AIMS Mathematics, 8(12): 29290–29306.
DOI: 10.3934/math.20231499
Received: 20 August 2023
Revised: 22 September 2023
Accepted: 06 October 2023
Published: 27 October 2023

http://www.aimspress.com/journal/Math

Research article

Reinforced concrete bridge damage detection using arithmetic optimization algorithm with deep feature fusion

Majdy M. Eltahir[1], Ghadah Aldehim[2], Nabil Sharaf Almalki[3], Mrim M. Alnfiai[4,*] and Azza Elneil Osman[5]

[1] Department of Information Systems, College of Science & Art at Mahayil, King Khalid University, Saudi Arabia
[2] Department of Computer Sciences, College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University, P.O. Box 84428, Riyadh, 11671, Saudi Arabia
[3] Department of Special Education, College of Education, King Saud University, Riyadh 12372, Saudi Arabia
[4] Department of Information Technology, College of Computers and Information Technology, Taif University, Taif P.O. Box 11099, Taif, 21944, Saudi Arabia
[5] Department of Computer and Self Development, Preparatory Year Deanship, Prince Sattam bin Abdulaziz University, AlKharj, Saudi Arabia

*** Correspondence:** Email: m.alnofiee@tu.edu.sa.

Abstract: Inspection of Reinforced Concrete (RC) bridges is critical in order to ensure its safety and conduct essential maintenance works. Earlier defect detection is vital to maintain the stability of the concrete bridges. The current bridge maintenance protocols rely mainly upon manual visual inspection, which is subjective, unreliable and labour-intensive one. On the contrary, computer vision technique, based on deep learning methods, is regarded as the latest technique for structural damage detection due to its end-to-end training without the need for feature engineering. The classification process assists the authorities and engineers in understanding the safety level of the bridge, thus making informed decisions regarding rehabilitation or replacement, and prioritising the repair and maintenance efforts. In this background, the current study develops an RC Bridge Damage Detection using an Arithmetic Optimization Algorithm with a Deep Feature Fusion (RCBDD-AOADFF) method. The purpose of the proposed RCBDD-AOADFF technique is to identify and classify different kinds of defects in RC bridges. In the presented RCBDD-AOADFF technique, the feature fusion process is performed using the Darknet-19 and Nasnet-Mobile models. For damage classification process, the attention-based

Long Short-Term Memory (ALSTM) model is used. To enhance the classification results of the ALSTM model, the AOA is applied for the hyperparameter selection process. The performance of the RCBDD-AOADFF method was validated using the RC bridge damage dataset. The extensive analysis outcomes revealed the potentials of the RCBDD-AOADFF technique on RC bridge damage detection process.

Keywords: sustainability; RC bridge; damage detection; arithmetic optimization algorithm; feature fusion
Mathematics Subject Classification: 11Y40

1. Introduction

Bridges are one of the main components of a road network and it plays a vital role in the functioning of the transportation system. Reinforced Concrete (RC) bridges are the most important arteries of the infrastructural network that are developed to help for a longer time [1]. However, the defects in bridges might reduce the life expectancy of the bridges. In general, the visual analysis of the RC bridges mainly depends on the perception and capability of the engineers who inspect the bridges manually [2]. Many a times, engineers may encounter problems in analyzing the parts of the bridges, which is complicated to conclude. Therefore, the chances are high for miscalculations and incorrect estimation of the bridge condition. In service life, the bridges are exposed to several deteriorative actions influenced by harsh meteorological conditions, destructive environmental activities, variable loading, and material ageing [3]. Thus, different forms of damage (for example, corrosion and crack) occur over time and damage the structural behavior of the bridges [4]. Hence, it is crucial to detect and estimate the damage at early stages in an accurate manner to avoid failure and retain the structural serviceability and protection of the bridges.

In recent years, Structural Health Monitoring (SHM) has gained much attention among the researchers and has been a subject of various efforts [5]. Several methods that depend on non-destructive analysis (e.g., ground-penetrating radar, ultrasonic and infrared methods), visual analysis, and sensors (For example, displacement sensors, velocimeters, and accelerometers) are employed for detection, localization and quantification of the damage in bridges. However, visual analysis has been the most important method for bridge quality assessment [6]. Skilled investigators execute *in-situ* analysis of the bridge elements based on the accepted guidelines and determine the conditions of the whole bridges. Nevertheless, this traditional analysis technique is time-consuming, laborious, and error-prone, because of the subjective decision of the examiners. Additionally, it also needs access equipment and vehicles to reach different parts of the bridge with lower availability that incur higher expenditure to monitor the operations [7].

Numerous vision-based approaches have been extensively applied for automatic detection of defects in various civil engineering projects [8]. These techniques include classical Machine Learning (ML) methods, Deep Convolutional Neural Networks (DCNNs) and Image Processing Techniques (IPTs). Cracks remain the major form of defects examined by the investigators in certain concrete defect identification designs. IPTs can be employed in the extraction of characteristic features of the cracks from input images using different morphological processes and filters. Then, the extracted features are fed into ML techniques for performing the classification tasks [9]. Though the IPTs offer handcrafted features for training purposes, the current systems have constrained learning abilities,

which do not characterize the challenging concrete texture and complex image acquisition conditions like camera movements, shading, and lighting. Alternatively, the DCNNs extract the features from a group of trained images using convolutional method and classify them using a single learning technique [10]. Due to their learning abilities and robust feature extraction nature, the DCNNs are extensively analysed in concrete defect classification research.

The current study develops an RC Bridge Damage Detection using an Arithmetic Optimization Algorithm with a Deep Feature Fusion (RCBDD-AOADFF) method. The purpose of the proposed RCBDD-AOADFF technique is to identify and classify different kinds of defects in RC bridges. In the presented RCBDD-AOADFF technique, the feature fusion process is performed with the help of the Darknet-19 and NASNet-Mobile models. For damage classification process, the Attention-based Long Short-Term Memory (ALSTM) model is used. To enhance the classification outcomes of the ALSTM model, the AOA is applied for the hyperparameter selection process. The performance of the RCBDD-AOADFF method was validated using the RC bridge damage dataset.

2. Related works

Bai et al. [11] developed the RC element mechanical damage image database primarily containing Component Damage Recognition Tasks (CDRTs). In this database, the EfficientNetV2 model was chosen as the baseline model for the execution of CDRTs. Then, the Discrete Wavelet Transform (DWT) method was introduced to incorporate the frequency data. At last, both elements and the structural defects were connected and structural protection fast assessment techniques were suggested based on the outcome of CDRTs. Kim and Cho [12] introduced a DL approach-based method named Mask RCNN, designed for sample segmentation. The framework of the Mask R-CNN model had three phases (such as classification, segmentation, and region proposal) and was optimized to identify various types of concrete defects. Wan et al. [13] proposed a DL technique named Bridge Detection Transformer (BR-DETR) based on Detection Transformer (DETR). In this study, the convolution was modified by Deformable Conv2D which presents the 2D offsets for normal grid sampling aspects of the traditional convolution. Further, convolution project attention was included after the self-attention layer. In every encoding and decoding layer, the Locally-enhanced Feedforward (LeFF) layer was employed for changing the FF.

In literature [14], the authors presented a new DL technique by employing one-dimensional CNN (1D-CNN) to utilize the raw EMA signature for automatic detection of small defects in concrete structures that remove the slow data pre-processing technique for network training and analysis. Two self-reliant EMA datasets, assessed by the smart piezoelectric sensors, were designed based on the proof-of-concept tests of small or serious mass-loss defect identification on a concrete cube. Zhang et al. [15] introduced a combination of 1D-CNN and LSTM algorithms in the image frequency domain. This technique was trained utilizing multiple images of non-cracked and cracked concrete bridge decks. Next, the flattened-frequency data was used to determine and measure the cracks. Zou et al. [16] suggested a quantitative structural defect level assessment approach. In this study, YOLOv4 network was utilized for the detection of various types of damages. Depthwise separable convolution was presented in this research work for minimizing the computational cost. Eventually, the defect identification and assessment techniques were incorporated within the GUI in order to facilitate the post-earthquake RC structural defect assessment.

Sun et al. [17] introduced a bridge defect identification and localization approach by employing

deflection or inclination measurements. Firstly, this analysis introduced a bridge defect identification and localization approach using an alteration in the deviation of extreme nodal load, evaluated by the CNN and PLSR algorithms. Secondly, the CNN technique was developed and a mathematical correlation was created between the measured excess nodal load and the output of the monitoring point response, which was fed as an input by training. Kumar et al. [18] proposed a DL approach named YOLOv3. This technique employed Jetson-TX2 as hardware to run the YOLOv3 model and was implemented on Pixhawk's hardware standards-based open-source hexacopter drone.

3. The proposed model

In the current study, a new RCBDD-AOADFF method has been introduced for automated and accurate RC bridge damage detection process. The primary aim of the RCBDD-AOADFF technique is to detect and classify different kinds of defects in RC bridges. In the presented RCBDD-AOADFF technique, three major processes are involved namely deep feature fusion, ALSTM classification and AOA-based hyperparameter tuning. Figure 1 illustrates the overall working flow of the RCBDD-AOADFF methodology.

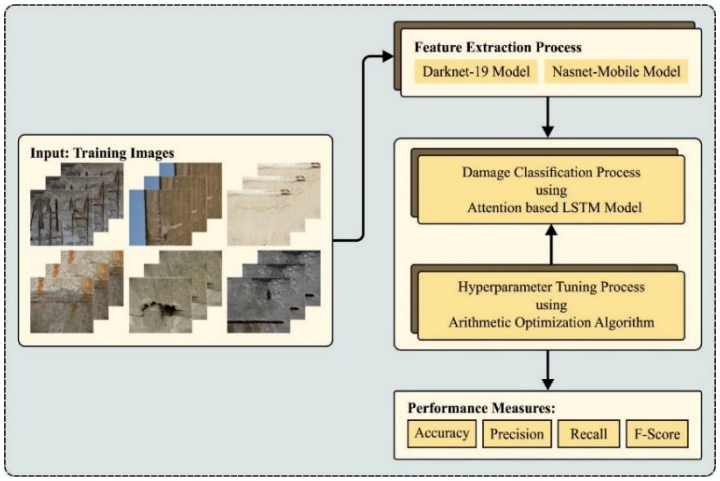

Figure 1. Overall flow of the RCBDD-AOADFF system.

3.1. Feature fusion process

For the feature fusion process, two models such as Darknet-19 and Nasnet-Mobile models are used in this study. Darknet-19 is a deep CNN (DCNN) structure that was devised for CV tasks including object detection and classification [19]. Being a special kind of Darknet structure, introduced by Joseph Redmon, it contains 19 layers in total and so it is named as Darknet_19. The Darknet_19 architecture comprises of 19 layers such as one FC layer and 18 convolution layers. It exploits the basic component as the bigger Darknet_53 structure yet it contains fewer layers, which makes it faster and computationally efficient in terms of training. In 2018, Google scholars proposed NASNet Mobile, a kind of CNN structure that was specifically created for the deployment of mobile devices. It is advantageous in terms of having a comparably low number of parameters than the rest of the CNN structures. These features make it appropriate in the deployment of mobile devices with

constrained computation resources. The structure is encompassed by different blocks of pooling and convolution layers, followed by the FC layer for classification. Furthermore, it also performs different optimization approaches namely learning rate scheduling, dropout, and batch normalization to improve the generalization performance of the network.

This model is initially trained using the ImageNet data set containing 1,000 different classes. The last FC layer is replaced by a novel dense layer to make the model, a fit one for the Hybrid-KTH dataset. Next, the adapted model is fine-tuned with the help of the Transfer Learning (TL) methods on this dataset. A ratio of 70:30 was exploited in this study for training and testing the datasets, correspondingly. The hyperparameters were configured by a dropout of 0.3, a mini-batch size of 16, 200 epochs and an initial learning rate of 0.01. Lastly, the deep model, fine-tuned with transfer learning, was trained.

2D Global Average Pooling $(2gap)$ is used for the extraction of the features from the Darknet_19 architecture. The extracted feature vector is $(N \times 1024)$, signified by V1. For NASNet-Mobile, the "Global Average Pooling $2d$ layer" (gap2) is used for the feature extraction process. The extracted feature vector is $(N \times 1056)$, signified by V2.

Feature fusion is a method that includes the integration of feature representations or maps, derived from different modalities or sources with an intention to improve the performance of the model. In this study, the authors used a modified relation extended serial technique to combine the Vec1 and Vec2 feature vectors. Consider that V1 belongs to X_a and V2 belongs to Y_b. The equation given below defines the relationship between a and b.

$$Corelation = \frac{\sum(X_a - \overline{X})(Y_b - \overline{Y})}{\sqrt{\sum(X_a - \overline{X})^2(Y_b - \overline{Y})^2}} \tag{1}$$

After using the equation, a feature with a positive relation is stored in the novel vector named V3, whereas negative or zero correlations are stored in V4. Then, the mean value of V4 is calculated whereas all the features are compared with these values based on Eq (2):

$$CT = \begin{cases} \tilde{V}_4 & forV_4 \geq \mu \\ ignore, otherwise \end{cases} \tag{2}$$

At last, V5 is attained after the fusion is implemented on V4 and V3 as follows.

$$V5(k) = \begin{pmatrix} V_3\ (k)_{u \times v} \\ \tilde{V}_4\ (k)_{u \times v} \end{pmatrix} \tag{3}$$

The concluding vector V5 is sized at $(N \times 2080)$.

3.2. Image classification process

In this stage, the ALSTM model is utilized for the image classification process. A combination of the attention mechanism and neural network is used here for training the algorithm [20]. In the LSTM model, the final hidden layer is used as an input sequence to signify the sentence. On the contrary, the authors used feedforward attention in the LSTM model with an attention to compute the weight amount of each hidden state from the LSTM layers. The concept of applying the attention layer is to emphasize the significant feature of the information by allocating weight to the input series. Heavy weight is

allocated to the most relevant feature. In NLP domain, this is usually applied for the linguistic components present in the text. This method is intended to simulate the human attention by taking the relevant data and investigating, whether the instance (i.e., a sentence or paragraph in NLP) belongs to a specific category. During the subjectivity analysis, the network analyzes whether the provided text is objective or subjective. On the other hand, the attention model captures the relevant data that specifies the sentence classification process. The attention module enables the NN to define the part of the sentence that needs to be heavily weighed; specifically, to which part most of the attention should be paid. The global attention mechanism is used in the current study. Consider the information from the hidden state in the procedure of context vector construction. During the current timestep, it exploits the output from the encoding and decoding parts. Figure 2 demonstrates the architecture of the ALSTM model.

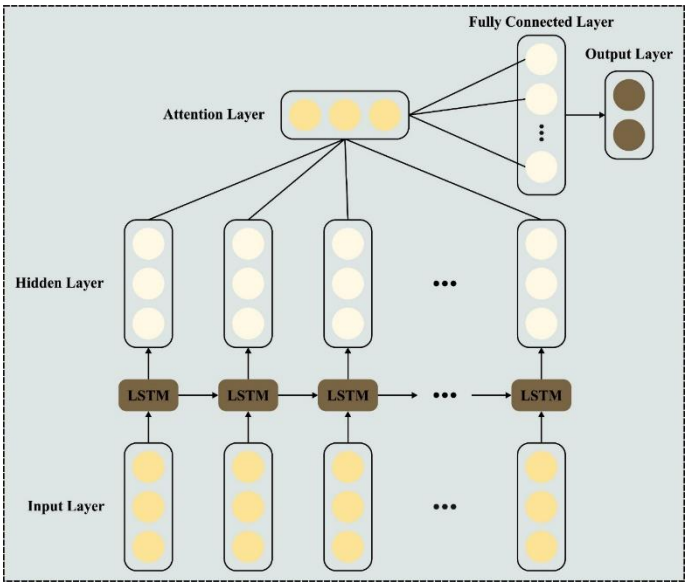

Figure 2. Architecture of the ALSTM model.

Firstly, the sentence is transformed into a sentence vector for x input. h_r hidden state is built for every time step t. W_w signifies the word-level weight matrices and b_w corresponds to the bias parameter. The vector is fed as an input to the single layer MLP that produces a hidden depiction u_t for the hidden vector h_t. Next, the value for h_t is evaluated to denote the significance of all the elements in the sentence, provided a word-level context vector is u_w and the value of the hidden depiction is u_t. The word-level context vector u_w is exploited as a depiction that specifies the significance of all the words. Next, the weighted mean of the h_t hidden state is evaluated by α_t, a softmax function. Here, T represents the length of the input series. At last, the sentence vector demonstration s_i is calculated as a weight amount of the annotations.

$$u_t = \tanh(W_w h_t + b_w), \tag{4}$$

$$\alpha_t = \frac{\exp\left(u_r^T u_w\right)}{\sum_{k=1}^{T_x} \exp(u_t^T u_w)}, \tag{5}$$

$$s_i = \sum_{i=1}^{n} \exp(\alpha_t h_t). \tag{6}$$

Then, $tanh$ is used as the activation function owing to its better non-linear ability. In the LSTM model, the hidden layer of the last cell is fed as an input to the FC softmax layer for classifying the sentence as their label class, y.

$$y = softmax(W_c + b_c). \tag{7}$$

In Eq (7), $y \in R^{\mathrm{k}}$ represents the probability of being subjective whereas W_c and b_c correspond to the trained parameters. After a series of LSTM hidden outputs, the attention layer is added.

3.3. Hyperparameter tuning

The AOA is used to adjust the hyperparameter values of the ALSTM model. The AOA comprises two search processes such as exploration and exploitation, inspired by the mathematical operators such as $-$, $+$, $*$ and $/$ [21]. At first, the AOA generates a set of N solutions (agent). Each one exhibits the agent to test the problem. Thus, the solution or agent represents the X population as given below.

$$X = \left[x_{N-1,1}, x_{N,1}, x_{2,1}, x_{1,1}, x_{N-1,j}, x_{N,j}, x_{2,j}, x_{1,j}, x_{N,n-1} x_{1,n-1}, x_{N-1,n}, x_{N,n}, x_{2,n}, x_{1,n}\right]. \tag{8}$$

$$X = \begin{bmatrix} x_{1,1} & \cdots & x_{1,j} & x_{1,n-1} & x_{1,n} \\ x_{2,1} & \cdots & x_{2,j} & \cdots & x_{2,n} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{N-1,1} & \cdots & x_{N-1,j} & \cdots & x_{N-1,n} \\ x_{N,1} & \cdots & x_{N,j} & x_{N,n-1} & x_{N,n} \end{bmatrix}. \tag{9}$$

Then, the FF of each solution is evaluated to detect the optimal one, X_b. Then, the AOA implements either exploration or exploitation process based on the Math Optimizer Accelerated (MOA) value [22]:

$$MOA(t) = \text{ Min } + t \times \left(\frac{\text{Max}_{MOA} - \text{Min}_{MOA}}{M_t}\right). \tag{10}$$

In Eq (10), M_t shows the overall number of iterations. Min_{MOA} and Max_{MOA} correspondingly indicate the minimal and maximal values of the accelerated function. Especially, the division (D) and multiplication (M) are utilized from the exploration phase of AOA and it can be attained using the succeeding equation:

$$X_{i,j}(t+1) = \begin{cases} X_{bj} \div (M_{OP} + \in) \times \left(\left(UB_j - LB_j\right) \times \mu + LB_j\right), & r_2 < 0.5, \\ X_{bj} \times M_{OP} \times \left(\left(UB_j - LB_j\right) \times \mu + LB_j\right), & otherwise. \end{cases} \tag{11}$$

In Eq (11), $\in$ shows the small integer value, UB_j and LB_j refer to lower and upper limitations of the searching area at j^{th} parameter. $\mu = 0.5$ represents the control function. Besides, Math Optimizer (M_{OP}) is described as follows.

$$M_{OP}(t) = 1 - \frac{t^{\frac{1}{\alpha}}}{M_t^{\frac{1}{\alpha}}}, \tag{12}$$

where $\alpha = 5$ determines the dynamic variables that resolve the accuracy of the exploitation phase during iteration.

Furthermore, subtraction (D) and addition (A) operators are used to implement the AOA exploitation phase [23].

$$x_{i,j}(t+1) = \begin{cases} X_{bj} - M_{OP} \times \left(\left(UB_j - LB_j\right) \times \mu + LB_j\right), r_3 < 0.5, \\ X_{bj} + M_{OP} \times \left(\left(UB_j - LB_j\right) \times \mu + LB_j\right), otherwise. \end{cases} \tag{13}$$

In Eq (13), r_3 shows the random value, produced within [0,1]. Next, the solution upgrade technique is implemented by the AOA operators. Briefly, Algorithm 1 shows the key steps involved in the AOA.

Algorithm 1. Steps included in the AOA

Input: The parameters of AOA such as the entire number of iterations M_t, dynamic exploitation parameter (α), and control function of solutions (N).
Generate the primary value of solution $X_i i = 1, \ldots N/\cdot$
While $(t < M_t)$ do
Compute the FF to each solution.
Determine the optimal solution X_b.
Upgrade the MOA and M_{OP} using Eq. (8) & (10)
for $i = 1$ to N do
for $j = 1$ to Dim do
Upgrade the value of r_1, r_2, and r_3.
If $r_1 > MOA$ then
Exploration phase
Using Eq. (8) to update the X_i.
Else
Exploitation phase
Using Eq. (10) to update the X_i.
End if
End for
End for

$$t = t + 1$$

End while
Output the better solution (feature subset) (X_b).

The AOA method derives the FF to accomplish high efficiency of classification. It defines a positive integer to signify the best outcomes of the solution candidate. The decline of the classification error rate is assumed to be the FF.

$$fitness(x_i) = ClassifierErrorRate(x_i) = \frac{No.of\ misclassified\ samples}{Total\ No.of\ samples} * 100 \tag{14}$$

4. Results and discussion

The damage detection performance of the RCBDD-AOADFF technique was validated using the defect dataset, comprising 10,850 samples under six classes, as exemplified in Table 1.

Table 1. Description of the dataset.

Class	No. of Samples
No Defect (Background)	2506
Corrosion (Stains)	1599
Crack	2507
Efflorescence	833
Exposed Bars	1507
Spallation	1898
Total Samples	**10850**

The confusion matrices, generated by the RCBDD-AOADFF method, are illustrated in Figure 3. The results indicate that the RCBDD-AOADFF technique properly recognized all the six types of damages.

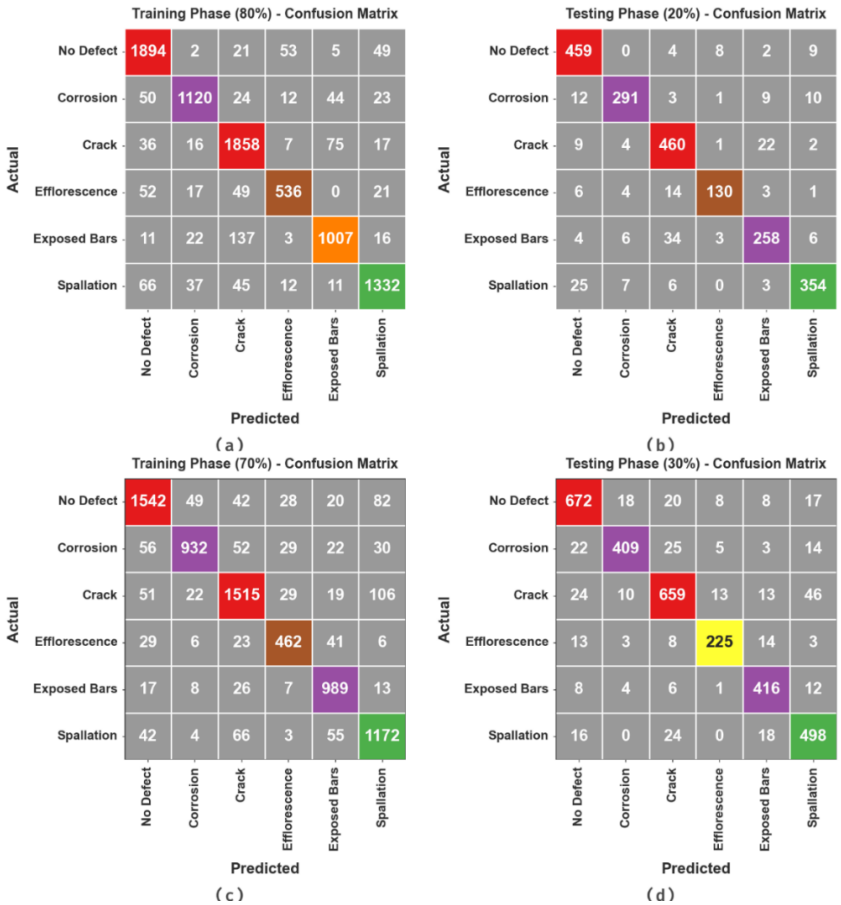

Figure 3. Confusion matrices of (a-b) 80:20 of TR set/TS set and (c-d) 70:30 of TR set/TS set.

In Table 2 and Figure 4, the damage detection outcomes of the RCBDD-AOADFF technique upon 80:20 of the TR set/TS set are portrayed. The experimental values infer that the RCBDD-AOADFF technique attained improved outcomes under all the measures. With 80% of the TR set, the RCBDD-AOADFF technique attained the average $accu_y$, $prec_n$, $reca_l$, and F_{score} values such as 96.42%, 89.12%, 87.71%, and 88.35% respectively. In addition to this, with 20% of the TS set, the RCBDD-AOADFF method accomplished the average $accu_y$, $prec_n$, $reca_l$, and F_{score} values such as 96.65%, 90.19%, 88.62%, and 89.33% correspondingly.

Table 2. Damage detection outcomes of the RCBDD-AOADFF method on 80:20 of TR set/TS set.

Class	$Accu_y$	$Prec_n$	$Reca_l$	F_{Score}
TR set (80%)				
No Defect	96.03	89.81	93.58	91.65
Corrosion	97.15	92.26	87.98	90.07
Crack	95.08	87.07	92.48	89.69
Efflorescence	97.40	86.04	79.41	82.59
Exposed Bars	96.27	88.18	84.20	86.14
Spallation	96.58	91.36	88.62	89.97
Average	**96.42**	**89.12**	**87.71**	**88.35**
TS set (20%)				
No Defect	96.36	89.13	95.23	92.08
Corrosion	97.42	93.27	89.26	91.22
Crack	95.44	88.29	92.37	90.28
Efflorescence	98.11	90.91	82.28	86.38
Exposed Bars	95.76	86.87	82.96	84.87
Spallation	96.82	92.67	89.62	91.12
Average	**96.65**	**90.19**	**88.62**	**89.33**

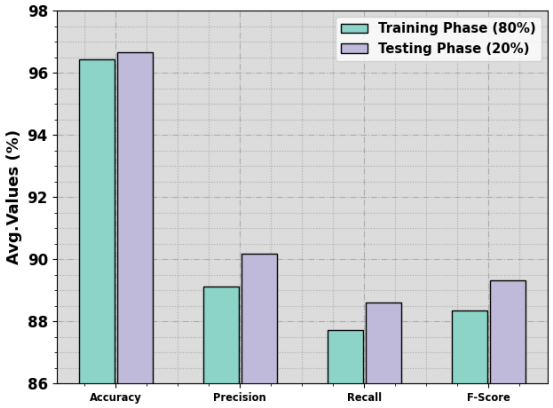

Figure. 4. Average outcomes of the RCBDD-AOADFF algorithm on 80:20 of TR set/TS set.

In Table 3 and Figure 5, the damage detection outcomes of the RCBDD-AOADFF technique upon 70:30 of the TR set/TS set are shown. The experimental values indicate that the RCBDD-AOADFF technique attained better outcomes under all the measures.

Table 3. Damage detection outcomes of the RCBDD-AOADFF algorithm on 70:30 of TR set/TS set.

Class	$Accu_y$	$Prec_n$	$Reca_l$	F_{Score}
TR set (70%)				
No Defect	94.52	88.77	87.46	88.11
Corrosion	96.34	91.28	83.14	87.02
Crack	94.26	87.88	86.97	87.42
Efflorescence	97.35	82.80	81.48	82.13
Exposed Bars	97.00	86.30	93.30	89.66
Spallation	94.64	83.18	87.33	85.21
Average	**95.69**	**86.70**	**86.61**	**86.59**
TS set (30%)				
No Defect	95.27	89.01	90.44	89.72
Corrosion	96.80	92.12	85.56	88.72
Crack	94.19	88.81	86.14	87.46
Efflorescence	97.91	89.29	84.59	86.87
Exposed Bars	97.33	88.14	93.06	90.53
Spallation	95.39	84.41	89.57	86.91
Average	**96.15**	**88.63**	**88.23**	**88.37**

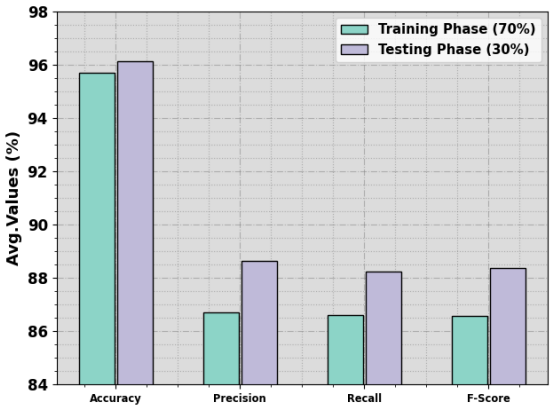

Figure 5. Average outcomes of the RCBDD-AOADFF algorithm on 70:30 of TR set/TS set.

With 70% of the TR set, the RCBDD-AOADFF method yielded the average $accu_y$, $prec_n$, $reca_l$ and F_{score} values such as 95.69%, 86.70%, 86.61%, and 86.59% correspondingly. Furthermore, with 30% of the TS set, the RCBDD-AOADFF method provided the average $accu_y$, $prec_n$, $reca_l$ and F_{score} values such as 96.15%, 88.63%, 88.23%, and 88.37% correspondingly.

Figure 6 shows the training accuracy TR_accu_y and VL_accu_y values of the RCBDD-AOADFF technique on 80:20 TR set/TS set. The TL_accu_y is calculated by evaluating the RCBDD-AOADFF technique on TR dataset, whereas the VL_accu_y is computed by evaluating the performance of the model using a separate testing dataset. The outcomes exhibit that the TR_accu_y and VL_accu_y values increased with an upsurge in the number of epochs. As a result, the performance of the RCBDD-AOADFF technique got improved on both TR and TS datasets with an increase in the number of epochs.

Figure 6. $Accu_y$ curve of the RCBDD-AOADFF algorithm on 80:20 of TR set/TS set.

In Figure 7, the TR_loss and VR_loss outcomes of the RCBDD-AOADFF method on 80:20 TR set/TS set are shown. The TR_loss corresponds to the error between the predictive performance and original values of the TR data. The VR_loss is a measure for analyzing the performance of the RCBDD-AOADFF technique on individual validation data. The results indicate that both TR_loss and VR_loss values tend to minimize with rising epochs. The outcomes portrayed the enhanced performance of the RCBDD-AOADFF technique and its capability to achieve accurate classification. The reduced TR_loss and VR_loss values demonstrate the enhanced performance of the RCBDD-AOADFF method in capturing the patterns and relationships.

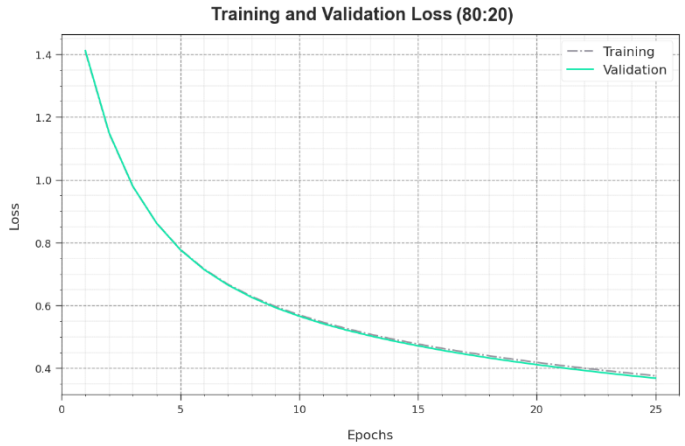

Figure 7. Loss curve of the RCBDD-AOADFF algorithm on 80:20 of TR set/TS set.

A brief Precision-Recall (PR) curve of the RCBDD-AOADFF system upon 80:20 of the TR set/TS set is illustrated in Figure 8. The results imply that the RCBDD-AOADFF method produced increasing PR values. Furthermore, the RCBDD-AOADFF method attained the maximum PR values on all the classes.

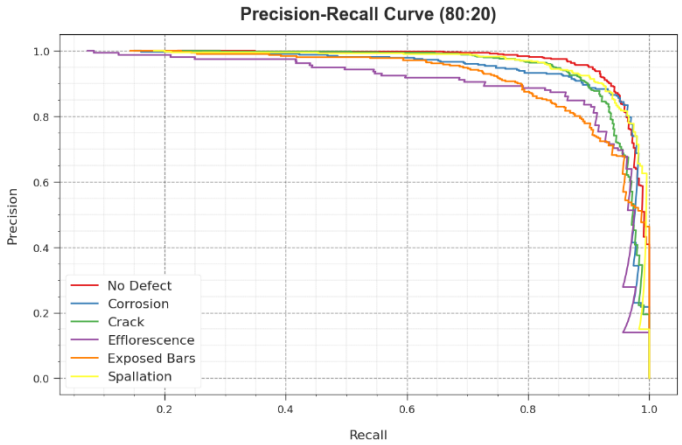

Figure 8. PR curve of the RCBDD-AOADFF method on 80:20 of TR set/TS set.

In Figure 9, the ROC curve for the RCBDD-AOADFF method is shown upon 80:20 TR set/TS set. The figure describes that the RCBDD-AOADFF technique achieved better ROC values. In addition to this, the RCBDD-AOADFF technique obtained better ROC values on all the classes.

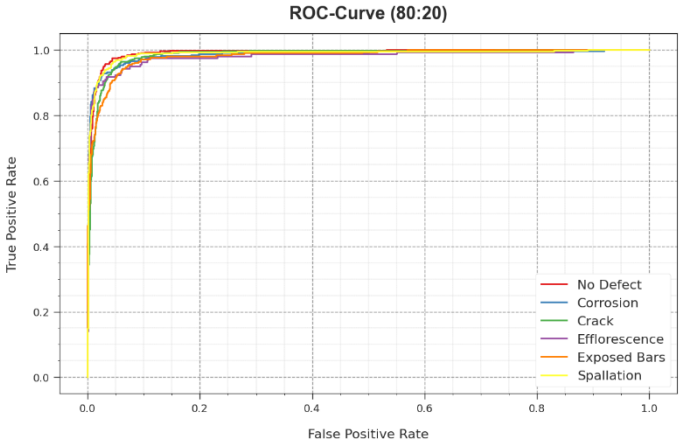

Figure 9. ROC of RCBDD-AOADFF algorithm on 80:20 of TR set/TS set.

In Table 4 and Figure 10, the comparative analysis outcomes between the RCBDD-AOADFF technique and other DL models are shown [24]. The results indicate that the RCBDD-AOADFF technique achieved enhanced performance. In terms of $accu_y$, the RCBDD-AOADFF method achieved the maximum $accu_y$ of 96.65% while the MetaQNN, ENAS, VGG16, Inception-v3, ResNet50, and Xception models obtained low $accu_y$ values such as 72.19%, 70.78%, 88.00%, 89.00%, 90.00%, and 94.95% respectively. With regards to $prec_n$, the RCBDD-AOADFF method accomplished the maximum $prec_n$ of 90.19% while the MetaQNN, ENAS, VGG16, Inception-v3, ResNet50, and Xception techniques attained low $prec_n$ values such as 84.23%, 86.09%, 86.75%, 89.6%, 81.83%, and 89.88% correspondingly. Finally, in terms of $reca_l$, the RCBDD-AOADFF technique accomplished the maximum $reca_l$1 of 88.62% while the MetaQNN, ENAS, VGG16, Inception-v3, ResNet50, and Xception methods attained low $reca_l$ values such as 83.05%, 84.58%, 81.25%, 84.06%, 85.41%, and 88.62% correspondingly.

Table 4. Comparative analysis outcomes of the RCBDD-AOADFF algorithm and other DL techniques [24].

Model	$Accu_y$	$Prec_n$	$Reca_l$
MetaQNN	72.19	84.23	83.05
ENAS	70.78	86.09	84.58
VGG16	88.00	86.75	81.25
Inception-V3	89.00	89.6	84.06
ResNet50	90.00	81.83	85.41
Xception	94.95	89.88	86.04
RCBDD-AOADFF	96.65	90.19	88.62

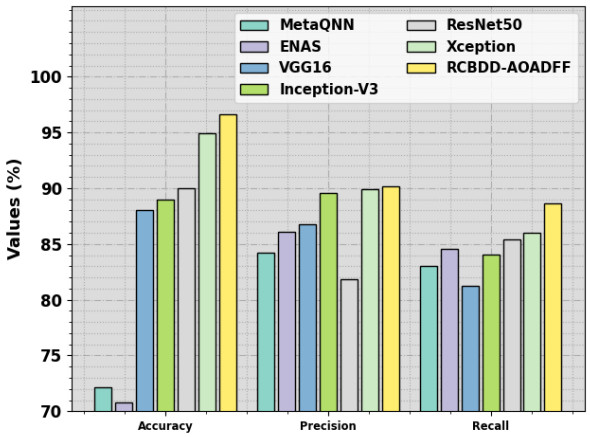

Figure 10. Comparative analysis outcomes of the RCBDD-AOADFF algorithm and other DL systems.

In Table 5 and Figure 11, the Computation Time (CT) results of the RCBDD-AOADFF technique are provided. The obtained values demonstrate that the RCBDD-AOADFF technique took the least CT of 1.53s. On the other hand, the metaQNN, ENAS, VGG16, Inception-v3, ResNet50, and Xception models consumed more CT values such as 4.53s, 2.67s, 2.92s, 2.72s, 4.20s, and 2.92s respectively. Thus, the RCBDD-AOADFF technique established its superior performance in damage detection process.

Table 5. CT outcomes of the RCBDD-AOADFF algorithm and other DL systems.

Model	Computational Time (sec)
MetaQNN	4.53
ENAS	2.67
VGG16	2.92
Inception-V3	2.72
ResNet50	4.20
Xception	2.92
RCBDD-AOADFF	1.53

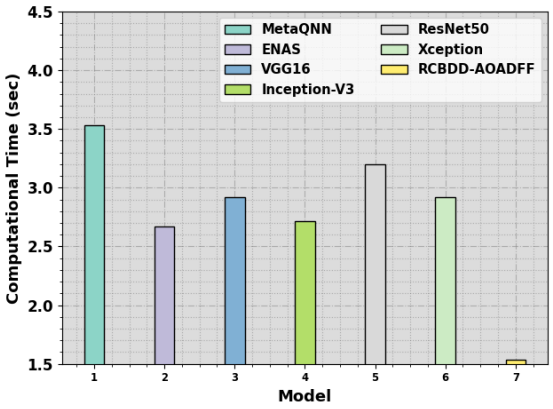

Figure 11. CT outcomes of the RCBDD-AOADFF algorithm and other DL systems.

5. Conclusions

In the current study, a new RCBDD-AOADFF technique has been introduced and validated for automated and accurate RC bridge damage detection. The major aim of the RCBDD-AOADFF technique is to detect and classify different kinds of defects in RC bridges. In the presented RCBDD-AOADFF technique, three major processes are involved namely deep feature fusion, ALSTM classification, and AOA-based hyperparameter tuning. For the feature fusion process, both Darknet-19 and Nasnet-Mobile models are used. In addition to this, the AOA with ALSTM model is utilized for the damage classification process. To enhance the classification results of the ALSTM model, the AOA is applied to the hyperparameter selection process. The performance of the RCBDD-AOADFF technique was validated using the RC bridge damage dataset. The extensive analyses outcomes establish the potential outcomes of the RCBDD-AOADFF technique upon RC bridge damage detection.

Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

The authors extend their appreciation to the Deanship of Scientific Research at King Khalid University for funding this work through a large group Research Project under grant number (RGP2/29/44). Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2023R387), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia. Research Supporting Project number (RSPD2023R521), King Saud University, Riyadh, Saudi Arabia. This study is supported via funding from Prince Sattam bin Abdulaziz University project number (PSAU/2023/R/1444).

Conflict of interest

The authors declare that they have no conflict of interest. The manuscript was written through the contributions of all authors. All authors have approved the final version of the manuscript.

References

1. L. Chen, W. Chen, L. Wang, C, Zhai, X. Hu, L. Sun, et al., Convolutional neural networks (CNNs)-based multi-category damage detection and recognition of high-speed rail (HSR) reinforced concrete (RC) bridges using test images, *Eng. Struct.*, **276** (2023), 115306. https://doi.org/10.1016/j.engstruct.2022.115306
2. X. W. Ye, S. Y. Ma, Z. X. Liu, Y. Ding, Z. X. Li, T. Jin, Post-earthquake damage recognition and condition assessment of bridges using UAV integrated with deep learning approach, *Struct. Control Hlth.*, **29** (2022), e3128. https://doi.org/10.1002/stc.3128
3. P. Santaniello, P. Russo, Bridge damage identification using deep neural networks on time–frequency signals representation, *Sensors*, **23** (2013), 6152. https://doi.org/10.3390/s23136152
4. P. Kumar, A. Sharma, S. R. Kota, Automatic multiclass instance segmentation of concrete damage using deep learning model, *IEEE Access*, **9** (2021), 90330–90345. https://doi.org/10.1109/ACCESS.2021.3090961
5. X. Yin, Z. Huang, Y. Liu, Bridge damage identification under the moving vehicle loads based on the method of physics-guided deep neural networks, *Mech. Syst. Signal Pr.*, **190** (2023), 110123. https://doi.org/10.1016/j.ymssp.2023.110123
6. Y. Jiang, D. Pang, C. Li, A deep learning approach for fast detection and classification of concrete damage, *Automat. Constr.*, **128** (2021), 103785. https://doi.org/10.1016/j.autcon.2021.103785
7. Y. Ni, J. Mao, Y. Fu, H. Wang, H. Zong, K. Luo, Damage detection and localization of bridge deck pavement based on deep learning, *Sensors*, **23** (2023), 5138. https://doi.org/10.3390/s23115138
8. H. X. He, J. C. Zheng, L. C. Liao, Y. J. Chen, Damage identification based on convolutional neural network and recurrence graph for beam bridge, *Struct. Health Monit.*, **20** (2021), 1392–1408. https://doi.org/10.1177/1475921720916928
9. B. Huang, S. Zhao, F. Kang, Image-based automatic multiple-damage detection of concrete dams using region-based convolutional neural networks, *J. Civil Struct. Health Monit.*, **13** (2023), 413–429.
10. A. Chehri, A. Saeidi, IoT and deep learning solutions for automated crack detection for the inspection of concrete bridge structures. In: *International Conference on Human-Centered Intelligent Systems*, 2021, Singapore: Springer. https://doi.org/10.1007/978-981-16-3264-8_11
11. Z. Bai, T. Liu, D. Zou, M. Zhang, A. Zhou, Y. Li, Image-based reinforced concrete component mechanical damage recognition and structural safety rapid assessment using deep learning with frequency information, *Automat. Constr.*, **150** (2023), 104839. https://doi.org/10.1016/j.autcon.2023.104839
12. B. Kim, S. Cho, Automated multiple concrete damage detection using instance segmentation deep learning model, *Appl. Sci.*, **10** (2020), 8008. https://doi.org/10.3390/app10228008

13. H. Wan, L. Gao, Z. Yuan, H. Qu, Q. Sun, H. Cheng, R. Wang, A novel transformer model for surface damage detection and cognition of concrete bridges, *Expert Syst. Appl.*, **213** (2023), 119019. https://doi.org/10.1016/j.eswa.2022.119019
14. D. Ai, F. Mo, J. Cheng, L. Du, Deep learning of electromechanical impedance for concrete structural damage identification using 1-D convolutional neural networks, *Constr. Build. Mater.*, **385** (2023), 131423. https://doi.org/10.1016/j.conbuildmat.2023.131423
15. Q. Zhang, K. Barri, S. K. Babanajad, A. H. Alavi, Real-time detection of cracks on concrete bridge decks using deep learning in the frequency domain, *Engineering*, **7** (2021), 1786–1796. https://doi.org/10.1016/j.eng.2020.07.026
16. D. Zou, M. Zhang, Z. Bai, T. Liu, A. Zhou, X. Wang, et al., Multicategory damage detection and safety assessment of post-earthquake reinforced concrete structures using deep learning, *Comput.-Aided Civ. Inf.*, **37** (2022), 1188–1204. https://doi.org/10.1111/mice.12815
17. H. Sun, L. Song, Z. Yu, A deep learning-based bridge damage detection and localization method, *Mech. Syst. Signal Pr.*, **193** (2023), 110277. https://doi.org/10.1016/j.ymssp.2023.110277
18. P. Kumar, S. Batchu, S. R. Kota, Real-time concrete damage detection using deep learning for high-rise structures, *IEEE Access*, **9** (2021), 112312–112331. https://doi.org/10.1109/ACCESS.2021.3102647
19. S. Khan, M. A. Khan, J. H. Shah, F. Shehzad, T. Kim, J. H. Cha, Suspicious activities recognition in video sequences using DarkNet-NasNet optimal deep features, *Comput. Syst. Sci. Eng.*, 2023. https://doi.org/10.32604/csse.2023.040410
20. A. Al Hamoud, A. Hoenig, K. Roy, Sentence subjectivity analysis of a political and ideological debate dataset using LSTM and BiLSTM with attention and GRU models, *J. King Saud Univ.-Comput. Inform. Sci.*, **34** (2022), 7974–7987. https://doi.org/10.1016/j.jksuci.2022.07.014
21. L. Abualigah, A. Diabat, S. Mirjalili, M. Abd Elaziz, A. H. Gandomi, The arithmetic optimization algorithm, *Comput. Method. Appl. Mech. Eng.*, 376 (2021), 113609. https://doi.org/10.1016/j.cma.2020.113609
22. H. Liu, X. Zhang, H. Zhang, C. Li, Z. Chen, A reinforcement learning-based hybrid Aquila optimizer and improved arithmetic optimization algorithm for global optimization, *Expert Syst. Appl.*, **224** (2023), 119898. https://doi.org/10.1016/j.eswa.2023.119898
23. X. Shi, X. Yu, M. Esmaeili-Falak, Improved arithmetic optimization algorithm and its application to carbon fibre-reinforced polymer-steel bond strength estimation, *Compos. Struct.*, **306** (2023), 116599. https://doi.org/10.1016/j.compstruct.2022.116599
24. M. Abubakr, M. Rady, K. Badran, S. Y. Mahfouz, Application of deep learning in damage classification of reinforced concrete bridges, *Ain Shams Eng. J.*, 2023, 102297. https://doi.org/10.1016/j.asej.2023.102297

AIMS Press