International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING

ISSN:2147-6799 www.ijisae.org Original Research Paper

A Fully Convolutional Neural Network Model Towards Internet of Things-Enabled Crack Detection in useful Structure: An Application to Structural Health Monitoring

Surajit Mohanty[1*], Subhendu Kumar Pani[2]

Submitted: 25/02/2024 **Revised**: 29/04/2024 **Accepted**: 15/05/2024

Abstract:

A highly advanced fully convolutional neural network (CNN) model is methodically proposed to classify bridge cracks. This paper explored Python libraries to create a simulation and training platform for the model. The proposed approach is observed to be an outstanding model for identifying bridge cracks effectively having comparatively less complex training with accuracy rates well over 90 percent and it is 82 percent efficient than the other compared approach. Here, intelligent detection methods have been proposed to optimize the bridge safety efficacy mitigating the associated risk factors. Moreover, in this study the significant impact of integrating IoT technology in structural health monitoring, especially in bridge crack detection has been highlighted.

Keywords: *Deep Learning; Structural Health Monitoring; Bridge Crack Detection; Image Classification;*

1. Introduction

The widespread availability, widespread application and many benefits of IoT have been facilitated by the development of information technology. These advances have particularly influenced the field of civil engineering, promoting the development of intelligent, complex and interconnected structures. Key considerations include crack detection, as previous studies show that bridge tragedies cause safety problems caused by cracks. Cracking can be caused by a number of hazards, such as mining. This is particularly worrying given these data. The maximum allowable width of 0.3 mm is set as the critical limit value for bridge cracks. Exceeding this limit can jeopardize the stability of the structure and lead to catastrophic collapse accidents. This study explores the potential of IoT to monitor bridge structures and accurately identify potential cracks. The goal is to provide valuable insights that can help structural disaster mitigation programs. The research begins with careful preprocessing of images of bridge cracks to ensure data quality. The intricate nature of critical structural frameworks like bridges and their process of construction, many risk factors in bridge projects are unavoidable. This is because of the continual expansion in the density of highway networks and the advent of large-span bridges in China's transportation industry [1]. There is a possibility that these risk factors could result in bad impacts and perhaps the collapse of bridges, which would threaten both people's lives and their property. As a result, a matter of utmost importance is to conduct damage detection and generate warning for bridge structures in order to swiftly monitor their current state of health. With the large number of highway bridges in China, many of which exhibit structural defects and varied degrees of damage [2], the diagnosis of modern bridges' health has been a focal point of research for both academia and engineering. This is especially true when taking into consideration the fact that the majority of these bridges are in China. Despite the fact that the significance of health diagnosis has been acknowledged since the 1950s, the research of the health diagnosis of bridge structures has received a greater amount of attention and urgency in recent decades. Particular attention has been paid to the development of efficient testing methods, damage control systems, and safety inspection systems [3].

1.Research Scholar, Department of Computer Science and Engineering, BijuPatnaik University of Technology, Rourkela, India, mohanty.surajit@gmail.com, ORCID Number: 0000-0002-7909-3478

2. Professor, Krupajal Computer Academy Odisha, Bhubaneswar, India, skpani.utkal@gmail.com, ORCID Number: 0000-0001-9595-0567

Corresponding author:mohanty.surajit@gmail.com

The existing approaches for identifying damage to structures can be roughly classified into two groups, depending on research and application perspectives: Static force-based approaches and dynamic-based methods [4]. Both of these groups are based on the characteristics of the structures themselves. These methods, on the other hand, frequently run into difficulties in real applications due to fluctuations in the loads that structures endure when they are in service and restrictions in algorithms. The implementation of Internet of Things (IoT) technology in the monitoring of structural security has become increasingly common due to advancements in intelligent identification terminal technology [5]. Through the use of intelligent terminals, Internet of Things technology makes it possible to do real-time monitoring of building sites. This makes it easier to process and analyze information in order to provide early warning signals for security purposes. Automatic identification and tracking, data transmission across networks, computerized alerts, digitalization, information construction, and complete real-time monitoring of the safety state of structures are all made possible by the Internet of Things (IoT)[6-8].

In recent years, the Internet of Things has emerged as the industry that is growing at the fastest rate. Plans that have been developed by the government indicate that there will be increasing investment in IoT technology in the years to come [9,10]. It is anticipated that innovations in connected industries would be brought about by the improvement of Internet of Things technology. As a result, the active study of Internet of Things technology applications in numerous industries, including engineering quality inspection, is gaining attention. This is because it improves the accuracy and the efficiency of the inspection operations, which represents a future development trend[11].

The rest part of the paper is organized as section 2 gives some related research in the domain, section 3 briefly describe the background of IoT in structural monitoring, section 4 illustrate the proposed framework, section 5 contain the result, and discussion, and finally section 6 highlights the conclusion of the research work.

2. Related Work

Zhang et. al [12] studied that Machine Learning (ML) based methods have been increasingly utilized in various stages of the entire life cycle in recent years .Fernández-Gómezet.al [13] said Deep Learning is widely acknowledged as a strategy that is consistently utilized, particularly in complex conditions that necessitate a substantial volume of data. Deep learning has gained significant importance in the construction sector, leading to several research that offer valuable insights into its various forms and applications. Panet.al[14] conducted a study that provided a summary of various architectures used in deep learning and their respective applications. The research analyzed seven traditional designs, namely Deep Neural Network, Convolution Neural Network, Recurrent Neural Network, Auto-encoder, Restricted Boltzmann Machine, Deep Belief Network, and Generative Adversarial Networks.

Among the various standard deep learning architectures that have been evaluated, the Convolutional Neural Network (CNN) stands out for its capacity to process images, particularly for tasks that involve matching images based on their width, height, and depth studied by Krizhevsky et.al [15]. The study by author Tabernik et.al [16] utilized an advanced deep learning network to identify cracks, resulting in enhanced precision in crack identification. Zhang et.al [17] conducted a study where they utilized a deep architecture of Convolutional neural networks (CNNs) to build a vision-based system for identifying concrete fractures. This method does not require the calculation of fault features. The CNN model is trained using a dataset of 40,000 photos with a size of 256 × 256 pixels. Its purpose is to identify fractures by categorizing individual regions individually.

Jeonget.al [18] applied pre-processing to the picture using a Naive Bayes ML classifier, and then identified fractures using a CNN. Mishraet.al [19] used a Deep Convolution Neural Network (DCNN) that is trained using the comprehensive ImageNet database, which contains a vast number of pictures. They applied this training to identify cracks in Hot-Mix Asphalt (HMA) and Portland Cement Concrete (PCC) surfaced pavement images, which also encompassed various non-crack abnormalities and flaws.Liu et.al [20] introduced an innovative network for semantic segmentation that utilizes deep convolution layers and is capable of incorporating contextual information for accurately identifying structural infrastructure cracks across different scenarios. The suggested approach utilizes a deep semantic segmentation network to accurately separate fractures in images of any size, without the need to retrain the network for future predictions. Additionally, a fusion technique that takes into account the context and utilizes local restrictions across different states and spaces is introduced. This approach aims to combine the predictions of picture patches. Dardouri et.al [21] utilized the U-Net model for the purpose of identifying concrete cracks. The evaluation function chosen is the focal loss function, and the optimization is performed using the Adam method. The trained U-Net has the capability to accurately detect the locations of cracks in the input raw photos, even under challenging settings such as varying illumination, cluttered backgrounds, and different crack widths. It demonstrates excellent efficacy and resilience in this task.

2.2. Contribution

In this work the major contributions are

(i) A fully convolutional neural network model has been proposed to classify bridge cracks.
(ii) Estimated the convolutional layer, gradient for down sampling.
(iii) Extensive result analysis has been carried out on enhanced imagery obtained after applying CNN.

3. Background of IoT for Structural Health Monitoring

Professor Kevin Ashton of the Massachusetts Institute of Technology is the first person to propose the idea of the Internet of Things in 1991. In 1999, the Center for Automatic Identification at the Massachusetts Institute of Technology (MIT) created a precise description of the concept behind the Internet of Things. This definition includes the transmission of all objects by radio frequency for intelligent detection and control. Devices that transmit information, such as RFID devices, infrared sensors, global positioning systems, and laser scanners, have been integrated into the Internet of Things (IoT) during its development. These devices can sense, compute and act, and together they form a massive network that connects everything to facilitate information transfer, coordination and processing in real time. With the ability of the Internet of Things to connect the digital and physical worlds, the networking of "things and things" can be achieved. This is achieved by extending internet objects from people to all objects. The structural foundation of the Internet of Things comprises four tiers: the perception layer, the network layer, the processing layer, and the application layer [7].

The perception layer, or the physical interface layer of the Internet of Things, implements the detection, recognition, tracking and data collection of objects using various sensors. The network layer brings together several different communication networks and the Internet, which are responsible for facilitating data transfer between the sensor layer and the processing layer. The processing layer, which is an intelligent processing layer, is responsible for the intelligent processing of huge amounts of data using units such as the control center of the application layer, which is built using the perception layer, the network layer and the processing layer. . , applications can achieve intelligence across the entire IoT ecosystem with a resource center, cloud computing platform and expert system. This layer also helps to facilitate the profound incorporation of information technology into a variety of different industries.

The widespread use of different identification technologies and the construction of a network that spreads across the Internet are two ways in which IoT technology differs from the traditional Internet. There are many sensors used in the Internet of Things, and they are all sources of information. These sensors record a wide variety of data content and formats and update their data in real time and regularly. The integration of wired and wireless networks forms the basis of the Internet of Things, which is based on the infrastructure of the Internet. This ensures that information about targets is sent in real time and accurately. In addition, the Internet of Things goes beyond sensor connections, as it has the ability to intelligently control objects. This is accomplished by integrating sensors with cognitive processing and utilizing In order to broaden its application domains, the IoT incorporates technologies such as cloud computing and pattern recognition

4. Proposed Fully Convolutional Neural Network Model towards Internet of Things-enabled Crack Detection

Figure 1 depicts a systematic workflow of the suggested framework, highlighting the many stages involved in the fully Convolutional Neural Network (CNN) architecture designed for detecting cracks in structural elements. Figure 2 describe the details of the proposed IoT based CNN structure for identifying crack. The several stages have been examined as follows:

1. Network Initialization: Begins with the initialization of the Convolutional Neural Network (CNN), which involves establishing the structure of the network and configuring the necessary parameters for the following phases.

2. Determining Convolution Layers: This step involves strategically deciding on the number of convolutional layers, which is a crucial factor that affects the network's capability to identify complex characteristics in the input data.

3. Selecting Functions and Parameters: The process of function and parameter selection involves carefully choosing activation functions and fine-tuning parameters in the CNN architecture to achieve optimal information extraction and feature representation.

4. Network Training: The CNN advances to the training phase, during which it learns and adjusts its parameters by being exposed to a dataset of crack images. This process improves its capacity to identify patterns that indicate structural deterioration.

5. Network Testing: Afterwards, the trained network is subjected to thorough testing using a distinct dataset to assess its performance and extend its ability to detect cracks in a broader context.

6. Result Condition Check: After a thorough examination, the findings are examined to assess the adequacy of the network's crack detection.

7. Decision Path: If the results satisfy predetermined criteria, the process advances to the "Mine Phase Diagram Recognition," indicating successful identification of cracks and further analysis. If the outcomes are not satisfactory, the workflow proceeds to the "Convolution Layer Adjustment Phase," which requires revisiting the design for more improvement.

8. Phase Diagram Recognition of Mine: Upon successful detection of cracks, this step entails obtaining useful insights and phase diagrams pertaining to the structural conditions, facilitating a thorough investigation.

9. Convolution Layer Adjustment Phase: If the results are not satisfactory, this phase comprises making changes to the convolutional layers, fine-tuning parameters, and iteratively enhancing the network architecture.

10. Exit: Marks the end of the workflow, indicating the successful completion of the CNN-based crack detection process.

This workflow guarantees a systematic way for fracture identification using CNNs, which includes iterative modifications and result verification to ensure reliable and precise assessments of structural health.

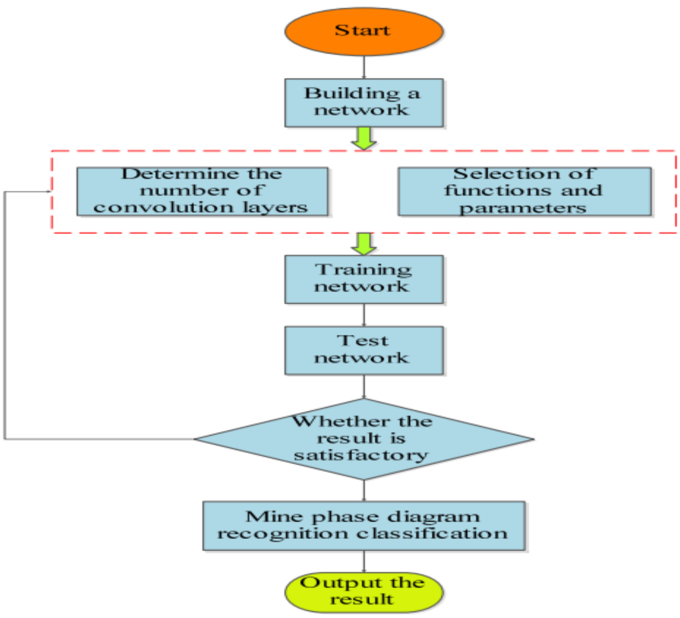

Fig 1: Workflow of the proposed fully CNN framework.

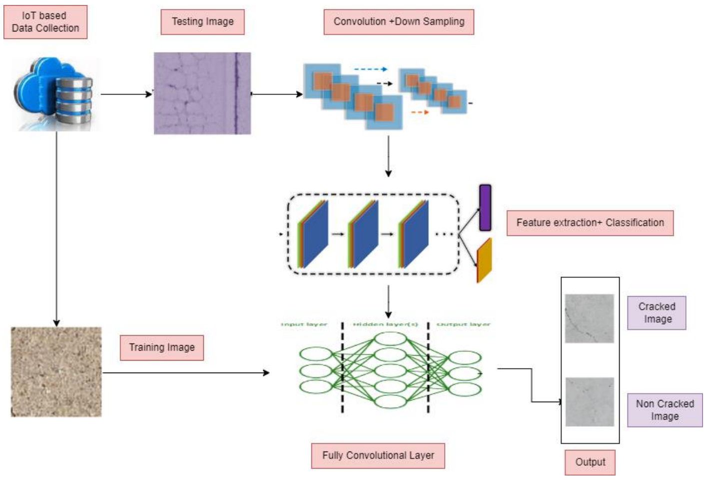

Fig 2: Details architecture of proposed CNN based IoT enabled crack detection

4.1 Construction of CNN Framework

The CNN is a ground-breaking research outcome that is predicated on the notions of artificial neural networks and is influenced by the principles of biological neuroscience. Due to the fact that it is capable of deep learning, it has become an important concentrate of research in the field of machine learning. CNN has broad adaptability, concurrent feature extraction and classification, excellent generalization, and global training parameter optimization, in compared to older approaches because of its capacity to perform all of these tasks simultaneously. The local sensation field is formed when the convolution neural network is used to the task of image categorization. This is accomplished by the set of tiny neurons of the network connecting with a particular region of the input image. This field improves the representation of the original image by utilizing groups of flat spreads that overlap one another in order to gain a more accurate representation of the image. It is because of this process that the network is able to accept distortions in the input image. This process is repeated throughout each layer.

A sub sampling layer that is referred to as the pool layer is also incorporated into the convolution neural network. This layer is designed to integrate the output of the neuron cluster through the utilization of a sub sampling approach. CNN uses the convolution operation with weight sharing in the convolution layer in order to overcome the issue of dealing with billions of parameters in layers that are fully connected. By recombining the same weights in each pixel of the layer, this method is able to effectively reduce the amount of memory that is required while also improving performance. When it comes to image identification or classification tasks, where the output 'tile' of neurons may be readily timed for picture analysis, certain time-delay neural networks adopt architectures that are comparable to those described above.

When compared to other algorithms for image classification, convolution neural networks require a comparatively little amount of preparation. This can be attributable to the fact that they are utilized in filter research, whereas traditional algorithms concentrate a large emphasis on the design of manual features. The independence of convolution neural networks from prior information and the relief of the obstacles associated with manual feature creation are the primary advantages that convolution neural networks have over traditional techniques with regard to their use.

4.2. Estimation of Convolutional Layer

Within the convolution layer, a learning convolution kernel is utilized to carry out a convolution operation on the feature map that is generated by the previously mentioned layer. Following that, the outcome are processed by an activation function, which finally results in the formation of a new feature map as the output. The output feature maps that are generated by various convolutional kernels are distinct from one another, and each output feature map may be the result of the combined convolution of many feature maps. When taking into consideration CNN as the proposed model for crack detection in bridges, the following is a full explanation of the calculation procedure for the convolution layer.

$$F_i^l = \alpha\left(\sum_{j\in M_i} F_j^{l-1} \times X_{ij}^l + \beta_i^l\right) \quad (1)$$

where, F_i^l is the notation that is used in the equation to represent the *i* feature map of layer *l*.

In Eq.(1), X_{ij}^l is the convolution kernel function, and it is represented by the letter X. Applying the activation function $\alpha()$ and activating the sigmoid function in classic convolutional neural networks are both examples of how the activation function is used. It is noted that the bias parameter β_i^l has been integrated, and M_i represents a collection of selected input feature maps. For every output feature map, a feature map is chosen from this set to serve as the input feature map respectively. There is a bias coefficient associated with each output feature map. The convolutional cores of each input feature map are combined for a particular output-holding graph. The sign c indicates that input feature maps of output feature graphs a and b have a similar set. On the other hand, they are produced from c by means of several separate convolutional cores being added together. In the context of CNN as the proposed model for fracture detection in bridges, this explanation provides an overview of the components that make up the equation.

4.3. Estimation of Gradient

The sequential design involves a convolution layer l followed by a subsequent down sampling layer *l + 1*. Thus, to adjust the weights in the convolution layer, it is imperative to gather the error signal γ for each neuron within that layer. This is done in accordance with the BP technique that is discussed earlier. In order to accomplish this, it is necessary to obtain the error signal γ^{l+1} of the succeeding layer. This is accomplished by adding up the errors recorded by neurons in the subsequent layer. This error signal is then multiplied by the weights that correspond to it, W, additionally, the error signal γ arises from the activation function α applied to the inputμ of the neuron in the *l*-layer.

In the *l*-layer, it is possible to compute *l* for each individual neuron. The error signals of neurons in the down sampling layer following the convolution layer are determined by the size of the sample window in the output feature map of the convolution layer. This is the case in the situation described above. As a consequence of this, every neuron that is presented in the feature map of this layer is related to a single neuron that is presented in the feature map that corresponds to the *l + 1* layer. It is important to conduct a sampling operation in order to get the error signal ε for layer γ, which corresponds to the feature map of the sampling layer. This method entails matching the findings from the previous phase with the error signal $\beta_i^l$of each feature graph *j* in the convolution layer. This iterative process can be done on a number of occasions in order to compute the error signals for every feature map within the convolution layer,

$$\gamma_i^l = \varepsilon_i^{l+1}\left(\alpha'\left(u_i^l\right) * up\left(\gamma_i^{l+1}\right)\right) \quad (2)$$

When a sampling operation is performed, the bias gradient is integrated with the error signal within the layer by applying the formula that is shown earlier.

$$\frac{\partial\varepsilon}{\partial\beta_I} = \sum_{u,v}\left(\gamma_i^l\right)_{uv} \quad (3)$$

Ultimately, the weight gradient of the convolution kernel can be calculated using the conventional backpropagation (BP) algorithm. Due to the numerous connections in convolutional neural networks being associated with weighted values, a gradient must be computed for all connections linked to a particular weight, and these gradients are eventually summed.

$$\frac{\partial\varepsilon}{\partial X_{ij}^l} = \sum_{u,v}\left(\partial_i^l\right)_{uv}\left(b_j^{l-1}\right)_{uv} \quad (4)$$

In this context, b_j^{l-1}represents a small block that is multiplied by the element X_{ij}^l during the convolution operation on X_j^{l-1}, where the value at position *(u, v)* of the output convolution feature map is obtained by multiplying a small block at the upper *(u, v)* position by an element of the convolution kernel. The convolution function in Python can handle this process, removing the need to manually track which segment of the output feature map corresponds to each neuron in the input feature map, thus enabling automation without human intervention.

$$\frac{\partial \varepsilon}{\partial X_{ij}^{l}} = \nabla_{180}\left(c_2\left(F_j^{l-1}, \nabla_{180}\left(\gamma_i^{l}\right)\right)\right) \tag{5}$$

In this context, the rotation feature map involves cross-correlation instead of convolution, and the resulting output is rotated back. During forward propagation, the convolution sends samples, and the convolution kernel operates in the predetermined direction.

4.4. Estimation of Gradient for Lower Sampling Layer

The principle behind the down sampling layer is straightforward: each output feature map's size is a scaled-down version of the input feature map's size.

$$F_i^{l} = \alpha\left(\varepsilon_i^{l} down\left(F_i^{l-1}\right) + \beta_i^{l}\right) \tag{6}$$

By using the down() function to do lower sampling and setting the sample window as n×n, the feature size of the output graph is reduced by a factor of n in order to ensure scaling invariance. This is done in order to minimize the resolution on the graph. The multiplicative offset parameter ε and the additive bias parameter β belonging to each output feature graph are unique to that particular graph.

Prior to calculating the error signal of the feature map in the down sampling layer, it is necessary to establish the correspondence between the small blocks in the sensitivity feature map of the current layer and the pixels in the sensitivity feature map of the subsequent layer. This knowledge enables the application of the reverse propagation algorithm for computation [13].

The erroneous signal of the currently active sub-sampling layer is obtained by a recursive process by employing the error message from the previous layer as,

$$\gamma_i^{l} = \alpha'\left(u_i^{l}\right) * c_2\left(\gamma_i^{l}, \nabla_{180}\left(X_i^{l-1}\right)\right) \tag{7}$$

To establish a connection between the convolution function and the computations, it is required to do a 180-degree rotation of the volume kernel prior to commencing the calculation. This will enable a convolution function to be complete for the whole duration of the convolution process. Solve the problem of the convolution boundary, which refers to the missing pixel needed to complete the value of 0. Subsequently, you will get a gradient of 'a' and 'b':

$$\frac{\partial \varepsilon}{\partial \beta_i} = \sum_{u,v}\left(\gamma_i^{l}\right)_{uv} \tag{8}$$

$$\frac{\partial \varepsilon}{\partial \varepsilon_i} = \sum_{u,v}\left(\gamma_i^{l} * down\left(F_i^{l-1}\right)\right)_{uv} \tag{9}$$

It is possible to achieve the weight update in the convolution neural network from time t to $t + 1$ instant by using the following calculation form, which is quite similar to the implementation of the BP algorithm:

$$\omega(t+1) = \omega(t) + \mu\gamma(t)x(t) \tag{10}$$

Where μ represents the rate of learning, $x(t)$ represents the neural input, and $\gamma(t)$ represents the error term.

4.5 Proposed approach

Step 1: Collect the crack image data from the IoT dataset. Send the testing data to the CNN

Step 2: The convolutional kernel function generated by equation 1 will identify the feature by down sampling.

Step 3: the classified image then feed to fully CNN with a weight value W, and error signal γ.

Step 4: repeat the step 3 to compute the error signals for every feature map within the convolution layer.

Step 5: The weight gradient of the convolution kernel can be calculated using the conventional backpropagation (BP) algorithm using equation 2.

Step 6: The convolution function defined in equation 4 can handle this process, removing the need to manually track which segment of the output feature map corresponds to each neuron in the input feature map, thus enabling automation without human intervention.

Step 7: send a training data to the fully CNN and it will identify the cracked and not cracked image.

5. Results and Discussions

In this section, we present the strategies adopted over the SDNET dataset [25-28] considered in this study and the experimental outcomes performed by making use of python functions to build the simulation environment.

5.1. Enhancement of Imagery

There are two primary classifications that may be applied to the process of image enhancement: the approach used in the frequency domain and the technique of spatial domain. The first method considers the image to be a signal that is only two dimensions in size, and it continues the process of signal enhancement by using the Fourier transform in two dimensions. The application of low-pass filtering, which is to say, only the low-frequency signal via) approach is able to eliminate the noise in the diagram. On the other hand, the high pass filtering method is able to significantly improve the edge and other high-frequency signals, thereby bringing clarity to the previously blurry picture. It is possible to eliminate or reduce the amount of noise by utilizing the representative algorithm that is included in the latter space domain approach. This procedure incorporates the technique of calculating the local average value, using the median filter method (which determines the median pixel value within the local neighborhood), and utilizing other comparable techniques.

The process of image enhancement involves either adding some additional data pertaining to the original picture or transforming the data through some means. Image enhancement may encompass the selective accentuation of desired features within the image or the suppression (concealment) of undesirable features. The goal of this process is to ensure the picture and its corresponding visual reaction characteristics are identical. It is not essential to do an analysis of the reason for the decline in image quality throughout the process of image enhancement, and the image that has been processed does not necessarily approximate the image that is originally captured. The technology that enhances images can be classified into two distinct categories: those that are based on the spatial domain, and those that are based on the frequency field. These classifications are made in accordance with the many spaces that are involved in the improved processing process. The gray level of an image is determined directly by spatial algorithm processing. On the other hand, the frequency domain algorithm is an indirect enhancement technique that modifies the transformation coefficients of the image in a certain transformation domain. Both of these techniques use the frequency domain.

In order to improve the contrast of the picture, this study makes use of a piecewise linear function because:

$$g(x,y)=\begin{cases}\frac{h(x,y)}{m}, h(x,y)<x_1\\ k[h(x,y)-x_1]+\frac{x_i}{k}, x_1\le h(x,y)\le x_2\\ \frac{h(x,y)+225(k-1)}{k}, h(x,y)>x_2\end{cases} \quad (11)$$

The formula uses the notation *g(x,y)* to represent the gray value of the output point and *h(x,y)* to represent the gray value of the input point, and the turning point of the two horizontal axes is denoted by x_1, x_2. The *k* value is what defines the overall value of the formula. The gradient of the interval function that encompasses many segments of the transformation.

5.2. Experimental Outcomes

The SDNET dataset for bridge crack detection stands as an annotated image dataset meticulously designed to facilitate the training, validation, and benchmarking of artificial intelligence-based algorithms specifically tailored for identifying cracks in concrete structures [25-28]. Encompassing a vast collection of over 56,000 images, this dataset encompasses diverse instances of both cracked and non-cracked concrete bridge decks, walls, and pavements. Notably, the dataset spans the spectrum of crack widths, ranging from a slender 0.06 mm to a substantial 25 mm. To enhance the dataset's realism and challenge algorithmic capabilities, it incorporates various environmental obstructions such as shadows, surface roughness, scaling effects, edges, holes, and background debris.

For data acquisition, a total of 230 images capturing both cracked and non-cracked concrete surfaces are meticulously obtained using a high-resolution 16 MP Nikon digital camera. The dataset comprises images from distinct structural elements, including 54 bridge decks, 72 walls sourced from the Russell/Wanlass Performance Hall building on the Utah State University (USU) campus, and 104 pavements extracted from roads and sidewalks within the USU campus. Each image undergoes segmentation into 256 × 256 px sub-images, allowing for a granular analysis. The labeling of each sub-image is methodically conducted, denoting it as 'Cracked' if a crack is present or 'Non-cracked' if devoid of any observable cracks. This comprehensive dataset serves as an invaluable resource for advancing the development of concrete crack detection algorithms, particularly those rooted in deep learning convolutional neural networks, aligning with the evolving landscape of structural health monitoring research.

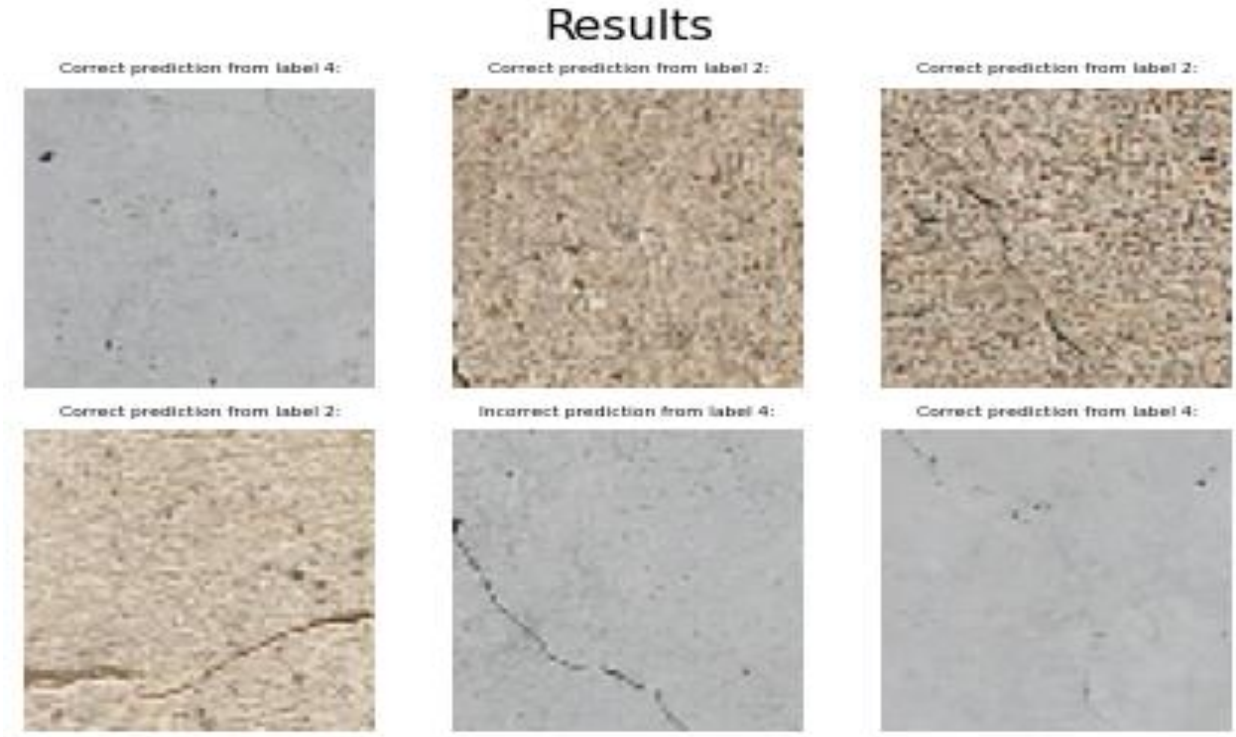

Fig 3 Visualization of cracked and un-cracked structures.

Figure 3 represents the prediction outcomes for cracks identified in the considered SDNET dataset after de-noising.

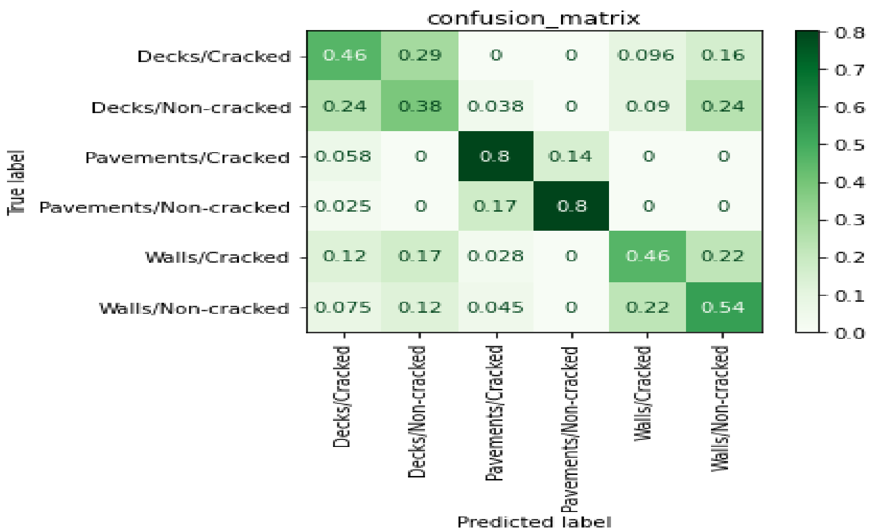

Fig 4: Confusion matrix for RF classifier.

In Figure 4, the confusion matrix for the RF approach has been presented over the considered dataset.

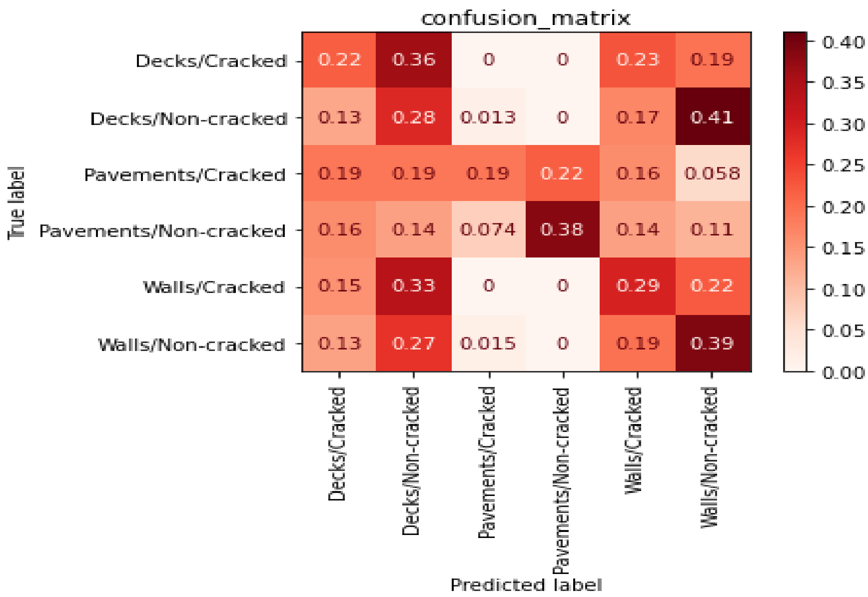

Fig 5: Confusion matrix for SVM.

In Figure 5, the confusion matrix for the SVM classifier is provided

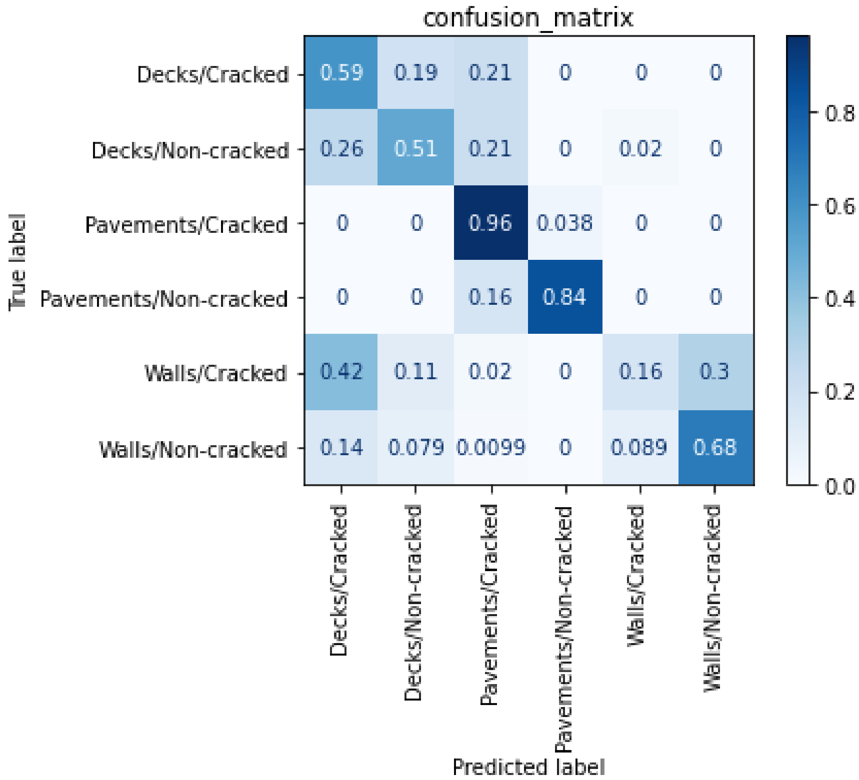

Fig 6: Confusion matrix for CNN.

In Figure 6, the Confusion matrix for the CNN is shown.

In Figure 7 class wise ROC curves for RF is shown. In Figure 8 Class-wise ROC curves for SVM is shown. In Figure 9,a Class-wise ROC curve for CNN is shown. Finally, in Figure 10, we depict the accuracy and loss value for the CNN algorithm for 500 epochs. It can be observed that the CNN shows rapid convergence in terms of accuracy after 100 epochs for both training and test instances. Further, the loss value for training and test instances also reduces considerably after 200 epochs.

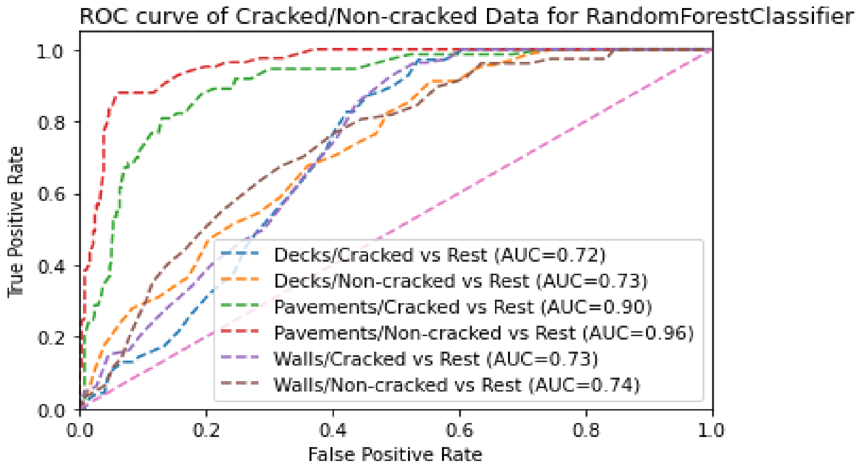

Fig 7: Class-wise ROC curves for RF.

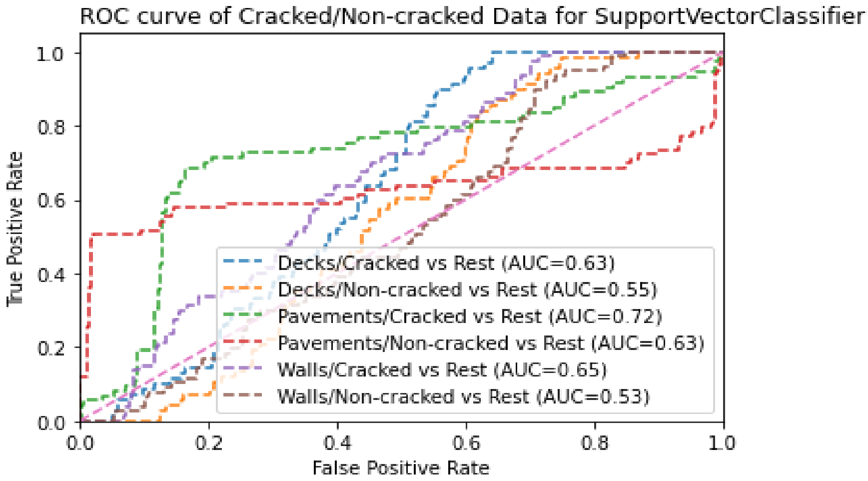

Fig 8: Class-wise ROC curves for SVM.

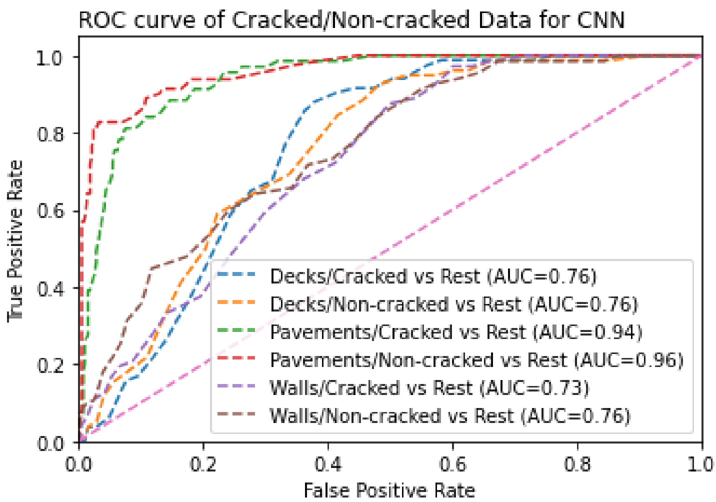

Fig 9: Class-wise ROC curves for CNN.

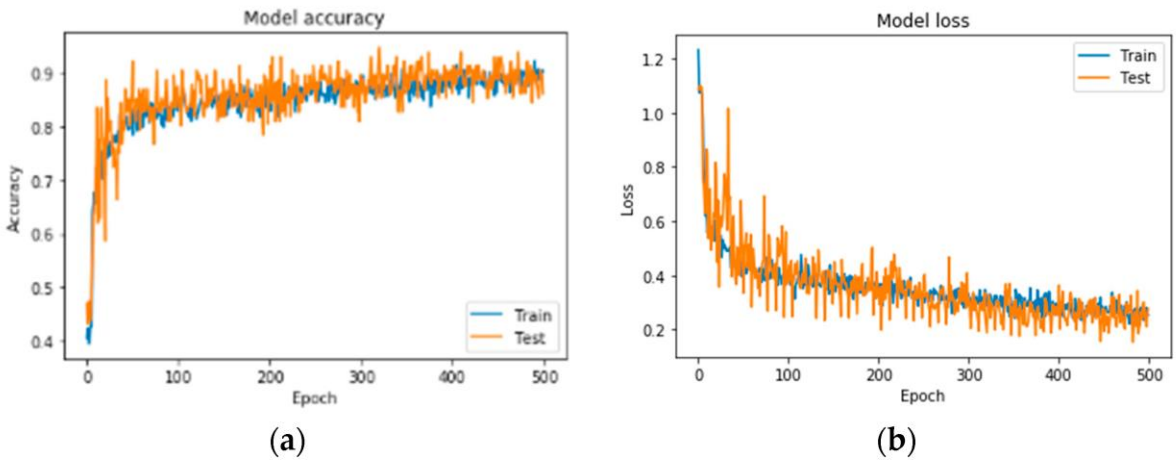

Fig 10: Accuracy and loss values for proposed CNN-based crack detection.

5.3 Discussion

In this study, we have compared the outcomes from our proposed CNN approach with two other well-known benchmark approaches like Random Forest (RF) and Support Vector Machine (SVM). Finally, It can be observed that the CNN outperforms the RF and SVM, by providing more accurate predicted class label indicating high scores in the confusion matrix. The proposed methodology encompasses a series of pre-processing steps applied to captured images of bridge cracks. These steps involve image enhancement techniques to improve image quality, wavelet denoising to eliminate unwanted noise, and image segmentation to isolate the cracks from the background. Subsequently, a meticulously formulated bridge crack CNN classification model, meticulously constructed using Python libraries, demonstrates remarkable capabilities in accurately classifying the various types of bridge cracks. The feasibility and effectiveness of this scheme are demonstrably validated by the real-world detection outcomes, exhibiting a loss value below 0.1 and an impressive total accuracy exceeding 90%. This innovative approach represents a significant stride towards enhancing the efficiency of fracture diagnosis and substantially reducing risk factors in domestic bridge safety inspections, ushering in a new era characterized by automation and intelligence.

6. Conclusions & Future work

This study delves deeply into the practical implementation of IoT technology for identifying cracks within bridge structures. We unveil a robust crack classification application system meticulously designed by considering both the inherent attributes of IoT and the unique structural characteristics of bridges. The proposed approach aims to address the prevailing reliance on manual visual inspection and outdated risk assessment practices for bridge crack detection. By leveraging digital and intelligent detection methods, we propose to optimize the efficacy of bridge safety diagnostics and successfully mitigate the associated risk factors. Building upon the promising results of this study, future work will focus on exploring the integration of diverse IoT sensors, such as strain gauges and vibration sensors, alongside image capture technology. This multi-sensor approach aims to provide a more comprehensive understanding of the bridge's health and potential crack development. The ultimate goal will be to develop a real-time monitoring system capable of continuously analyzing sensor data and images to detect anomalies in real-time.

Funding

This research received no external funding.

Conflicts of interest

The authors declare no conflict of interest.

Authors contribution statement

SM, SKP: Conceptualization; SM: Data curation; SM, SKP: Formal analysis; SM, SKP: Investigation; SKP: Methodology; SKP: Project administration; SM, SKP: Resources; SM: Software; SKP: Supervision; SM, SKP: Validation; SM: Visualization; SM, SKP: Roles/Writing – original draft; SM: Writing – review & editing.

References

[1] Mishra, M., Lourenço, P. B., &Ramana, G. V. (2022). Structural health monitoring of civil engineering structures by using the internet of things: A review. *Journal of Building Engineering*, *48*, 103954.

[2] Kong, X., &Hucks, R. G. (2023). Preserving our heritage: A photogrammetry-based digital twin framework for monitoring deteriorations of historic structures. *Automation in Construction*, *152*, 104928.

[3] Luo, J., Huang, M., & Lei, Y. (2022). Temperature effect on vibration properties and vibration-based damage identification of bridge structures: A literature review. *Buildings*, *12*(8), 1209.

[4] Wang, D., Ren, B., Cui, B., Wang, J., Wang, X., & Guan, T. (2021). Real-time monitoring for vibration quality of fresh concrete using convolutional neural networks and IoT technology. *Automation in Construction*, *123*, 103510.

[5] Wang, D., & Zhang, W. (2023). Damage detection combining principal component analysis and deep convolutional neural network with dynamic response from FBG arrays. *Journal of Civil Structural Health Monitoring*, *13*(1), 101-115.

[6] Scuro, Carmelo, et al. "IoT for structural health monitoring." *IEEE Instrumentation & Measurement Magazine* 21.6 (2018): 4-14.

[7] Hou, Y., Qian, S., Li, X., Wei, S., Zheng, X., & Zhou, S. (2023). Application of Vibration Data Mining and Deep Neural Networks in Bridge Damage Identification. *Electronics*, *12*(17), 3613.

[8] Mohammed Abdelkader, E. (2022). On the hybridization of pre-trained deep learning and differential evolution algorithms for semantic crack *detection* and recognition in ensemble of infrastructures. *Smart and Sustainable Built Environment*, *11*(3), 740-764.

[9] Yessoufou, F., & Zhu, J. (2023). One-Class Convolutional Neural Network (OC-CNN) Model for Rapid Bridge Damage Detection Using Bridge Response Data. *KSCE Journal of Civil Engineering*, *27*(4), 1640-1660.

[10] Jia, J., & Li, Y. (2023). Deep Learning for Structural Health Monitoring: Data, Algorithms, Applications, Challenges, and Trends. *Sensors*, *23*(21), 8824.

[11] Qureshi, K. N., Kaiwartya, O., Jeon, G., &Piccialli, F. (2022). Neurocomputing for internet of things: object recognition and detection strategy. *Neurocomputing*, *485*, 263-273.

[12] Zhang, J., Zhang, K., An, Y., Luo, H., & Yin, S. (2023). An integrated multitasking intelligent bearing fault diagnosis scheme based on representation learning under imbalanced sample condition. *IEEE Transactions on Neural Networks and Learning Systems*.

[13] Fernández-Gómez, A. M., Gutiérrez-Avilés, D., Troncoso, A., &Martínez-Álvarez, F. (2023). A new Apache Spark-based framework for big data streaming forecasting in IoT networks. *The Journal of Supercomputing*, 1-23.

[14] Pan, Y., & Zhang, L. (2023). Integrating BIM and AI for smart construction management: Current status and future directions. *Archives of Computational Methods in Engineering*, *30*(2), 1081-1110.

[15] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, *25*.

[16] Tabernik, D., Šuc, M., &Skočaj, D. (2023). Automated detection and segmentation of cracks in concrete surfaces using joined segmentation and classification deep neural network. *Construction and Building Materials*, *408*, 133582.

[17] Zhang, J., Liu, X., Zhang, X., Xi, Z., & Wang, S. (2023). Automatic Detection Method of Sewer Pipe Defects Using Deep Learning Techniques. *Applied Sciences*, *13*(7), 4589.

[18] Jeong, J. H., & Jo, H. (2023). Toward Real-world Implementation of Deep Learning for Smartphone-Crowdsourced Pavement Condition Assessment. *IEEE Internet of Things Journal*.

[19] Mishra, A., Gangisetti, G., EftekharAzam, Y., &Khazanchi, D. (2024). Weakly supervised crack segmentation using crack attention networks on concrete structures. *Structural Health Monitoring*, 14759217241228150.

[20] Liu, H., Nehme, J., & Lu, P. (2023). An application of convolutional neural network for deterioration modeling of highway bridge

components in the United States. *Structure and Infrastructure Engineering*, *19*(6), 731-744.

[21] Dardouri, S., BuHamdan, S., Al Balkhy, W., Dakhli, Z., Danel, T., &Lafhaj, Z. (2023). RFID platform for construction materials management. *International Journal of Construction Management*, *23*(14), 2509-2519.

[22] Fan, W., Chen, Y., Li, J., Sun, Y., Feng, J., Hassanin, H., &Sareh, P. (2021). Machine learning applied to the design and inspection of reinforced concrete bridges: Resilient methods and emerging applications. In *Structures* (Vol. 33, pp. 3954-3963). Elsevier.

[23] Santarsiero, G., Masi, A., Picciano, V., &Digrisolo, A. (2021). The Italian guidelines on risk classification and management of bridges: Applications and remarks on large scale risk assessments. *Infrastructures*, *6*(8), 111.

[24] Lei, X., Dong, Y., &Frangopol, D. M. (2023). Sustainable life-cycle maintenance policymaking for network-level deteriorating bridges with a convolutional autoencoder–structured reinforcement learning agent. *Journal of Bridge Engineering*, *28*(9), 04023063.

[25] Elghaish, F., Talebi, S., Abdellatef, E., Matarneh, S. T., Hosseini, M. R., Wu, S., ...& Nguyen, T. Q. (2022). Developing a new deep learning CNN model to detect and classify highway cracks. *Journal of Engineering, Design and Technology*, *20*(4), 993-1014.

[26] Ai, D., Jiang, G., Lam, S. K., He, P., & Li, C. (2023). Computer vision framework for crack detection of civil infrastructure—A review. *Engineering Applications of Artificial Intelligence*, *117*, 105478.

[27] Baduge, S. K., Thilakarathna, S., Perera, J. S., Arashpour, M., Sharafi, P., Teodosio, B., ...&Mendis, P. (2022). Artificial intelligence and smart vision for building and construction 4.0: Machine and deep learning methods and applications. *Automation in Construction*, *141*, 104440.

[28] Bazrafshan, P., On, T., Basereh, S., Okumus, P., &Ebrahimkhanlou, A. (2023). A graph-based method for quantifying crack patterns on reinforced concrete shear walls. *Computer-Aided Civil and Infrastructure Engineering*.

[29] Abubakr, M., Rady, M., Badran, K., & Mahfouz, S. Y. (2024). Application of deep learning in damage classification of reinforced concrete bridges. *Ain Shams engineering journal*, *15*(1), 102297.