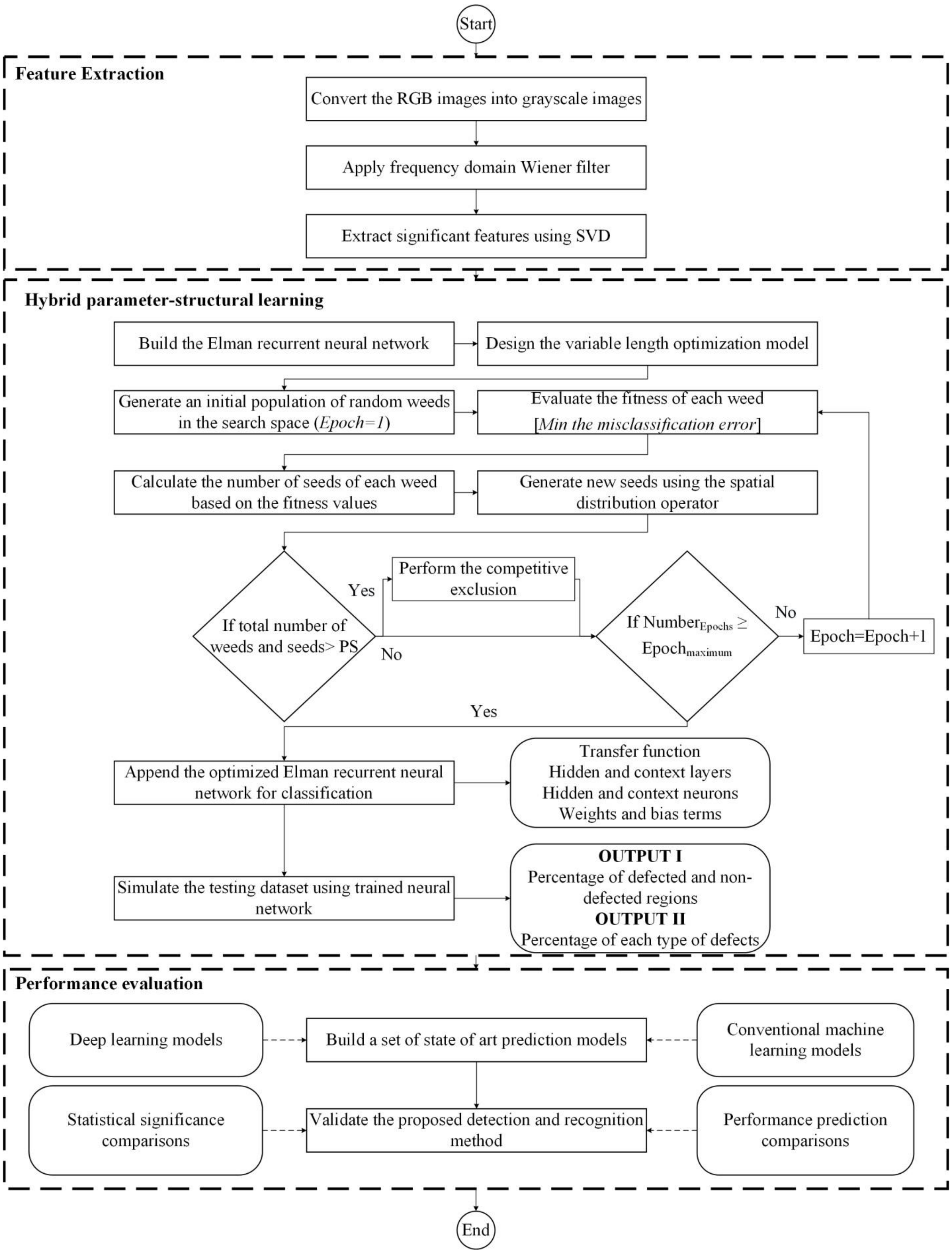


Figure 1: Framework of the proposed defects' detection and recognition method

4. MODEL DEVELOPMENT

This section delineates the algorithms and techniques presented in the "Proposed Method" section.

4.1 Singular Value Decomposition

Singular value decomposition is a powerful tool that has many applications such as data compression and pattern recognition. SVD enables robust and reliable matrix factorization in order to extract the algebraic and geometric invariant features of an image. SVD factorizes a square or non-square matrix into two orthogonal matrices and a singular value matrix. The spatial domain features of an image of size 100×100 can be modelled using singular value decomposition by a feature vector set of size 1×100 (see Figure 2). This is expected to speed up the computational process by eliminating insignificant features meanwhile preserving as much as possible information in the image. The singular value decomposition of a rectangular real complex matrix A is expressed as follows [79-80].

$$A = U\Sigma V^T = \begin{bmatrix} u_{11} & \cdots & u_{1m} \\ u_{21} & \dots & u_{2m} \\ \vdots & \ddots & \vdots \\ u_{m1} & \cdots & u_{mm} \end{bmatrix}_{m\times m} \times \begin{bmatrix} s_1 & 0 & \cdots & 0 \\ 0 & s_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & s_m \end{bmatrix}_{m\times n} \times \begin{bmatrix} v_{11} & \cdots & v_{1n} \\ v_{21} & \dots & v_{2n} \\ \vdots & \ddots & \vdots \\ v_{n1} & \cdots & v_{nn} \end{bmatrix}_{n\times n}^{T} \tag{1}$$

Such that;

$$U\,U^T = I \tag{2}$$

$$V\,V^T = I \tag{3}$$

$$s_1 \geq s_2 \geq s_3 \dots \dots . \geq s_m \tag{4}$$

Where;

A is a $m \times n$ matrix. U is a $m \times m$ orthonormal matrix. V is a $n \times n$ orthonormal matrix. Σ is a diagonal matrix of size $m \times n$ which is composed of singular values of A such that it holds non-negative numbers. The diagonal entries of the Σ matrix represent the singular values and they have higher values compared to the entries of U and V such that a matrix of size $m \times n$ can be reduced to a vector of size n. The singular values are ranked in a descending order, whereas the first entries of the singular value matrix contain the most substantial information while the last entries at the vector contain the least significant information. The singular values contain the energy information while the orthogonal matrices contain the intrinsic information. U^T and V^T are the transpose of matrices U and V, respectively. I is the identity matrix. The columns of U are called left singular vectors of A while the columns of V are called the right singular vectors of A.

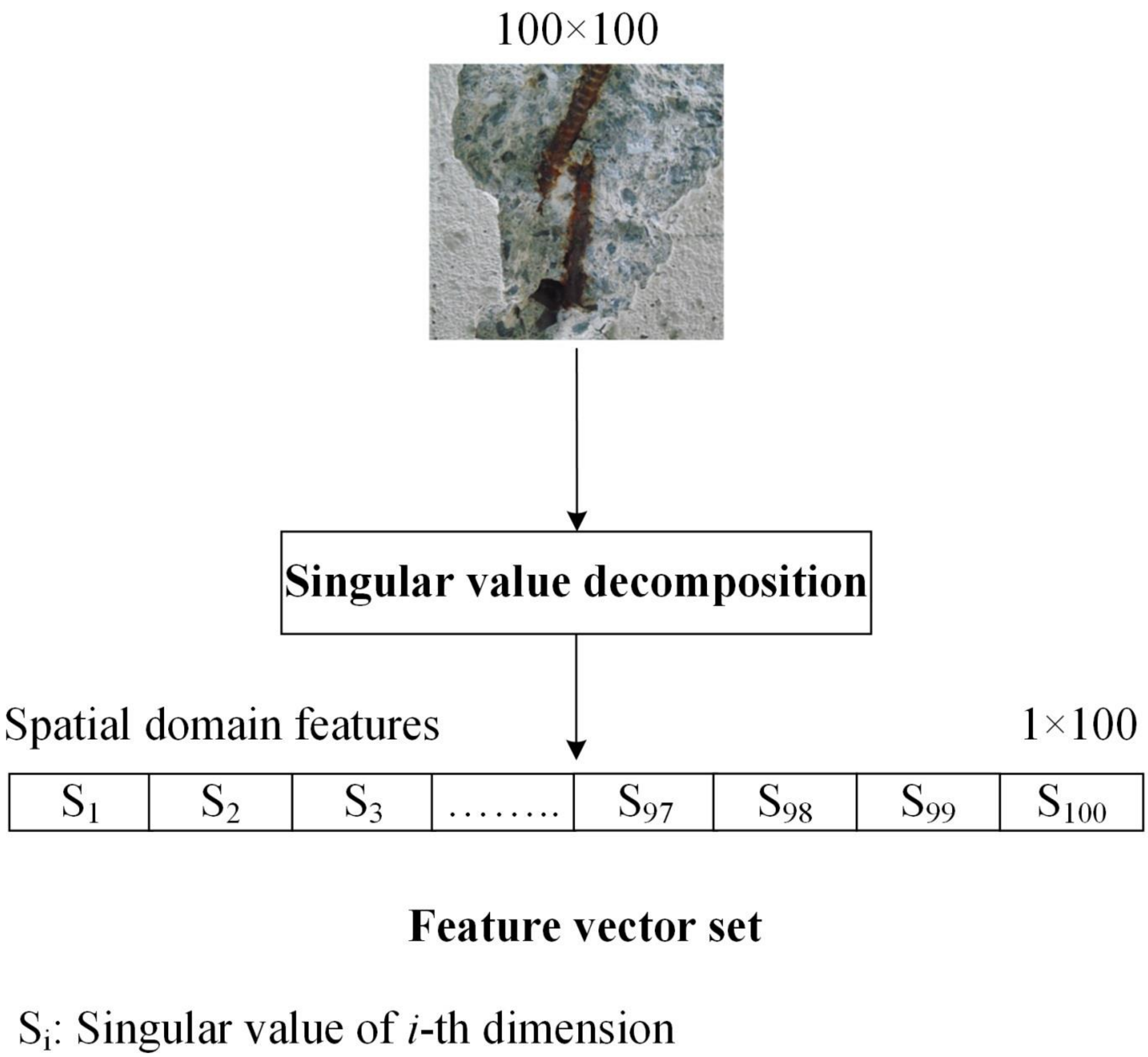


Figure 2: Extracted spatial domain features using singular value decomposition

4.2 Elman Recurrent Neural Network

Elman neural network is one of the types of recurrent neural network that was introduced by scientist Jeffrey Locke Elman in 1990 [81]. It is delineated by the presence of additional context layers, which aid in establishing a memory for the computations conducted so far in the network. The main distinct feature between the recurrent neural network and feedforward neural network is that the output at each time step relies on the previous inputs and computations through memorizing preceding events in the case of recurrent neural networks. However, the network outputs are independent of each other and they depend solely on the current time step in the case of feedforward neural networks. It is worth mentioning that the connections between neurons alongside the dependencies between layers establish a directed cycle that aids in maintaining a state between subsequent time steps [82-83].

Elman neural network comprises input layer, hidden layer, context layer, and output layer, such that number of neurons in the context layer equals to number of neurons in the hidden layer. The neurons in each layer are designated for the propagation of information from one layer to the subsequent layers. In the ENN, the connections of hidden layers entering the context layer are not weighted. Nevertheless, the connections of the context layer entering the hidden layer are weighted. Elman neural network is one of the recurrent neural networks because it is characterized by the presence of feedback loop. This in return is expected to induce a significant implication on improving the learning and recognition capability of the network. It should be mentioned that the feedback loop encompasses the utilization of unit-delay element (Z^{-1}), which provides non-linear dynamism to the behavior of Elman recurrent neural network.

The output of the hidden layer is being sent to both context and output layers. In this context, the output from the hidden layer is fed into the context to be appended in order to capitalize on this information in the subsequent interactions. Then, it is fed through the weighted connections back to the hidden layers in the succeeding steps. In this manner, the ENN is repeatedly remembering the hidden layer output of the previous iterations which enables the ENN to preserve its short term memory which can enhance the learning performances of ENN. It should be stated that the connections between the context layer output and hidden layer are weighted while the connections between the hidden layer output and context layer are un-weighted [84-85].

The hidden layer output and output layer output can be obtained using Equations 5 and 6, respectively.

$$X(k) = f(W_2 X_c(k) + W_1 U(k-1)) \tag{5}$$

$$Y(k) = g(W_3 X(k)) \tag{6}$$

Give that:

$$X_c(k) = X(k-1) \tag{7}$$

Where;

W_1denotes the weight of the input later to the hidden layer. W_2 indicates the weight of the context layer to the hidden layer. $X_c(k)$ stands for the output of the context layer. $X(k)$ denotes the output of the hidden layer. $U(k-1)$ stands for the input of the neural network. $Y(k)$ denotes the output of the neural network. f represents the transfer activation function at the hidden layer. g represents the transfer function at the output layer.

Gradient descent is the most commonly utilized algorithms to train the Elman neural network and back propagation neural networks. The networks are called “back propagation” because the error of the prediction or classification accuracy is computed at the output layer based on the targeted and predicted output values for each input instance (image). The calculated errors propagate backwards through the network layers all the way from the output to the hidden layers and then further to the input layer in an attempt to generate as much as possible close values to the desired output. Gradient descent algorithm capitalizes on computing the partial derivative of the error function in order to update the weights of the connection between different layers during each training epoch (k). In this regard, the modified weights during the training process as per the error partial derivative function can be obtained using Equation 8. In the present study, the optimum weights of connections are derived based on minimizing the error cost function of sum of squared error (SSE) between the predicted and desired classification values during each training epoch. The error cost function can be defined using Equation 9 [86].

$$W_{ij}(k+1) = W_{ij}(k) + \Delta W_{ij}((k) = W_{ij}(k) - \eta \times \frac{\partial E(k)}{\partial W_{ij}} \quad (8)$$

$$E = \sum_{t=1}^{N2} (P_t - O_t)^2 \quad (9)$$

Where;

$\Delta W_{ij}((k)$ denotes the adjustment or increment in the weights of connections between layers. $W_{ij}(k+1)$ and $W_{ij}(k)$ represent the updated and current weights, respectively. η depicts the learning rate. $\frac{\partial E(k)}{\partial W_{ij}}$ represents the error partial derivative with respect to the weights. E denotes the error cost function. P_t and O_t stand for the predicted and targeted values, respectively.

The architecture of the Elman neural network for defects recognition is depicted in Figure 3. The defects' recognition model is formulated as a multi-class classification problem, whereas the number of input neurons equal to the number of singular values obtained from the feature extraction module. Additionally, the number of output neurons is equal to the number of classes or defects. In this regard, one-hot encoding algorithm is applied for the purpose of more realistic classification of surface defects through the efficient modeling of categorical values. It is selected over the label encoding or integer encoding algorithm because in the label encoding, the higher the categorical value the more important the category is, which may lead to confusion and misinterpretation of the encoded variables by the machine learning model [87-88]. In the one hot encoding algorithm, the categorical values are encoded as binary vector. In this transformation, each class label or surface defect has its own register bits. As such, for M class labels, there will be M mutually exclusive binary vectors. In the binary vector, a class label is signified by the value one for the index of the integer while other elements take the value zero. For instance, cracking, spalling and scaling are of class labels 1, 2 and 3, respectively. In this regard, cracking, spalling and scaling are hot encoded using the binary vectors [1 0 0], [0 1 0] and [0 0 1], respectively. The model output is the probability an instance or a sample belongs to each particular class. In this context, the image is assigned to the class associated with the highest probability [89-90].

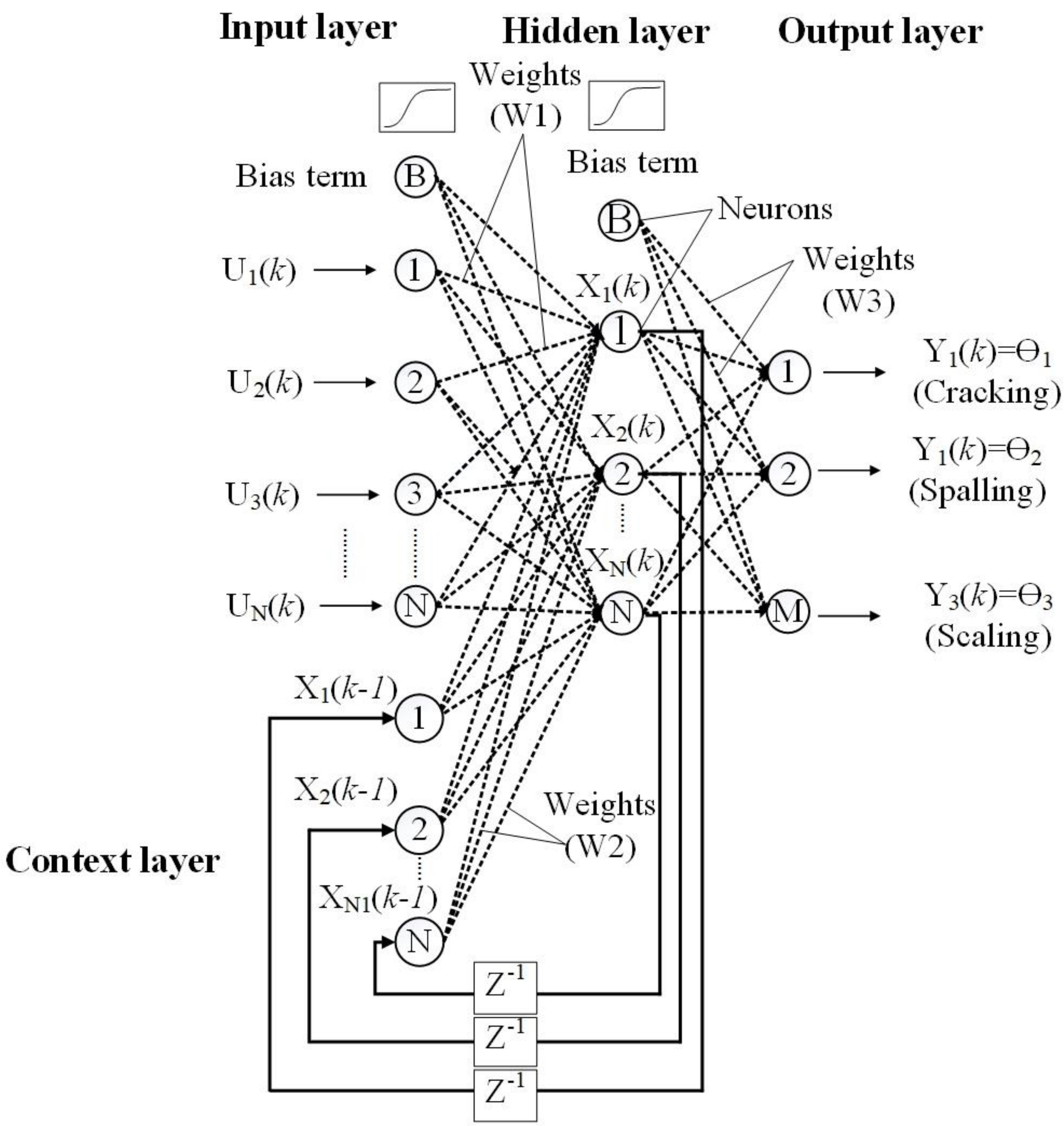


Figure 3: Architecture of the Elman recurrent neural network for bridge defects recognition

4.3 Autonomous Training of Elman neural network

As stated earlier, the ultimate objective of the present study is to develop an automated method for bridge defects detection and recognition. The bridge defects detection is formulated as a binary classification problem to classify images based on the existence of defects present in images. The output of this model is whether the images contain defects or not. The bride defects recognition is articulated in the form of three-point classification problem. Its output is whether the image contains cracking, spalling or scaling. A crack is a linear fracture, which extends partly or completely through the concrete member because of the tensile stresses. The tensile stresses are primarily carried out by the steel reinforcement and concrete. When the tensile stresses surpass the structural capacity of the concrete, the concrete starts to crack and the tensile stresses are transformed completely to the steel reinforcement. Spalling is a problematic surface defect, which is induced in the form of a fragment of concrete detached from a larger concrete. It can cause serious structural damage and sometimes it can significantly contribute to the structural collapse of the concrete structure. Scaling is a surface deterioration mechanism which can be defined as flaking or peeling of finished hardened concrete surface more often due to the exposure to cycles of freezing and thawing and the utilization of de-icer chemicals. When concrete pores near the surface thaws and freezes as a result of the temperature fluctuations. It affects the functional performance of the structural element because it influences the riding quality and safety of traffic.

In the present study, the invasive weed optimization algorithm is utilized instead of the gradient descent algorithm to train the neural network for the following two reasons: inferior accuracy of the gradient descent and manual tuning of hyper parameters. The learning of gradient descent neural network is vulnerable to slow convergence, local minima entrapment and over-fitting issues as described earlier. In the same context, there are a wide range of hyper parameters, which significantly affect the learning performance of the neural network. These parameters are highly sensitive to their initial values, whereas their initial setting is always variable from one case to the other. For instance, there is no exact method to systematically identify the number of hidden neurons and layers, whereas most of the equations present in the literature are case dependent and cannot be generalized. So, if the number of hidden neurons is less than the optimum number, then the accuracy will be so much affected. Furthermore, if the number of hidden neurons is more than optimum number, this will consume long processing and training time. In this regard, the blindness in the determination of the parameters of Elman neural network may result in the network to be trapped in an inferior solution. Additionally, this may cause lengthy computational time of the training process and slow convergence. Thus, a self-adaptive method is formulated for the sake of autonomous and dynamic tuning the input parameters based on the present dataset.

Invasive weed optimization algorithm is employed to train the Elman neural network by optimizing both the weights and structure of the ENN simultaneously in an attempt to amplify its learning capacity. The structural training includes both the topological structure and the transfer functions of the ENN model. Eight types of transfer activation are analyzed namely, log-sigmoid transfer function, hyperbolic tangent sigmoid transfer function, Elliot symmetric sigmoid transfer function, positive linear transfer function, radial basis transfer function, triangular basis transfer function, linear transfer function and normalized radial basis transfer function. The parameter learning encompasses optimizing both the values of weights and bias terms. The structural and parameter training is conducted based on minimizing the single objective function of misclassification error of the total instances during each training epoch as follows.

$$\mathrm{MC\_ERR} = \min \frac{\mathrm{FAL\_CLASS}}{\mathrm{TOT\_ISNT}} \tag{10}$$

Where;

MC_ERR denotes the misclassification error. FAL_CLASS indicates number of falsely classified instances. TOT_ISNT represents total number of instances in the training dataset. It should be highlighted that misclassification error is preferred over other performance indicators since it is a well-known good performing performance indicator, unitless, and un-biased performance metric. Furthermore, it is usually more practical and efficient to deal with error cost functions in machine learning.

Optimality theory is primarily capitalized on the fixed-length assumption, whereas most of the optimization algorithms encompass a fixed length vector of decision variables to represent a particular solution in the design space. Nevertheless, some few reported cases in the literature which encompass variable-length optimization problems. In it, the number of decision variables is changing iteratively over the training epochs. It should be mentioned that the variable-length optimization problems are of more complex nature and they require more computational time and resources during the training process when compared against the fixed-length optimization problems. There is no clear definition for the gradient vector of the variable-length problem in the variable-length optimization problems. Hence, gradient-based methods are inefficient in dealing with such type of problems. One of the approaches to deal with the variable-length optimization models is to assume a fixed length for the decision variables and to tune iteratively the decision variable that causes variability in length. Nevertheless, this approach often leads to suboptimal solutions. Additionally, it is inefficient and impractical method especially in the presence of wide ranges of decision variables. This necessitates the formulation of a new approach which enables the estimation of the varying length of vector of candidate solutions in each iteration [91].

In the present study, a self-adaptive optimization method is designed to handle the variability in the length of the optimization problem because the length of the optimization problem changes iteratively based on the number of hidden layers, number of context layers and number of hidden neurons. In order to be able to address the problem in hand, the variable length of the vector of solutions has to be known during the optimization process using a predefined function, i.e., the total number of connection weights has to be known during the training process. The total number of weights and bias terms can be computed using Equation 11. As shown in Equation 11, the optimization model gives the user the flexibility to design a multi-hidden layer neural network and a multi-context layer neural network based on the input dataset of images.

$$\mathrm{Num} = \big((\mathrm{I}+1)\times \mathrm{N}\big) + ((\mathrm{N}\times \mathrm{C}\times \mathrm{P}) + \big((\mathrm{N}+1)\times \mathrm{N}\times (\mathrm{P}-1)\big) + \big((\mathrm{N}+1)\times \mathrm{O}\big) \tag{11}$$

Where;

Num stands for the total number of weights and bias terms. I represents the number of input neurons. N indicates the number of hidden neurons. C represents the number of neurons in the context layer. P represents number of hidden and context layers. O depicts the number of output neurons. In this regard, the number of context layers is assumed to be equal to the number of hidden layers for simplification purposes.

4.4 Invasive Weed Optimization

Invasive weed optimization is a meta-heuristic bio-inspired search algorithm that was proposed by Mehrabian and Lucas in 2006. IWO emulates the natural and invasive behavior of weeds in colonizing and exploring the search space in an attempt to find the most optimum place for growth and reproduction. Weeds are aggressive, fast and robust plants which grown spontaneously and they exhibit a harmful serious effect on cultivated crops. IWO is primarily based on four core phases that will be discussed in the following lines [92-94].

The first phase is to generate an initial population of weeds that are spread in the i-dimensional search space. The position of each weed represents a solution. Then, the fitness of each weed within the population is calculated based on a designated predefined objective function. The second step is the reproduction, whereas in this phase each weed in the population is allowed to produce seeds based on its own fitness function value, maximum and minimum fitness values within the population in addition to the maximum and minimum number of seeds. It is worth mentioning that the higher the fitness of the weed, the more seeds it is allowed to produce. The reproduction process of the seeds is depicted in Figure 4. The number of seeds which are allowed to be produced by a particular weed can be computed as follows.

$$\mathrm{Seed_i} = \frac{\mathrm{f_i} - \mathrm{f_{min}}}{\mathrm{f_{max - f_{min}}}} \times (\mathrm{s_{max}} - \mathrm{s_{min}}) + \mathrm{s_{min}} \quad (12)$$

Where;

$\mathrm{Seed_i}$ denotes number of seeds associated with the $\mathrm{i - th}$ weed. $\mathrm{f_i}$ depicts the current fitness of the weed. $\mathrm{f_{max}}$, and $\mathrm{f_{min}}$ represent the maximum and minimum fitness in the current population, respectively. $\mathrm{s_{max}}$, and $\mathrm{s_{min}}$ denote the maximum and minimum allowable number of seeds, respectively.

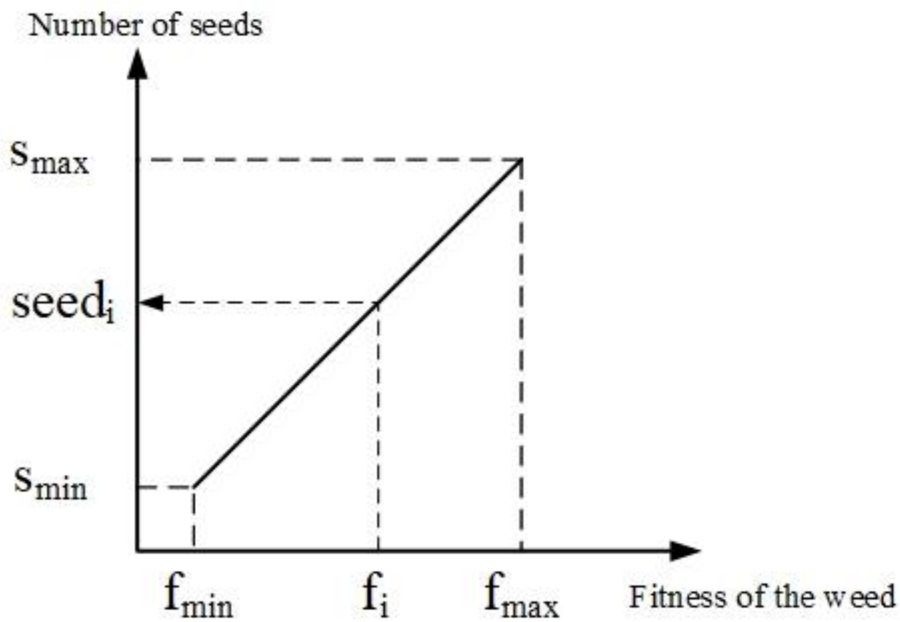


Figure 4: Reproduction process based on the weed's fitness

The third phase is the spatial dispersion, whereas the seeds are randomly scattered in the solution space based on a normal distribution of a mean equal to zero and an adaptive varying standard deviation. This phase guarantees that the seeds are accumulated around the weed plant, which leads to a local search around each parent weed. The standard deviation of the seed dispersion starts from a predefined initial maximum value and it is reduced to a final predefined minimum value using non-linear function. In this regard, the probability of observing a seed existing far from the weed plant is high at the beginning of the search process and it decreases within the iterations. The standard deviation of a particular iteration can be computed as follows.

$$\sigma_i = \sigma_{min} + \left(\frac{iter_{max} - iter}{iter_{max} - iter_{min}}\right)^p \times (\sigma_{max} - \sigma_{min}) \quad (13)$$

Where;

σ_i stands for the standard deviation of the current iteration. σ_{max}, and σ_{min} describe the initial and final standard deviation of the optimization process, respectively. $iter_{max}$ denotes the maximum number of iterations. p represents non-linear modulation index that is a number between two and three. It should be noted that the high standard deviation at the beginning is essential to allow better exploration of the solution space. It decreases by increasing the number of iterations to facilitate the exploitation of design search space.

The final phase is the competitive exclusion, whereas it is performed because the number of weeds and seeds reaches the maximum population size due to the fast and exponential reproduction. In this stage, the parent weeds in addition the seeds are sorted based on the fitness value for the purpose of eliminating the solutions with the least fitness values to keep the number of the weed plants and seeds within the maximum allowable population size. The seeds and their parent weeds with higher fitness are allowed to survive, and become reproductive. The process continues until the convergence criteria are met (reaching the maximum number of iterations).

5. TRANSFER LEARNING-BASED NEURAL NETWORKS

Pre-trained networks are deep learning architectures that were previously trained with very large datasets such as ImageNet and Places365. In this regard, transfer learning is fine-tuning a pre-trained deep network on a target dataset. There are two main ways of transfer learning namely, fine-tuning and feature extraction. In the first way, the weights of the pre-trained network serve base line or initial values of the learning process. In this regard, part of the network uses directly the pre-trained weights while the other is trained from scratch. Normally, the first layers of the network preserve their weights as the features of these layers are usually generic and can be applied elsewhere. On the other hand, the late layers provide more specific features and they need to be adjusted for the sake of establishing more adaptive model to the training dataset.

In the second way, the pre-trained filters are utilized as feature extractor in the workflow. It is carried out by removing the last output layer of deep convolutional neural network and dealing with the output of the second last layer as the extracted features (usually a high dimensional vector). In this regard, the pre-trained filters are applied to extract features from the new dataset, such that the obtained encoded features can be analyzed by any other classification model. It is worth mentioning the feature extraction approach is preferred over the fine-tuning if the new research dataset is different from the original training dataset of the pre-trained deep network in terms of size and content [95-96]. A set of well-known deep convolutional neural network architectures such as AlexNet and VGG19 are discussed in the following lines. More information about VGG16 and caffeNet can be found in Simonyan and Zisserman [97], and Jia et al. [98].

5.1 AlexNet

AlexNet architecture was developed by krizhevsky et al. [99] which won the ImageNet Large Scale Visual Recognition Challenge in 2012. The architecture is composed of five convolutional layers and three fully connected layers such that, the input size of the image is 224×224×3. In the first convolutional layer, 96 kernel filters of size 11×11×3 and a stride of 4 pixels are applied to generate a stack of 96 feature maps. The second convolutional layer receives the output of the first convolutional layer and filters it with 256 kernels of size 5×5×48. The third, fourth and fifth convolutional layers utilize kernels of size 3×3 and these layers are connected without intervening normalization or pooling layers. Each of the fully connected layers encompasses 4096 neurons.

5.2 VGG19

VGG19 is a deeper pre-trained architecture than VGG16. It is composed of 19 trainable layers with learnable weights: 16 convolutional layers and 3 fully connected layers. Its input is images of size 224×224 and three channels. The number of filter kernels starts with 64 and increases until reaching 512. In this regard, the convolutional layers utilize small kernels of size 3×3 with 1 pixel for stride and padding. The convolution operations are followed by rectified linear unit to add the dimension of non-linearity to the network structure. The max pooling layers uses kernels of size 2×2 and stride of 2 pixels for downsizing the feature map. The stack of convolutional layers is followed by three fully-connected layers, whereas the first two layers encompass 4096 nodes while the last one involves 1000 nodes. The last fully-connected layer is followed by softmax layer with the same number of outputs to provide the probabilities of a certain input instance belongs to each of the 1000 classes of the ImageNet [97]. It is worth mentioning that VGG19 is a larger network than VGG16 and more computationally expensive to train.

6. PERFORMANCE EVALUATION

There are several performance measures for evaluating the performance of classifiers. No single measure can provide an accurate insight of the classifier performance. Hence, a set of performance metrics are utilized to assess the recognition capacities of different classifiers. The confusion matrix includes information about the predicted and actual classifications. Based on the information retrieved, the performance of different classifiers can be evaluated. The micro-averaging of different performance metrics is utilized to evaluate the average performance of classifiers over different classes except for the area under curve, which is computed using macro-averaging. One of the main objectives of the present study is to establish a comprehensive evaluation of the bridge defects detection (binary classification) model and bridge defects recognition (multi-class classification) model. In order to fulfill this objective, a set of performance evaluation metrics are utilized for the sake of analyzing the capabilities of the classification model. In this regard, a set of well-known performance measures reported for their efficiency in judging the success of predictive models, are employed in the present study. This involves the indicators of overall accuracy and F-measure [100-101]. The present study encompasses imbalanced dataset which may cause the afore-mentioned measures to be misleading, overoptimistic and inducing highly-inflated results. A dataset is called imbalanced if it contains different number of instances in each class. This may lead to misinterpretation of the evaluation of classifiers. In this regard, Kappa coefficient, balanced accuracy, Matthews's correlation coefficient and area under curve are utilized as more appropriate measures to better reflect the performances of the classifiers [102-103].

The overall accuracy, F-measure, Kappa coefficient, balanced accuracy, Matthews's correlation coefficient and area under curve for multi-class classification can be computed using Equations 14, 15, 16, 17, 18 and 19, respectively [104-108]. Overall accuracy is the most commonly-utilized classification metric, whereas it can be defined as the ratio between the correctly classified instances to the total number of instances. Sensitivity is sometimes called "recall" or "true positive rate", whereas it is used to evaluate the capability of the classifier to map the positive instances. Sensitivity is expressed as the number of positive correctly classified instances to the total number of positive instances. Specificity is sometimes called "inverse recall" or "true negative rate", whereas it is used to measure the capacity of the classifier to identify the negative instances. Specificity is defined as the number of the negative correctly classified instances to the total number of negative instances.

Precision is used to measure the agreement of instances with positive labels, whereas it is expressed as the number positive correctly classified instances to the total number of predicted instances. F-measure is the harmonic mean of the precision and recall. It offers trade-off between the two metrics because in some cases one of the metrics dominates the other. Kappa coefficient is utilized to measure the agreement between the predicted instances and the actual instances. Kappa coefficient is a robust index because it considers the probability that an instance is classified by chance. It measures the fraction of correctly classified instances after the omission of probability of by chance agreement.

Balanced accuracy is the average of the sensitivity and specificity. In the case of highly imbalanced data, accuracy can be a misleading performance metric. Thus, the balanced accuracy can solve the bias in the imbalanced data, and thus can stand as a better performance metric to compare the different classifiers. Matthews's correlation coefficient is a correlation coefficient that measures the similarities between the observed and predicted classification, Moreover, sometimes it stands as a better-balanced performance evaluation than averaged percentages because it considers all fields of the confusion matrix. Area under curve is derived from receiver operating characteristic (ROC) curve. The ROC curves are used to visually compare the performance of the classifiers. The ROC curve is a 2-Dimensional curve such that the true positive rate (sensitivity) is plotted on the vertical axis while the false positive rate (1-specificity) is plotted on the horizontal axis. ROC curve, which is close to the diagonal implies random guessing. ROC curve, which is close to the top left corner, implies perfect performance of the classifier.

$$\text{Overall Accuracy} = \frac{\sum_{i=1}^{R} X_{ii}}{\sum_{i=1}^{R}\sum_{i=1}^{R} X_{ii}} \tag{14}$$

$$F - \text{measure} = \frac{2 \times \text{sensitivity} \times \text{precision}}{\text{sensitivity} + \text{precision}} \tag{15}$$

$$\text{Kappa coefficient} = \frac{\text{over\_agre} - \exp\ \text{\_agre}}{1 - \exp\ \text{\_agre}}$$

$$= \frac{[N \times \sum_{i=1}^{R} X_{ii}] - [\sum_{i=1}^{R} X_{i+} \times X_{+i}]}{N^2 - [\sum_{i=1}^{R} X_{i+} \times X_{+i}]} \tag{16}$$

$$\text{Balanced accuracy} = \frac{\text{Sensitivity} + \text{Specificity}}{2} \tag{17}$$

$$\text{MCC} = \frac{(TP \times TN) - (FP \times FN)}{\sqrt{(TP + FP) \times (TP + FN) \times (TN + FP) \times (TN + FN)}} \tag{18}$$

$$\text{AUC} = \int_a^b f(x) \tag{19}$$

Such that;

$$\text{Precision} = \frac{\sum_{i=1}^{C} TP}{\sum_{i=1}^{C} TP + \sum_{i=1}^{C} FP} \quad (20)$$

$$\text{Sensitivity (recall)} = \frac{\sum_{i=1}^{C} TP}{\sum_{i=1}^{C} TP + \sum_{i=1}^{C} FN} \quad (21)$$

$$\text{Specificity} = \frac{\sum_{i=1}^{C} TN}{\sum_{i=1}^{C} TN + \sum_{i=1}^{C} FP} \quad (22)$$

Where;

C indicates number of classes. TP, TN, FP and FN denote the number of true positive instances, true negative instances, false positive instances and false negative instances, respectively. TP, TN, FP and FN represent the instances correctly identified, correctly rejected, incorrectly identified and incorrectly rejected, respectively. $over\_agre$ and $exp\ \_agre$ indicate the overall agreement and expected agreement, respectively. N denotes the total number of instances. R represents number of rows or columns (classes). X_{ii} represents the element of $i-th$ row and $i-th$ column in the confusion matrix (diagonal elements). X_{i+} and X_{+i} indicate the total number of instances in the $i-th$ column and total number of instances in the $i-th$ row, respectively. MCC holds values between -1 and +1, whereas +1 indicates perfect agreement, 0 indicates similarity to random prediction and -1 indicates total disagreement between the observed and predicted instances. AUC holds values between 0 and +1 such that +1 implies perfect diagnostic test. $f(x)$ represents the ROC curve. The lower and upper bounds of the ROC curve are denoted by a and b, respectively. The higher the Overall Accuracy, $F-\text{measure}$, Kappa coefficient, Balanced accuracy, MCC and AUC, the better the classifier performance is.

7. METHOD IMPLEMENTATION

The proposed method is validated using dataset I and dataset II to test its robustness towards different types and natures of images.

7.1 Dataset I

7.1.1 Dataset description

The developed $SVD-ENN-IWO$ method is utilized for the defects detection and recognition. In this regard, two datasets are generated from dataset I for the sake of defects detection and recognition. For the defects detection model, a dataset comprising of 265 real-world images are used as an input to experiment the proposed method such that 200 images are used for training while the remaining 65 images are used for testing. For the defects recognition, the data set is composed of 264 images, whereas 215 and 49 images are utilized for training and testing the model, respectively. These images were captured from three bridge decks in Montreal and Laval, Canada using Sony DSC-H300 digital camera of 20.1 megapixel resolution. All the calculations and optimization algorithms took place on a laptop with an Intel Core i7 CPU, 2.2 GHz and 16 GB of memory. The images are resized to 100×100 to speed up the computation process. Sample of the distress images is shown in Figure 5. The images were captured in different weather conditions for the purpose of establishing automated detection and recognition models that are invariant to the lighting conditions.

(a) Spalling- Image “A”

(b) Spalling- Image “B”

(c) Scaling- Image “C”

(d) Cracking- Image “D”

Figure 5: Sample of the defects images of dataset I

7.1.2 Baseline implementation

In the defects detection model, the images are labeled to defected and non-defected ones. In this regard, the dataset of bridge defects detection comprises 239 and 26 of defected and non-defected images. With respect to the conventional machine learning models and deep learning models, images of size 100×100 are utilized as an input to feed the prediction models. The developed method is compared against a set of conventional machine learning models and different deep learning architectures for its validation. Each model is governed by a set of adjustable parameters that were determined based on iterative trial and error procedure. For the defects detection, linear discriminant analysis (LDA) is utilized to perform the discriminant analysis. The number of nearest neighbors in the KNN model is assumed three. In the decision tree, the minimum numbers of branch node observations and leaf node observations are set to 10 and 1, respectively with 10 maximum category levels. With respect to the SVM model, Gaussian radial basis function is utilized with scaling factor and sigma of one. The number of bags is assumed 20 in the random forest model. For the ANN model, the number of hidden layers and hidden neurons are assumed two. The learning rate and momentum coefficient are assumed 0.001 and 0.8, respectively. Furthermore, log-sigmoid transfer function is utilized. In the Elman neural network, the number of hidden layers and context layers are set to three while the number of hidden neurons and context neurons are assumed two.

With regard to deep learning models, the architecture of CONVNET is composed of two convolutional layers whereas the first one utilizes 16 filters of size 3×3 and padding of 1 pixel. The second uses 32 filters of size 3×3 and padding of 1 pixel. The first convolutional layer is followed by a max pooling layer of window size 2×2 and stride of 2 pixels. Each convolutional layer is followed by a rectified linear unit activation function to establish more effective training through mapping the negative values to zero and maintaining the positive values. Finally, one fully connected layer of two neurons followed by a softmax layer are used for binary classification purpose. CONVNET model alongside the set of pre-trained models are trained using the stochastic gradient descent algorithm. The learning rate and MiniBatchSize are assumed 1×10^{-6} and 10, respectively. The number of epochs is assumed 500 for the different defects detection models. Training performance of AlexNet for bridge defects detection is reported in Table 1. It reports the elapsed time of training and the generated classification accuracy.

Table 1: Sample of training performance of AlexNet for bridge defects detection

Epoch	Iteration	Time Elapsed (seconds)	Mini-batch loss	Mini-batch accuracy	Base learning rate
1	1	0.1	6.4643	60%	1.00e-06
50	1000	30.49	9.5654	40%	1.00e-06
100	2000	61.56	6.377	60%	1.00e-06
420	8400	273.48	3.1885	80%	1.00e-06
500	10000	331.27	3.1885	80%	1.00e-06

For the defects recognition model, the images were either classified to cracking or spalling or scaling. The dataset used for defects recognition consists of 20, 174 and 70 images of cracking, spalling and scaling. This model utilizes the same setting of hyper parameters utilized in the defects detection model. The architecture of random forest for bridge defects recognition is presented in Figure 6. The resulting tree map demonstrates how the random forest can be utilized to classify images as per the type of defects. For instance, if x(5660) is less than 145.5 and x(4856) is more than 187. Thus, there cracking defect is the most dominant defect in the present image. It is worth mentioning that the input feature vector used to train the conventional machine learning model is of size 1×10000. The dimensions herein denote the indices of the image pixel. Table 2 reports sample of training course of CONVNET model designated for bridge defects recognition.

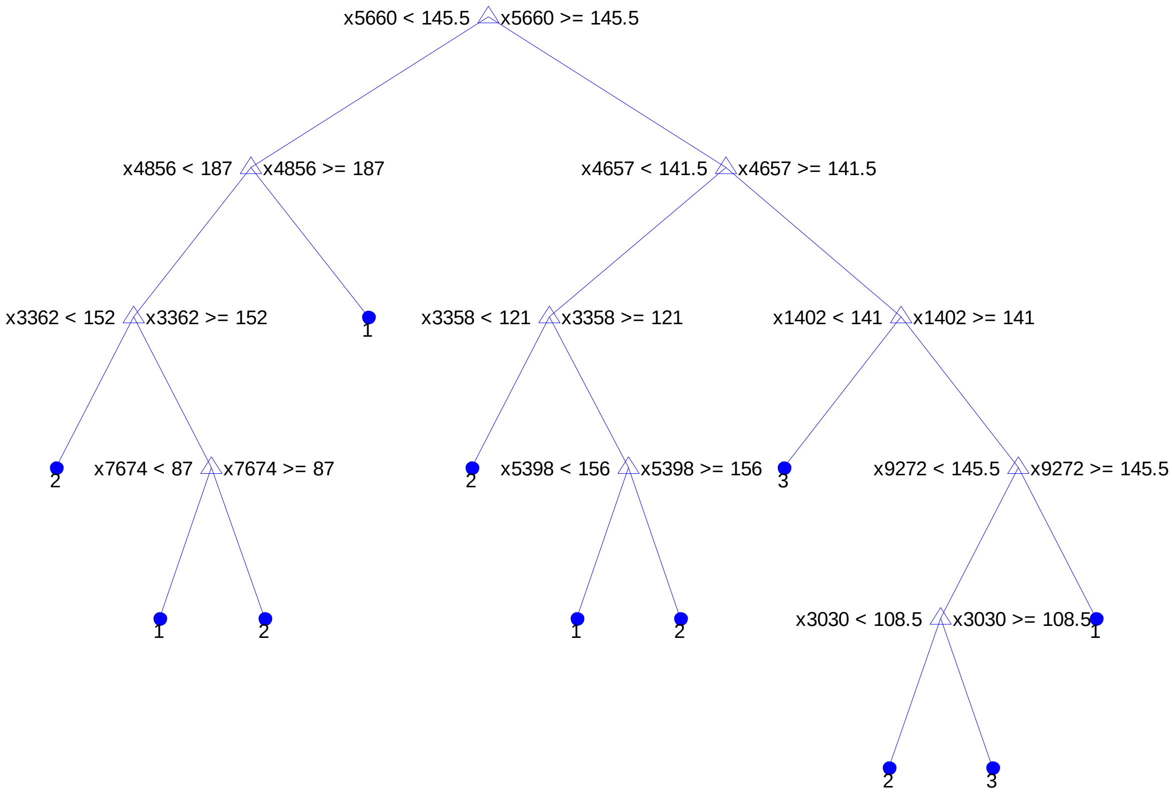


Figure 6: knowledge-based random forest for bridge defects recognition

Table 2: Sample of training performance of CONVNET for bridge defects recognition based on dataset I

Epoch	Iteration	Time Elapsed (seconds)	Mini-batch loss	Mini-batch accuracy	Base learning rate
1	1	0.09	1.0913	40%	1.00e-06
50	1050	12.08	0.9806	70%	1.00e-06
100	2100	24.4	0.7539	70%	1.00e-06
400	8400	96.84	0.3859	80%	1.00e-06
500	10500	131.60	0.3328	100%	1.00e-06

7.1.3 Proposed model implementation

The developed $\mathrm{SVD}-\mathrm{ENN}-\mathrm{IWO}$ method utilizes singular value decomposition for the purpose of dimensionality reduction and extracting the most significant features in images. In the singular value decomposition, an input image of size 100×100 can be reduced to a feature vector of size 1×100. For instance, the distribution of singular values of image 5.a is depicted in Figure 7. In this regard, the first few diagonal elements contain the most considerable amount of information while the tail end of the feature vector incorporates lesser information. It can be inferred that the first 50 dimensions are able to preserve substantial amount of information (approximately 95% of the total inform present in the image). The interface of the feature extraction model in the computerized platform is shown in Figure 8. By clicking “View” button, the singular values vector for the different images are displayed and By clicking “Plot”, the distribution of singular values are plotted.

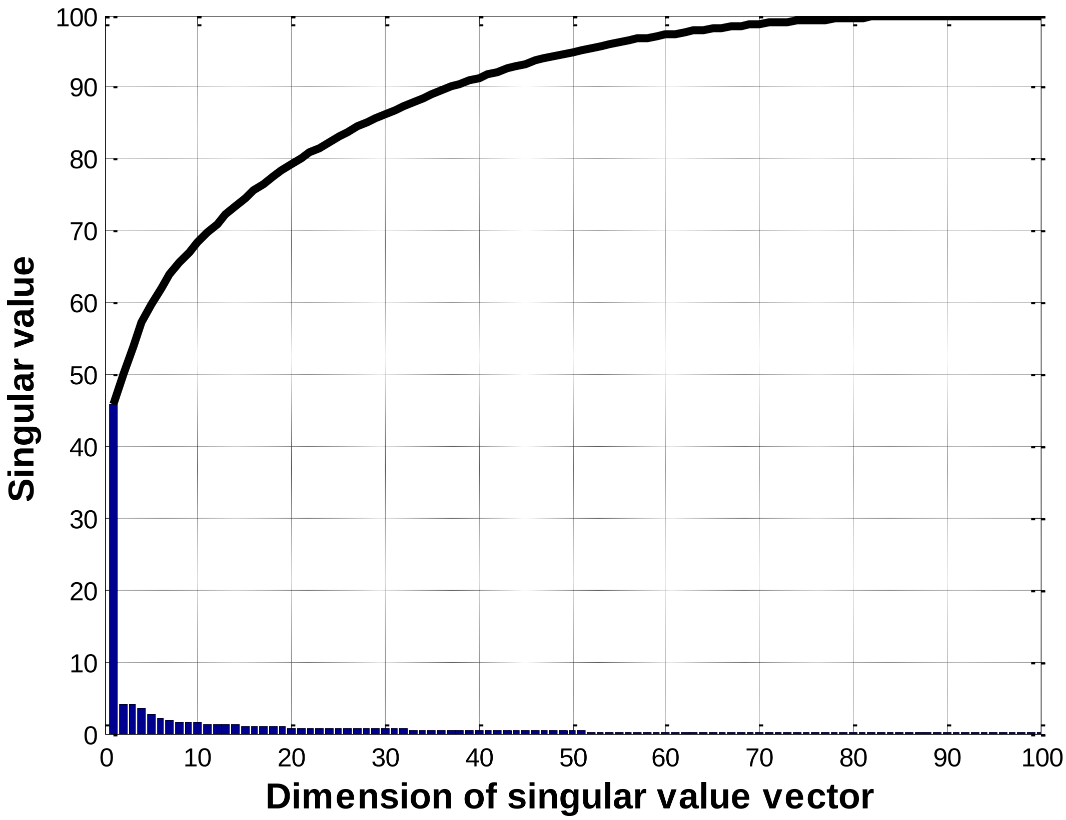


Figure 7: Distribution of the singular values for image 4.a

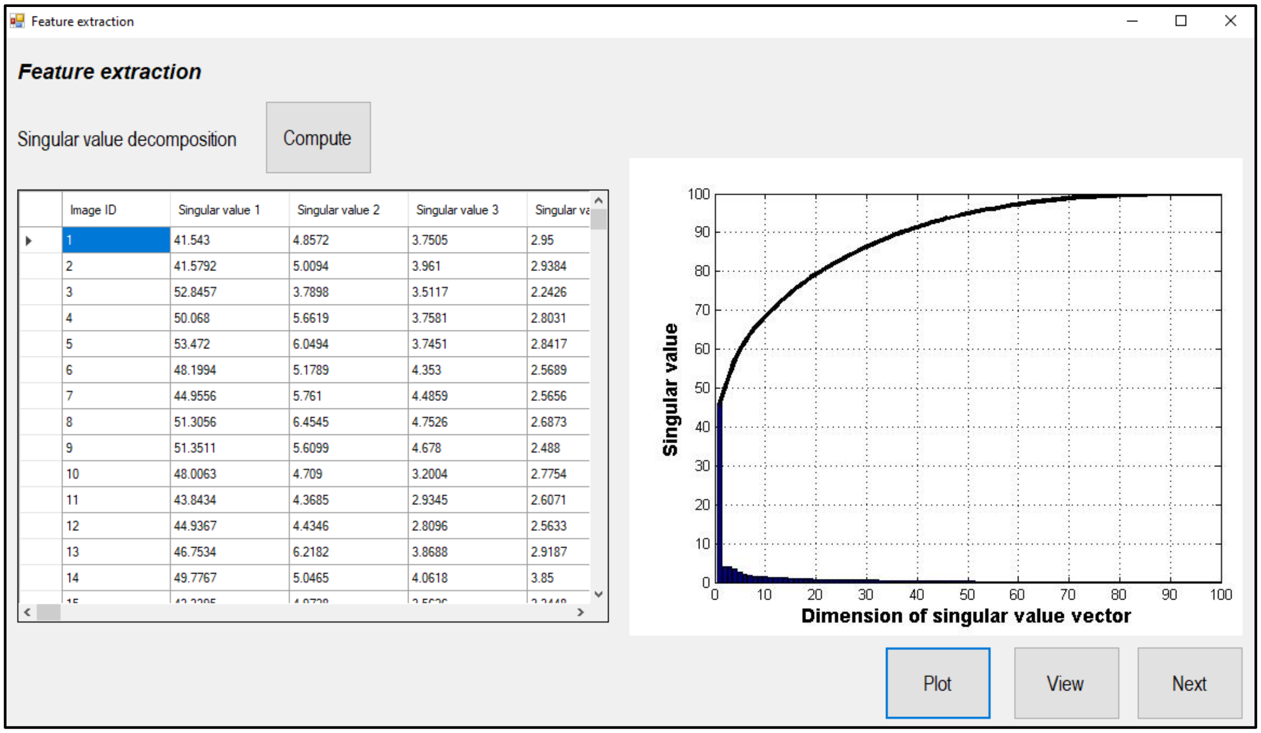


Figure 8: Interface of the developed feature extraction model

In the SVD – ENN – IWO model, the feature vector of 100 singular values is used as an input to establish its training. Since the performance of the Elman neural network is significantly influenced by its parameters such as number of hidden layers, number of context layers, number of hidden neurons, number of context neurons, type of transfer functions, moment value, bias terms and weights of the connections between neurons. The present study relies on the IWO to establish a proper setting for the tuning parameters of the Elman neural network. The optimization parameters of the developed SVD – ENN – IWO model for defects detection are listed in Table 3. The maximum number of hidden and context layers are 5. Also, the maximum number of hidden and context neurons are five. Eight transfer functions are investigated and the values of weights and bias terms are real numbers between -1 and 1. Therefore, the maximum length of the optimization problem is 759. The parameters of the IWO algorithm are as follows: the number of iterations and the initial population size are assumed 500 and 250, respectively. The maximum and minimum numbers of seeds are 5 and 0, respectively. The initial and final standard deviations are assumed 0.5 and 0.001, respectively. The convergence of the developed SVD – ENN – IWO model for defects detection is shown in Figure 9. The least misclassification error achieved by SVD – ENN – IWO model equals to zero. Moreover, the optimization model stabilizes at iteration 120 which demonstrates the superior search capability of the IWO algorithm. The optimum structure of the ENN is as follows: the optimum numbers of hidden and context layers are one while the optimum number of hidden and context neurons are one. The optimum transfer function is the Elliot symmetric sigmoid transfer function.

Table 3: The optimization parameters of the Elman neural network and their corresponding ranges for defects detection

Decision Variable	Range
Number of hidden and context neurons	$1 \leq$ integer number ≤ 5
Number of hidden layers and context layers	$1 \leq$ integer number ≤ 5
Type of transfer function	$1 \leq$ integer number ≤ 8
Weights and bias terms	$-1 \leq$ real number ≤ 1

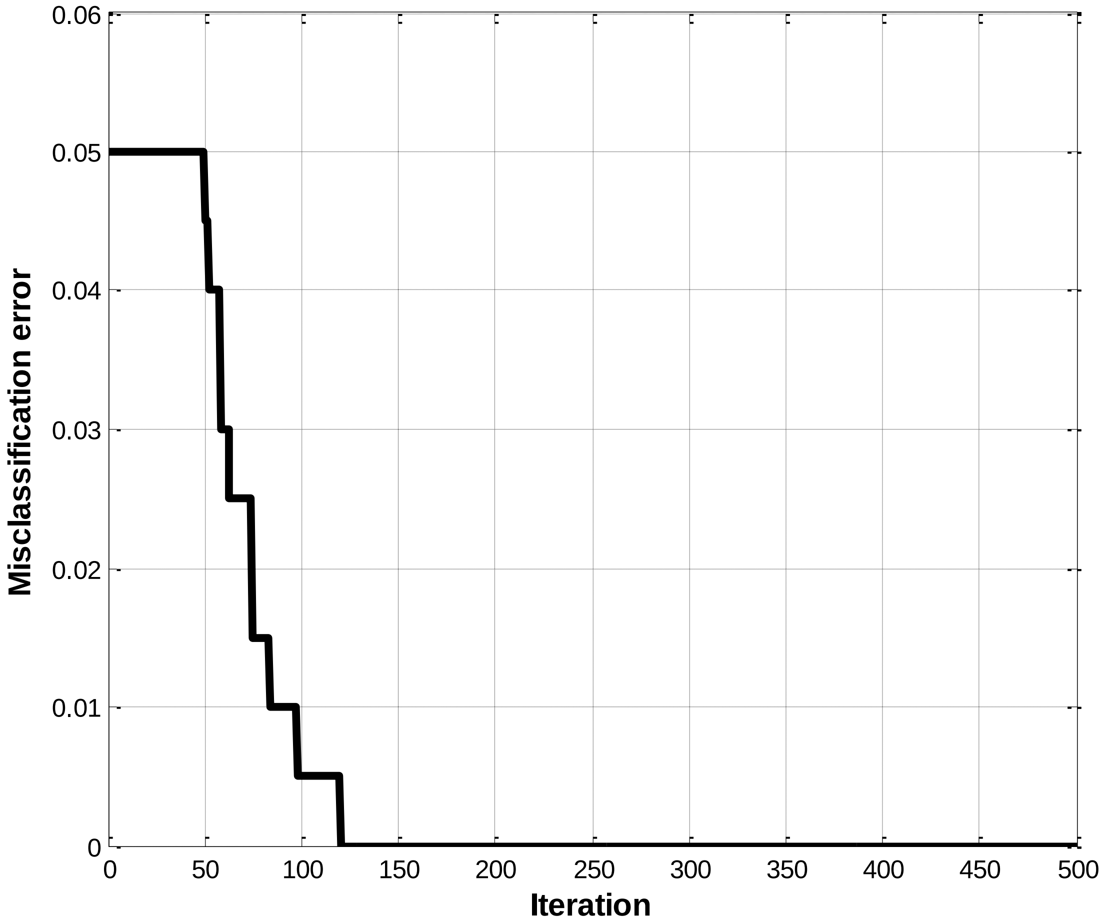


Figure 9: Convergence of the SVD − ENN − IWO model for defects detection based on dataset I

The interface of the developed SVD − ENN − IWO model for defects recognition is depicted in Figure 10. In the computerized platform, the user is asked to identify the ranges of the optimization hyper parameters of Elman neural network alongside the parameters of invasive weed optimization algorithm. As can be seen, the upper bounds of number of hidden layers, context layers, hidden neurons and context neurons are assumed 15. Other setting of optimization hyper parameters and parameters of IWO algorithm are the same as the ones used in the bridge defects model. The Elman neural network of this model is composed of three output neurons as a result of the investigation of three defects. The output of this model is obtained by pressing the “View” button. This encompasses the maximum length of the variable-length optimization model, minimum mean absolute percentage error, and optimum hyper parameters of Elman neural network. In this regard, the maximum length of the optimization problem is 8301, which constitutes a large search space that entails deploying an exhaustive exploration mechanism to deal with the wide-ranging and variability of the search space. The optimum topology of the ENN is as follows: optimum numbers of hidden and context layers are one while the optimum number of hidden and context neurons are eight. The optimum transfer function is the hyperbolic tangent sigmoid transfer function. The convergence of developed SVD − ENN − IWO model for defects recognition is depicted in Figure 11. The minimum misclassification error achieved by the SVD − ENN − IWO model equals to 0.0372. Furthermore, the IWO algorithm stabilizes at iteration 292, which illustrates the capability of the IWO algorithm in exploring the search space.

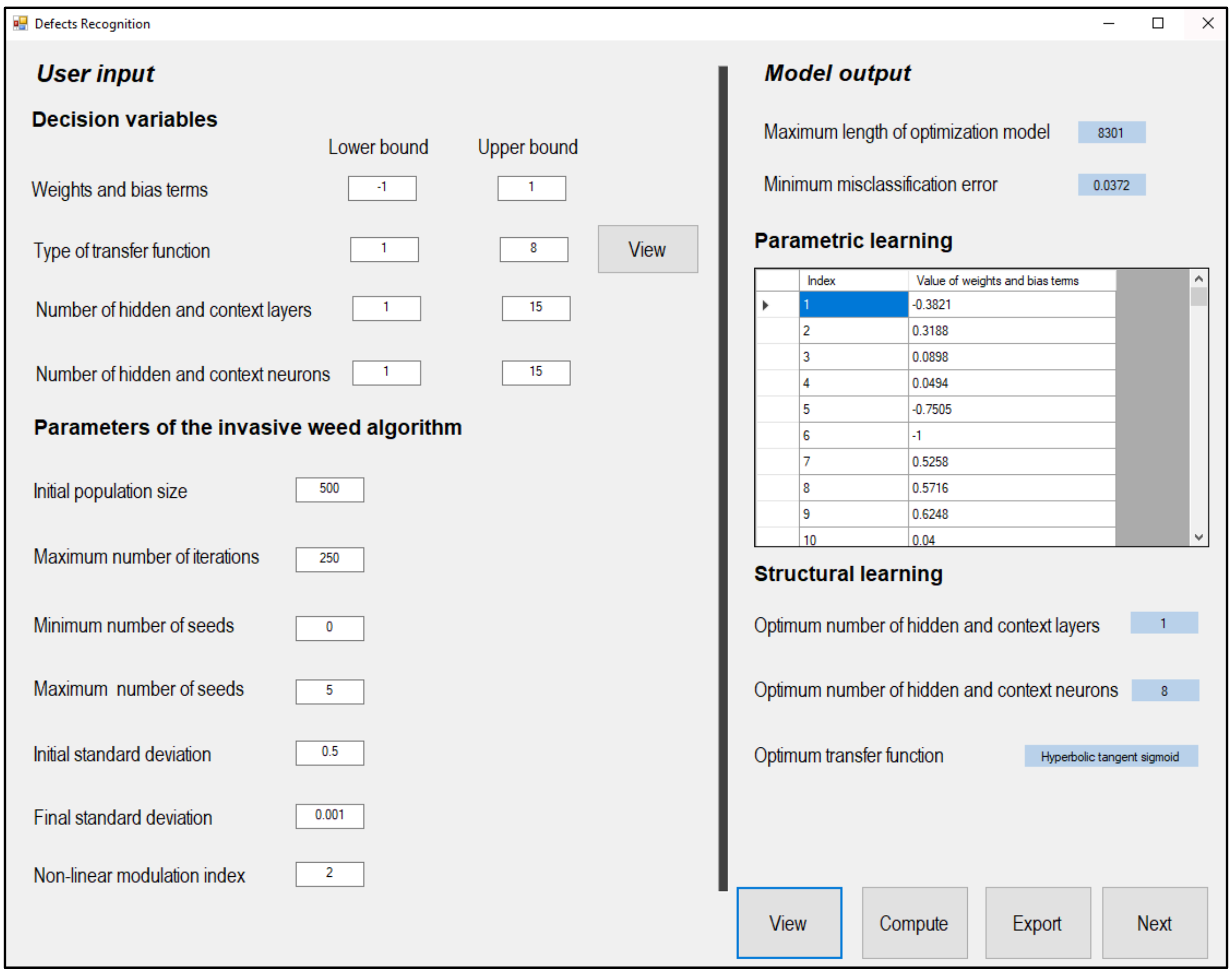


Figure 10: Interface of the developed SVD − ENN − IWO model for bridge defects recognition

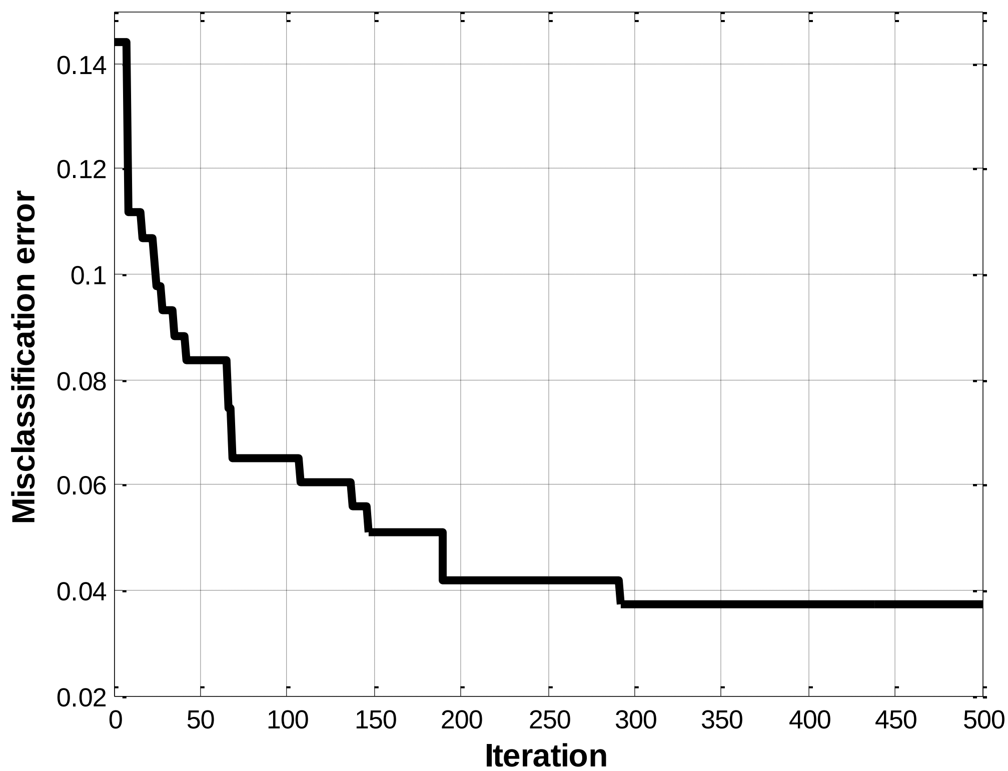


Figure 11: Convergence of the SVD − ENN − IWO model for defects recognition

7.1.4 Analysis and Discussion

The confusion matrices of the thirteen prediction models are reported in Table 4. An image with defects was classified as true positive if it was detected by the prediction model that it contains defects. An image with no defects was classified as true negative if no defects were detected by the prediction model. If a prediction model failed to detect defects in a defected image, it would be classified as false negative. If an image with no defects, was detected as defected image by the prediction model, this is considered as false positive. Evaluation of them demonstrates that the SVD − ENN − IWO managed to achieve the highest correctly classified defected images. Additionally, it provided acceptable accuracy in detecting the non-defected images. Conversely, it can be interpreted that KNN could hardly detect the defected images while managing to detect the non-defected images efficiently. KNN, AlexNet and caffeNet achieved the highest correctly classified non-defected images. In this context, the number of correctly classified defected images and non-defected images attained by the developed model are 52 and 9, respectively. It can be also inferred that the CONVNET model was able to detect the defects better than the pre-trained deep neural networks. This can be explained by the existence of versatile lighting condition images in the defects detection dataset. It is expected that such conditions created negative transfer due to the absence of sufficient similarity between the source domain and target domain, which in return undermined the learning performance of the deep neural networks.

Table 4: Confusion matrices of the prediction models for bridge defects detection based on testing dataset I

Prediction model	Predicted class	Actual class	
		No defect	Defect
Discriminant analysis	**No defect**	37	1
	Defect	17	10
K-nearest neighbors	**No defect**	31	0
	Defect	23	11
Random Forest	**No defect**	47	4
	Defect	7	7
Support vector machines	**No defect**	43	2
	Defect	11	9
Decision tree	**No defect**	50	9
	Defect	4	2
Artificial neural network	**No defect**	45	7
	Defect	9	4
Elman neural network	**No defect**	47	5
	Defect	7	6
CONVNET	**No defect**	51	3
	Defect	3	8
AlexNet	**No defect**	27	27
	Defect	0	11
VGG16	**No defect**	29	0
	Defect	25	11
VGG19	**No defect**	32	1
	Defect	22	10
CaffeNet	**No defect**	23	0
	Defect	31	11
SVD − ENN − IWO	**No defect**	52	2
	Defect	2	9

The performance accuracies of the prediction models for the testing dataset based on split validation and for the entire dataset based on 10-fold cross validation are recorded in Table 5 and Table 6, respectively. It is worthy to note that developed model outperformed other prediction models based on split validation and 10-fold cross validation. With regards to split validation, it provided overall accuracy, F-measure, Kappa coefficient, balanced accuracy, Matthews's correlation coefficient and area under curve of 0.939, 0.963, 0.781, 0.891, 0.781 and 0.891, respectively. As per the split validation, decision tree provided the lowest prediction accuracies achieving overall accuracy, F-measure, Kappa coefficient, balanced accuracy, Matthews's correlation coefficient and area under curve of 0.8, 0.885, 0.132, 0.554, 0.149 and 0.554, respectively. With regards to cross validation, artificial neural network yielded the lowest prediction outcome. In the context, the developed model managed to establish an average improvement in the performance indicators by 24.85% when compared against the ENN model. The prediction accuracies of the cross validation and testing accuracies are close to each other which evinces that the developed method doesn't suffer from over fitting. The receiver operating characteristics curves for the different prediction models are depicted in Figure 12. The ROC curves are utilized to visually compare the performance of the prediction models. A larger area under ROC curve indicates a better performance of the prediction model. In this context, the ROC curve of the $\mathrm{SVD-ENN-IWO}$ lies above other classifiers. This implies that AUC for $\mathrm{SVD-ENN-IWO}$ is larger than other classifiers. Thus, the developed model provides better classification performance than other models. This evinces the significant enhancement in the classification accuracy achieved by the implementation of the developed model.

Table 5: Performance evaluation of the prediction models for defects detection based on testing dataset I

Prediction model	overall accuracy	F-measure	Kappa coefficient	Balanced accuracy	Matthews correlation coefficient	Area under curve
Discriminant analysis	0.723	0.804	0.376	0.797	0.452	0.797
K-nearest neighbors	0.646	0.729	0.313	0.787	0.431	0.787
Random Forest	0.831	0.895	0.457	0.753	0.462	0.753
Support vector machines	0.8	0.869	0.463	0.807	0.499	0.807
Decision tree	0.8	0.885	0.132	0.554	0.149	0.554
Artificial neural network	0.754	0.849	0.184	0.599	0.185	0.599
Elman neural network	0.815	0.887	0.388	0.709	0.39	0.709
CONVNET	0.907	0.944	0.672	0.836	0.672	0.836
AlexNet	0.585	0.667	0.253	0.645	0.38	0.645
VGG16	0.615	0.698	0.282	0.769	0.405	0.769
VGG19	0.646	0.736	0.285	0.751	0.376	0.751
CaffeNet	0.523	0.597	0.201	0.713	0.334	0.713
SVD − ENN − IWO	0.939	0.963	0.781	0.891	0.781	0.891

Table 6: Performance evaluation of the prediction models for defects detection based on entire dataset I using 10-fold cross validation

Prediction model	overall accuracy	F-measure	Kappa coefficient	Balanced accuracy	Matthews correlation coefficient	Area under curve
Discriminant analysis	0.918	0.947	0.685	0.931	0.712	0.931
K-nearest neighbors	0.801	0.872	0.418	0.884	0.512	0.884
Random Forest	0.934	0.957	0.688	0.837	0.689	0.837
Support vector machines	0.934	0.959	0.747	0.926	0.758	0.926
Decision tree	0.893	0.936	0.425	0.695	0.427	0.695
Artificial neural network	0.859	0.917	0.236	0.608	0.237	0.608
Elman neural network	0.908	0.944	0.588	0.824	0.593	0.824
CONVNET	0.957	0.974	0.811	0.886	0.812	0.89
AlexNet	0.761	0.845	0.357	0.863	0.464	0.863
VGG16	0.758	0.842	0.352	0.861	0.461	0.861
VGG19	0.596	0.712	0.182	0.583	0.341	0.583
CaffeNet	0.724	0.817	0.31	0.843	0.427	0.843
SVD − ENN − IWO	0.977	0.984	0.906	0.949	0.907	0.949

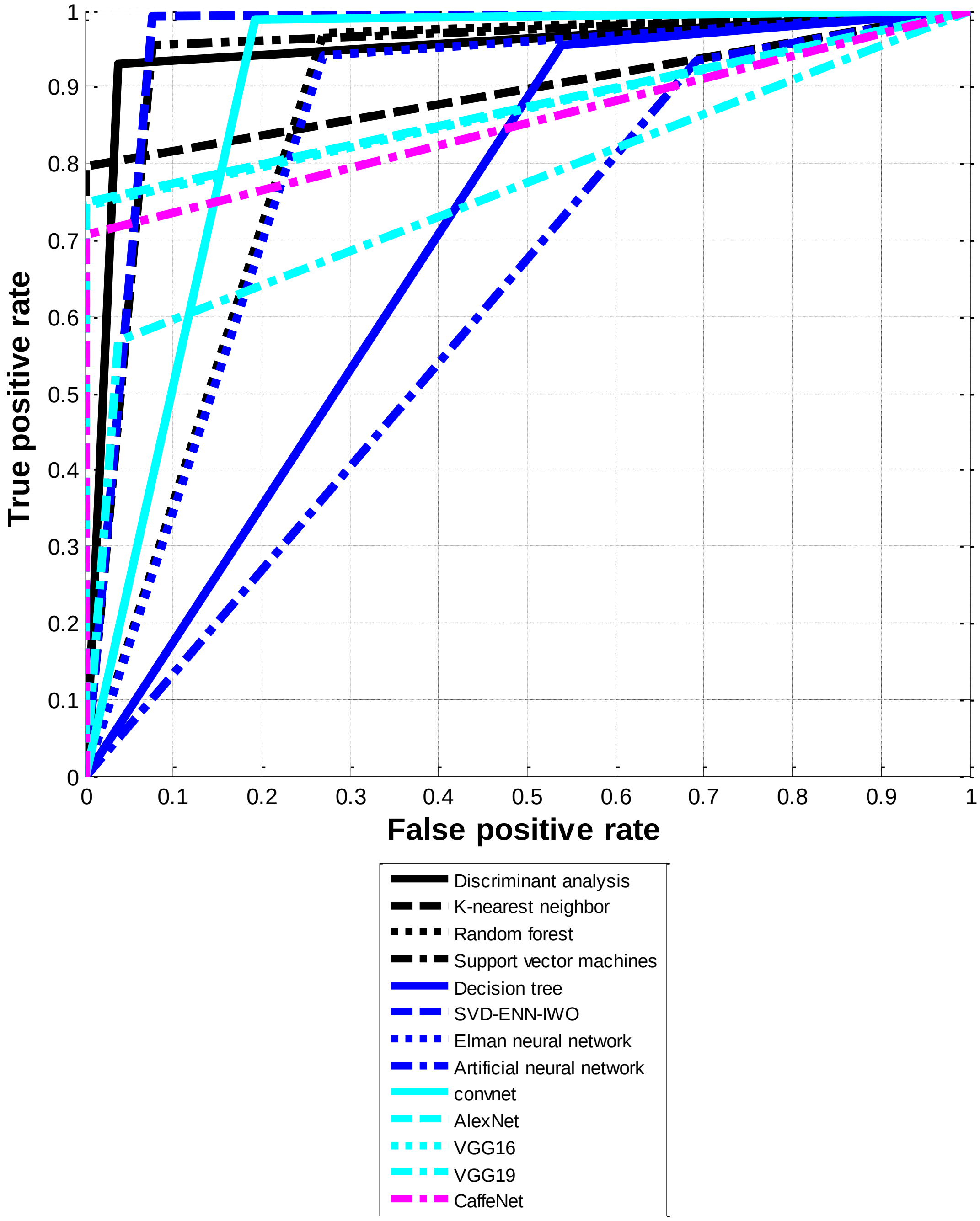


Figure 12: Receiver operating characteristics curves of the prediction models for defects detection based on entire dataset I

The confusion matrices of the prediction models for bridge defects recognition are reported in Table 7 and Table 8. It enables to analyze the classifications of data in depth. With respect to the Elman neural network, it misclassified six instances. Out of them, four cracking images were classified as spalling. In the support vector machines, the highest confusion was between spalling and scaling. It can be also inferred that the developed model reduced the confusion between cracking and spalling when compared against Elman neural network. Additionally, it substantially reduced the misclassified spalling and scaling. It is worth mentioning that the developed model misclassified two cracking images as spalling. In this context the developed model has the highest success rate in the detection of spalling and scaling images. It managed to accurately detect cracking, spalling and scaling with 3, 38 and 6, instances. Conversely, K-nearest neighbors had the lowest accuracy in the detection of spalling with 23 instances. It had one correctly classified cracking image. Additionally, it can be observed that it failed to detect scaling images.

The classification performances evaluations of the prediction models for bridge defects recognition based on split validation and cross validation are recorded in Table 9 and Table 10, respectively. With respect to the classification performance of testing dataset, it can be noted that the developed model provided the highest prediction accuracies achieving overall accuracy, F-measure, Kappa coefficient, balanced accuracy, Matthews's correlation coefficient and area under curve of 0.959, 0.959, 0.882, 0.969, 0.939 and 0.903, respectively. However, K-nearest neighbors yielded the lowest classification performance accomplishing overall accuracy, F-measure, Kappa coefficient, balanced accuracy, Matthews's correlation coefficient and area under curve of 0.531, 0.531, 0.033, 0.647, 0.293 and 0.534, respectively. A the level of cross validation, the developed $\mathrm{SVD}-\mathrm{ENN}-\mathrm{IWO}$ model outperformed other prediction models attaining overall accuracy, F-measure, Kappa coefficient, balanced accuracy, Matthews's correlation coefficient and area under curve of 0.955, 0.955, 0.914, 0.965, 0.937 and 0.904, respectively. It managed to improve the classification performance indicators by values ranging from 10.7% to 37.44% when compared against the artificial neural network. The ROC curves of the developed model and Elman neural network are presented in Figure 13 and Figure 14, respectively. They enable to establish a rigorous visual performance evaluation of the prediction model with respect to each class of defects. For the $\mathrm{SVD}-\mathrm{ENN}-\mathrm{IWO}$ model, the areas under curve of cracking, spalling and scaling are 0.775, 0.969 and 0.987, respectively. For the Elman neural network, the areas under curve of cracking, spalling and scaling are 0.861, 0.787 and 0.785, respectively. In this regard, the highest successful detection rate was in scaling followed by spalling and then cracking. In the Elman neural network, its highest success rate was in the detection of scaling followed by spalling and finally scaling. This demonstrates that the developed model managed to significantly enhance the detection accuracy of spalling and scaling.

Table 7: Confusion matrices of the prediction models for bridge defects recognition based on testing dataset

Prediction model	Predicted class	Actual class		
		Cracking	Spalling	Scaling
Discriminant analysis	Cracking	1	4	0
	Spalling	1	34	3
	Scaling	0	6	0
K-nearest neighbors	Cracking	3	15	1
	Spalling	2	23	5
	Scaling	0	0	0
Random Forest	Cracking	0	1	0
	Spalling	4	37	6
	Scaling	1	0	0
Support vector machines	Cracking	1	0	0
	Spalling	4	38	6
	Scaling	0	0	0
Decision tree	Cracking	0	0	0
	Spalling	5	38	6
	Scaling	0	0	0
Artificial neural network	Cracking	0	0	0
	Spalling	2	37	3
	Scaling	3	1	3
Elman neural network	Cracking	1	1	0
	Spalling	4	37	1
	Scaling	0	0	5

Table 8: Confusion matrices of the prediction models for bridge defects recognition based on testing dataset (Continued)

Prediction model	Predicted class	Actual class		
		Cracking	Spalling	Scaling
CONVNET	Cracking	0	0	0
	Spalling	2	38	0
	Scaling	3	0	6
AlexNet	Cracking	5	2	0
	Spalling	0	34	0
	Scaling	0	2	6
VGG16	Cracking	5	0	0
	Spalling	0	35	0
	Scaling	0	3	6
VGG19	Cracking	5	1	0
	Spalling	0	33	0
	Scaling	0	4	6
CaffeNet	Cracking	5	1	0
	Spalling	0	33	0
	Scaling	0	4	6
SVD − ENN − IWO	Cracking	3	0	0
	Spalling	2	38	0
	Scaling	0	0	6

Table 9: Performance evaluation of the prediction models for defects recognition based on testing dataset

Prediction model	overall accuracy	F-measure	Kappa coefficient	Balanced accuracy	Matthews correlation coefficient	Area under curve
Discriminant analysis	0.714	0.714	0.021	0.786	0.571	0.516
K-nearest neighbors	0.531	0.531	0.033	0.647	0.293	0.534
Random Forest	0.755	0.755	0.108	0.816	0.631	0.503
Support vector machines	0.795	0.795	0.143	0.847	0.694	0.548
Decision tree	0.776	0.776	0	0.832	0.663	0.5
Artificial neural network	0.816	0.816	0.427	0.861	0.722	0.654
Elman neural network	0.878	0.878	0.616	0.908	0.816	0.755
CONVNET	0.898	0.898	0.701	0.923	0.845	0.791
AlexNet	0.918	0.918	0.809	0.939	0.877	0.967
VGG16	0.939	0.939	0.852	0.954	0.908	0.975
VGG19	0.898	0.898	0.768	0.923	0.846	0.958
CaffeNet	0.898	0.898	0.768	0.923	0.846	0.958
SVD − ENN − IWO	0.959	0.959	0.882	0.969	0.939	0.903

Table 10: Performance evaluation of the prediction models for defects recognition based on entire dataset using 10-fold cross validation

Prediction model	overall accuracy	F-measure	Kappa coefficient	Balanced accuracy	Matthews correlation coefficient	Area under curve
Discriminant analysis	0.805	0.805	0.597	0.85	0.714	0.783
K-nearest neighbors	0.782	0.782	0.633	0.834	0.68	0.865
Random Forest	0.858	0.858	0.715	0.891	0.794	0.828
Support vector machines	0.892	0.892	0.791	0.917	0.846	0.854
Decision tree	0.896	0.896	0.79	0.919	0.851	0.845
Artificial neural network	0.833	0.833	0.665	0.872	0.754	0.756
Elman neural network	0.859	0.859	0.696	0.892	0.795	0.743
CONVNET	0.89	0.89	0.768	0.915	0.841	0.766
AlexNet	0.936	0.936	0.879	0.949	0.91	0.922
VGG16	0.944	0.944	0.895	0.956	0.922	0.929
VGG19	0.933	0.933	0.873	0.947	0.905	0.921
CaffeNet	0.941	0.941	0.889	0.953	0.917	0.909
SVD − ENN − IWO	0.955	0.955	0.914	0.965	0.937	0.904

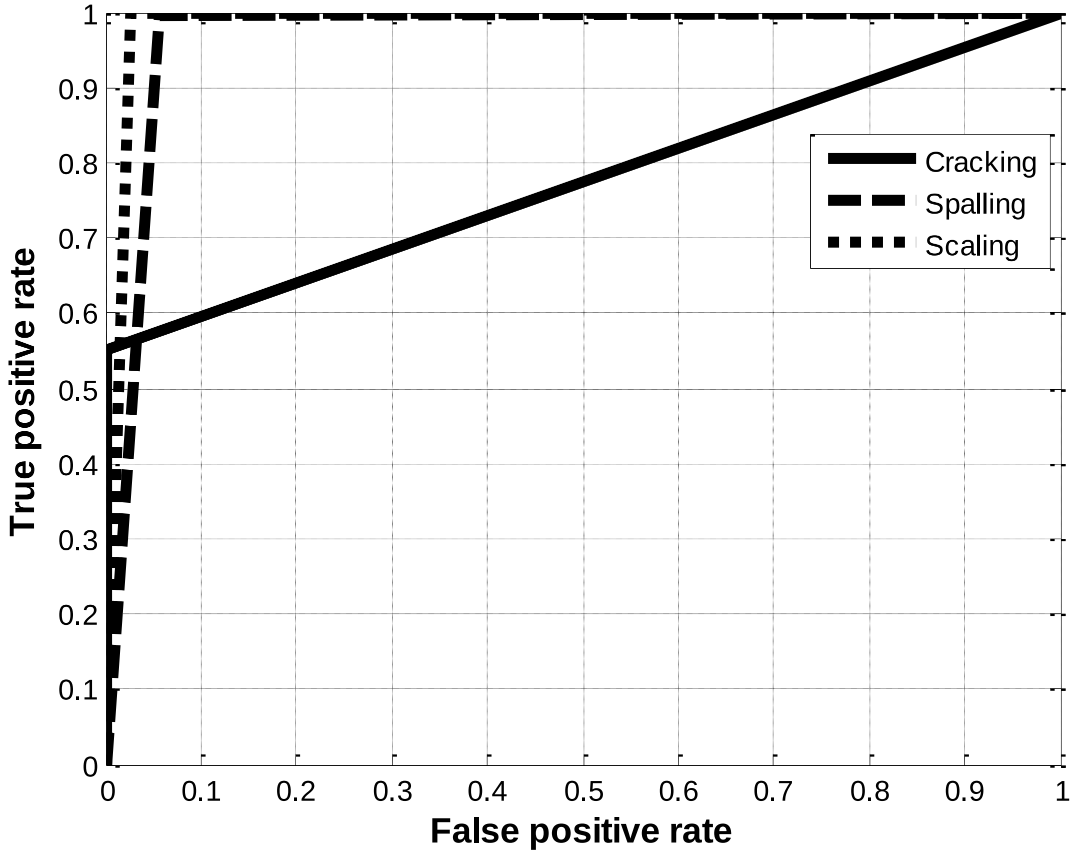


Figure 13: Receiver operating characteristics curves of the developed $\mathbf{SVD-ENN-IWO}$ model for bridge defects recognition

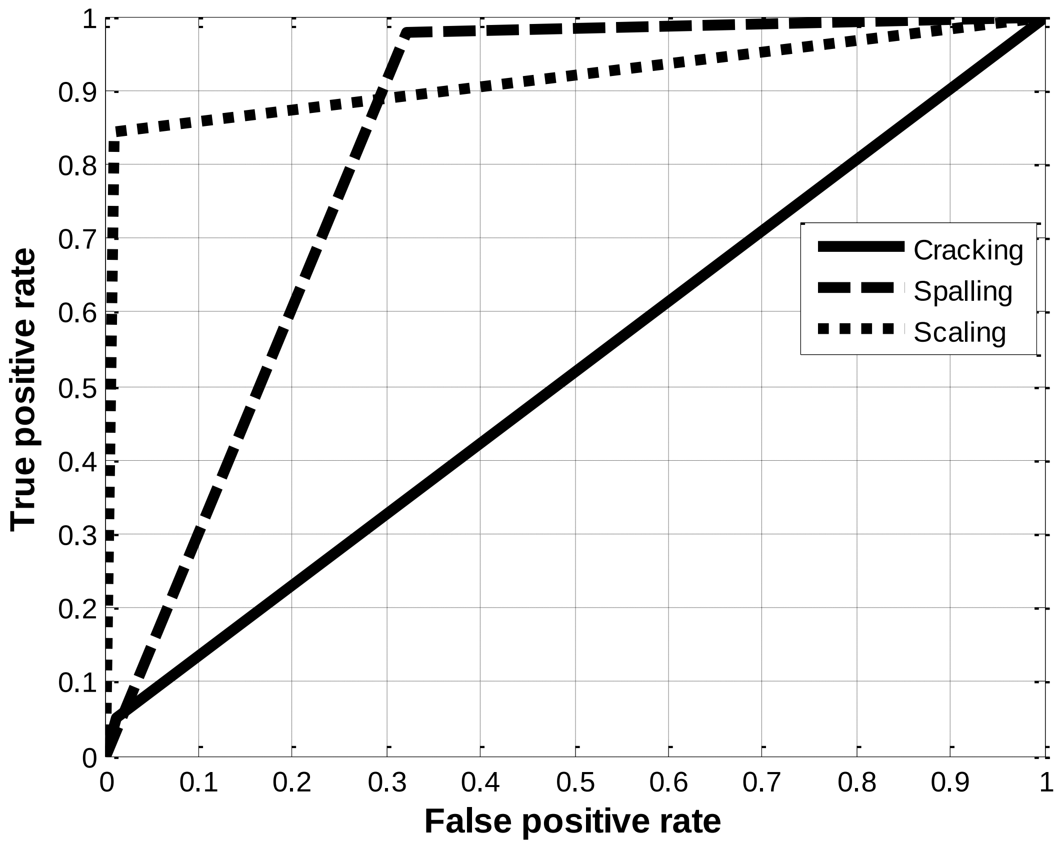


Figure 14: Receiver operating characteristics curves of Elman neural network for bridge defects recognition

7.2 Dataset II

7.2.1 Dataset description

A further analysis is conducted to study the performance of the developed model capitalizing on a larger dataset. The SDNET2018 dataset is composed of 56,000 annotated images of non-cracked and cracked images of pavement, walls and concrete bridge decks. In the present study, 2,921 images are utilized to train and test the prediction models. In this context, dataset II is composed of 306 cracked images and 2,615 non-cracked images. The present study utilizes 2,338 and 583 images for training and testing the prediction models, respectively.

7.2.2 Baseline implementation

This study utilizes the same previous architecture of CONVNET. The number of epochs is assumed 100 for CONVNET and all pre-trained networks. A sample of the training performance of VGG16 for bridge defects detection is reported in Table 11.

Table 11: Sample of training performance of VGG16 for bridge defects detection based on dataset II

Epoch	Iteration	Time Elapsed (seconds)	Mini-batch loss	Mini-batch accuracy	Base learning rate
1	1	0.01	8.8105	20%	1.00e-06
20	4450	139.16	0.9557	80%	1.00e-06
50	11450	366.62	0	100%	1.00e-06
80	18550	599.1	0	100%	1.00e-06
100	23300	761.72	0	100%	1.00e-06

7.2.3 Proposed model implementation

The developed $\mathrm{SVD}-\mathrm{ENN}-\mathrm{IWO}$ model utilizes a maximum number of hidden layers, context layers, hidden neurons and context neurons of five. The number of iterations and initial population size are assumed 100 and 20, respectively. In this context, the maximum length of the optimization model is 759. The convergence of the developed $\mathrm{SVD}-\mathrm{ENN}-\mathrm{IWO}$ model for bridge defects detection based on dataset II is displayed in Figure 15. As can be seen, the developed model is capable of accomplishing a low training misclassification error of 0.56%. The optimum topology encompasses two hidden layers, two context layers, one hidden neuron and one context neuron. Hyperbolic tangent sigmoid is the optimum transfer function.

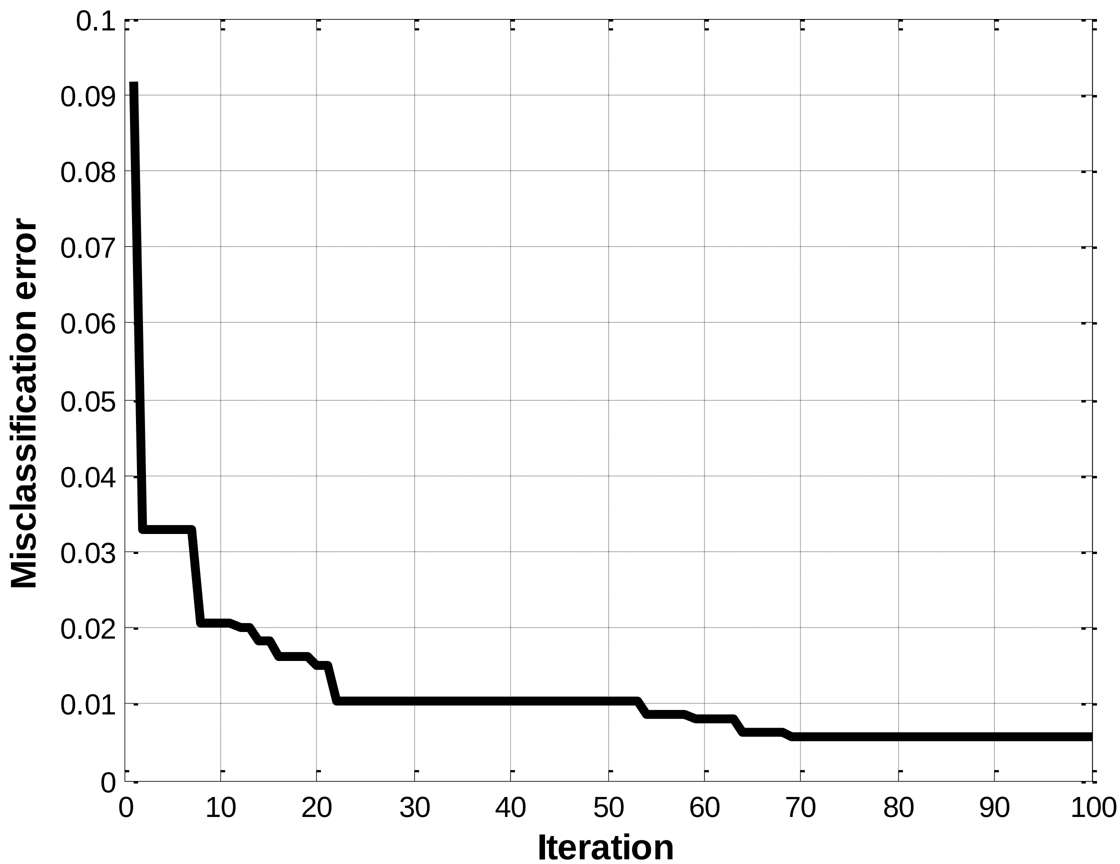


Figure 15: Convergence of the SVD − ENN − IWO model for defects detection based on dataset II

7.2.4 Analysis and Discussion

The confusion matrices of the prediction models for bridge defects detection based on testing dataset II are presented in Table 12. It can be deduced that SVD − ENN − IWO achieved the highest correctly classified cracked images with 39 instances. Additionally, it provided the second highest correctly classified non-cracked images with 528 instances after CONVNET. AlexNet provided the lowest correctly classified cracked images with 39 instances while CaffeNet yielded the lowest successful detection rate of non-cracked images with 413 instances. Table 13 and Table 14 report the performance evaluations of the prediction models for bridge defects detection based on testing dataset II using split validation and entire dataset II using 10-fold cross validation. With regards to split validation, it can be observed that the developed model generated the highest classification accuracies accomplishing overall accuracy, F-measure, Kappa coefficient, balanced accuracy, Matthews's correlation coefficient and area under curve of 0.997, 0.982, 0.98, 0.998, 0.99 and 0.998, respectively. CONVNET outperformed the pre-trained networks achieving overall accuracy, F-measure, Kappa coefficient, balanced accuracy, Matthews's correlation coefficient and area under curve of 0.976, 0.848, 0.835, 0.868, 0.847 and 0.868, respectively. In this regard, the developed model established an average enhancement in the classification indicators of 13.68% when compared against the trained from scratch network.

With respect to the cross validation, the developed model outperformed other prediction models such that it achieved overall accuracy, F-measure, Kappa coefficient, balanced accuracy, Matthews's correlation coefficient and area under curve of 0.995, 0.973, 0.969, 0.981, 0.97 and 0.981, respectively. VGG19 outperformed $\mathtt{CONVNET}$ and other pre-trained networks yielding overall accuracy, F-measure, Kappa coefficient, balanced accuracy, Matthews's correlation coefficient and area under curve of 0.954, 0.814, 0.788, 0.957, 0.8 and 0.957, respectively. In this context, the developed model generated an average improvement in the classification performance evaluation of 12.18% with reference to VGG16. The ROC curves of the prediction models for bridge defects detection are presented in Figure 16. As can be seen, the ROC curve of the developed model lies above other prediction models while the ROC curve of $\mathtt{CONVNET}$ lies beneath other prediction models. This implies that the developed model provides a higher area under curve with respect to other prediction models, which in return indicates that it provided higher detection accuracy.

Table 12: Confusion matrices of the prediction models for bridge defects detection based on testing dataset II

Prediction model	Predicted class	Actual class	
		No defect	Defect
$\mathbf{CONVNET}$	No defect	39	0
	Defect	14	530
AlexNet	No defect	38	115
	Defect	15	415
VGG16	No defect	42	115
	Defect	11	415
VGG19	No defect	45	116
	Defect	8	414
CaffeNet	No defect	41	117
	Defect	12	413
$\mathbf{SVD-ENN-IWO}$	No defect	53	2
	Defect	0	528

Table 13: Performance evaluation of the prediction models for defects detection based on testing dataset II

Prediction model	overall accuracy	F-measure	Kappa coefficient	Balanced accuracy	Matthews correlation coefficient	Area under curve
CONVNET	0.976	0.848	0.835	0.868	0.847	0.868
AlexNet	0.777	0.369	0.27	0.75	0.327	0.75
VGG16	0.784	0.4	0.306	0.788	0.373	0.788
VGG19	0.787	0.421	0.329	0.815	0.405	0.915
CaffeNet	0.779	0.389	0.292	0.776	0.358	0.776
SVD − ENN − IWO	0.997	0.982	0.98	0.998	0.99	0.998

Table 14: Performance evaluation of the prediction models for defects detection based on entire dataset II using 10-fold cross validation

Prediction model	overall accuracy	F-measure	Kappa coefficient	Balanced accuracy	Matthews correlation coefficient	Area under curve
CONVNET	0.924	0.422	0.397	0.642	0.485	0.642
AlexNet	0.954	0.811	0.786	0.951	0.796	0.951
VGG16	0.953	0.811	0.785	0.957	0.798	0.956
VGG19	0.954	0.814	0.788	0.957	0.8	0.957
CaffeNet	0.953	0.811	0.785	0.959	0.798	0.958
SVD − ENN − IWO	0.995	0.973	0.969	0.981	0.97	0.981

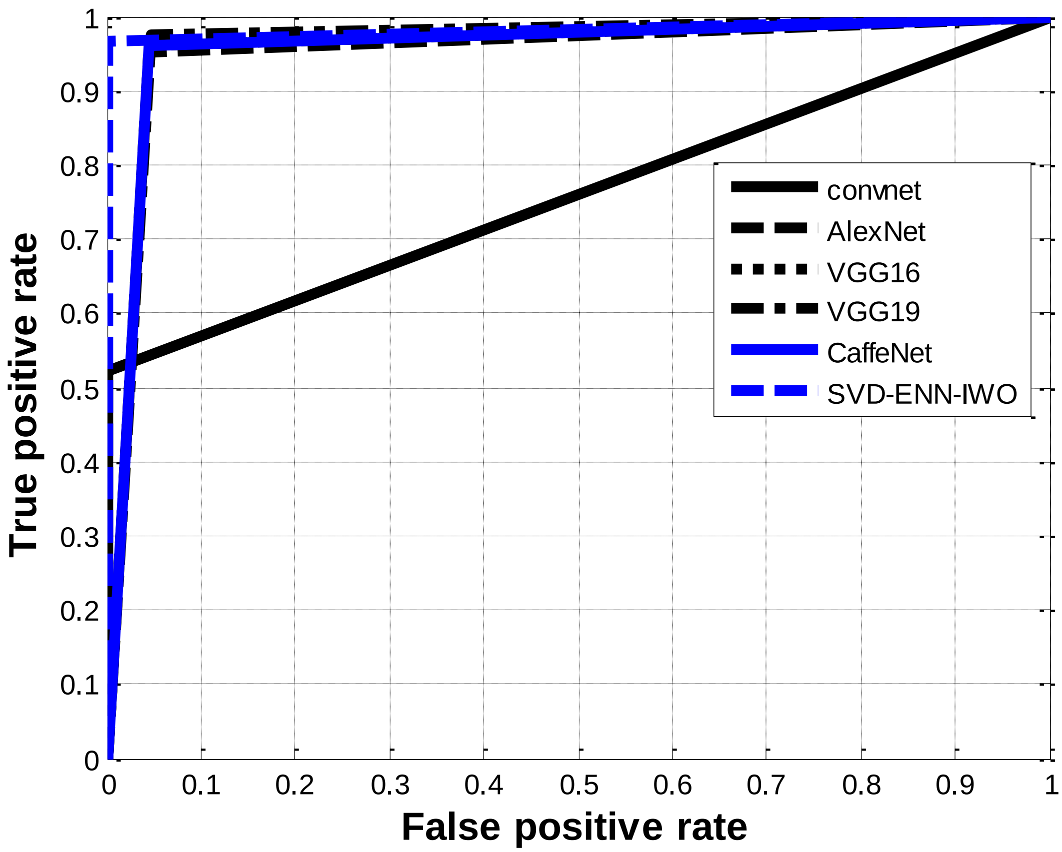


Figure 16: Receiver operating characteristics curves of the prediction models for defects detection based on entire dataset II

The box plots of the overall accuracies achieved by the prediction models are depicted in Figure 17. They enable to synthesize the robustness of the prediction models capitalizing on mapping the distribution and skewness of the numerical data of overall accuracies. Figure 17 displays the minimum, first quartile, third quartile and maximum values of the multiple runs of the prediction models. The height of the box signifies the robustness of the prediction model, whereas lower spread implies a more robust prediction model. It can be observed that the developed model alongside the transfer learning-based deep neural network provides more stable and highly consistent prediction accuracies within the different folds. Conversely, CONVNET experiences more perturbations in the prediction accuracies across the multiple runs.

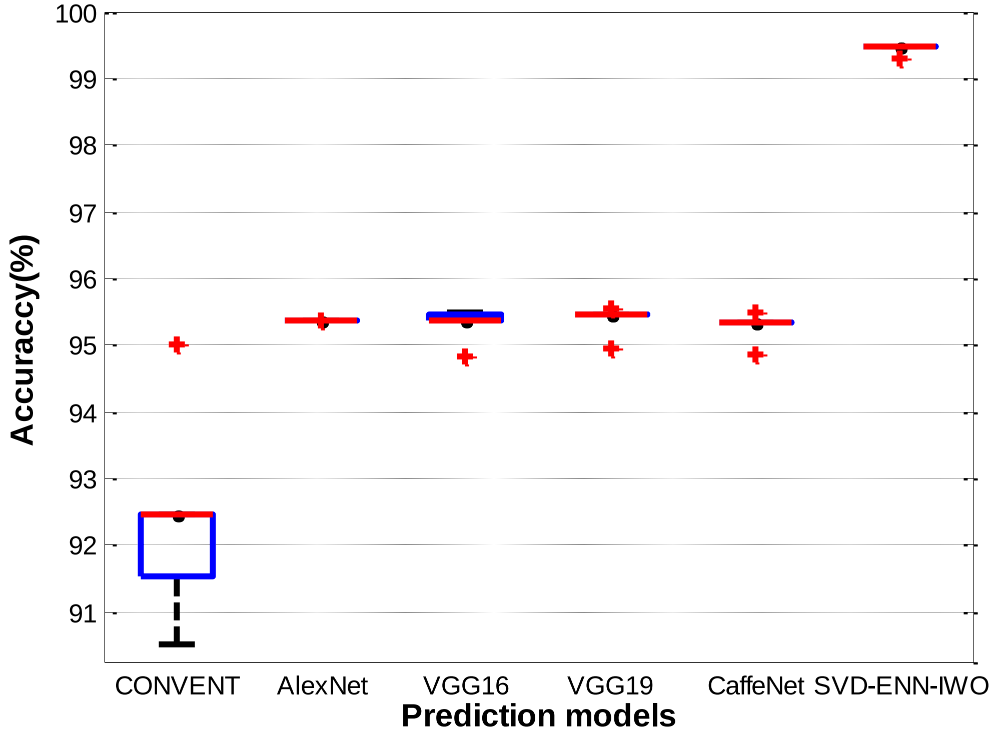


Figure 17: Box plots of the overall accuracy obtained by the prediction models

A further comparison is carried out to evaluate the statistical significance levels of the generated prediction accuracies. In this context, Shapiro-Wilk test is performed to study the normality of the overall prediction accuracies produced by the different folds of the prediction models at significance level of 0.05. It analyzes the null hypothesis (H_0), which implies that the random variable follows a normal distribution. On the other hand, the alternative hypothesis (H_1) implies that the random variable doesn't follow a normal distribution. Hence, if the $P-$value is less than the significance level, then the overall prediction accuracies don't follow normal distribution. Nonetheless, if the $P-$value is more than the significance level, then the overall prediction accuracies follow normal distribution. $P-$values of the overall accuracies of the prediction models using Shapiro-Wilk test are recorded in Table 15. As can be seen, all the $P-$values are less than 0.05, which indicate that the null hypothesis is rejected and therefore overall accuracies of the prediction models don't follow normal distributions.

In this regard, non-parametric testing needs to be applied for their statistical significance analysis at significance level of 0.05. Table 16 reports the $P-$values of the developed model against other prediction models using a set of non-parametric tests. The comparison is carried out based on the overall accuracies of the different folds. Additionally, the non-parametric tests encompass Wilcoxn test, Mann-Whitney-U test, Kruskal–Wallis test, binomial sign test and Mood's median test. The conducted tests examine the null hypothesis (H_0), which is that there is no significant difference between the classification capacities of the prediction models. On the other hand, the alternative hypothesis (H_1) assumes that there is a significant difference between the classification performances of the prediction models. As shown in Table 16, the $P-$values of the pairs ($SVD-ENN-IWO$, $CONVNET$), ($SVD-ENN-IWO$, AlexNet), ($SVD-ENN-IWO$, VGG16), ($SVD-ENN-IWO$, VGG19) and ($SVD-ENN-IWO$, CaffeNet) are less than 0.05. This demonstrates that the developed model significantly outperformed $CONVNET$ and other pre-trained deep networks.

Table 15: $P-$values of the overall accuracies of the prediction models using Shapiro-Wilk test for normality

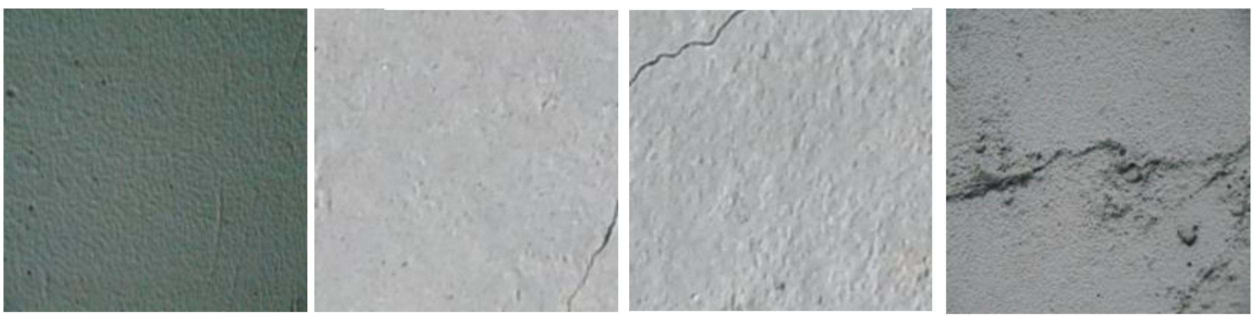
Prediction model	$\mathbf{P-value}$
$CONVNET$	4.53×10^{-2} (H_1)
AlexNet	٩×10^{-٥} (H_1)
VGG16	$٢,٦\times10^{-٤}$ (H_1)
VGG19	٠ (H_1)
CaffeNet	6.9×10^{-4} (H_1)
$SVD-ENN-IWO$	0 (H_1)

Table 16: Statistical comparison of the developed $\mathbf{SVD-ENN-IWO}$ against other prediction models using non-parametric tests

Pair of prediction learning models	Wilcoxn	Mann-Whitney-U	Kruskal–Wallis	Binomial sign	Mood's median
$(\mathrm{SVD-ENN-IWO}, \mathrm{CONVNET})$	$\mathrm{H_1}$ ($\mathrm{P-value}$ $=4.7\times10^{-3}$)	$\mathrm{H_1}$ ($\mathrm{P-value}$ $=0$)	$\mathrm{H_1}$ ($\mathrm{P-value}$ $=0$)	$\mathrm{H_1}$ ($\mathrm{P-value}$ $=1.01\times10^{-4}$)	$\mathrm{H_1}$ ($\mathrm{P-value}$ $=0$)
$(\mathrm{SVD-ENN-IWO}$, AlexNet)	$\mathrm{H_1}$ ($\mathrm{P-value}$ $=3.58\times10^{-3}$)	$\mathrm{H_1}$ ($\mathrm{P-value}$ $=0$)	$\mathrm{H_1}$ ($\mathrm{P-value}$ $=0$)	$\mathrm{H_1}$ ($\mathrm{P-value}$ $=4.73\times10^{-5}$)	$\mathrm{H_1}$ ($\mathrm{P-value}$ $=0$)
$(\mathrm{SVD-ENN-IWO}$, VGG16)	$\mathrm{H_1}$ ($\mathrm{P-value}$ $=4.92\times10^{-3}$)	$\mathrm{H_1}$ ($\mathrm{P-value}$ $=0$)	$\mathrm{H_1}$ ($\mathrm{P-value}$ $=0$)	$\mathrm{H_1}$ ($\mathrm{P-value}$ $=1.01\times10^{-4}$)	$\mathrm{H_1}$ ($\mathrm{P-value}$ $=0$)
$(\mathrm{SVD-ENN-IWO}$, VGG19)	$\mathrm{H_1}$ ($\mathrm{P-value}$ $=4.7\times10^{-3}$)	$\mathrm{H_1}$ ($\mathrm{P-value}$ $=0$)	$\mathrm{H_1}$ ($\mathrm{P-value}$ $=0$)	$\mathrm{H_1}$ ($\mathrm{P-value}$ $=8.63\times10^{-5}$)	$\mathrm{H_1}$ ($\mathrm{P-value}$ $=0$)
$(\mathrm{SVD-ENN-IWO}$, CaffeNet)	$\mathrm{H_1}$ ($\mathrm{P-value}$ $=4.7\times10^{-3}$)	$\mathrm{H_1}$ ($\mathrm{P-value}$ $=0$)	$\mathrm{H_1}$ ($\mathrm{P-value}$ $=0$)	$\mathrm{H_1}$ ($\mathrm{P-value}$ $=8.63\times10^{-5}$)	$\mathrm{H_1}$ ($\mathrm{P-value}$ $=0$)

Sample of the correctly and wrongly classified images is depicted in Figure 18. Figure 18.a describes the correctly classified images by the developed $\mathrm{SVD-ENN-IWO}$ method for bridge defects detection and recognition. As can be seen, the developed method succeeded in detecting the defects and recognizing their type under various lighting conditions. However, few images are misclassified by the developed method, which are displayed in Figure 18.b. In this regard, versatile lighting conditions sometimes create wide variations in the intensity values and illuminance across the image pixels such as the first one to the left imported from dataset I. This caused the incapability of the developed method to detect the presence of defects in the image. The second and third images obtained from dataset II encompassed light and thin cracks which generated low contrast between them and background which lead the developed detection method to wrongly misclassify them as non-defected images. The failure of the developed recognition method in the accurate identification of the type of defect in the fourth image can be explained by the nature of dataset I which is imbalanced towards spalling and scaling in addition to the presence of wide irregularities on the surface.

(a) Correctly classified images



(b) Wrongly classified images

Figure 18: Sample of the correctly and wrongly classified images

The average ranking method is adopted for the sake of establishing an integrative representation of the performances of the prediction models for bridge defects detection and recognition based on datasets I and II. The mean and standard deviation of rankings of the prediction models are reported in Table 17. It is revealed that the developed model achieved the highest rank and satisfactory standard deviation of rankings. In this regard, its mean and standard deviation are equal to 1.22 and 0.92, respectively. Results also demonstrate that VGG16 achieved the second ranking followed by AlexNet and then CaffeNet. It is also found that the transfer learning-based deep neural networks generated an overall prediction performance better than the trained from scratch network. In the light of the previous analysis, it is revealed that the developed method was capable of providing a holistic and significant better classification scheme than other prediction models for both defects detection and recognition. Thus it can aid in establishing more efficient image-based condition assessment models and more reliable maintenance planning actions.

Table 17: Mean and standard deviation of rankings obtained by the prediction models

Prediction model	Mean of rankings	Standard deviation of rankings
CONVNET	4.39	1.74
AlexNet	3.61	0.68
VGG16	3	0.94
VGG19	4.28	1.66
CaffeNet	3.67	1
$\mathrm{SVD} - \mathrm{ENN} - \mathrm{IWO}$	1.22	0.92

8. CONCLUSION

Routine inspections are diagnostic methods to quantity the degradation of the bridges. However, current visual inspection-based methods are labour-intensive and exhibit subjective and imprecise judgments, which eventually lead to error-prone deterioration models and maintenance actions. As such, the present study proposes a newly-developed $\mathrm{SVD} - \mathrm{ENN} - \mathrm{IWO}$ method for the purpose of automated detection and recognition of surface defects. The first tier is envisioned on singular value decomposition which aims at dimensionality reduction of the image through capturing the most significant spatial domain features. In the second tier, invasive weed optimization algorithm is deployed for the purpose of fine-tuning the hyper parameters of Elman neural network in an attempt to alleviate the shortcomings of gradient descent algorithm. In this context, a variable-length optimization model is designed for the purpose of both parameter and structural learning of the Elman neural network. The third tier is designated for evaluating the developed method stepping on a set of performance prediction and statistical significance comparisons. These comparisons encompass a set of conventional machine learning and deep neural networks well-known for their higher classification accuracies. Non-parametric statistical testing involves five type, namely Wilcoxn test, Mann-Whitney-U test, Kruskal–Wallis test, binomial sign test and Mood's median test.

Results demonstrated that the developed method significantly outperformed other prediction models based on split validation and cross validation. With respect to the bridge defects recognition, it was found that the developed method achieved overall accuracy, F-measure, Kappa coefficient, balanced accuracy, Matthews's correlation coefficient and area under curve of 0.955, 0.955, 0.914, 0.965, 0.937 and 0.904, respectively. This demonstrated that the developed model was capable of improving the classification accuracies by values ranging from 10.7% to 37.44% with reference to the back propagation artificial neural network. It was also found that the developed method managed to provide an average improvement in the detection accuracies of 13.68% with respect to CONVNET based on dataset II. A holistic evaluation of the prediction models was carried out using average ranking method. In this regard, it was derived that the developed method could provide a better and robust prediction performance achieving mean and standard deviation for rankings of 1.22 and 0.92, respectively. As such, it is expected that the developed method can lead to better evaluation of severities of surface defects, which eventually aids in establishing more reliable bridge maintenance intervention actions.

REFERENCES

1. Felio, G. (2016). "*Canadian Infrastructure Report Card"*. Canadian Construction Association, Canadian Public Works Association, Canadian Society for Civil Engineering, and Federation of Canadian Municipalities, Canada. <www.canadainfrastructure.ca/downloads/Canadian_Infrastructure_Report_2016.pdf> (06.05.2016).
2. Statistics Canada. (2009). "*Age of Public Infrastructure: A Provincial Perspective*". < http://www.statcan.gc.ca/pub/11-621-m/11-621-m2008067-eng.htm> (20.12.2016).
3. Sennah, K., Juette, B., Witt, C., & Combar, P. M. (2011). "Vehicle Crash Testing On a GFRP-Reinforced PL-3 Concrete Bridge Barrier". In *Proceedings of the 4th International Conference on Durability and Sustainability of Fibre Reinforced Polymer Composites for Construction and Rehabilitation*, Québec City, Canada, 20-22 June.
4. Al-Salih, H., Juno, M., Collins, W., Bennett, C., Li, J., & Sutley, E. J. (2019). Evaluation of a digital image correlation bridge inspection methodology on complex distortion-induced fatigue cracking. *Procedia Structural Integrity*, *17*, 682–689.
5. Guo, L., Li, R., Shen, X., & Jiang, B. (2019). Crack and Noncrack Damage Automatic Classification from Concrete Surface Images using Broad Network Architecture. In *Proceedings of the 31st Chinese Control and Decision Conference*, 1-6, Nanchang, China, 3-5 June.
6. Kim, I. H., Jeon, H., Baek, S. C., Hong, W. H., & Jung, H. J. (2018). Application of crack identification techniques for an aging concrete bridge inspection using an unmanned aerial vehicle. *Sensors*, *18*(6), 1–14.
7. Dorafshan, S., & Azari, H. (2020). Evaluation of bridge decks with overlays using impact echo, a deep learning approach. *Automation in Construction*, *113*, 1-12.
8. Shokouhi, P., Wöstmann, J., Schneider, G., Milmann, B., Taffe, A., & Wiggenhauser, H. (2011). Nondestructive detection of delamination in concrete slabs. *Transportation Research Record*, *2251*(1), 103–113.
9. Sultan, A. A., & Washer, G. A. (2018). Reliability Analysis of Ground-Penetrating Radar for the Detection of Subsurface Delamination. *Journal of Bridge Engineering*, *23*(2), 1–11.
10. Yao, C., Tao, M., Xiaojie, W., & Feng, L. (2016). "A Bridge Crack Image Detection and Classification Method Based On Climbing Robot". *Proceedings of the 35th Chinese Control Conference*, , 4037–4042, Chengdu, China, 27-29 July.
11. Tong, X., Guo, J., Ling, Y., & Yin, Z. (2011). "A New Image-Based Method for Concrete Bridge Bottom Crack Detection." *2011 International Conference on Image Analysis and Signal Processing*, 1-4, Wuhan, China, 21-23 October.
12. Moon, H., & Kim, J. (2011). "Inteligent Crack Detecting Algorithm on The Concrete Crack Image Using Neural Network", *28th International Symposium on Automation and Robotics in Construction (ISARC)*, 1-7, Seoul, Korea, 29 June-2 July.
13. Xuejun, W. & Yan, Z. (2017). "The Detection and Recognition of Bridges' Cracks Based on

Deep Belief Network." *2017 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC)*, 768–771, Guangzhou, China, 21-24 July.

14. Chen, B., Zhang, X., Wang, R., Li, Z., and Deng, W. (2019). Detect concrete cracks based on OTSU algorithm with differential image. *The Journal of Engineering*, *2019*(23), 9088–9091.
15. Dawood, T., Zhu, Z., and Zayed, T. (2017). "Machine vision-based model for spalling detection and quantification in subway networks". *Automation in Construction*, *81*, 149–160.
16. Bu, G. P., Chanda, S., Guan, H., Jo, J., Blumenstein, M., & Loo, Y. C. (2015). "Crack Detection using a Texture Analysis-based Technique for Visual Bridge Inspection." *Electronic Journal of Structural Engineering*, 14(1), 41–48.
17. Cen, J., Zhao, J., Xia, X., & Liu, C. (2017). "Application Research on Convolution Neural Network for Bridge Crack Detection". *Advances in Computer Science Research*, 74, 150–156.
18. Li, Y., Zhao, W., Zhang, X., & Zhou, Q. (2018). "A Two-Stage Crack Detection Method for Concrete Bridges Using Convolutional Neural Networks". *IEICE Transactions on Information and Systems*, E101(12), 3249–3252.
19. Yeboah, Y., Wu, W., Jie, W. J., & Yu, Z. L. (2015). "Robotic Crack Detection and Classification via AdaBoost-RVM Implementation". *International Journal of Mechanical Engineering and Robotics Research*, 4(4), 361–367.
20. Kruachottikul, P., Cooharojananone, N., Phanomchoeng, G., Chavarnakul, T., Kovitanggoon, K., Trakulwaranont, D., & Atchariyachanvanich, K. (2019). "Bridge Sub Structure Defect Inspection Assistance by using Deep Learning". In *2019 IEEE 10th International Conference on Awareness Science and Technology*, Morioka, Japan, 23-25 October.
21. Xu, H., Su, X., Wang, Y., Cai, H., Cui, K., & Chen, X. (2019). "Automatic Bridge Crack Detection Using a Convolutional Neural Network." *Applied Sciences*, *9*(14), 1–14.
22. Kim, I. H., Jeon, H., Baek, S. C., Hong, W. H., & Jung, H. J. (2018). "Application of crack identification techniques for an aging concrete bridge inspection using an unmanned aerial vehicle". *Sensors*, *18*(6), 1–14.
23. Cha, Y. J., Choi, W., & Büyüköztürk, O. (2017). "Deep Learning-Based Crack Damage Detection Using Convolutional Neural Networks." *Computer-Aided Civil and Infrastructure Engineering*, 32, 361-378.
24. Wang, Q., Li, B., & Nie, X. (2019). "Bridge Crack Identification Based on Feature Fusion of Convolutional Neural Networks." *2019 International Conference on Computation and Information Sciences (ICCIS)*, 1-11, Aljouf, Kingdom of Saudi Arabia, 3-4 July.
25. Słoński, M. (2019). "A comparison of deep convolutional neural networksfor image-based detection of concrete surface cracks". *Computer Assisted Methods in Engineering and Science*, *26*(2), 105–112.
26. Dorafshan, S., Thomas, R. J., Coopmans, C., & Maguire, M. (2018). "Deep Learning Neural Networks for sUAS-Assisted Structural Inspections: Feasibility and Application." In *2018 International Conference on Unmanned Aircraft Systems*, 1-9, Dallas, United States of America, 12-15 June.
27. Dung, C. V., & Anh, L. D. (2019). "Autonomous concrete crack detection using deep fully convolutional neural network." *Automation in Construction*, 92, 52-58.
28. Liang, H., & Zou, J. (2020). "Rock Image Segmentation of Improved Semi-supervised SVM–FCM Algorithm Based on Chaos." *Circuits, Systems, and Signal Processing*, *39*(2), 571–585.

29. Xie, Q., & Zhou, K. (2019). “An Optimized Ant Colony Algorithm for Text Edge Extraction.” *Proceedings - 2019 International Conference on Artificial Intelligence and Advanced Manufacturing*, 289–292, Dublin, Ireland, 16-18 October.
30. Mohammed Abdelkader, E., Moselhi, O., Marzouk, M. & Zayed, T. (2019). “A Multi-objective Invasive Weed Optimization Method for Segmentation of Distress Images.” *Intelligent automation and soft computing*, 1-20.
31. Hoang, N. (2018). “Detection of Surface Crack in Building Structures Using Image Processing Technique with an Improved Otsu Method for Image Thresholding”. *Advances in Civil Engineering,* Article ID 3924120, 10 pages.
32. Lei, B., Wang, N., Xu, P., & Song, G. (2018). “New Crack Detection Method for Bridge Inspection Using UAV Incorporating Image Processing.” *Journal of Aerospace Engineering*, *31*(5), 1–13.
33. Zhang, H., Yu, H., Zhang, H., & Yang, W. (2017). “Accurate Extraction Of Cracks On The Underside Of Concrete.” *2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, 2342-2344, Texas, United States of America, 23-28 July.
34. Yang, J., Qu, X. dong, & Chang, M. (2019). “An intelligent singular value diagnostic method for concrete dam deformation monitoring.” *Water Science and Engineering*, *12*(3), 205–212.
35. Mucolli, L., Krupinski, S., Maurelli, F., Mehdi, S. A., & Mazhar, S. (2019). “Detecting cracks in underwater concrete structures: An unsupervised learning approach based on local feature clustering.” *OCEANS 2019 MTS/IEEE SEATTLE*, 1-8, Seattle, United States of America, 27-31 October.
36. Sysyn, M., Gruen, D., Gerber, U., Nabochenko, O., & Kovalchuk, V. (2019). “Turnout monitoring with vehicle based inertial measurements of operational trains: A machine learning approach.” *Communications - Scientific Letters of the University of Zilina*, *21*(1), 42–48.
37. Agahi, H., & Mahmoodzadeh, A. (2020). “Decision fusion scheme for bearing defects diagnosis in induction motors.” *Electrical Engineering*, 1–11.
38. Paradarami, T. K., Bastian, N. D., & Wightman, J. L. (2017). “A hybrid recommender system using artificial neural networks.” *Expert Systems with Applications*, *83*, 300–313.
39. Li, F., Ge, L., Dan, W., Gu, Y., He, Q., & Sun, K. (2019). “Application of improved variable learning rate back propagation neural network in energy dispersion X-ray fluorescence quantitative analysis.” *2019 IEEE 4th International Conference on Cloud Computing and Big Data Analytics*, *ICCCBDA 2019*, 428–432, Chengdu, China, 12-15 April.
40. Mohammed Abdelkader, E., Marzouk, M., & Zayed, T. (2020). “A self-adaptive exhaustive search optimization-based method for restoration of bridge defects images.” *International Journal of Machine Learning and Cybernetics*, *11*, 1659–1716.
41. Dehuri, S., & Cho, S. (2010). “Evolutionarily optimized features in functional link neural network for classification.” *Expert Systems With Applications*, *37*(6), 4379–4391.
42. Singh, P., & Borah, B. (2013). “Indian summer monsoon rainfall prediction using artificial neural network.” *Stochastic Environmental Research and Risk Assessment*, *27*, 1585–1599.
43. Jabin, S. (2014). “Stock market prediction by using artificial neural network.” *International Journal of Computer Applications*, *99*(9), 718–722.
44. Dai, Q., & Liu, N. (2012). “Alleviating the problem of local minima in Backpropagation through competitive learning.” *Neurocomputing*, *94*, 152–158.
45. Dundar, M., Kou, Q., Zhang, B., He, Y., & Rajwa, B. (2015). “Simplicity of kmeans versus deepness of deep learning: A case of unsupervised feature learning with limited data.”

Proceedings of 2015 IEEE 14th International Conference on Machine Learning and Applications, , 883-888, Miami, United States of America, 9-11 December.

46. Shang, S., Lin, S., & Cong, F. (2020). "Zebrafish larvae phenotype classification from bright-field microscopic images using a two-tier deep-learning pipeline." *Applied Sciences*, *10*(4), 1-12.
47. Al Najada, H., Mahgoub, I., and Mohammed, I. (2019). "Cyber Intrusion Prediction and Taxonomy System Using Deep Learning and Distributed Big Data Processing." *Proceedings of the 2018 IEEE Symposium Series on Computational Intelligence*, 631-638, Bangalore, India, 18-21 November.
48. Albelwi, S., & Mahmood, A. (2017). "A framework for designing the architectures of deep Convolutional Neural Networks." *Entropy*, *19*(6), 1-20.
49. Hetherington, J., Lessoway, V., Gunka, V., Abolmaesumi, P., & Rohling, R. (2017). "SLIDE: automatic spine level identification system using a deep convolutional neural network." *International Journal of Computer Assisted Radiology and Surgery*, *12*(7), 1189–1198
50. Zhang, J., Li, Y., Xiao, W., & Zhang, Z. (2020). "Non-iterative and Fast Deep Learning: Multilayer Extreme Learning Machines."*Journal of the Franklin Institute*, 1-31.
51. Hassan, M., & Bermak, A. (2016). "Robust Bayesian Inference for Gas Identification in Electronic Nose Applications by Using Random Matrix Theory." *IEEE Sensors Journal*, *16*(7), 2036–2045.
52. Alhamdoosh, M., & Wang, D. (2014). "Fast decorrelated neural network ensembles with random weights." *Information Sciences*, *264*, 104–117.
53. Al-Allaf, O. N. A. (2011). "Fast BackPropagation Neural Network algorithm for reducing convergence time of BPNN image compression." *2011 International Conference on Information Technology and Multimedia*, 1-6, Kuala Lumpur, Malaysia, 14-16 November.
54. Dhillon, A., & Verma, G. K. (2019). "Convolutional neural network: a review of models, methodologies and applications to object detection." *Progress in Artificial Intelligence*, 1–28.
55. Stamate, C., Magoulas, G. D., & Thomas, M. S. C. (2015). "Transfer learning approach for financial applications." *ArXiv:1509.02807*, 1–5.
56. Zhu, J., & Song, J. (2020). Weakly supervised network based intelligent identification of cracks in asphalt concrete bridge deck. *Alexandria Engineering Journal*, *59*(3), 1307–1317.
57. Prasanna, P., Dana, K., Gucunski, N., & Basily, B. (2012). "Computer Vision Based Crack Detection and Analysis." *Proceedings of SPIE 8345, Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems, California, United States of America.*
58. Chaiyasarn, K., Khan, W., Ali, L., Sharma, M., Brackenbury, D., & DeJong, M. (2018). "Crack detection in masonry structures using convolutional neural networks and support vector machines." *35th International Symposium on Automation and Robotics in Construction, 1-8,* Berlin, Germnay, 20-25 July.
59. Prasad, R., Ali, M., Xiang, Y., & Khan, H. (2020). "A double decomposition-based modelling approach to forecast weekly solar radiation." *Renewable Energy*, *152*, 9–22.
60. Bhandari, S., Thakur, B., Kalra, A., Miller, W. P., Lakshmi, V., & Pathak, P. (2019). "Streamflow Forecasting Using Singular Value Decomposition and Support Vector Machine for the Upper Rio Grande River Basin." *Journal of the American Water Resources Association*, *55*(3), 680–699.

61. Mulimani, M., & Koolagudi, S. G. (2019). "Segmentation and characterization of acoustic event spectrograms using singular value decomposition." *Expert Systems with Applications*, *120*, 413–425.
62. Yu, C., Li, H., & Wang, X. (2019). "SVD-based image compression, encryption, and identity authentication algorithm on cloud." *IET Image Processing*, *13*(12), 2224–2232.
63. Guo, Q., Zhang, C., Zhang, Y., & Liu, H. (2016). "An Efficient SVD-Based Method for Image Denoising." *IEEE Transactions on Circuits and Systems for Video Technology*, *26*(5), 868–880.
64. Chitsaz, N., Azarnivand, A., & Araghinejad, S. (2016). "Pre-processing of data-driven river flow forecasting models by singular value decomposition (SVD) technique." *Hydrological Sciences Journal*, *61*(12), 2164–2178.
65. Tanwar, S., Ramani, T., & Tyagi, S. (2018). "Dimensionality reduction using PCA and SVD in big data: A comparative case study." *International Conference on Future Internet Technologies and Trends*, , 116–125, Surat, India, 31 August-2 September.
66. Asgari, H. R., Haddad, O. B., Pazoki, M., & Loáiciga, H. A. (2016). "Weed optimization algorithm for optimal reservoir operation." *Journal of Irrigation and Drainage Engineering*, *142*(2), 1–11.
67. Goharnejad, H., Mohamadi Naser, M., & Niri, M. Z. (2017). "The Optimization of Energy Supply Systems by Sequential Streamflow Routing Method and Invasive Weed Optimization Algorithm; Case Study: Karun II Hydroelectric Power Plant." *Journal of Hydraulic Structures*, *3*(1), 71–82.
68. Nagaraju, T. V., Durga Prasad, C., & Murthy, N. G. K. (2020). "Invasive Weed Optimization Algorithm for Prediction of Compression Index of Lime-Treated Expansive Clays." *Soft Computing for Problem Solving*, 317-324.
69. Goli, A., Tirkolaee, E. B., Malmir, B., Bian, G. Bin, & Sangaiah, A. K. (2019). "A multi-objective invasive weed optimization algorithm for robust aggregate production planning under uncertain seasonal demand." *Computing*, *101*(6), 499–529.
70. Uyar, K., & Ülker, E. (2017). "B-spline curve fitting with invasive weed optimization." *Applied Mathematical Modelling*, *52*, 320–340.
71. Razmjooy, N. & Khalilpour, M. (2015). "A new design for PID controller by considering the operating points changes in Hydro-Turbine Connected to the equivalent network by using Invasive Weed Optimization (IWO) Algorithm." *International Journal of Information, Security, and Systems Management*, *4*, 468–475.
72. Rathi, V. P. G. P., & Palani, D. S. (2012). "Brain tumor mri image classification w ith feature selection and extraction using linear discriminant analysis". *International Journal of Information Sciences and Techniques*, *2*(4).
73. Yang, C. C., Soh, C. S., & Yap, V. V. (2018). "A systematic approach in appliance disaggregation using k-nearest neighbours and naive Bayes classifiers for energy efficiency". *Energy Efficiency*, *11*, 239–259.
74. Feng, C., Ju, S., & Huang, H. (2016). "Using a Simple Soil Spring Model and Support Vector Machine to Determine Bridge Scour Depth and Bridge Safety". *Journal of Performance of Constructed Facilities*, *30*(4), 1–14.
75. Ahmad, M. W., Mourshed, M., & Rezgui, Y. (2017). "Trees vs Neurons : Comparison between random forest and ANN for high-resolution prediction of building energy consumption". *Energy & Buildings*, *147*, 77–89.

76. El-Zahab, S., Mohammed Abdelkader, E. & Zayed, T. (2018). "An Accelerometer-Based Leak Detection System". Mechanical Systems and Signal Processing, *108*, 276-271.
77. Dorafshan, S., Thomas, R. J., & Maguire, M. (2018). "SDNET2018: An annotated image dataset for non-contact concrete crack detection using deep convolutional neural networks. *Data in Brief*, *21*, 1664–1668.
78. Yu, D., Hong, J., Zhang, J., & Niu, Q. (2018). "Multi-Objective Individualized-Instruction Teaching-Learning-Based Optimization Algorithm." *Applied Soft Computing*, *62*, 288–314.
79. Chang, B., Tsai, H., and Yen, C. (2016). "Engineering Applications of Artificial Intelligence SVM-PSO based rotation-invariant image texture classification in SVD and DWT domains". *Engineering Applications of Artificial Intelligence*, *52*, 96–107.
80. Jha, R. K., and Chouhan, R. (2014). "Noise-induced contrast enhancement using stochastic resonance on singular values". *Signal, Image and Video Processing*, 8(2), 339–347.
81. Singh, N. K., Singh, A. K., and Tripathy, M. (2014). "A Comparative Study of BPNN , RBFNN and ELMAN Neural Network for Short-Term Electric Load Forecasting : A Case Study of Delhi Region." *9th International Conference on Industrial and Information Systems (ICIIS)*, 1-6, Gwalior, India, 15-17 December.
82. Kurach, K., and Pawlowski, K. (2016). "Predicting Dangerous Seismic Activity with Recurrent Neural Networks." *Proceedings of the Federated Conference on Computer Science*, 239–243, Gdansk, Poland, 11 - 14 September.
83. Bianchi, F. M., Maiorino, E., Kampffmeyer, M. C., Rizzi, A., and Jenssen, R. (2017). "An overview and comparative analysis of Recurrent Neural Networks for Short Term Load Forecasting". *arXiv preprint*, 1-41.
84. Köker, R. (2013). "A Genetic Algorithm Approach to a Neural-network-based Inverse Kinematics Solution of Robotic Manipulators Based on Error Minimization." *Information and Control*, 222, 528–543.
85. Wang, J., Zhang, W., Li, Y., Wang, J., and Zhangli Dang. (2014). "Forecasting Wind Speed Using Empirical Mode Decomposition and Elman Neural Network." *Applied Soft Computing*, 23, 452–459.
86. Yu, F., and Xu, X. (2014). "A Short-term Load Forecasting Model of Natural Gas Based on Optimized Genetic Algorithm and Improved BP Neural Network". *Applied Energy*, *134*, 102–113.
87. Ma, Y., & Zhang, Z. (2020). "Travel Mode Choice Prediction Using Deep Neural Networks with Entity Embeddings." *IEEE Access*, *8*, 64959–64970.
88. R. M., S. P., , Maddikunta, P. K. R., Parimala, M., Koppu, S., Gadekallu, T. R., Chowdhary, C. L., & Alazab, M. (2020). "An effective feature engineering for DNN using hybrid PCA-GWO for intrusion detection in IoMT architecture." *Computer Communications*, *160*, 139–149.
89. Chan, F. T. S., Wang, Z. X., Patnaik, S., Tiwari, M. K., Wang, X. P., & Ruan, J. H. (2020). "Ensemble-learning based neural networks for novelty detection in multi-class systems." *Applied Soft Computing Journal*, *93*, 1-14.
90. Abu Saleem, R., Radaideh, M. I., & Kozlowski, T. (2020). "Application of deep neural networks for high-dimensional large BWR core neutronics." *Nuclear Engineering and Technology*, 1–8.
91. Ryerkerk, M. L., Averill, R. C., Deb, K., and Goodman, E. D. (2016). "Solving Metameric Variable-length Optimization Problems Using Genetic Algorithms." *Genetic Programming and Evolvable Machines*, *128*(2), 247-277.

92. Abu-al-Nadi, D. I., Alsmadi, O. M. K., Abo-hammour, Z. S., Hawa, M. F., and Rahhal, J. S. (2013). "Invasive Weed Optimization for Model Order Reduction of Linear MIMO Systems." *Applied Mathematical Modelling*, 37(6), 4570–4577.
93. Zhou, Y.-Q., and Xidian, H. C. (2014). "Invasive Weed Optimization Algorithm for Optimization No-Idle Flow Shop Scheduling Problem". *Neurocomputing*, 137, 285–292.
94. Azizipour, M., Ghalenoei, V., Afshar, M. H., and Solis, S. S. (2016). "Optimal Operation of Hydropower Reservoir Systems Using Weed Optimization Algorithm". *Water Resources Management*, *30*, 3995–4009.
95. Xu, G., Zhu, X., Fu, D., Dong, J., & Xiao, X. (2017). "Automatic land cover classification of geo-tagged field photos by deep learning." *Environmental Modelling and Software*, *91*, 127–134.
96. Özsert Yiğit, G., & Özyildirim, B. M. (2018). "Comparison of convolutional neural network models for food image classification." *Journal of Information and Telecommunication*, *2*(3), 347–357.
97. Simonyan, K., & Zisserman, A. (2015). "Very deep convolutional networks for large-scale image recognition." *3rd International Conference on Learning Representations*, 1–14, San Diego, United States of America, 7-9 May.
98. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S. & Darrell, T. (2014). "Caffe: Convolutional architecture for fast feature embedding." *Proceedings of the 22nd ACM International Conference on Multimedia***,** 675–678, Florida, United States of America.
99. Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). "ImageNet Classification with Deep Convolutional Neural Networks." *Advances in Neural Information Processing Systems*, 1097–1105.
100. Gibril, M. B. A., Kalantar, B., Al-Ruzouq, R., Ueda, N., Saeidi, V., Shanableh, A., Mansor,S., & Shafri, H. Z. M. (2020). "Mapping heterogeneous urban landscapes from the fusion of digital surface model and unmanned aerial vehicle-based images using adaptive multiscale image segmentation and classification." *Remote Sensing*, *12*(7), 1-27.
101. Zhan, X., & Ukkusuri, S. V. (2019). "Spatial dependency of urban sprawl and the underlying road network structure." *Journal of Urban Planning and Development*, *145*(4), 1–8.
102. Chicco, D., & Jurman, G. (2020). "The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation." *BMC Genomics*, *21*(1), 1–13.
103. Chang, C. C., & Chen, S. H. (2019). "Developing a Novel Machine Learning-Based Classification Scheme for Predicting SPCs in Breast Cancer Survivors." *Frontiers in Genetics*, *10*, 1–6.
104. Mocherla, S., Danehy, A., Impey, C., & Mocherla, S. (2018). "Evaluation of Naive Bayes and Support Vector Machines for Wikipedia Evaluation of Naive Bayes and Support Vector Machines for Wikipedia". *Applied Artificial Intelligence*, 31, 733-744.
105. Zhou, J., Li, X., & Mitri, H. S. (2016). "Classification of Rockburst in Underground Projects : Comparison of Ten Supervised Learning Methods". *Journal of Computing in Civil Engineering*, *30*(5), 1–19.
106. Jain, S., Kotsampasakou, E., & Ecker, G. F. (2018). "Comparing the performance of meta-classifiers — a case study on selected imbalanced data sets relevant for prediction of liver toxicity". *Journal of Computer-Aided Molecular Design*, *32*(5), 583–590.

107. Shang, X., Li, X., Morales-esteban, A., & Chen, G. (2017). “Improving microseismic event and quarry blast classification using Artificial Neural Networks based on Principal Component Analysis”. *Soil Dynamics and Earthquake Engineering*, *99*, 142–149.

108. Jiao, Y., & Du, P. (2016). “Performance measures in evaluating machine learning based bioinformatics predictors for classifications”. *Quantitative Biology*, *4*(4), 320–330.