


Precision crack analysis in concrete structures using CNN, SVM, and KNN: a machine learning approach

Mohammed Nabi Anwar Basha Gulshan Taj¹, Nuha Alruwais², Haya Mesfer Alshahrani³,
Jagadeesan Vijayalakshmi⁴, Narayanan Shanmugapriya⁵, Sridhar Jayaprakash⁶ 

¹Sona College of Technology, Department of Civil Engineering. Salem, Tamilnadu, India.

²King Saud University, College of Applied Studies and Community Services, Department of Computer Science and Engineering. 22459, Saudi Arabia.

³Princess Nourah bint Abdulrahman University, College of Computer and Information Sciences, Department of Information Systems. 84428, Riyadh, Saudi Arabia.

⁴Kongu Engineering College, Department of Electronics and Communication Engineering. Perundurai, India.

⁵School of Engineering and Technology Dhanalakshmi Srinivasan University, Department of Computer Science and Engineering. Trichy, Tamilnadu, India.

⁶GMR Institute of Technology, Department of Civil Engineering. Rajam, Andhrapradesh, India.

e-mail: drgulshaniit@gmail.com, n.rowais@ksu.edu.sa, hshahrani@pnu.edu.sa, vijaya.jagadeesh09@gmail.com, shanmugapriyan.set@dsuniversity.ac.in, sridharjayaprakash599@gmail.com

ABSTRACT

Cracks in structures are discontinuities that occur due to stress, material degradation, or design flaws, compromising structural integrity. Detecting and analyzing cracks is crucial for assessing safety and determining maintenance needs. Various methods like image processing, machine learning, and deep learning are employed for accurate crack identification and classification. Advanced algorithms enhance detection accuracy and efficiency, leading to improved structural maintenance strategies. Addressing cracks promptly ensures prolonged infrastructure lifespan and safety. The goal of this research is to completely change the field of concrete cracking analysis through the use of cutting-edge machine learning technologies. The main objective is to design and implement a crack analysis system that uses cutting-edge techniques such as Convolutional Neural Network (CNN), support vector machine (SVM), and the k-nearest-neighbor (KNN). By harnessing the power of machine learning, the project aims to achieve unprecedented accuracy in crack detection, enabling precise identification of crack characteristics including length, width, and depth. Through meticulous coding and collaboration utilizing platforms like Google Colab, the study seeks to establish a robust framework for effectively and efficiently classifying cracks in structures. This innovative approach holds promise for enhancing structural integrity assessment and facilitating timely maintenance interventions, thus contributing to safer and more resilient infrastructure.

Keywords: Concrete crack analysis; Machine learning; Structural integrity; Classification; Google Colab.

1. INTRODUCTION

Concrete structures form the backbone of modern infrastructure, supporting everything from buildings to bridges. However, over time, these structures are susceptible to various forms of degradation, with cracks being one of the most common and critical issues. Detecting and analyzing cracks in concrete is paramount for ensuring structural integrity and safety [1]. Traditional methods of crack analysis have often been labour-intensive, time-consuming, and prone to human error. Machine learning has been used widely these days to solve various real world problems [2]. Usually, cracks emerge at a microscopic level on the surface of the infrastructure component. These cracks make the component weak, reduce its loading capacity and lead to discontinuities on the surface. If such cracks are detected at an early stage, further damage can be reduced [3]. Manual methods of crack analysis involve experts who examine the component visually and the use of specific tools to identify any deficiency in the component. Automatic crack analysis deals with using technologies to identify cracks from infrastructures. Some of the primary benefits encompass exceptional accuracy, automation, swiftness, cost-effectiveness, early anomaly detection, continuous monitoring, data integration, remote surveillance, historical data analysis, and minimized

downtime. Noteworthy drawbacks involve data quality and quantity issues, intricate model complexity, data privacy and security concerns, difficulties in generalization, model robustness under varying conditions, regulatory and compliance considerations, and the need for ongoing maintenance and updates.

Machine learning has made significant advancements in the domain of crack detection and structural health monitoring [4]. There are many algorithms available in ML which is given in Figure 1. Key applications of machine learning in this context encompass visual inspection and image analysis, remote sensing, automated crack segmentation, utilization of sensor networks, data fusion from diverse sources, pattern recognition, real-time monitoring, geospatial crack mapping, deployment of deep learning techniques, and the augmentation of traditional non-destructive testing (NDT) methods [5].

This research aims to capitalize on these advancements by integrating advanced machine learning algorithms into crack analysis systems.

The main goal is to develop and execute an advanced framework utilizing cutting-edge techniques like Convolutional Neural Networks (CNN), support vector machine (SVM), and K-Nearest Neighbors (KNN). Supervised learning in machine learning refers to a kind of learning where the algorithm is trained on a labeled dataset, which means that every input data point has a label assigned to it. The goal is for the algorithm to learn a mapping from inputs to outputs based on the patterns present in the labeled data [6]. Deep learning has emerged as a powerful tool for image analysis. In this context, CNNs are particularly well-suited for crack detection as they can effectively capture intricate patterns and features present in images, enabling accurate identification of cracks even in complex and cluttered environments [7]. By harnessing the power of machine learning, this project seeks to achieve unprecedented accuracy in crack detection. Unlike traditional methods, which rely heavily on manual inspection, the proposed system aims to automate the process, enabling rapid and precise identification of crack characteristics, including length, width, and depth [8]. This automated approach not only enhances efficiency but also reduces the likelihood of human error, thereby improving the reliability of crack assessment.

The primary objective is to improve the way that cracks in structures are classified by using cutting-edge machine learning techniques. Traditional methods of crack classification often rely on manual inspection and subjective interpretation, leading to inconsistencies and inaccuracies [9]. In contrast, machine learning algorithms offer the potential for automated and objective crack classification, improving efficiency and reliability. To facilitate the development and implementation of the crack analysis system, meticulous coding and collaboration will be conducted, utilizing platforms such as Google Colab. By leveraging cloud computing resources and collaborative tools, the study aims to establish a robust framework for effectively and efficiently classifying cracks in structures. This collaborative approach enables the pooling of expertise and resources, fostering innovation and accelerating progress in the field of concrete crack analysis. It contributes to safer and more resilient infrastructure by enhancing structural integrity assessment and facilitating timely maintenance interventions [10]. By deploying advanced machine learning techniques, we aim to empower engineers and stakeholders with the tools and insights needed to address the challenges posed by concrete cracks effectively. This innovative approach holds promise for transforming the way we analyze and mitigate the impact of cracks in concrete structures, thereby laying the foundation for a more sustainable built environment.

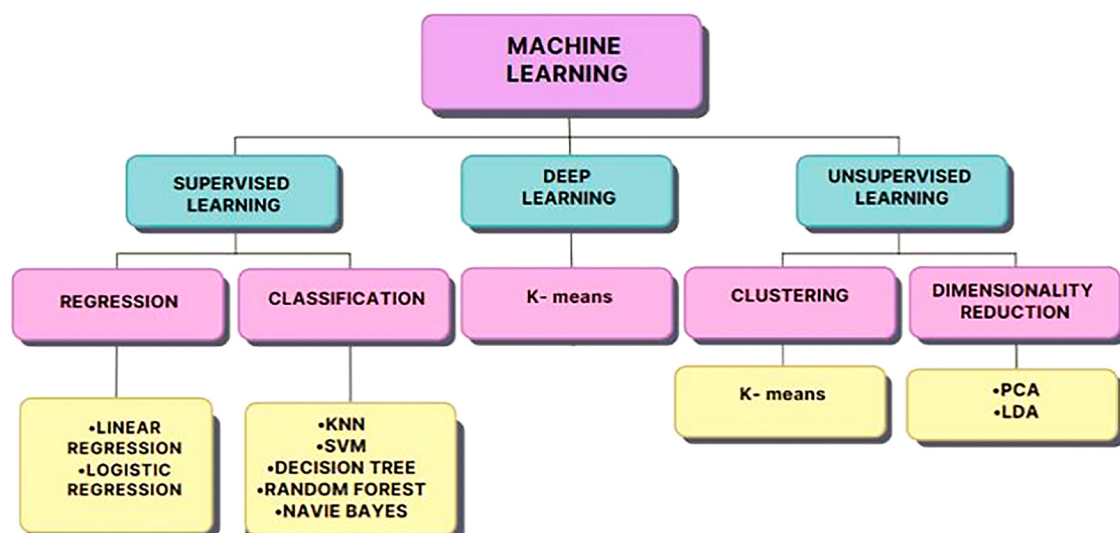


Figure 1: Machine learning algorithm.

2. ML/DL ALGORITHM STUDY

Studying machine learning algorithms for crack analysis can be a fascinating and impactful endeavor, especially in fields like Civil engineering, materials science, and infrastructure maintenance. Here's a general approach you can take to study crack analysis using the Support vector machine (SVM), convolutional neural network (CNN), and k-nearest-neighbor (KNN) algorithms. Since there is a possibility of diverse category of cracks that are to be present in various structural components, the present algorithm study made an attempt to train the data so the type and other parameters of the cracks can be easily estimated. Also the algorithm ensure the quick and efficient analysis procedure using various machine learning process.

2.1. K-nearest neighbour

K-Nearest Neighbors (KNN) algorithm, its application in crack analysis for concrete structures, and how it can be utilized for crack detection along with estimating the length of the crack. A straightforward and user-friendly machine learning approach for regression and classification is K-Nearest Neighbors (KNN) is shown in Figure 2. Its fundamental idea is that comparable data points in the feature space tend to be found close to one another [11]. KNN finds the K nearest neighbors for a given query instance in the training dataset using a distance metric, often Euclidean distance. In KNN classification tasks, the query instance is allocated the majority class label among the K nearest neighbors. KNN averages, or weights, the target values of the K nearest neighbor to obtain the goal value for the query instance while performing regression tasks. Features can be extracted from images of concrete structures containing cracks. These features might include texture descriptors, color histograms, or morphological characteristics of the cracks [12].

Collect a dataset consisting of images of concrete structures, where each image is labeled with information about the presence and extent of cracks. Train a KNN model using the extracted features as input and the length of the cracks as the target variable. Since KNN works well with small datasets, it's suitable for scenarios where the dataset size is limited. Given a new image of a concrete structure, use the trained KNN model to predict the length of the cracks.

To detect cracks, segment the image to isolate regions containing cracks, then use the KNN model to estimate the length of each detected crack. The formula calculates the distance (d) between two points $(x1, y1)$ and $(x2, y2)$ in a coordinate system is given in Equation 1. Fine-tune the model parameters, such as the choice of K and the feature extraction methods, to optimize performance. Integrate the trained KNN model into a software system or application for real-time crack analysis in concrete structures. Ensure scalability and efficiency of the deployed system, especially if dealing with large-scale infrastructure monitoring.

$$d = \sqrt{(x2 - x1)^2 + (y2 - y1)^2} \quad (1)$$

where,

$x1, x2$ = Co-ordinates of one data point

$y1, y2$ = Co-ordinates of another data point

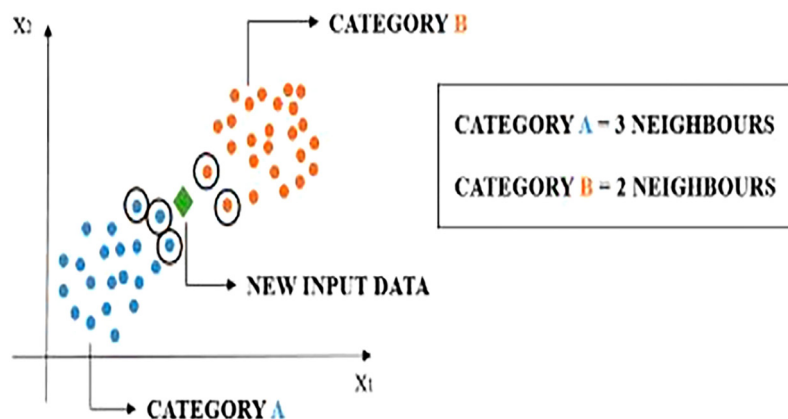


Figure 2: K-Nearest Neighbors (KNN) algorithm.

2.2. Support vector machine

The supervised learning algorithm Support Vector Machine (SVM), which is utilized for regression and classification problems, is seen in Figure 3. Boundary line is placed which divides the plane into positive and negative domain. Since many data points are used, the positive and negative planes create a margin boundary so the data point can be segregated well manner. Optimum hyperplane is considered as support vectors since they find the maximum margin between the data points. Many fields, such as image processing and structural health monitoring, employ it extensively, and it performs especially well in high-dimensional spaces [13]. Let's explore how SVM can be applied to crack analysis in concrete structures, focusing on crack detection and estimation of crack properties like length and width.

The Support Vector Machine (SVM), is a classifier using discrimination that operates by identifying the hyperplane in the feature space that best divides the classes. The objective is to minimize classification mistakes and maximize the margin between the classes. The way SVM works is that it takes input data, maps it into a feature with high dimension space, and then determines which hyperplane best divides the data points into distinct classes. SVM finds the best hyperplane to optimize the margin between classes in classification problems. The goal of support vector machines (SVM) in regression problems is to fit a hyperplane that best captures the connection between the target variable and the input information [14]. Using kernel functions like the polynomial or radial basis function (RBF) kernel, SVM may effectively handle non-linearly separable data.

Extract relevant features from images of concrete structures containing cracks. These features could include texture features, intensity histograms, or morphological characteristics of cracks. Prepare a labelled dataset consisting of images of concrete structures, with corresponding labels indicating the presence and properties of cracks (e.g., length, width). Train an SVM model using the extracted features as input and the crack properties (length, width) as the target variables. Use appropriate kernel functions and regularization parameters to optimize the SVM model for crack analysis tasks.

Given a new image of a concrete structure, use the trained SVM model to detect cracks and estimate their properties (length, width). Morphological operations such as erosion, dilation, and opening can be applied to enhance crack detection and segmentation in the images [15]. Generate histograms of crack intensity along the length and width of detected cracks to visualize their properties with respect to intensity.

Evaluate the performance of the SVM model using suitable evaluation metrics such as accuracy, precision, recall, and F1-score. Fine-tune the SVM parameters, including the choice of kernel function and regularization parameter, to improve model performance [16].

Integrate the trained SVM model into a software system or application for automated crack analysis in concrete structures. Ensure scalability and efficiency of the deployed system, especially when dealing with large-scale infrastructure monitoring [17]. Few of the other studies related to such works are also mentioned in the articles. Further interested readers are advised to refer the reference part [18–24].

Support Vector Machine (SVM) algorithm can be effectively utilized for crack analysis in concrete structures, including crack detection and estimation of crack properties like length and width. The linear hyperplane for the SVM can be written in Equation 2. By leveraging appropriate feature extraction techniques, kernel functions, and regularization parameters, SVM can provide valuable insights for infrastructure maintenance and safety assessment. computes the distance (d_i) of a data point (x_i) from a hyperplane defined by weight vector

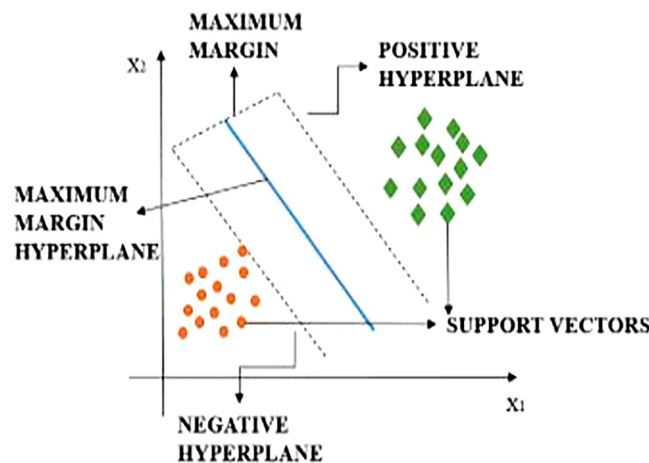


Figure 3: Support Vector Machine (SVM) algorithm.

(w), bias (b), and the norm of w ($\|w\|$) is given in Equation 3. The SVM Classifier can be determining by using the equation given in Equation 4.

The equation for the linear hyperplane can be written as

$$wTx + b = 0 \quad (2)$$

Where,

w = normal vector

b = offset

The distance between a data point x_i and the decision boundary can be calculated as

$$d_i = \frac{w^T x_i + b}{\|w\|} \quad (3)$$

Where,

$\|w\|$ = Euclidean norm of the normal vector W

For Linear SVM classifier:

$$\hat{y} = \begin{cases} 1; wTx + b \geq 0 \\ 0; wTx + b < 0 \end{cases} \quad (4)$$

2.3. Convolutional neural networks

Convolutional Neural Networks (CNNs) are a great tool for crack diagnosis in concrete buildings because, as Figure 4 illustrates, they can automatically learn hierarchical characteristics from raw picture data. Higher accuracy in crack identification and property estimate tasks can be attained by utilizing pre-trained models such as VGG16, which have been trained on extensive datasets and are able to identify a broad variety of characteristics [25]. In the present study, VGG16 is used as a convolutional neural network (CNN) model to perform image classification tasks. In general, it consists of 16 layers that would process the information from image and hence improve the accuracy. Let's examine how CNN's can be used to solve this issue, starting with crack identification and using color bars to visualize crack characteristics like length and width.

Specifically created for processing and evaluating visual data, including images, CNNs are deep learning models. Feature extraction is accomplished by convolutional layers, while spatial down sampling is accomplished by pooling layers, among other layers [26]. Edges, textures, and patterns at various spatial scales are extracted from the input image using convolutional layers by the application of convolution procedures. While maintaining the most crucial information, pooling layers shrink the feature maps' spatial size. The collected characteristics for tasks involving regression or classification are combined in fully connected layers at the conclusion of the network [27]. VGG16 and other pre-trained CNN models can be adjusted for particular tasks using comparatively little datasets because they were trained on large datasets like ImageNet. CNNs automatically

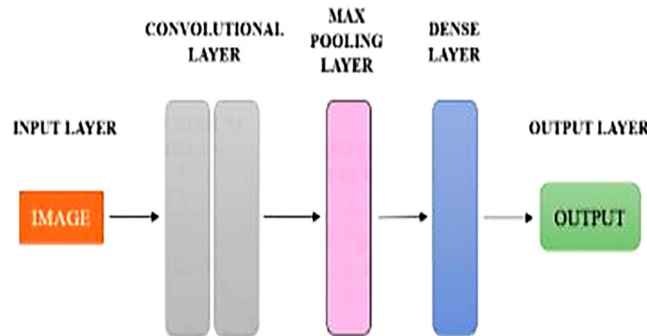


Figure 4: Convolutional Neural Networks (CNNs) algorithm.

learn hierarchical features from raw pixel values, eliminating the need for handcrafted feature engineering [28]. Collect a dataset of images of concrete structures containing cracks, with labels indicating the presence and properties of cracks (e.g., length, width).

Convolutional layers are retained whereas fully linked layers are removed when employing the pre-trained VGG16 model as a feature extractor. To make the VGG16 model more suited to the particular characteristics of concrete crack photographs, use the crack analysis dataset to refine it. Apply CNN's trained model for detecting cracks in concrete structures. Partitioning the images to find the regions with cracks is necessary to do this [29]. Crack delineation and detection can be enhanced by utilizing edge map features, thresholding, and kernel techniques. Estimate the length and width of discovered cracks by measuring the spatial dimensions of crack segments using the relevant methodologies. Examine the CNN model's fracture detection capabilities and the precision of its length and width estimations [30]. The Mean value of the Data can be determined by using the Equation 5.

Compare the performance of the CNN model with traditional machine learning algorithms like KNN and SVM, considering factors such as accuracy, computational efficiency, and robustness. Standard Deviation of the data can be analysed using the Equation 6. Integrate the trained CNN model into a software system or application for automated crack analysis in concrete structures. For Normalize the input data we use adequate formulae which is given in Equation 7. Ensure scalability and efficiency of the deployed system, especially for real-time or large-scale infrastructure monitoring [31]. In particular when combined with transfer learning using pre-trained models like VGG16, offer a powerful approach for crack analysis in concrete structures. Transfer learning is a general AI technique that uses pre-trained models to improve the speed of the process in addition to increase the efficiency. It also best suited to recognize similar objects in set of images given as input.

The problem is complex, and starting from a pre-trained model increases the chances of success By leveraging automatic feature learning and advanced techniques for crack detection and property estimation, CNNs can provide accurate and efficient solutions for infrastructure maintenance and safety assessment [32]. Additionally, visualizing crack properties using color bars enhances the interpretability of the analysis results. Finally, the output result of the BN layer can be identified using the Equation 8. In Boundary layer, a set of latent physical parameters are connected with neural networks that can easily map the extracted features of the image along with the other physical influencing parameters mentioned in the study.

First calculate the mean value of the data

$$\mu B = \frac{1}{m} \sum_{i=1}^m x_i \quad (5)$$

Where,

x_i = input data

m = data set

B = Mean value of data set

The above formula calculates the average value of mini-batch. Then calculate the standard deviation of minibatch as shown in equation

$$\sigma_B^2 = \frac{1}{m} \sum_{i=1}^m (x_i - \mu_B)^2 \quad (6)$$

Then normalize the input data as shown in equation

$$\hat{x}_i = \frac{(x_i - \mu_B)^2}{\sqrt{\sigma_B^2 + \varepsilon}} \quad (7)$$

Output result of the BN (boundary layer) can be obtained as

$$y_i = \gamma \hat{x}_i + \beta = BN_{\gamma\beta}(x_i) \quad (8)$$

Where,

β = offset coefficient

γ = Scaling Factor

The CNN-based approach offers several advantages, including scalability, adaptability to different types of cracks, and potential for real-time deployment. By automating the crack analysis process, it can significantly reduce inspection time, human error, and maintenance costs, thereby improving the overall efficiency and safety of civil infrastructure management.

3. PROPOSED METHOD

In the context of an image dataset, the main goal of crack classification is to determine the classification to which the crack belongs. To identify the classification of crack the proposed solution is based on a transfer learning method and call back method by using Convolutional Neural Networks (CNN) algorithm is shown in Figure 5. Major difference between the various call back methods used in the present analysis is that model checkpoint will perform early stopping procedure while the other methods perform stopping in intervals.

3.1. Transfer learning method

In machine learning, transfer learning is a valuable technique, especially for image classification applications. Convolutional neural network (CNN) with transfer learning used to crack classification use a pre-trained CNN model [33]. Usually, this pre-trained model was built for a different purpose, such general image recognition, using a sizable dataset.

A number of primary fracture forms are recognized and categorized, including transverse, longitudinal, water-heaved, shrinkage, expansion, overloading, settling, and durability cracks. Used convolutional neural networks (CNNs) for crack classification. The present study utilized transfer learning to exploit the learned characteristics of pre-trained models, such as ResNet152v2 and VGG16. ResNet152V2 is a model which constitutes 152 convolutional layers with a Global average pooling layer. It also holds a dense layer with 64 neurons, and a dense layer followed by a convolution layer. In the present approach, CNN models used Keras and TensorFlow which are well documented by other researchers [34]. Dataset has been augmented with data to increase the robustness and diversity. The following settings were applied during the algorithm process: shearing, rotation, horizontal flipping, zooming, rescaling, and changes to width and height. Dataset has been split into three categories: 70% for training, 20% for validation, and 10% for testing. One epoch (one pass through the entire dataset) was completed for each category. Used a batch size of 32 samples per iteration for training the model.

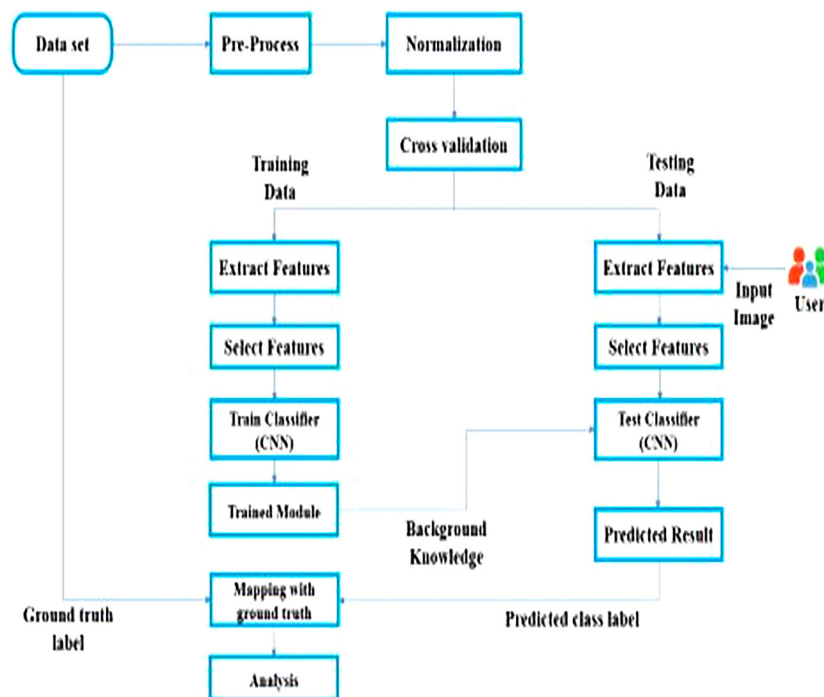


Figure 5: Proposed method for classification.

Three variables were used to create the CNN model: the loss function (categorical cross-entropy), an evaluation metric, and an optimizer. A flattening layer (X_Flatten) was incorporated to flatten the multi-dimensional feature maps into a one-dimensional vector for the subsequent fully connected layers. The flatten layer used in present algorithm acts as a bridge between the convolutional/pooling layers in-order to extract special features of the data set to perform regression and other classification tasks. Unlike retraining the CNN model from start, which can be expensive computationally and data-intensive, transfer learning allows us to fine-tune the pre-trained model by retraining only the last layers [35].

3.2. Callback method

In the context of Convolutional Neural Networks (CNNs) for crack classification, callback methods play a crucial role in monitoring and optimizing the training process [36]. Callbacks are functions that are called at specific points during training, allowing for dynamic adjustments, logging, and control over the training process. This callback saves the model's weights during training. It allows you to save the best model based on certain criteria, such as validation accuracy or loss. This ensures that you can retrieve the best performing model even if training is interrupted. This callback stops training when a monitored metric has stopped improving during the flow process. For example, one can monitor validation loss and stop training when it stops decreasing, preventing overfitting and saving computation time. Learning Rate Reduction on Plateau is a callback that reduces the learning rate when a monitored metric has stopped improving. This can help fine-tune the model's parameters more effectively, especially in later stages of training when progress is slower. In the present algorithm, early stopping with callback method to monitor the validation loss has been implemented and stopped the process training when the conditions were met, preventing overfitting and optimizing model performance.

4. ANALYTICAL EVALUATION

Analytical evaluation of machine learning (ML) algorithms for crack detection and classification involves assessing the performance, effectiveness, and suitability of these algorithms for accurately identifying and categorizing cracks in images [37]. Begin by assembling a comprehensive dataset containing images or data samples with cracks of various types, sizes, orientations, and conditions. Ensure that the dataset is appropriately labeled with ground truth annotations for crack detection and classification. Extract relevant features from the dataset that are informative for distinguishing cracks from non-crack areas and categorizing different types of cracks. These features may include texture patterns, edge information, shape characteristics, color contrasts, etc. Commonly used algorithms include K-Nearest Neighbors (KNN), Support Vector Machines (SVMs), Convolutional Neural Networks (CNNs). These algorithms include factors such as algorithm complexity, computational efficiency, and scalability [38]. KNN is a relatively simple and interpretable algorithm, but its performance can be sensitive to the choice of the number of neighbors (k) and the distance metric used. Number of images uploaded by the user could be 250–350 and can be fixed as maximum upper limit. Meanwhile the good quality images with standard pixel values (72ppi- pixel per inch) is expected to be upload in the programming. Experimenting with different values of k and distance metrics may yield improvements in accuracy as shown in Figure 6. In total, 250 samples were used to extract the results in the present algorithm.

Analytical evaluation of a K-Nearest Neighbors (KNN) algorithm for crack detection, focusing solely on crack length due to limited dataset availability, yielded an accuracy of 69%. This assessment provides insights into the algorithm's performance in identifying cracks based solely on their length within the constrained dataset.

The limited dataset size may impact the algorithm's ability to generalize effectively to unseen crack instances. With a small number of samples, the algorithm may struggle to capture the full diversity of crack



Figure 6: Detection for KNN algorithm.

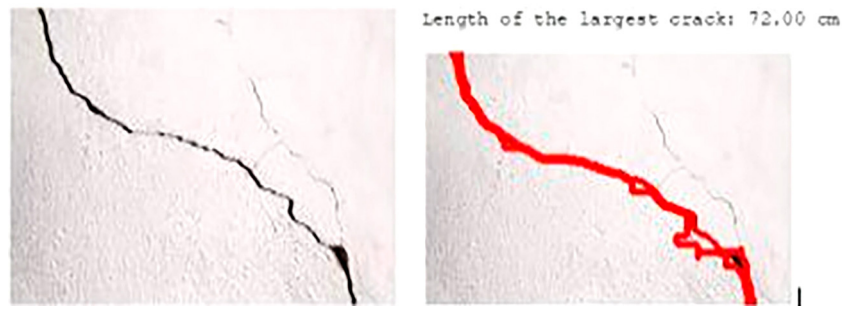


Figure 7: Detection for SVM algorithm.

Table 1: Accuracy for crack detection.

TYPE OF ALGORITHM	ACCURACY
KNN Algorithm	69%
SVM Algorithm	78%
CNN Algorithm	86%

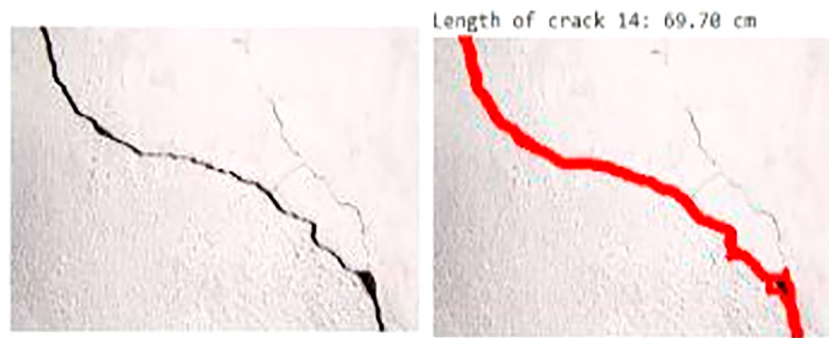


Figure 8: Detection for CNN algorithm.

lengths and variations present in real-world scenarios. Incorporating additional features could enhance the algorithm's discriminative power and improve overall performance.

Analytical evaluation of a Support Vector Machine (SVM) algorithm for crack detection, incorporating both crack length and width, demonstrated improved performance compared to the K-Nearest Neighbors (KNN) algorithm. The SVM algorithm achieved an accuracy of 78% on the same dataset size. Additionally, a histogram graph was utilized to visualize crack width distribution in relation to intensity and frequency for comprehensive crack analysis. Incorporating both crack length and width as features provides a more comprehensive representation of crack characteristics compared to considering only crack length. By capturing additional dimensions of crack geometry, the SVM algorithm can potentially better discriminate between different types of cracks. Visualizing crack width distribution using a histogram graph offers valuable insights into the relationship between crack width, intensity, and frequency. The SVM algorithm's accuracy of 78% surpasses that of the KNN algorithm 69%, indicating its superior performance in crack detection and classification tasks. SVMs are known for their effectiveness in handling high-dimensional data and separating complex decision boundaries, which may contribute to the observed improvement in accuracy is shown in Figure 7.

Analytical evaluation of a Convolutional Neural Network (CNN) algorithm for crack detection, incorporating both crack length and width as features, demonstrated significant performance improvement compared to both the K-Nearest Neighbors (KNN) and Support Vector Machine (SVM) algorithms. The CNN algorithm achieved an impressive accuracy of 86%, surpassing the accuracies achieved by both KNN and SVM algorithms, which were previously reported as 69% and 78%, respectively is given in Table 1. Additionally, the width of the crack was visualized using a color bar representation for comprehensive crack analysis in machine learning.

By considering both crack length and width as features, the CNN algorithm benefits from a more comprehensive representation of crack characteristics. CNNs are adept at learning hierarchical representations of complex features, making them well-suited for analyzing multi-dimensional data such as crack geometry. Utilizing a color bar representation to visualize crack width enhances the interpretability of the results and facilitates deeper insights into the relationship between crack dimensions and intensity as shown in Figure 8.

The CNN algorithm achieved a significantly higher accuracy of 86% compared to both KNN and SVM algorithms. This improvement underscores the effectiveness of deep learning approaches, particularly CNNs, in capturing intricate patterns and relationships within crack data, leading to more accurate classification results [39].

Analytical evaluation for crack classification using Convolutional Neural Networks (CNN) algorithm with transfer learning and callback methods, achieved 86% accuracy. Leveraging pre-trained CNN models like ResNet152v2 and VGG16 for crack classification tasks. Transfer learning allows the model to benefit from features learned on large-scale image datasets, enhancing performance on the crack classification task. Call-back methods such as Model Checkpoint, Early Stopping, and Tensor Board are employed to optimize the training process and to monitor model performance. These callbacks ensure efficient training, prevent overfitting, and provide insights into the training dynamics. Considering precautions specific to each type of crack during model training and evaluation was also performed. This involves ensuring balanced representation of different crack types in the dataset and adjusting model parameters accordingly. Generally, as the number of epochs increases, the loss should be decreased, indicating improved model performance. This relationship can be visualized through graphs by plotting epoch against accuracy and loss as shown in Figure 9.

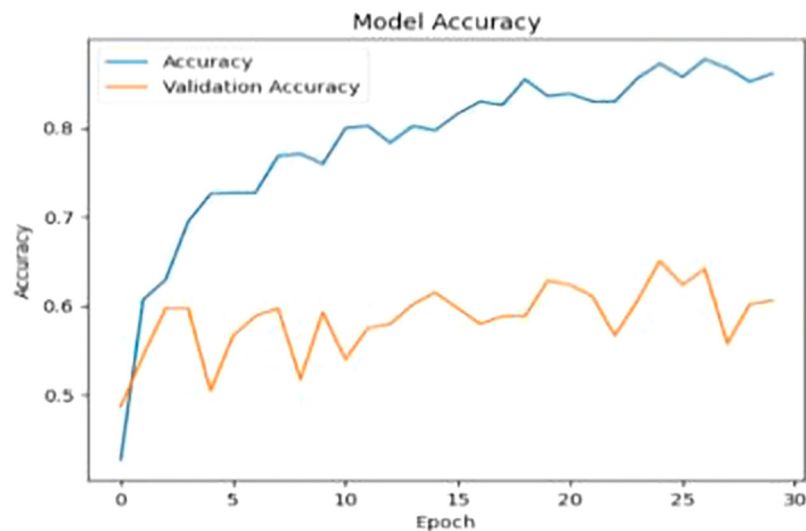


Figure 9: Model accuracy testing graph for crack classification.

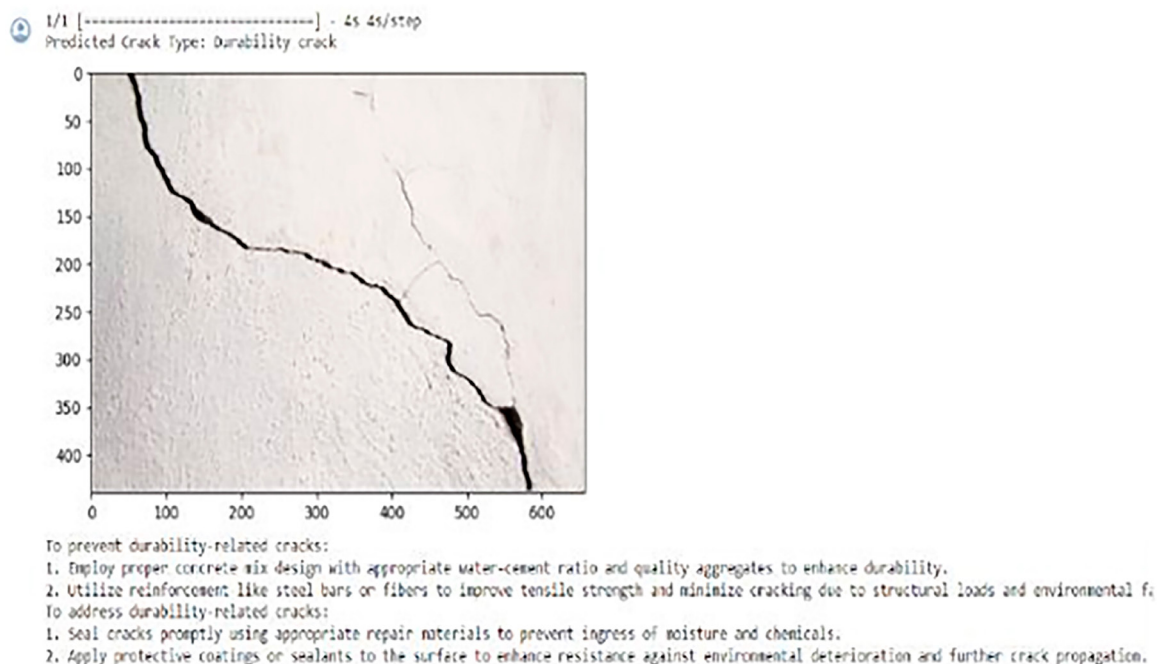


Figure 10: Result for crack classification.

Table 2: Accuracy for crack classification.

TYPE OF ALGORITHM	ACCURACY
CNN Algorithm	86%

To control the output verbosity verbose parameter in the training function is utilized. This allows for better monitoring of the training process and debugging if necessary. Implementing both ResNet152v2 and VGG16 models from transfer learning to evaluate their performance in crack classification. In the present case study done by the authors, both the models showcased similar result pattern irrespective the algorithm developed by programming. Trained model was employed to predict the type of crack present in images and remedial measures are provided accordingly is shown in Figure 10. This involves categorizing cracks into 8 types and offering appropriate recommendations or actions for each type. Table 2 Accuracy for crack classification.

The analytical evaluation ensures robust crack classification using CNN algorithms with transfer learning and callback methods. The integration of multiple models, precautions for different crack types, and remedial measures enhance the practical utility of the model in real-world applications such as infrastructure inspection and maintenance.

5. WEB DEVELOPMENT

5.1. ASP.NET Framework

ASP.NET is a comprehensive web application framework developed by Microsoft, designed to simplify and streamline the process of building dynamic web pages, websites, and web applications. It operates within the broader .NET ecosystem and is particularly advantageous for developers working with languages like C# or Visual Basic.

One of ASP.NET's key strengths is its adherence to the model-view-controller (MVC) architecture, which promotes a structured approach to development by separating the application's logic, presentation, and data layers. This separation of concerns enhances code organization and maintainability. Within ASP.NET, developers have access to a rich set of server controls such as buttons, text boxes, and data grids, facilitating the creation of interactive and data-driven web pages. Data access is supported through ADO.NET, enabling connectivity to various databases, including popular ones like SQL Server, MySQL, and Oracle.

Security features are robust in ASP.NET, with built-in mechanisms for authentication and authorization, making user management and access control straightforward. Session management capabilities allow developers to maintain state across multiple page requests, ensuring a seamless user experience. ASP.NET also integrates with other Microsoft technologies like Azure for cloud hosting and services, offering scalability and flexibility. Development is further streamlined with Visual Studio, an integrated development environment (IDE) that provides tools for coding, debugging, and testing ASP.NET applications and the one employed in the present research work is depicted in Figure 11.

5.2. CODE IMPLEMENTATION PLATFORM

In Visual Studio Code Ultimate, an ASP.NET web application can be developed integrating JavaScript, SQL for data management, and Darknet for crack classification. JavaScript enhances client-side interactivity, while SQL facilitates database operations for storing crack data and associated precautions. Darknet, a deep learning framework, enables efficient crack classification using neural networks. The application can display crack classifications and corresponding precautions based on user inputs or uploaded images. Visual Studio Code Ultimate offers a comprehensive development environment with features for code editing, debugging, and version control, optimizing the development process. Overall, this setup enables the creation of a sophisticated web application for crack classification, incorporating advanced technologies and functionalities seamlessly. The below are the steps for implementing the web development for the crack classification Which leads to the user to check their need by uploading the image in the website. The proposed application is web based and not part of android mobile application and hence the users are directed to go for web based processing. Moreover, the data used is stored in the cloud space with security systems and hence the possibility of data breach is less.

STEP 1: A user login page is designed to authenticate users accessing a crack classification system, ensuring secure and personalized access to the platform's features and data. It provides a gateway for authorized users to contribute to and benefit from the crack classification functionalities offered by the system.

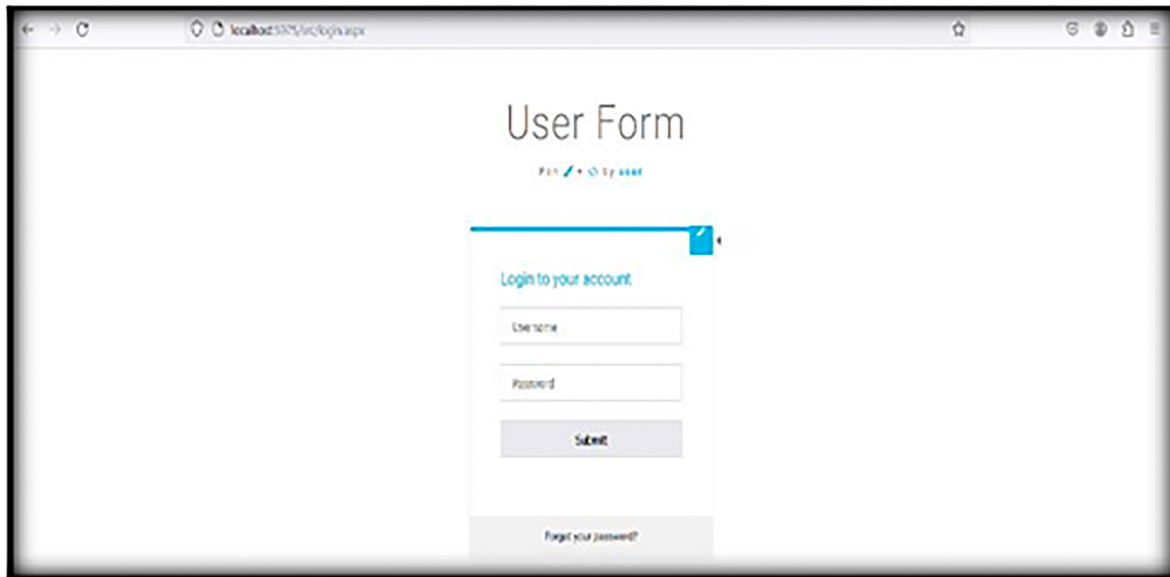


Figure 11: User login for web development.

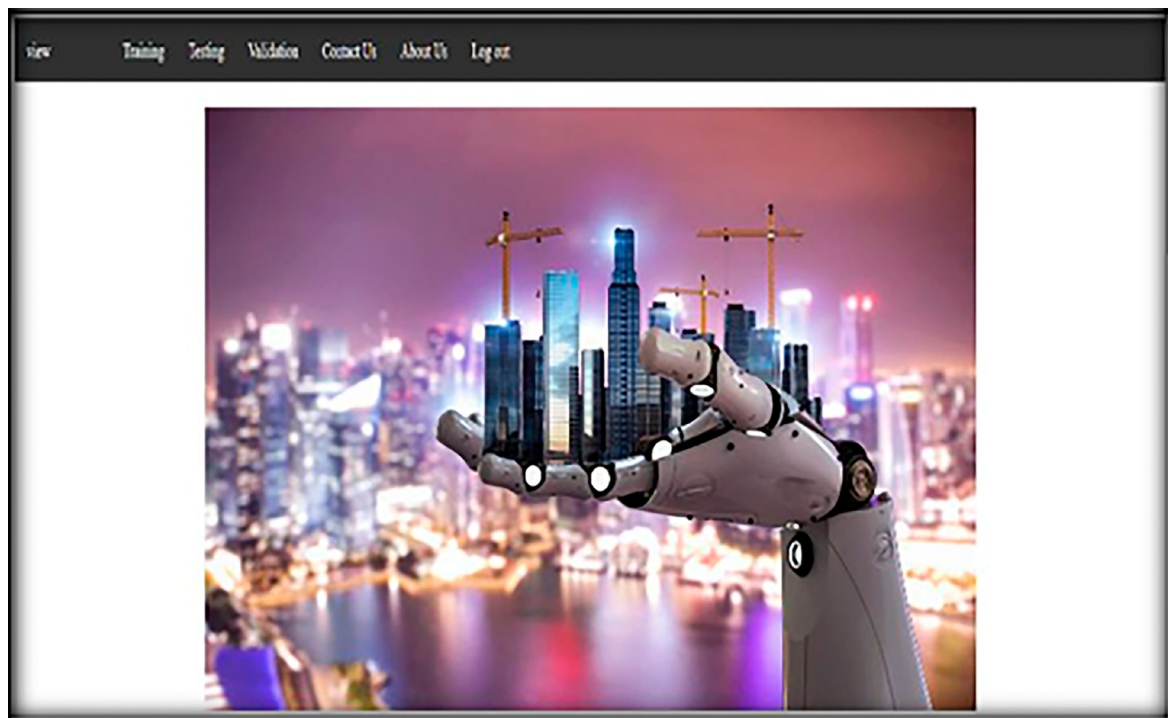


Figure 12: Graphical user interface for web development.

STEP 2: After logging in, users are directed to a graphical user interface page where they can access options like uploading crack images for classification, facilitating user interaction and data input for the system's analysis and processing (Figure 12).

STEP 3: Upon accessing the upload page from the interface, users can upload crack images, and upon submission, the system provides results identifying the type of crack depicted and corresponding precautions, enhancing user understanding and promoting informed decision-making for crack management. For illustration purpose and considering the scope of the study, the present paper demonstrated only the length of the crack and width of the crack also estimated in the same manner. The category and nature of crack also furnished below the Figure 13 for better understanding.



Figure 13: Image results page for crack classification.

6. CONCLUSION

In this analysis study, we explored the effectiveness of three different algorithms, namely K-Nearest Neighbor (KNN), Support Vector Machine (SVM), and Convolutional Neural Network (CNN), for crack detection. Our findings demonstrate that CNN outperformed KNN and SVM in terms of accuracy, indicating its superior capability in crack classification tasks.

As discussed in the previous chapter, out of the three algorithms CNN yields accurate because of the algorithm supervision and accuracy scale. The CNN algorithm exhibited higher accuracy due to its ability to automatically learn hierarchical features from raw data, such as images of cracks, without the need for manual feature extraction. This deep learning approach allows CNN to capture intricate patterns and variations in crack images, leading to more precise classification results compared to traditional machine learning algorithms like KNN and SVM.

7. ACKNOWLEDGMENT

Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2024R237), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia. Research Supporting Project number (RSPD2024R608), King Saud University, Riyadh, Saudi Arabia.

8. BIBLIOGRAPHY

- [1] ZOU, Q., CAO, Y., LI, Q., *et al.*, “CrackTree: automatic crack detection from pavement images”, *Pattern Recognition Letters*, v. 33, n. 3, pp. 227–238, 2012. doi: <http://doi.org/10.1016/j.patrec.2011.11.004>.
- [2] MOON, H., JUNG, H., LEE, C., *et al.*, “Crack detection on pressed panel products using image processing techniques with camera system”, *PHM Society Asia-Pacific Conference*, v. 1, n. 1, pp. 462–465, 2017. doi: <http://doi.org/10.36001/phmap.2017.v1i1.1834>.
- [3] PRASANNA, P., DANA, K.J., GUCUNSKI, N., *et al.*, “Automated crack detection on concrete bridges”, *IEEE Transactions on Automation Science and Engineering*, v. 13, n. 2, pp. 591–599, 2014. doi: <http://doi.org/10.1109/TASE.2014.2354314>.
- [4] YAO, Y., TUNG, S.-T.E., GLISIC, B., “Crack detection and characterization techniques: an overview”, *Structural Control and Health Monitoring*, v. 21, n. 12, pp. 1387–1413, 2014. doi: <http://doi.org/10.1002/stc.1655>.
- [5] SUN, L., KAMALIARDAKANI, M., ZHANG, Y., “Weighted neighborhood pixels segmentation method for automated detection of cracks on pavement surface images”, *Journal of Computing in Civil Engineering*, v. 30, n. 2, p. 04015021, 2016. doi: [http://doi.org/10.1061/\(ASCE\)CP.1943-5487.0000488](http://doi.org/10.1061/(ASCE)CP.1943-5487.0000488).

- [6] OSISANWO, F.Y., AKINSOLA, J.E.T., AWODELE, O., *et al.*, “Supervised machine learning algorithms: classification and comparison”, *International Journal of Computer Trends and Technology*, v. 48, n. 3, pp. 128–138, 2017. doi: <http://doi.org/10.14445/22312803/IJCTT-V48P126>.
- [7] CHA, Y.-J., CHOI, W., BÜYÜKÖZTÜRK, O., “Deep learning-based crack damage detection using convolutional neural networks”, *Computer-Aided Civil and Infrastructure Engineering*, v. 32, n. 5, pp. 361–378, 2017. doi: <http://doi.org/10.1111/mice.12263>.
- [8] AI, D., JIANG, G., KEI, L.S., *et al.*, “Automatic pixel-level pavement crack detection using information of multi-scale neighborhoods”, *IEEE Access: Practical Innovations, Open Solutions*, v. 6, pp. 24452–24463, 2018. doi: <http://doi.org/10.1109/ACCESS.2018.2829347>.
- [9] SILVA, W., LUCENA, D., “Concrete cracks detection based on deep learning image classification”, *Proceedings*, v. 2, n. 8, p. 489, 2018.
- [10] KANG, S.-M., CHUN, C.-J., SHIM, S.-B., *et al.*, “Real time image processing system for detecting infrastructure damage: crack”, In: *Proceedings of the 2019 IEEE International Conference on Consumer Electronics (ICCE)*, pp. 1-3, Las Vegas, NV, USA, 2019. doi: <http://doi.org/10.1109/ICCE.2019.8661830>.
- [11] PARTHASAARATHI, R., BALASUNDARAM, N., NAVEEN ARASU, A., *et al.*, “Analysing the Impact and Investigating Coconut Shell Fiber Reinforced Concrete (CSFRC) under varied loading conditions”, *Journal of Advanced Research in Applied Sciences and Engineering Technology*, v. 35, n. 1, pp. 106–120, 2024.
- [12] NAGAR, R., SINGH, Y., “A literature survey on machine learning algorithms”, *Journal of Emerging Technologies and Innovative Research*, v. 6, n. 4, pp. 471–474, 2019.
- [13] AKAGIC, A., BUZA, E., OMANOVIC, S., “Pothole detection: an efficient vision based method using rgb color space image segmentation”, In: *Proceedings of the 2017 40th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pp. 1104–1109, Opatija, Croatia, 2017. doi: <http://doi.org/10.23919/MIPRO.2017.7973589>.
- [14] ARBAOUI, A., OUAHABI, A., JACQUES, S., *et al.*, “Concrete cracks detection and monitoring using deep learning-based multiresolution analysis”, *Electronics*, v. 10, n. 15, p. 1772, 2021. doi: <http://doi.org/10.3390/electronics10151772>.
- [15] KADHAR, S.A., GOPAL, E., SIVAKUMAR, V., *et al.*, “Optimizing flow, strength, and durability in high-strength self-compacting and self-curing concrete utilizing lightweight aggregates”, *Matéria*, v. 29, n. 1, e20230336, 2024. doi: <http://doi.org/10.1590/1517-7076-rmat-2023-0336>.
- [16] SRINIVASAN, S.S., MUTHUSAMY, N., ANBARASU, N.A., “The structural performance of fiber-reinforced concrete beams with nanosilica”, *Matéria*, v. 29, n. 3, e20240194, 2024. doi: <http://doi.org/10.1590/1517-7076-rmat-2024-0194>.
- [17] SOMVANSHI, M., CHAVAN, P., TAMBADE, S., *et al.*, “A review of machine learning techniques using decision tree and support vector machine”, In: *Proceedings of the 2016 International Conference on Computing Communication Control and Automation (ICCUBEA)*, pp. 1-7, 2016.
- [18] YADAV, P.S., PATIL, J., PATIL, P., *et al.*, “Crack detection using image processing”, *International Journal of Engineering Research & Technology*, v. 7, n. 3, p. 1, 2018.
- [19] MOHAN, A., POOBAL, S., “Crack detection using image processing: a critical review and analysis”, *Alexandria Engineering Journal*, v. 57, n. 2, pp. 787–798, 2018. doi: <http://doi.org/10.1016/j.aej.2017.01.020>.
- [20] BRACKENBURY, D., BRILAKIS, I., DEJONG, M., “Automated defect detection for masonry arch bridges”, In: *International Conference on Smart Infrastructure and Construction 2019 (ICSIC)*, pp. 3-9, 2019. doi: <http://doi.org/10.1680/icsic.64669.003>.
- [21] PARTHASAARATHI, R., BALASUNDARAM, N., NAVEEN ARASU, A., “A stiffness analysis of treated and non-treated meshed coir layer fibre reinforced cement concrete”, *AIP Conference Proceedings*, v. 2861, n. 1, p. 1, 2023. doi: <http://doi.org/10.1063/5.0158672>.
- [22] KUMAR, S., NAVEEN ARASU, A., NATARAJAN, M., “A comprehensive microstructural analysis for enhancing concrete’s longevity and environmental sustainability”, *Journal of Environmental Nanotechnology*, v. 13, n. 2, pp. 368–376, 2024. doi: <http://doi.org/10.13074/jent.2024.06.242584>.
- [23] REN, J., XU, L., “On vectorization of deep convolutional neural networks for vision tasks”, *Proceedings of the AAAI Conference on Artificial Intelligence*, v. 29, n. 1, pp. 1840–1846, 2015. doi: <http://doi.org/10.1609/aaai.v29i1.9488>.

- [24] ZHANG, L., YANG, F., ZHANG, Y., *et al.*, “Road crack detection using deep convolutional neural network”, In: *Proceedings of the 2016 IEEE International Conference on Image Processing (ICIP)*, pp. 3708-3712, Phoenix, AZ, USA, 2016. doi: <http://doi.org/10.1109/ICIP.2016.7533052>.
- [25] DOULAMIS, A., DOULAMIS, N., PROTOPAPADAKIS, E., *et al.*, “Combined convolutional neural networks and fuzzy spectral clustering for real time crack detection in tunnels”, In: *Proceedings of the 2018 25th IEEE International Conference on Image Processing (ICIP)*, pp. 4153-4157, Athens, Greece, 2018. doi: <http://doi.org/10.1109/ICIP.2018.8451758>.
- [26] GIBB, S., LA, H.M., LOUIS, S., “A genetic algorithm for convolutional network structure optimization for concrete crack detection”, In: *Proceedings of the 2018 IEEE Congress on Evolutionary Computation (CEC)*, pp. 1-8, Rio de Janeiro, Brazil, 2018. doi: <http://doi.org/10.1109/CEC.2018.8477790>.
- [27] ÖZGENEL, Ç.F., GÖNENÇ SORGUÇ, A., “Performance comparison of pretrained convolutional neural networks on crack detection in buildings”, In: *Proceedings of the 35th International Symposium on Automation and Robotics in Construction*, p. 1-8, Berlin, Germany, 2018. doi: <http://doi.org/10.22260/ISARC2018/0094>.
- [28] DORAFSHAN, S., THOMAS, R.J., MAGUIRE, M., “Comparison of deep convolutional neural networks and edge detectors for image-based crack detection in concrete”, *Construction & Building Materials*, v. 186, pp. 1031–1045, 2018. doi: <http://doi.org/10.1016/j.conbuildmat.2018.08.011>.
- [29] ARASU, A., NAVEEN, N., MUTHUSAMY, B., *et al.*, “Optimization of high performance concrete composites by using nano materials”, *Research on Engineering Structures and Materials*, v. 9, n. 3, pp. 843–859, 2023.
- [30] KANG, S.-M., CHUN, C.-J., SHIM, S.-B., *et al.*, “Real time image processing system for detecting infrastructure damage: crack”, In: *Proceedings of the 2019 IEEE International Conference on Consumer Electronics (ICCE)*, pp. 1-3, Las Vegas, NV, USA, 2019. doi: <http://doi.org/10.1109/ICCE.2019.8661830>.
- [31] GEETHALAKSHMI, S., “A survey on crack detection using image processing techniques and deep learning algorithms”, *International Journal of Pure and Applied Mathematics*, v. 118, n. 8, pp. 215–220, 2018.
- [32] TURKAN, Y., HONG, J., LAFLAMME, S., *et al.*, “Adaptive wavelet neural network for terrestrial laser scanner-based crack detection”, *Automation in Construction*, v. 94, pp. 191–202, 2018. doi: <http://doi.org/10.1016/j.autcon.2018.06.017>.
- [33] DUNG, C.V., ANH, L.D., “Autonomous concrete crack detection using deep fully convolutional neural network”, *Automation in Construction*, v. 99, pp. 52–58, 2019. doi: <http://doi.org/10.1016/j.autcon.2018.11.028>.
- [34] HSIEH, Y.-A., TSAI, Y.J., “Machine learning for crack detection: review and model performance comparison”, *Journal of Computing in Civil Engineering*, v. 34, n. 5, p. 04020038, 2020. doi: [http://doi.org/10.1061/\(ASCE\)CP.1943-5487.0000918](http://doi.org/10.1061/(ASCE)CP.1943-5487.0000918).
- [35] FLAHL, M., SULEIMAN, A.R., NEHDI, M.L., “Classification and quantification of cracks in concrete structures using deep learning image-based techniques”, *Cement and Concrete Composites*, v. 114, p. 103781, 2020. doi: <http://doi.org/10.1016/j.cemconcomp.2020.103781>.
- [36] BOROS, N., KALLÓS, G., BALLAGI, A., “Implementation of trajectory planning algorithms for track serving mobile robot in ROS 2 ecosystem”, *Tehnicki Vjesnik*, v. 30, n. 4, pp. 1020–1028, 2023.
- [37] SUBHANI, T., ALGHAMDI, A.S., KHALIQ, A., *et al.*, “Investigating the post-sintering thermal and mechanical treatments on the properties of alumina reinforced aluminum nanocomposites”, *Tehnicki Vjesnik*, v. 30, n. 4, pp. 1003–1009, 2023.
- [38] ARASU, A., NATARAJAN, M., BALASUNDARAM, N., *et al.* “Development of high-performance concrete by using nanomaterial graphene oxide in partial replacement for cement”, *AIP Conference Proceedings*, v. 2861, n. 1, p. 050008, 2023.
- [39] NGUYEN, S.D., TRAN, T.S., TRAN, V.P., *et al.*, “Deep learning-based crack detection: a survey”, *International Journal of Pavement Research and Technology*, v. 16, n. 4, pp. 943–967, 2023. doi: <http://doi.org/10.1007/s42947-022-00172-z>.