

Automated crack detection and crack depth prediction for reinforced concrete structures using deep learning



Laxman K C^a, Nishat Tabassum^b, Li Ai^{a,*}, Casey Cole^b, Paul Ziehl^{a,c}

^a Department of Civil and Environmental Engineering, University of South Carolina, Columbia, SC, USA

^b Department of Computer Science and Engineering, University of South Carolina, Columbia, SC, USA

^c Department of Mechanical Engineering, University of South Carolina, Columbia, SC, USA

ARTICLE INFO

Keywords:
 Concrete crack
 Inspection
 Deep learning
 CNN
 Random Forest
 XGBoost

ABSTRACT

Automatic inspection for crack detection and estimation of the crack depth is critical in assessing the damage and determining the appropriate method of repair in concrete structures. Most of the studies which have employed deep learning models for automatic inspection are limited to the detection and estimation of the width, length, area, and direction of cracks. The innovation of this study lies in developing a comprehensive automated crack detection and crack depth evaluation framework for concrete structures using images taken from portable devices. Firstly, a binary-class Convolutional Neural Network (CNN) model was developed to automatically detect the cracks on a concrete surface. Secondly, an integrated CNN model combining the convolutional feature extraction layers and regression models (RF and XGBoost) was developed to automatically predict the depth of the cracks. The proposed framework has been validated on a reinforced concrete (RC) slab. Results indicate the models are accurate and reliable for automated inspection of the cracks which could help in evaluating the condition of a concrete structure and choosing suitable repair methods.

1. Introduction

Concrete is a widely used component in the construction of infrastructures such as bridges, and buildings. Concrete structures degrade over time due to harsh environmental conditions, overloading, and material deterioration [1–3]. Cracks ranging from micro-cracks to macro-cracks are the first sign of distress in these structures. The propagation of the cracks may reduce the strength of the concrete structure and may affect its structural integrity [4,5]. Water and corrosive chemicals can invade the concrete through these cracks. When the cracks reach the rebar level, corrosion of the rebars can be induced. Delamination and spalling of the concrete can occur due to the development of corrosion, undermining the safety and serviceability of the concrete structure [6–8]. Inspection for crack detection in concrete structures is crucial to determining the presence of damage and condition assessment [9,10]. Assessment of the crack depth is vital in determining the appropriate repair method to prevent substantial damage and ensure public safety.

Inspection of concrete structures is usually accomplished by manual visual inspection of concrete surfaces. However, it is time-consuming, labor-intensive, and poses a risk to the safety of inspectors [11,12].

This method is also subjective to the skills, expertise, and experience of the engineers in charge of the inspection. To overcome these limitations, automated methods of inspection have been explored using unmanned aerial vehicles and machine learning algorithms. Automation in inspection has the advantage of being cost-effective, efficient, using less labor and reducing the risk of accidents in the workplace [13,14]. Moreover, it can be objective and more reliable since the condition assessment is designated through a computer algorithm.

Many studies have investigated the possibility of automated inspection of concrete structures using image processing techniques (IPT) and deep learning methods. Abdel-Qader et al., (2003) [15] implemented an edge detection method based on the IPT to automatically detect the presence of cracks in the images from a concrete bridge. Algorithms for edge detection were analyzed and their effectiveness in crack detection was compared with the highest accuracy of 86 % for crack identification. However, edge detection methods are susceptible to noise in the image caused by texture, shadows, blemishes, and illumination because of the sharp change in brightness [16]. A percolation-based technique of image processing was proposed for crack detection [17]. As the method depends on the shape and brightness of the cracks, it could detect long cracks, but small and unclear ones were detected as

* Corresponding author.

E-mail address: ail@email.sc.edu (L. Ai).

noise. Image binarization was implemented for the identification of cracks in the image of a concrete surface and to estimate the width of the crack [18]. Although crack widths were estimated with an error of less than 11 %, the method was not able to detect small cracks present in the blurred images. Kim et al., (2017) [19] applied a hybrid image binarization technique to detect small cracks and extract the width and length of the crack. However, there is no standardized method to set a threshold and choose the best parameters for feature extraction making it a tedious, inefficient, and costly process.

To overcome the limitations of IPTs, deep learning has been employed for the automated inspection of cracks in concrete structures [20–22]. It is capable of automatically extracting features from images and detecting defects with greater accuracy and robustness than traditional IPTs [23,24]. A deep learning method, CNN, is efficient for image classification [25–27]. It can differentiate and accurately classify images belonging to multiple classes. It is successful in large-scale object detection problems and requires lower computations [28–30]. The integrated CNN, developed based on CNN, is a hybrid deep learning algorithm where CNNs are integrated with alternate classifiers to detect and quantify cracks in concrete surfaces [31,32]. Random Forest (RF) and extreme gradient boosting (XGBoost) are used as the alternate classifiers for crack inspection due to their simplicity, effectiveness, ease of implementation, and outstanding generalization performance [33,34].

In concrete structures, CNNs and integrated CNNs have shown success in the detection and quantification of cracks [35,36]. Cha et al. (2017) [37] implemented CNNs for crack detection and compared their performance with traditional methods. The Sobel and Canny edge detection algorithms produced no meaningful results, an elevated level of noise, and inaccurate results in detecting the crack. These techniques were found to be dependent on lighting conditions and the shape of the crack. CNN showed better performance under different lighting conditions and produced clear information on the cracks. Kim and Cho (2018) [38] proposed an automated crack detection method to classify concrete surface images obtained from the site based on CNNs and transfer learning. A CNN classifier was developed that had an accuracy of 98 % during validation. To verify its effectiveness, images taken from an on-site concrete structure using smartphone and DSLR cameras were evaluated. An average accuracy of 97 %, was achieved for all the test images. A CNN architecture was used to classify the concrete damage from the images collected from the internet [39]. Images were preprocessed based on their suitability; a larger dataset was obtained by data augmentation. A pre-trained model based on CNN classified these images with an accuracy of 92.57 %. A CNN integrated with an RF classifier was used for crack detection in concrete structures [40]. The CNN with a softmax layer detected the cracks with an accuracy of 88 %, while the integrated CNN had an accuracy of 89 % for crack detection.

Along with the detection of cracks in the images, quantitative information about the crack, such as its length, width, depth, and orientation, is required to investigate and make an informed decision on the level of the damage [41–43]. Teng and Chen (2022) [44] proposed a fast and high-performing CNN model for detecting cracks and calculating the physical features, such as length, width, area, and cracking ratio. The proposed model detected the presence of the crack with an accuracy of 96.5 % for images with cracks and 98.7 % for images without cracks. The CNN model quantified the crack features, with an error of 7 % in determining length, 2 % in extracting average width, 22 % in measuring maximum width, 27 % in finding the area, and 25 % in quantifying cracking ratio. Zhao et al., (2022) [45] implemented feature pyramid networks (FPN) for the quantification of the crack in the concrete structure. Crack-FPN effectively detected cracks having different widths and estimated the width of the cracks with an error of 5 %. An automatic method based on CNN was developed to calculate the width of the cracks from the images collected from the concrete structure [46]. The method is highly accurate in calculating the width of the crack, with a relative error of 3.87 %.

Automatic inspection for crack detection is important in assessing the condition of a concrete structure [47,48]. Inspection of the depth of the crack is important to determine its structural integrity. The depth of the concrete crack is also necessary to determine and apply an appropriate repair technique (surface treatment using coatings or surface sealers, injecting epoxy or grouts, or an extensive repair with replacement concrete or shotcrete) to avoid further damage to the concrete structure. Various algorithms based on IPTs have been developed and implemented for inspection. To address the limitations of traditional IPTs, CNN, and integrated CNN have been used for concrete crack inspection and quantification. However, these algorithms have only focused on crack detection and the determination of the length, width, area, and orientation of the crack. To the best of the author's knowledge, there are no autonomous inspection techniques that estimate the depth of the crack from the images of concrete structures. To fill this gap, this study aims to explore the possibility of using a CNN model to automatically detect the presence of cracks and develop an integrated CNN model with regression models to predict the depth of the crack.

2. Overview of the complete framework

Detecting cracks and estimating the depth of the cracks are critical steps in preventing further damage and ensuring public safety. This study focuses on the autonomous inspection of concrete cracks without the need for human intervention while improving efficiency, rationality, and accuracy. Machine learning methods (CNN, RF, and XGBoost) are explored to develop a comprehensive framework to detect cracks and predict the depth of the crack.

First, images of concrete surfaces are obtained from a public database. A binary-class CNN model was trained, validated, and tested on these images for crack detection. An accurate model was developed which was then implemented to detect the cracks from the images collected from a damaged RC slab. The reliability of the developed model is checked by evaluating its performance in classifying these new images.

An integrated CNN model with regression models is developed for crack depth prediction. The first network for crack detection uses a CNN architecture to extract and learn the features from the images of the concrete surface and classifies the images to their respective labels. The second network for crack depth prediction uses the same CNN architecture to extract the features from the images of concrete surfaces. The images with known crack depths are linked to their corresponding crack depths. The alternate classifiers, Random Forest and XGBoost are then used to train, validate and test the images for depth prediction.

A framework with two models for crack detection and crack depth

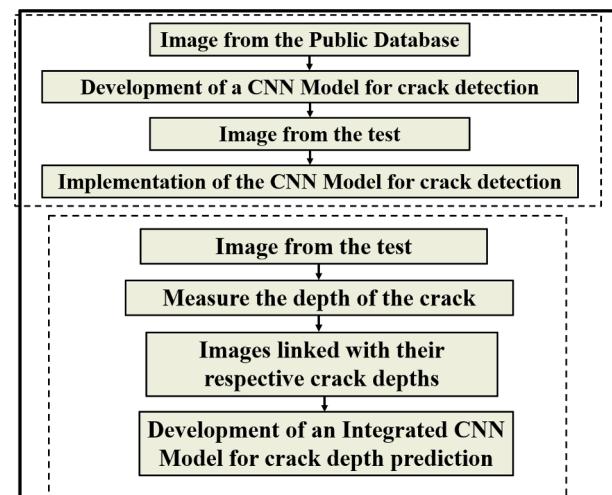


Fig. 1. Overview of the complete framework.

prediction is developed as illustrated in Fig. 1.

3. Machine learning models

3.1. Convolutional neural network model

A CNN model is developed for the classification of images by feature extraction and learning [49,50]. It consists of an input layer, feature extraction and classification layers of CNN architecture, and an output layer, as shown in Fig. 2. The image dataset is fed to the model in the input layer. The images go through the CNN architecture, where the features are extracted and learned by the neural network. The features of the images are interpreted by the last layer and classified into the respective output labels.

A CNN architecture is comprised of a convolution layer, pooling layer, activation layer, auxiliary layer, fully connected layer, and softmax layer. A convolution layer consists of a kernel that is multiplied on an element-to-element basis with an input image. It slides with a pre-determined slide on the image regions, extracting the features from an input image to create a feature map. The feature map then goes to the pooling layer for additional feature extraction and data filtering. The layer reduces the size of the feature map and lowers the cost of computation by taking maximum values or mean values, referred to as “max pooling” and “mean pooling”, respectively. Nonlinearity in the architecture to extract the nonlinear features and improve the CNN architecture is introduced in the activation layer. Sigmoid, tanh, and rectified linear unit (ReLU) are some of the typical nonlinear activation functions. Auxiliary layers consist of drop-out layers and batch normalization to overcome the issue of overfitting and train the image dataset efficiently and effectively. The fully connected layer flattens the extracted features to a single vector and produces a label for the image. The softmax layer predicts the class labels and classifies the input image to their respective labels based on their features.

3.2. An integrated CNN model

An integrated CNN model consists of a CNN architecture that is integrated with a regression model. Input data is fed into the CNN architecture, which extracts data features. These features with their corresponding labels are input into the regression models. The labels are important in feeding the regression model so that the model can understand the relationship between the independent and dependent variables. The independent variables input into the regression models in this case are the features from extracted by CNN architecture, and the outputs are the predicted labels, which are the depths of the cracks. The regression models utilized in this study are Random Forest and Extreme Gradient Boosting (XGBoost). Details of the regression models are presented in Sections 3.3 and 3.4.

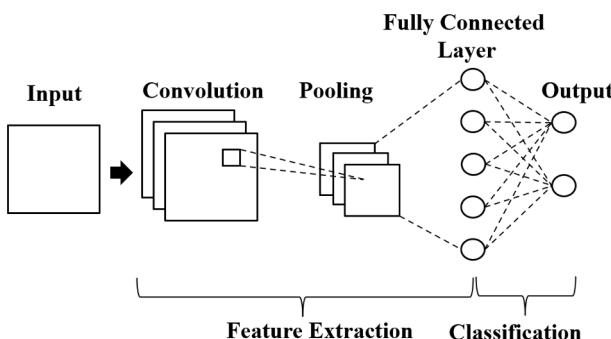


Fig. 2. Overall architecture of a CNN model.

3.3. Random Forest

Random Forest is an ensemble learning method leveraging the bagging technique [51,52]. The model combines the mean and average forecasts of numerous decision trees to produce more accurate predictions than a single model or tree. In this study, the bootstrapping process is used to construct an input dataset for each decision tree by randomly selecting data from the features dataset extracted by the CNN structure. Some samples may be chosen frequently, while others may not be chosen at all. As a result, the input data for each tree is unique. This increases the diversity of the individual decision trees. Taking a vote or averaging the results generated by these decision trees can reduce the risk of overfitting from a single decision tree when the final prediction is made. The structure of the RF model is illustrated in Fig. 3.

3.4. Extreme gradient boosting

XGBoost is another type of ensemble technique that utilizes the gradient boosting technique [53,54]. Boosting is a technique that reduces the bias of the model in training and minimizes overall prediction error. The decision trees in XGBoost are created in sequential order, and the weight of the variables for each ensemble member is not the same throughout, whereas the weights in bagging in RF were the same. As more decision trees are added, the model focuses on the variables that produced errors and places a higher weight on those variables. The focus is on the higher weight variables which are then fed again to a subsequent decision tree to fit and correct the prediction errors made in the previous model. The residuals from the previous trees are used in the subsequent trees to rectify the performance of the model. The sum of the results of all the trees is used to make the final prediction. The structure of a XGBoost is shown in Fig. 4.

4. Execution and performance evaluation of the models

Keras and TensorFlow were used to develop and implement the deep learning models in this study. Keras was selected because the framework is easy to use and it also gives clear feedback when there is a user error, so it makes it easy to learn [55].

A training and validation loss plot, a training and validation accuracy plot, and a confusion matrix obtained from the CNN model are plotted. These plots are used to analyze the performance of the model for crack detection.

The training and validation loss plot measures how well the model

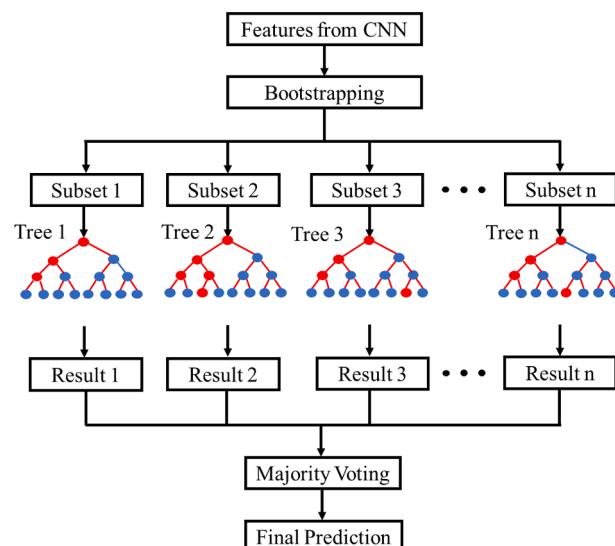


Fig. 3. Architecture of a RF.

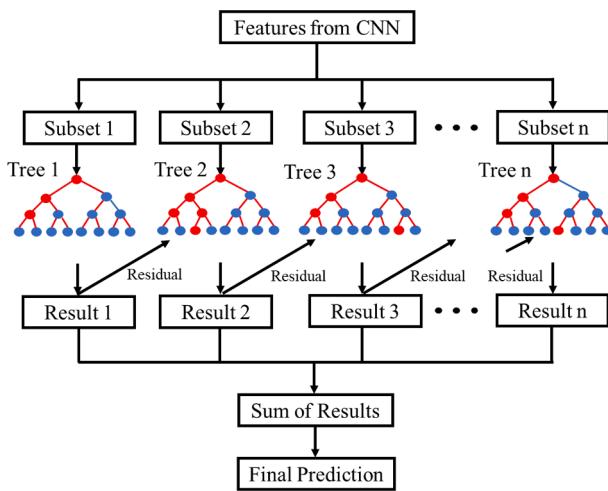


Fig. 4. Architecture of a XGBoost.

fits the training and validation dataset. The training loss shows how well the model fits the training dataset and the validation loss shows how well the model fits new data. The classification model's training and validation accuracy track the training and validation accuracy across epochs. Confusion matrices show the actual values versus the predicted values in solving classification problems. Recall, precision, and overall accuracy are calculated to evaluate the performance of the model. Recall is the ratio of the number of images correctly classified to a data label to the total number of images in that label. Precision is the ratio of the number of images correctly classified to a data label to the total number of images classified to that label. The accuracy is the ratio of the total number of correctly classified images to the total number of images.

For the analysis of the results, two metrics, mean square error (MSE), and correlation coefficient (R^2) values are used. A plot of actual versus predicted values is also produced to compare the accuracy of the predictions to the actual values. The MSE represents the average of the squared distance between the original and predicted values in the dataset, and it measures the variance of the residuals. Residuals are how far away the data points are in respect to the regression line. The R^2 value represents how well the model accounts for the variance to see how many points fall onto the regression line, so the higher the R^2 value, the better the predictions.

5. Development of a binary-class CNN model for crack detection

Images of concrete surfaces were obtained from a public dataset. A binary-class CNN model is developed for the classification of the images based on the presence of a crack. An optimized, accurate model is obtained by training, validating, and testing the model on these images for crack detection. The summary of the process for crack detection is presented in Fig. 5.

5.1. Image database

For crack detection, a binary-class CNN model was developed to recognize the features of images with and without cracks. An online

public dataset containing images of concrete surfaces with and without cracks was used for training, validating, and testing the model. These images were captured in campus buildings at the Middle East Technical University and retrieved from Kaggle [56]. There are 40,000 images of 227x227 pixel resolution with RGB channels. These images were split into two sets: the images without cracks are labeled "Undamaged," and the images with cracks are labeled "Cracked." Each set contains 20,000 images. Some of the typical images with and without cracks that are used in the study are exhibited in Fig. 6.

5.2. Binary-class CNN model

In the model, the dimension of the input layer is $120 \times 120 \times 3$, where 120 is the target size of images and three represents the red, green, and blue color modes. The dimension of the input layer was reduced to 120×120 to improve efficiency and reduce the computational time of the model. The kernel size of 3x3 was selected to limit the number of unrelated features that could be filtered. The filter size is sixteen and thirty-two for the first and second layer, respectively. In the first layer, sixteen lower-level features (edges or lines in the image) are extracted. In the second layer, thirty-two higher level features (crack patterns, shapes, and sizes) are extracted from the previous sixteen features. The maximum element from the region of the feature map under the filter was calculated using a max pooling layer; the result of the max pooling layer provides a feature map containing the most prominent features from the preceding feature map. A feature map is the outcome of extracting features from input photos using the filter. A global average pooling layer was also used, which computes the average value of all feature map elements [57]. The Adaptive Moment Optimization (Adam) optimizer is one of the two arguments needed to compile the Keras model. The Adam optimizer was chosen for its adaptability to smaller datasets, faster computational time, and fewer parameter needs for tuning [58]. The final activation function used was the sigmoid function. Binary cross entropy was used for the loss function, which is specifically tuned for binary classification [59]. After the model is compiled, the model is fitted to the training set.

5.3. Results of crack detection

A binary-class CNN model is developed with the images of the concrete surfaces as the input, a CNN architecture for feature extraction and classification, and an output layer that assigns the image output labels (Fig. 7). The model was trained with 22,400 images, validated with 5,600 images, and tested with 12,000 images. The images are fed to the model, which extracts and learns the respective features and classifies the images to their respective labels— "Undamaged" and "Crack."

The training and validation loss plot in Fig. 8(a) demonstrates that the model is well-fitted to the new data in the validation dataset; therefore, the binary-class CNN model produced fine results for crack detection. The training loss is lower than the validation loss, which is expected because the validation data set is new, so it does not perform better than the training dataset. As the number of epochs increases, the training and validation loss both decrease; therefore, the model is better fitting as the number of epochs increases.

The training and validation accuracy increases as the number of epochs increases (Fig. 8 (b)). The validation accuracy is slightly lower than the training accuracy and does not vary. It indicates that the model

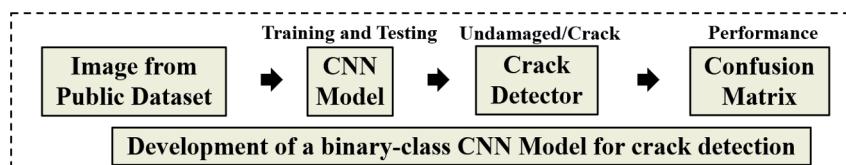


Fig. 5. Summary of crack detection.

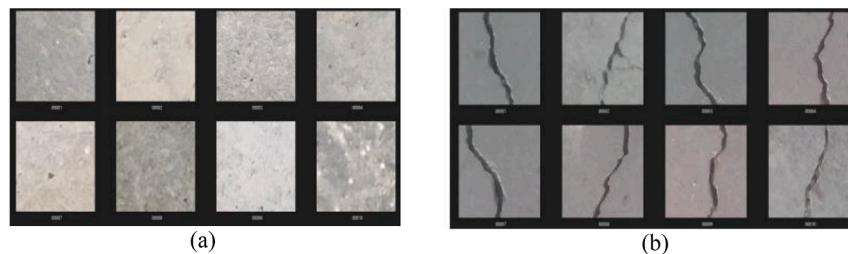


Fig. 6. Representation of images from Kaggle (a) without cracks, Undamaged (b) with cracks, Cracked [42].

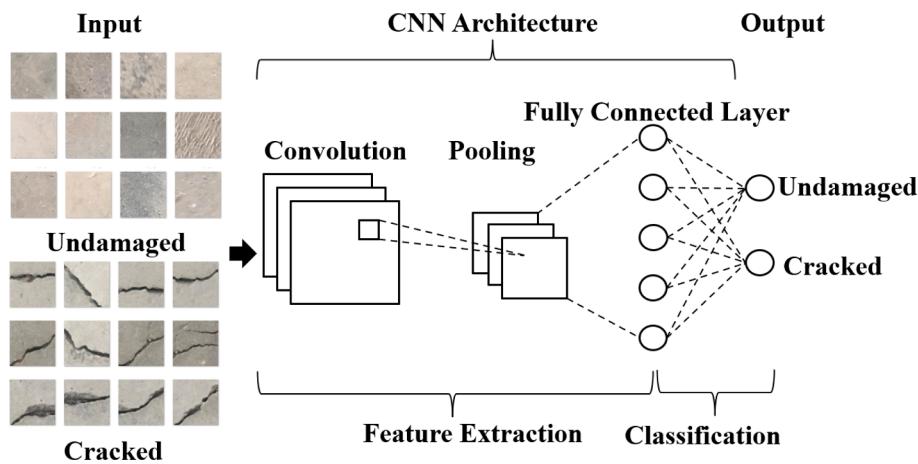


Fig. 7. Development of a binary-class CNN model for crack detection on the images from Kaggle [42].

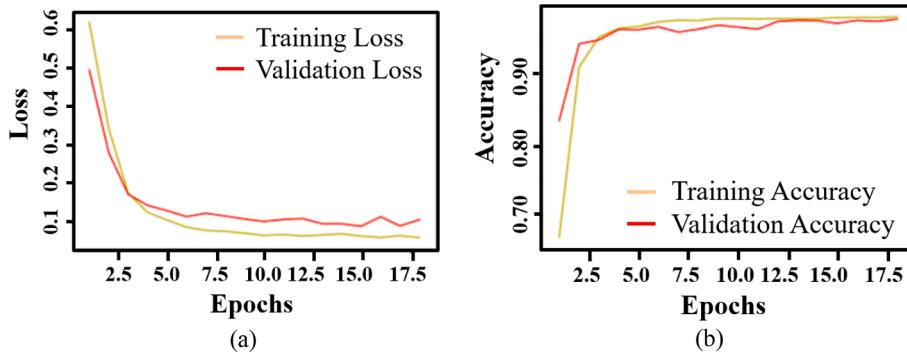


Fig. 8. Relationship of (a) Training and Validation Loss (b) Training and Validation Accuracy.

is accurate in classifying new data in the validation dataset based on what it learned in the training dataset. Both accuracies are converging to one as the epochs increase, so it shows that the training dataset trained the model well and produced accurate predictions for crack detection in the validation dataset. These graphs demonstrate that the binary classification model produces reliable results in detecting the cracks in the input images.

The trained model is tested with 12,000 images from the image dataset. A confusion matrix shown in Fig. 9 demonstrates the accuracy of the predictions. Out of 6,000 images labeled as "Undamaged", the model correctly predicted 5,996 images as "Undamaged", whereas 4 images were incorrectly classified as "Cracked". Out of 6,000 images labeled as "Cracked", the model correctly predicted 5,992 images as "Cracked" and incorrectly classified 8 images as "Undamaged". A total of 11,988 images out of 12,000 were correctly predicted by the binary-class CNN model producing a test accuracy of 99.9 % on crack detection.

6. Implementation of the binary-class CNN model

The binary-class CNN model developed for crack detection is implemented for crack detection in the new images acquired from a slab tested in the lab. The reliability of the model is investigated by testing the model with the new images obtained from the slab. The summary of the process is shown in Fig. 10.

6.1. Image database

A slab with length, width, and depth of 427 cm, 152 cm, and 21 cm was tested under monotonic loading in the laboratory at the University of South Carolina (Fig. 11 (a)). The loading procedure generated cracks in the slabs. The cracks were of varying lengths, widths, depths, and shapes (Fig. 11 (b)). The images of the surface of the slab with and without the cracks were taken using a cell phone. Some of the typical images are shown in Fig. 12 (a) and Fig. 12 (b). 2,574 images of the concrete surface were equally divided into two folders namely,

Actual State	Recall		
	Undamaged	Cracked	Precision
Undamaged	5996	4	99.8%
Cracked	8	5992	99.9%
Precision	99.9%	99.8%	99.9%

Undamaged Cracked
Predicted State

Fig. 9. Confusion Matrix of the binary-class CNN model during development.

“Undamaged” and “Cracked”, based on the presence of the crack in the images. These images were then input to the binary-class CNN model for classification. The reliability of the model was checked based on its ability to classify the new images.

6.2. Results for the implementation of the binary-class CNN model for crack detection

The model was implemented to classify the images obtained from the damaged slab (Fig. 13). Images obtained in the lab were input into the model that assigned them their respective labels based on the presence of cracks. The model classified 2,574 images and assigned the images “Undamaged” or “Cracked”. The ability of the model to classify these new images was used to evaluate the reliability of the model.

A confusion matrix shown in Fig. 14 demonstrates the performance of the binary-class CNN model. Out of 1,287 images labeled as Undamaged, the model correctly predicted 1,132 images as “Undamaged” whereas, 155 images were incorrectly classified as “Cracked”. Out of 6,000 images labeled as “Cracked”, the model correctly predicted 1,280 images as “Cracked” and incorrectly classified seven images as “Undamaged”. A total of 2,412 images out of 2,574 images were correctly

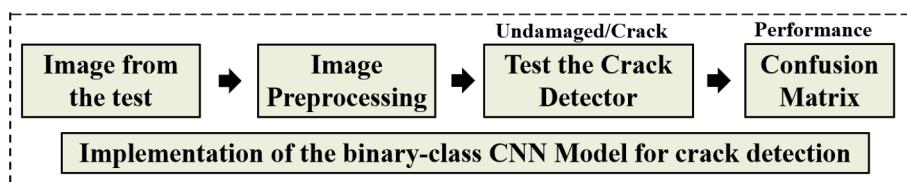


Fig. 10. Summary of crack detection.



(a)



(b)

Fig. 11. (a) Test setup of the RC slab (b) Damaged condition of the slab after the test.



Fig. 12. Images obtained from the damaged RC slab (a) without cracks (b) with cracks.

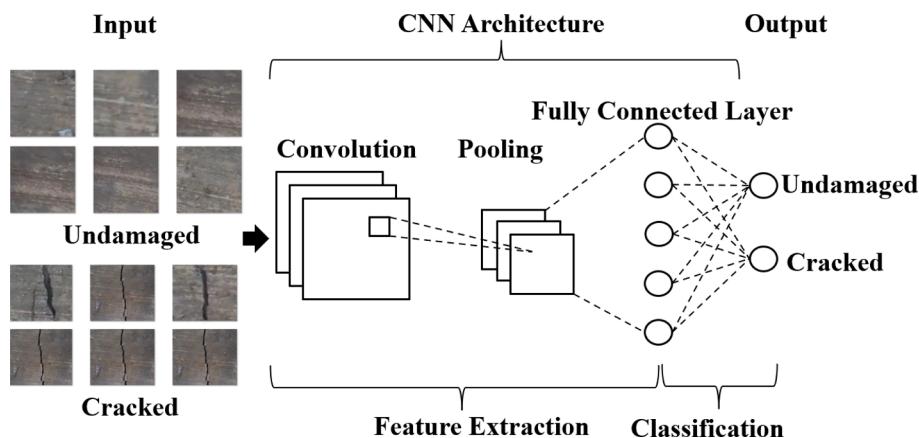


Fig. 13. Implementation of a binary-class CNN model for crack detection.

		Recall		
Actual State	Undamaged	1132	155	87.9%
	Cracked	7	1280	99.5%
Precision	99.4%	89.2%	93.7%	
	Undamaged	Cracked		
Predicted State				

Fig. 14. Confusion Matrix of the binary-class CNN model during implementation.

predicted, with an accuracy of 93.7 % in classifying the new images taken from the damaged RC slab. Hence, a reliable model was developed with an accuracy of 93 % to detect the presence of cracks.

In the study, a binary-class CNN model was developed for automatic crack detection. The model was developed by training and testing the images in a public dataset. Since the dataset has smooth images with perfect lighting conditions and without any noise, the model has high accuracy. To check the reliability of the developed model, it was implemented for the automatic detection of cracks in the images collected from a damaged RC slab. Even though the model was not trained or tested on these images, it could classify the images with an accuracy of more than 90 %. Hence, the model is reliable to be used for the automatic detection of cracks in the RC slabs.

7. Development of an integrated CNN model for crack depth prediction

After the crack detection, an integrated CNN model with regression models is used to predict the depth of the crack in the concrete images. The image dataset generated with the images from the damaged RC slab in the lab is used to develop the model for crack depth prediction. The images associated with their respective crack depths are used to train the models. The features of the images are extracted using a CNN model. These features are then fed to the regression models, which learn the trends for crack depth prediction. The two regression models that are

implemented in the study are: RF and XGBoost. The extracted features were used for training the model, making the predictions, and adjusting weights according to the data values to achieve higher accuracy. The process to obtain a model for crack depth prediction is illustrated in Fig. 15.

7.1. Image database

The image database was created with the images taken from the RC slab tested in the lab. There are multiple cracks of varying lengths and depths running along the surface of the slab. The cracks spanned across the width of the slab, so the lengths of the cracks are 130 cm to 152 cm inches long. The cracks were measured at the deepest part; however, the cracks do not have a uniform depth and varies throughout the length of the crack.

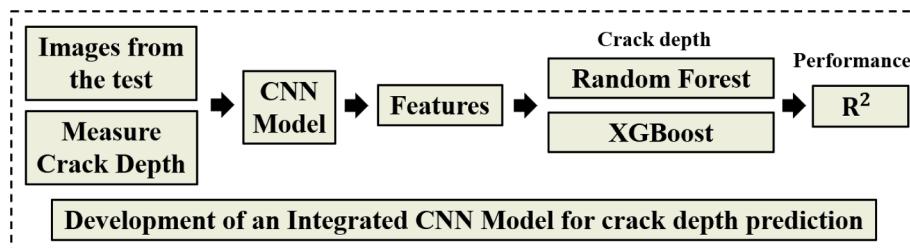
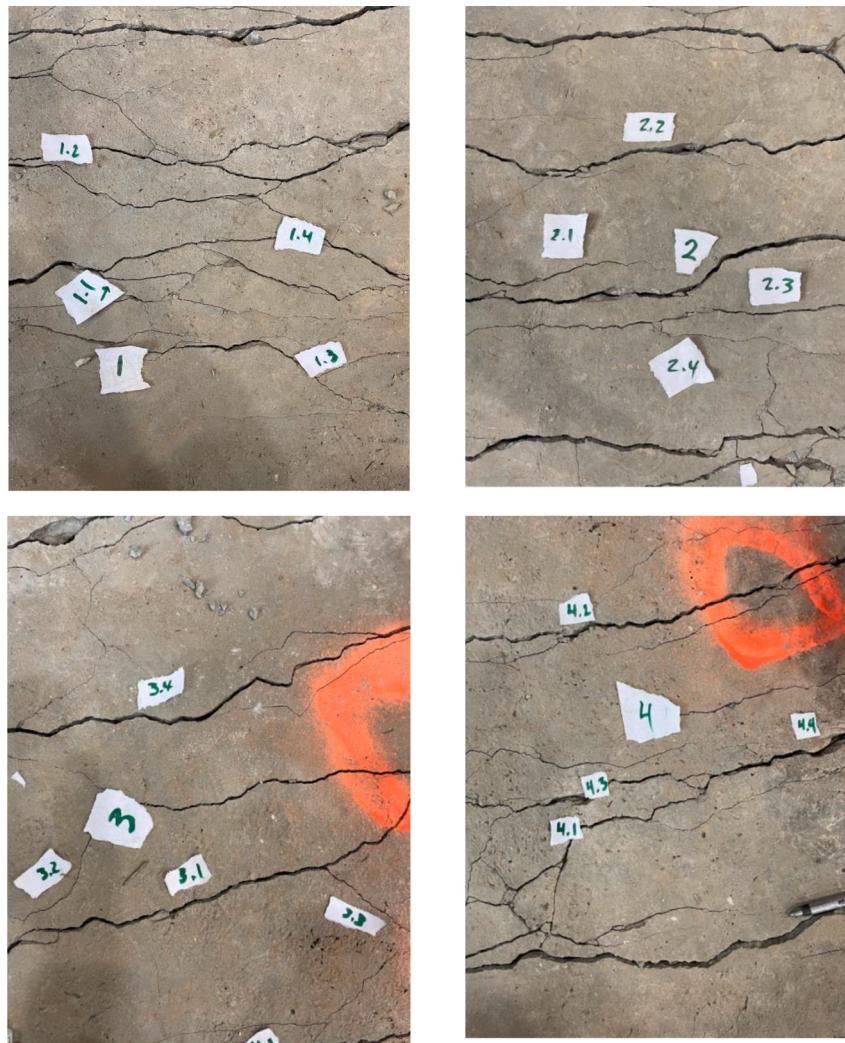
On each of the primary cracks, four secondary cracks (i.e., smaller cracks that stem or propagate from the primary crack) are chosen and labeled with decimals such that the secondary crack on the primary crack is labeled 1.1 to 1.4. This is demonstrated in Fig. 16.

Each primary and secondary crack is manually annotated to record the depth of the crack at a certain point. It was measured by taking a slip of paper and inserting it into the crack until it reached the bottom and marking it; then the length of the submerged part of the paper was measured. This process was repeated for five times in each place and the maximum depth of the crack was obtained. The repetition of the process was done to minimize the error and guarantee the depth estimation results. The depth values of each crack with its reference are depicted in Table 1.

Images of the concrete surface were collected from an RC slab using a cell phone. The phone was kept horizontally at 1.5 feet away from the surface of the cracks while taking the images. The images were taken during the day with sufficient lighting conditions. A total of fifteen raw images of the cracks in the slab were taken with an original pixel resolution of 3024×4032 . The cracks chosen on the slab had varying depths to account for a wider range in the dataset of images. This was done to increase accuracy for a better representation of the model with the cracks. Four primary cracks were chosen and labeled 1 to 4.

The fifteen raw images were cropped into smaller images to extract individual cracks from the large group of cracks and expanded to generate a larger database of images, as shown in Fig. 17. From the raw images, the individual primary and secondary cracks were cropped into smaller images, with the single crack itself. This process was repeated for all the raw images. This is done for the primary and secondary cracks of a crack and is demonstrated in the figure below.

The preprocessed dataset of the images of the slab is used in the integrated CNN model for crack depth prediction. After preprocessing and generating the dataset of images, the images of the crack with their

**Fig. 15.** Crack depth prediction.**Fig. 16.** Visual representation of primary crack and secondary cracks.

corresponding depths were used to train the integrated CNN model to recognize the features and predict the depth of the crack.

7.2. An integrated CNN model for crack depth prediction

The image dataset of the cracks with their corresponding depths is fed into a CNN model. The feature extraction layer of the CNN model extracts the features from the images. These features are input to the regression models for training and testing. The depth of the crack is predicted by the integrated CNN model (Fig. 18).

7.3. Results of an integrated CNN model for crack depth prediction

Two plots were created to show the relationship between the actual values and predicted values of crack depth using RF and XGBoost (Fig. 19). The figures show the relationship between the predicted depth values of the crack from the model and the actual known values. The closer the points are to the regression line, the more accurate the predictions are. Since XGBoost had the higher R^2 value, it indicates that more points in the plot of XGBoost are closer to the line of regression compared to the plot of actual versus predicted in RF.

The metric results for both models are shown in Table 2. The regression models were compared for their ability to predict the depth of the cracks from a dataset of images with known crack depths. Both

Table 1
Measurements of Crack Depths.

Crack Reference	Crack Depth (cm)
1	1.1
1.1	0.1
1.2	2.5
1.3	0.2
1.4	1.8
2	3.1
2.1	1.5
2.2	3.3
2.3	2
2.4	0.1
3	1
3.1	1.2
3.2	0.1
3.3	0.5
3.4	2.1
4	0.4
4.1	1.4
4.2	2.3
4.3	1.4
4.4	2.5

models produced satisfactory results with a MSE of 14.3 % and R^2 value of 0.88 for RF, and a MSE of 14 % and R^2 value of 0.89 for XGBoost while predicting the depth of the cracks. Both models are reliable in predicting crack depths from the images.

An integrated CNN model was developed for crack depth prediction. The model can predict the depth of the cracks from the images. The depth of the crack can be used to decide on an appropriate repair method. If a crack is shallow, it may be possible to repair it using a surface treatment method. If the crack is deeper, more extensive

methods may be opted for repair.

8. Summary and conclusions

A comprehensive framework implementing deep learning models was developed in the study for the inspection of RC structures for crack detection in the concrete surface and crack depth prediction. Images from a public dataset and images taken from a RC slab using a phone camera were utilized for training, validation, and testing the deep learning models. A binary-class CNN model was developed and implemented to detect the cracks in the images of concrete. Based on the features obtained from the feature extraction layer of the CNN model, an integrated CNN model with regression models, RF and XGBoost, was developed to predict the depth of the crack. The conclusions of the study are listed below:

1. A binary-class CNN model can be implemented to automatically detect the presence of cracks in RC structures. The model obtained an accuracy of 99.9 % on the public image dataset. In addition, the model trained on the public dataset could also detect the cracks on a damaged RC slab in the laboratory with an accuracy of 93.7 %, indicating a good generalization ability to implement it on a damaged structure.
2. An integrated CNN model can be employed to automatically predict the depth of the cracks in the images of RC structures. An integrated CNN model with RF predicted the depths of the cracks from the dataset with the MSE and R^2 values of 14.3 % and 0.88, respectively. Using an integrated CNN model with XGBoost the depths of the cracks from the dataset were predicted with the MSE and R^2 values of 14 % and 0.89, respectively.

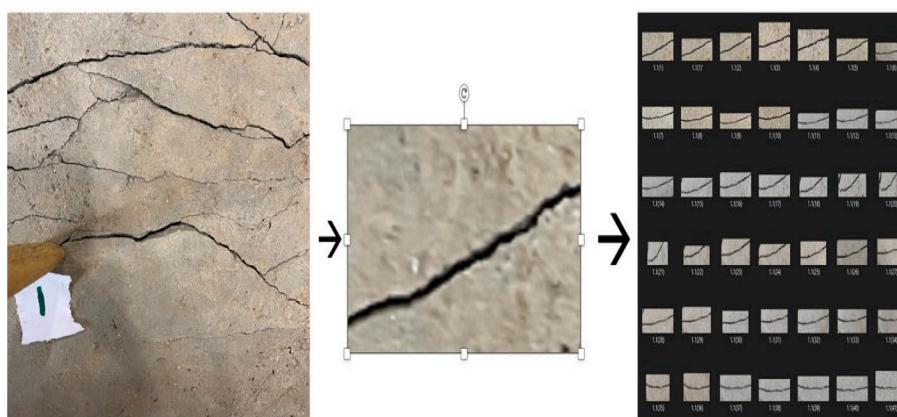


Fig. 17. Image Dataset.

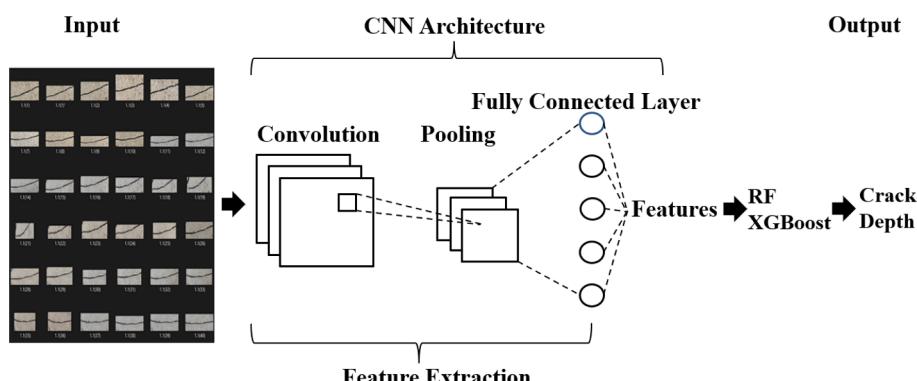


Fig. 18. Development of an Integrated CNN model for crack depth prediction.

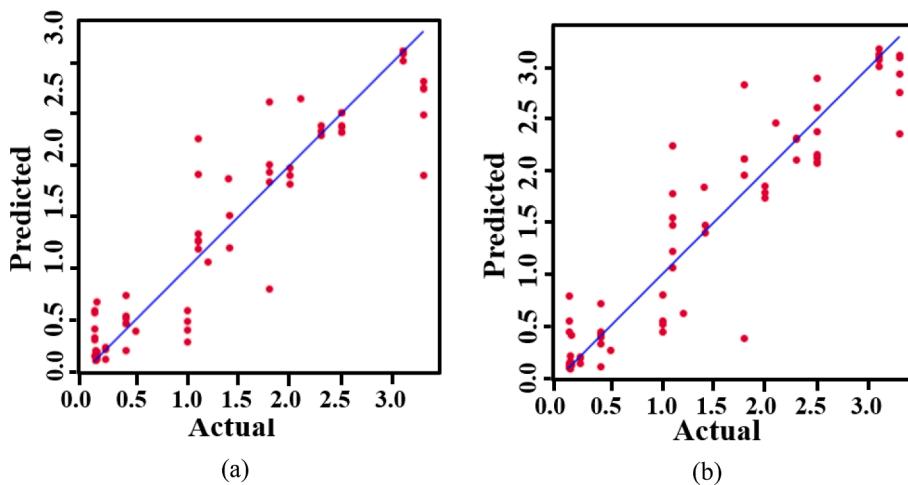


Fig. 19. Actual vs Predicted values for crack depth (a) RF (b) XGBoost model.

Table 2
Results of the RF and XGBoost Model.

Results:	RF	XGBoost
MSE	14.3 %	14.0 %
R ²	0.88	0.89

The current study has some limitations. The models were trained to detect and predict the crack depths generated by monotonic loading only. Studies are needed to validate the model on cracks generated due to other types of loadings such as cyclic loading. The models were trained and tested on an image dataset with a limited number of images, taken under good lighting conditions and with background noise eliminated. It is recommended that future studies consider a variety of illumination conditions and background noise. In this study, the maximum crack depth was assumed to be uniform along the length of the crack. However, the depth of the crack may vary along its length. More studies are needed in the future to investigate the actual crack depth along the crack length direction.

CRediT authorship contribution statement

Laxman K.C.: Conceptualization, Methodology, Formal analysis, Investigation, Writing – original draft. **Nishat Tabassum:** Conceptualization, Methodology, Investigation, Formal analysis, Writing – original draft. **Li Ai:** Conceptualization, Methodology, Investigation, Formal analysis, Writing – review & editing. **Casey Cole:** Conceptualization, Methodology, Investigation, Formal analysis. **Paul Ziehl:** Conceptualization, Methodology, Supervision.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

This research was partially supported by the South Carolina Department of Transportation (SCDOT) under contract number SPR No.752 and the Center for Connected Multimodal Mobility (C2M2) (Tier

1 University Transportation Center) Grant, under grant number 69A3551747117.

References

- [1] C. Koch, K. Georgieva, V. Kasireddy, B. Akinci, P. Fieguth, A review on computer vision based defect detection and condition assessment of concrete and asphalt civil infrastructure, *Adv. Eng. Inf.* 29 (2) (2015) 196–210.
- [2] X. Han, Z. Zhao, L. Chen, X. Hu, Y. Tian, C. Zhai, L. Wang, X. Huang, Structural damage-causing concrete cracking detection based on a deep-learning method, *Constr. Build. Mater.* 337 (2022), 127562, <https://doi.org/10.1016/J.CONBUILDMAT.2022.127562>.
- [3] L. K c, A. Ross, L.i. Aj, A. Henderson, E. Elbatanouny, M. Bayat, P. Ziehl, Determination of vehicle loads on bridges by acoustic emission and an improved ensemble artificial neural network, *Constr. Build. Mater.* 364 (2023) 129844.
- [4] H. Kim, E. Ahn, M. Shin, S.-H. Sim, Crack and Noncrack Classification from Concrete Surface Images Using Machine Learning, *Struct. Health Monit.* 18 (2019) 725–738, <https://doi.org/10.1177/1475921718768747>.
- [5] X. Tan, Y. Bao, Measuring crack width using a distributed fiber optic sensor based on optical frequency domain reflectometry, *Measurement* 172 (2021), 108945, <https://doi.org/10.1016/J.MEASUREMENT.2020.108945>.
- [6] S. Lin, Y. Wang, Crack-Depth Estimation in Concrete Elements Using Ultrasonic Shear-Horizontal Waves, *J. Perform. Constr. Facil.* 34 (2020), [https://doi.org/10.1061/\(asce\)cf.1943-5509.0001473](https://doi.org/10.1061/(asce)cf.1943-5509.0001473).
- [7] X.Y. Long, S.K. Zhao, C. Jiang, W.P. Li, C.H. Liu, Deep learning-based planar crack damage evaluation using convolutional neural networks, *Eng. Fract. Mech.* 246 (2021) 107604.
- [8] J. Tian, X. Wu, X. Tan, W.W. Wang, S. Hu, Y. Du, J. Yuan, W. Huang, X. Huang, Experimental study and analysis model of flexural synergistic effect of reinforced concrete beams strengthened with ECC, *Constr. Build. Mater.* 352 (2022), 128987, <https://doi.org/10.1016/J.CONBUILDMAT.2022.128987>.
- [9] J. Valen  a, I. Puente, E. J  lio, H. Gonz  lez-Jorge, P. Arias-S  nchez, Assessment of cracks on concrete bridges using image processing supported by laser scanning survey, *Constr. Build. Mater.* 146 (2017) 668–678.
- [10] X. Tan, L. Fan, Y. Huang, Y. Bao, Detection, visualization, quantification, and warning of pipe corrosion using distributed fiber optic sensors, *Autom. Constr.* 132 (2021), 103953, <https://doi.org/10.1016/J.AUTCON.2021.103953>.
- [11] T. Nishikawa, J. Yoshida, T. Sugiyama, Y. Fujino, Concrete Crack Detection by Multiple Sequential Image Filtering, *Comput. Aid. Civil. Infrastruct. Eng.* 27 (2012) 29–47, <https://doi.org/10.1111/j.1467-8667.2011.00716.x>.
- [12] W. Wang, C. Su, Semi-supervised semantic segmentation network for surface crack detection, *Autom. Constr.* 128 (2021) 103786.
- [13] N. Metni, T. Hamel, A UAV for bridge inspection: Visual servoing control law with orientation limits, *Autom. Constr.* 17 (2007) 3–10, <https://doi.org/10.1016/J.AUTCON.2006.12.010>.
- [14] S. Teng, Z. Liu, G. Chen, L.i. Cheng, Concrete crack detection based on well-known feature extractor model and the YOLO_v2 network, *Appl. Sci. (Switzerland)*, 11 (2) (2021) 813.
- [15] I. Abdel-qader, O. Abudayyeh, M.E. Kelly, Analysis of Edge-Detection Techniques for Crack Identification in Bridges, 17 (2003) 255–263.
- [16] D. Lattanzi, G.R. Miller, Robust Automated Concrete Damage Detection Algorithms for Field Applications, *J. Comput. Civ. Eng.* 28 (2) (2014) 253–262.
- [17] Y. Tomoyuki, H. Shuji, Improved percolation-based method for crack detection in concrete surface images, *Proc. Int. Conf. Pattern Recogn.* (2008), <https://doi.org/10.1109/icpr.2008.4761627>.
- [18] H. Kim, E. Ahn, S. Cho, M. Shin, S. Sim, Cement and Concrete Research Comparative analysis of image binarization methods for crack identification in

- concrete structures, *Cem. Concr. Res.* 99 (2017) 53–61, <https://doi.org/10.1016/j.cemconres.2017.04.018>.
- [19] H. Kim, J. Lee, E. Ahn, S. Cho, M. Shin, S.-H. Sim, Concrete crack identification using a UAV incorporating hybrid image processing, *Sensors (Switzerland)* 17 (9) (2017) 2052.
- [20] I.-H. Kim, H. Jeon, S.-C. Baek, W.-H. Hong, H.-J. Jung, Application of crack identification techniques for an aging concrete bridge inspection using an unmanned aerial vehicle, *Sensors (Switzerland)* 18 (6) (2018) 1881.
- [21] S.E. Park, S.H. Eem, H. Jeon, Concrete crack detection and quantification using deep learning and structured light, *Constr. Build. Mater.* 252 (2020), 119096, <https://doi.org/10.1016/J.CONBUILDMAT.2020.119096>.
- [22] S. Dorafshan, R.J. Thomas, M. Maguire, Comparison of deep convolutional neural networks and edge detectors for image-based crack detection in concrete, *Constr. Build. Mater.* 186 (2018) 1031–1045, <https://doi.org/10.1016/J.CONBUILDMAT.2018.08.011>.
- [23] A. Zhang, K.C.P. Wang, Y. Fei, Y. Liu, J.Q. Li, C. Chen, Automated Pixel-Level Pavement Crack Detection on 3D Asphalt Surfaces Using a Deep-Learning Network, 32 (2017) 805–819. <https://doi.org/10.1111/mice.12297>.
- [24] Y. Ren, J. Huang, Z. Hong, W. Lu, J. Yin, L. Zou, X. Shen, Image-based concrete crack detection in tunnels using deep fully convolutional networks, *Constr. Build. Mater.* 234 (2020), 117367, <https://doi.org/10.1016/J.CONBUILDMAT.2019.117367>.
- [25] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, *Nature* 521 (7553) (2015) 436–444.
- [26] W. Rawat, Z. Wang, Deep convolutional neural networks for image classification: A comprehensive review, *Neural Comput.* 29 (9) (2017) 2352–2449.
- [27] P.Y. Simard, D. Steinkraus, J.C. Platt, Best practices for convolutional neural networks applied to visual document analysis, in: Proceedings of the International Conference on Document Analysis and Recognition, ICDAR, 2003. <https://doi.org/10.1109/ICDAR.2003.1227801>.
- [28] A. Krizhevsky, I. Sutskever, G.E. Hinton, ImageNet classification with deep convolutional neural networks, *Commun. ACM* 60 (6) (2017) 84–90.
- [29] P. Sermanet, D. Eigen, OverFeat : Integrated Recognition , Localization and Detection using Convolutional Networks arXiv : 1312 . 6229v4 [cs . CV] 24 Feb 2014, ArXiv. (2014).
- [30] K. He, X. Zhang, S. Ren, J. Sun, Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition; Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition, *IEEE Trans. Pattern Anal. Mach. Intell.* 37 (9) (2015) 1904–1916.
- [31] Tang, Y. (2013). Deep learning using linear support vector machines. arXiv preprint arXiv:1306.0239.
- [32] C. Kyal, M. Reza, B. Varu, S. Shreya, Image-Based Concrete Crack Detection Using Random Forest and Convolution Neural Network, in: 2022. https://doi.org/10.1007/978-981-16-2543-5_40.
- [33] S. Mangalathu, H. Jang, S.-H. Hwang, J.-S. Jeon, Data-driven machine-learning-based seismic failure mode identification of reinforced concrete shear walls, *Eng. Struct.* 208 (2020) 110331.
- [34] M.J. Park, J. Kim, S. Jeong, A. Jang, J. Bae, Y.K. Ju, Machine Learning-Based Concrete Crack Depth Prediction Using Thermal Images Taken under Daylight Conditions, *Remote Sens. (Basel)* 14 (2022) 1–12, <https://doi.org/10.3390/rs14092151>.
- [35] V.P. Golding, Z. Gharineiat, H.S. Munawar, F. Ullah, Crack Detection in Concrete Structures Using Deep Learning, *Sustainability (Switzerland)* 14 (13) (2022) 8117.
- [36] G. Li, Q. Liu, S. Zhao, W. Qiao, X. Ren, Automatic crack recognition for concrete bridges using a fully convolutional neural network and naive Bayes data fusion based on a visual detection system, *Meas. Sci. Technol.* 31 (7) (2020) 075403.
- [37] Y.-J. Cha, W. Choi, O. Büyüköztürk, Deep Learning-Based Crack Damage Detection Using Convolutional, *Neural Netw.* 32 (5) (2017) 361–378.
- [38] B. Kim, S. Cho, Automated vision-based detection of cracks on concrete surfaces using a deep learning technique, *Sensors (Switzerland)* 18 (10) (2018) 3452.
- [39] P.D. Hung, N.T. Su, V.T. Diep, Surface Classification of Damaged Concrete Using Deep Convolutional Neural Network, 29 (2019) 676–687. <https://doi.org/10.1134/S1054661819040047>.
- [40] K. Chaiyasarn, A. Buatik, S. Likitlersuang, Concrete crack detection and 3D mapping by integrated convolutional neural networks architecture, *Adv. Struct. Eng.* 24 (2021) 1480–1494, <https://doi.org/10.1177/1369433220975574>.
- [41] P. Miao, T. Srimahachota, Cost-effective system for detection and quantification of concrete surface cracks by combination of convolutional neural network and image processing techniques, *Constr. Build. Mater.* 293 (2021), 123549, <https://doi.org/10.1016/J.CONBUILDMAT.2021.123549>.
- [42] W. Qiao, H. Zhang, F. Zhu, Q. Wu, P. Lonetti, A Crack Identification Method for Concrete Structures Using Improved U-Net Convolutional Neural Networks, *Math. Probl. Eng.* 2021 (2021) 1–16.
- [43] X. Tan, P. Guo, X. Zou, Y. Bao, Buckling detection and shape reconstruction using strain distributions measured from a distributed fiber optic sensor, *Measurement* 200 (2022), 111625, <https://doi.org/10.1016/J.MEASUREMENT.2022.111625>.
- [44] S. Teng, G. Chen, Deep Convolution Neural Network-Based Crack Feature Extraction, Detection and Quantification, *J. Fail. Anal. Prev.* 22 (2022) 1308–1321, <https://doi.org/10.1007/s11668-022-01430-9>.
- [45] W. Zhao, Y. Liu, J. Zhang, Y.i. Shao, J. Shu, Automatic pixel-level crack detection and evaluation of concrete structures using deep learning, *Struct. Control Health Monit.* 29 (8) (2022).
- [46] L. Song, H. Sun, J. Liu, Z. Yu, C. Cui, Automatic segmentation and quantification of global cracks in concrete structures based on deep learning, *Measurement (Lond)* 199 (2022), 111550, <https://doi.org/10.1016/j.measurement.2022.111550>.
- [47] X. Tan, A. Abu-Obieidah, Y. Bao, H. Nassif, W. Nasreddine, Measurement and visualization of strains and cracks in CFRP post-tensioned fiber reinforced concrete beams using distributed fiber optic sensors, *Autom. Constr.* 124 (2021), 103604, <https://doi.org/10.1016/J.AUTCON.2021.103604>.
- [48] H. Bai, D. Guo, W. Wang, X. Tan, M. Yan, G. Chen, Y. Bao, Experimental investigation on flexural behavior of steel-concrete composite floor slabs with distributed fiber optic sensors, *J. Build. Eng.* 54 (2022), 104668, <https://doi.org/10.1016/J.JOBEE.2022.104668>.
- [49] L. Ai, V. Soltanghae, P. Ziehl, Developing a heterogeneous ensemble learning framework to evaluate Alkali-silica reaction damage in concrete using acoustic emission signals, *Mech. Syst. Signal Process.* 172 (2022), 108981, <https://doi.org/10.1016/J.YMSSP.2022.108981>.
- [50] S. Albawi, T.A. Mohammed, S. Al-Zawi, Understanding of a convolutional neural network, in: Proceedings of 2017 International Conference on Engineering and Technology, ICET 2017, 2018. <https://doi.org/10.1109/ICEngTechnol.2017.8308186>.
- [51] L. Ai, V. Soltanghae, M. Bayat, B. Greer, P. Ziehl, Source localization on large-scale canisters for used nuclear fuel storage using optimal number of acoustic emission sensors, *Nucl. Eng. Des.* 375 (2021), 111097, <https://doi.org/10.1016/J.NUCENGDES.2021.111097>.
- [52] L. Breiman, Random forests, *Mach. Learn.* 45 (2001) 5–32.
- [53] Y. Wang, Z. Pan, J. Zheng, L. Qian, M. Li, A hybrid ensemble method for pulsar candidate classification, *Astrophys. Space Sci.* 364 (2019), <https://doi.org/10.1007/s10509-019-3602-4>.
- [54] L.i. Ai, V. Soltanghae, M. Bayat, M. Van Tooren, P. Ziehl, Detection of impact on aircraft composite structure using machine learning techniques, *Meas. Sci. Technol.* 32 (8) (2021) 084013.
- [55] K. Ramasubramanian, A. Singh, Deep Learning Using Keras and TensorFlow, *Mach. Learn. Using R* (2019), https://doi.org/10.1007/978-1-4842-4215-5_11.
- [56] Ç.F. Özgenel, Concrete crack images for classification, *Mendeley Data* 1 (1) (2018), <https://doi.org/10.17632/5y9wdsg2zt.2>.
- [57] T.-Y. Hsiao, Y.-C. Chang, H.-H. Chou, C.-T. Chiu, Filter-based deep-compression with global average pooling for convolutional networks, *J. Syst. Archit.* 95 (2019) 9–18.
- [58] V.-H. Nhu, N.-D. Hoang, H. Nguyen, P.T.T. Ngo, T. Thanh Bui, P.V. Hoa, P. Samui, D. Tien Bui, Effectiveness assessment of Keras based deep learning with different robust optimization algorithms for shallow landslide susceptibility mapping at tropical area, *Catena (Amst)* 188 (2020) 104458.
- [59] U. Ruby, Dr.A, Binary cross entropy with deep learning technique for Image classification, *Int. J. Adv. Trends Comput. Sci. Eng.* 9 (2020), <https://doi.org/10.30534/ijatcse/2020/175942020>.