



# CrackSpot: deep learning for automated detection of structural cracks in concrete infrastructure

R. Shashidhar<sup>1</sup> · D. Manjunath<sup>1</sup> · S. M. Shanmukha<sup>1</sup>

Received: 23 May 2023 / Accepted: 1 June 2023 / Published online: 27 June 2023  
© The Author(s), under exclusive licence to Springer Nature Switzerland AG 2023

## Abstract

To maintain structural integrity and avoid structural failures that could harm neighboring infrastructure, pollute the environment, and even result in fatalities, routine inspection and repair of concrete infrastructure are required. Throughout the structure's operational life, routine visual inspections are typically undertaken to detect various problems caused by environmental exposure (such as cracks, loss of material, rusting of metal bindings, etc.). Visual examination can yield a variety of data that may enable the cause of distress to be positively identified. Its effectiveness is subject to human error and depends on the investigator's skill and experience and because of their size and difficult-to-reach features, huge structures like dams, bridges, and tall skyscrapers can be prohibitively dangerous. The approach presented here uses deep learning techniques to identify the structural cracks on concrete surfaces to achieve easy detection of the cracks and high accuracy. Here, we propose an integrated Tensflow CNN and image processing-based crack-finding method to detect cracks with high precision. Thousands of photos of cracked and non-cracked structure surface datasets are considered while developing the model. Image features are extracted during pre-processing to increase training effectiveness. The developed model has a 97.11% accuracy rate and an F1-score of 97%. The results show that the designed model is highly precise and effective in identifying cracks in structures and more accurate than many implemented techniques.

**Keywords** Crack detection · Structural damage · Image processing · Structural load · Feature extraction · Convolutional neural network

## Introduction

This section gives a glimpse of what our proposed method is all about and it introduces the basics and the overview of the proposed technique. It also includes the motive for the proposed method and the necessity for the model implementation with objectives.

Effective damage detection techniques are necessary to identify damage quickly and avoid catastrophic failure. The examination includes detecting cracks in structures. In most

cases, structural cracks are examined by humans. During this type of examination, the cracks are noted along with the surrounding anomalies. The human method lags in the quantitative analysis because it is completely based on the experience and method of the examining person (Shan & Dewhurst, 1998). Thus, automated identification of cracks becomes one of the most important methods in examining the structures and at the same time a challenge for developing intelligent maintenance systems (Zhou et al., 2021).

Automated fracture detection has shown to be a difficult endeavor due to issues including noises in the captured images, non-uniformity, and other complex topologies.

This study introduces a convolutional neural network that can train on deep convolutional features in the images, improving the discriminant of captured image features along other complex conditions (Han et al., 2018). With the right data processing, it is possible to extract useful information from structures and use it for spotting cracks. This helps in taking early measures for maintenance and preventing any major damage to the structures. The

✉ R. Shashidhar  
shashidhar.r@jssstuniv.in

D. Manjunath  
manjunath3divakar@gmail.com

S. M. Shanmukha  
shanmukhasm03@gmail.com

<sup>1</sup> Department of Electronics and Communication Engineering,  
JSS Science and Technology University, Mysore 570006,  
India

experimental findings show that our detection approach produces good performance with a precision of above 98.00% after we train and analyze the data on our Dataset. This method is highly effective when analyzed with other existing methods and approaches.

The motivation is that, currently, there are many uses for image processing. Being involved in this field now is fascinating and thrilling. Visual information that is delivered in the form of a digital image is more important in today's society. In the future, image processing may be widely used to identify and make people aware of cancerous tumors, which aids in disease prevention by raising awareness. Digital image processing is a branch of signals and systems, where the images are manipulated and it concentrates mostly on pictures and visuals. The input for the system is an image, which is processed by effective algorithms and methods to generate the required output.

We can see that there are numerous instances of buildings collapsing owing to deterioration and fractures in the walls, and numerous accidents on the highways take place for the same reason. This can occasionally result in numerous casualties and losses. In the circumstances mentioned above, early damage or crack identification enables us to take preventative action to lessen or avert damage and potential failure. This technique may be enhanced and made accessible to everyone.

The goal of the research is to locate fractures in concrete surfaces. Builders may quickly determine the strength of any concrete building using this computed information and take any required adjustments right away. Concrete has a low tensile strength and quasi-brittleness. Tensile stress can arise in concrete as a result of applied loads, harmful chemical reactions, and environmental influences. The concrete will break if these tensile stresses are greater than its tensile strength. The size and number of fractures affect how well bridges and structures function. Although this cracking may be minimized by carefully choosing the elements that makeup concrete, some cracking is unavoidable.

The primary goal is to construct and create a software tool for crack detection with Python and algorithms, as well as ideas from machine learning and deep neural networks.

The contribution and objectives of the proposed work are.

- To implement localization methods of signal processing in the detection of cracks.
- To use the fundamentals of image processing for crack detection.
- To collect datasets.
- To develop an algorithm for analyzing the dataset.
- To apply the mathematics to implement a software-based process Framework to the sampled data.

“[Different methods used in crack detection](#)” deals with studies and research made by some of the certified researchers throughout the world on crack or damage detection and related works. “[Methodology](#)” gives the required procedure to detect cracks using image processing and also deals with the explanation of the block diagram which is used for implementation with the necessary evaluation models.

“[Result and discussion](#)” deals with the final step of our project, in this section we discuss the accuracy and losses of the training and testing model that we have implemented with the validation loss over time. We will also have a look at the model classification reports.

In “[Conclusion and future scope](#)”, we conclude our work by explaining the challenges that are present in surface crack detection and future scope of the proposed work.

## Existing work

Many efforts are made to examine the structural properties by use of an image processing approach with different cameras for highly efficient examination of the structures. Many methods have been developed over time for the detection of structural damages or cracks using various methods.

Cracks or damages on concrete structures are one of the early signs, and concrete quality is a key indicator in assessing the quality of construction projects (Arun & Poobal, 2018). It's important to find cracks and insect holes in structures since they frequently influence the quality of concrete surfaces (Hassene et al. 2017; Sun et al., 2021). One of the biggest challenges in the industry today is to maintain the quality of the surface of the structure which directly indicates the durability and maintenance of the buildings (Yao et al., 2019). Nevertheless, because of uneven light, deformation, possible shadows, and other factors, it is challenging to precisely identify pavement fractures in complex formations (Qu et al., 2020). Due to the fast increase in traffic, many older bridges' load designs are no longer able to accommodate the necessary loads, significantly jeopardizing the safety of the structure. Thus, it is essential to regularly check infrastructure like tunnels, bridges, etc., and identify any possible structural damage to guarantee operational safety (Zinno et al., 2022; Dong et al., 2019).

## Using image processing and convolution neural network

Many image processing methods have been put into place to find civil infrastructures and partially replace on-site inspections that are handled by humans. Yet, very

different real-world circumstances (such as lightning and shadow) might provide difficulties (Dong et al., 2019). Cracks that are not identified at the earliest or structures that are less maintained can lead to major damages. While identifying the cracks using image processing, one of the major problems is noise-captured images. So, based on a discretized variation minimization model with constraints, Amir et al. provided gradient-based solutions for picture denoising and deblurring issues (Beck & Teboulle 2009; Cha et al., 2017). A deep architecture using a convolutional neural network and infrared thermal image processing-based fracture detection assist overcome these difficulties (Mao et al. 2020). Vision-based algorithms are also often utilized in crack detections, as the human inspection takes more time and includes many other difficulties (Lionnie et al. 2022; Yang et al., 2019; Yeum & Dyke, 2015).

Shaoqing et al. suggested using RPN (region proposal network) that uses complete image-based convolutional network features including the network of crack detection, so, making an almost cost-free region proposals network (Ren et al., 2017). Liyan Zhang et al. suggested a convolutional neural network-based method combined with the Internet of Things with an accuracy of more than 90% (Zhang et al., 2018). Kaiming He et al. proposed spatial pyramid pooling with CNN based technique with accuracy much better than the R-CNN model (He et al., 2015). Jun Yang et al. proposed infrared thermal image crack detection with an accuracy of 95.52% (Yang et al., 2019). Chen et al. put forward a multi-task enhanced faster RCNN approach and the results with the K-MABtrA method were useful when dealing with multiple or small objects and were able to get 80.02 average precision (mAP) (Zou et al., 2012).

### Different methods used in crack detection

As cracks are of high non-uniformity and topological complexity using a multiscale feature attention network or multiscale dilated convolution model yields better efficiency in crack detection (Song et al., 2019). Pavement crack detection can also be done using a Gabor filter which is very much a potential and possible technique for crack detection in various directions (Salman et al., 2013). Lee et al. proposed a bridge inspection system using a robot, as conventional bridge inspection has a lot of challenges (Oh et al., 2009). Adaptive thresholding and deep convolutional neural network model can be implemented for crack detection to yield better accuracy (Fan et al., 2019). Hyunwoo Cho et al. put forward a structural damage detection approach that depends on edge finding with many intermediate steps. It is based on CWT (crack width transformation) algorithm for damage detection (Cho et al., 2018). Dhanajitha et al. proposed crack detection in buildings using drones and which were able to inspect the

cracks in high-rise buildings from various angles and were able to get 90.67% accuracy (Danajitha et al., 2022).

Kaveh worked on damage detection with different technologies and he introduce different optimization technologies his primary goal is to determine the position and severity of multiple damages in buildings or structures (Kaveh & Maniat, 2015). To achieve this, natural frequencies and mode shapes are utilized in constructing the necessary objective function (Kaveh, 2017). The authors presented an alternative technique for detecting structural damage in beams and frames by leveraging natural frequencies (Kaveh & Dadras, 2018; Kaveh & Zolghadr 2012).

Kaveh and Maniat researched identifying structural damage in skeletal structures when only incomplete data is available (Kaveh & Maniat, 2014, Kaveh & Zolghadr 2017a) and the authors proposed the tug-of-war algorithm for the detection of structural damage (Kaveh & Zolghadr, 2017b).

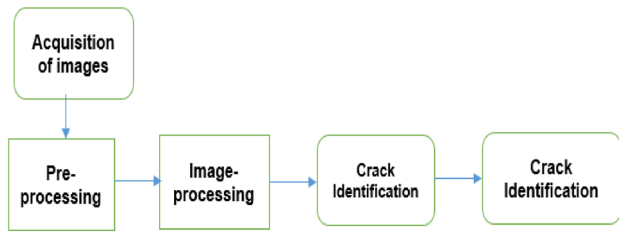
Here, we analyzed numerous approaches and fracture detection systems used to concrete civil constructions. The research we have done gives a thorough analysis of various technologies with different approaches used to identify cracks in concrete buildings. This investigation also sheds insight into the difficulties associated with fracture identification on concrete buildings and the potential avenues for future research. In conclusion, the analysis of the crack-detecting method demonstrates significant advancement in several areas. Although coping with varying camera resolutions has not been a major barrier for these studies, a bargain in the middle between the accuracy of the system and the complexity of an algorithm still exists which has to be addressed.

### Methodology

This section gives the required procedure to detect cracks using image processing and also deals with the explanation of the block diagram which is used for implementation with the necessary evaluation models.

The main benefit of employing image processing instead of traditional manual methods for crack identification is that the results are more accurate with image-based analysis. One of the major challenges in damage detection is the size of the image. The image resolution of modern digital cameras exceeds 10 megapixels.

The ability to capture detailed photographs of concrete surfaces is made possible by the improvement in resolution. The outspread view of the structure surface is captured on a single image utilizing modern commercial cameras. A broad



**Fig. 1** Block diagram for crack identification utilizing image processing

range of pictures is employed for realistic fracture detection in low-cost applications.

The overall framework of the cracks identification model by utilizing image processing is shown above. The block diagram (Fig. 1) includes the following:

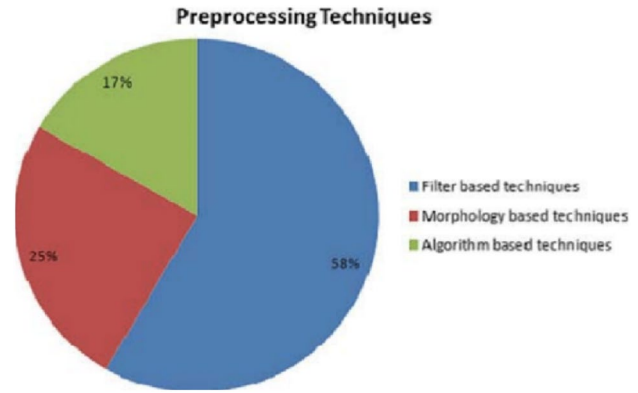
- Using a camera or other source, first, get a picture of the required structure which will be used in the fracture identification technique.
- Succeeding the image acquisition, the gathered pictures are pre-processed to perform techniques like segmentation and make the image processing process more effective.
- To process the subtracted picture sample, certain image processing methods are implemented.
- Using the output from image processing, and fracture detection on the structure can be observed here.

### Image acquisition

Illuminating the surface with the requisite light is required to obtain a picture of the fracture surface that is of high quality. For the broken surface to receive stable light, the light source is also crucial. Some of the light sources used are tungsten (wide spectrum), light emitting diodes, Xeon and sunshine, etc. They are utilized to shine a light on the surfaces of cracks in steel and concrete civil constructions. Digital cameras are typically utilized in concrete construction. Digital camera resolution power and lighting conditions determine the accuracy and precision of the images.

### Pre-processing and image processing

The majority of pre-processing approaches are based on filtering techniques and these are used in differentiating the crack feature from the surroundings. Multi sequential image filter is implemented to filter out background noises and spot fuzzy fractures. By removing of input picture from smoothed picture, the smoothing filter gets rid of the uneven lighting conditions and shading effects (Cao et al., 2020). Line emphasis filters are used to eliminate the noise from



**Fig. 2** Types of pre-processing techniques

the input image. Morphology- and algorithm-based pre-processing methods are additional crucial methods. Figure 2 displays a pie chart with the various pre-processing procedure types displayed on it.

The pixel value of the support region is operated linearly by a filter known as a linear filter (or weighted summation). So, the “filter matrix” designates the support zone, which is represented as  $H(i, j)$ . The filter region’s size is referred to as  $H$ , and the coordinate system for the filter matrix is unique with  $I$  denoting the index in column and  $j$  the index in row. Its center serves as the origin point and is called a hot spot. Noise is removed with a smoothing filter (a linear filter), which produces a blurred image structure, line, and edge. This problem was solved using non-linear filters, which operate nonlinearly.

Following steps are used to apply the filter to the image.

- Adjust the filter matrix so that  $I$  and  $H(0, 0)$  match the image’s current location  $(u, v)$ .
- Multiply each of the filter’s coefficients  $H(i, j)$  by the associated picture component  $I(u + i, v + j)$ .
- Calculate the result for the present location  $I(u, v)$  by averaging all the results from the previous step.

The equation below can be used to describe each stage.

$$I'(u, v) \leftarrow \sum_{i=-1}^1 \sum_{j=-1}^1 I(u + i, v + j) \cdot H(i, j) \quad (1)$$

One of the operations associated with linear filters is linear convolution. The convolution operation is described as the equation for the two-dimensional function  $I$  and  $H$ .

$$I'(u, v) \leftarrow \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} I(u - i, v - j) \cdot H(i, j) \quad (2)$$

$$I' = I * H \quad (3)$$

when the convolution technique is used. If you look at the equation, you'll notice that this procedure yields results that are comparable to those of linear filters that have filter functions that take into account Vertical axes and also in horizontal axes. The kernel is convolution matrix  $H$ .

One type of non-linear filter is the minimum and maximum filter which generate the least and maximum value respectively in moving region  $R$  of the original image. Defining these filters,

$$I'(u, v) \leftarrow \min \{I(u + i, v + j) | (i, j) \in R\} \quad (4)$$

$$I'(u, v) \leftarrow \max \{I(u + i, v + j) | (i, j) \in R\} \quad (5)$$

Another type of Non-linear filter, the median filter's output is the median of each value in  $R$  (moving area). Moreover, these filters are frequently implemented to eliminate pepper and salt noises in pictures. This filter's definition is

$$I'(u, v) \leftarrow \text{median}\{I(u + i, v + j) | (i, j) \in R\} \quad (6)$$

Filters can be implemented where the quality of the image has to be improved (e.g., noise removal). It may be used to sharpen the image and find edges.

## Crack detection

There are various crack identification methods constructed on image processing. The first sign of a structure deteriorating is when cracks start to show on its concrete surface. Regular fracture revelation will result in the structure being rigorously destroyed. To solve these issues, it is crucial to spot the fissures as soon as possible. Nevertheless, voids and delamination, as well as other elements, make fracture identification on a concrete picture challenging. This section discusses numerous techniques for fracture detection on the surface of concrete. Several image processing techniques may be used to find fractures in concrete constructions (Gui & Li, 2020). Based on the sort of technology used to find cracks in concrete buildings, crack identification methods are divided into 3 groups.

- Model-dependent approaches.
- Thresholding-based strategies.
- Pattern-based approaches.

Image processing methods implemented in crack identification and analyzed datasets are specifically

designed for certain pictures. For fresh photos and datasets gathered under diverse lighting circumstances and in the presence of shadows, the approaches might not produce correct findings. In some cases, crack detection methods are also classified into three groups as Filter dependent approaches, Machine learning-dependent approaches, and the last one which is machine learning and filtering-dependent methods (Sizyakin et al., 2020).

## Procedure

**Annotations and Labelling:** The images were taken from a Kaggle dataset having  $227 \times 227$  dimensions. The final dataset had a total of 6000 images labelled as cracks. Images with cracks were labelled as 'Cracked' and images without cracks were labelled as 'Uncracked' and based on that datasets were divided into file paths and labels that contain the file path of the images with labeling from the starting index.

Here, we first build a model using Keras layers which are already defined in the network, and compile it. Then training of the model is carried out.

Although the model's structure is similar to that of VGG-16, it has fewer layers and a considerably more compact input picture size. Three convolutional blocks make up the model, after which fully connected layers along with an output layer. Spatial dimensions of activation maps after each convolutional block were also shown in Fig. 3. This illustration helps us understand how the Keras layer works.

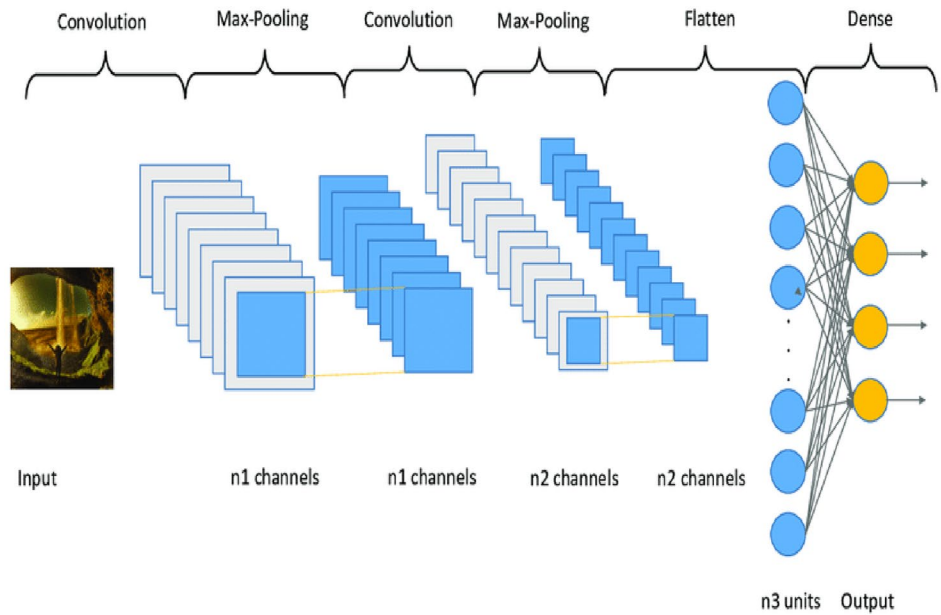
As we have only considered binary, with only two classes, binary classification methods employ Sigmoid, whereas multiclass issues require SoftMax. So, here we have used sigmoid activation after the dense layer.

The default behavior has no padding therefore the convolutional layer's output will have a spatial dimension that is somewhat less than its input if we don't specifically specify this padding option. Except for the output layer, we apply a ReLU activation function across the board throughout the network. In addition to the consideration of Convolutional neural networks, the convolution of the kernel is a crucial part of many other Computer Vision approaches. In the procedure, the kernel (a small number matrix) or filter is utilized to change the image depending on filter values. The formula, which is used to produce subsequent feature map values, represents the input picture as the letter  $f$  and the kernel as the letter  $h$ . The result matrix's indices for the rows and the columns are denoted by the symbols  $m$  and  $n$ .

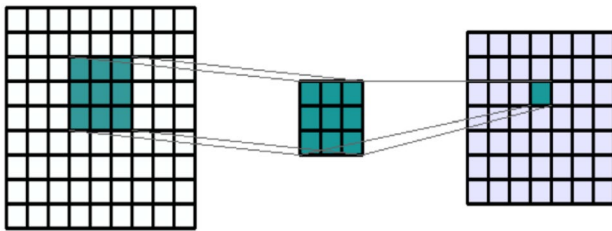
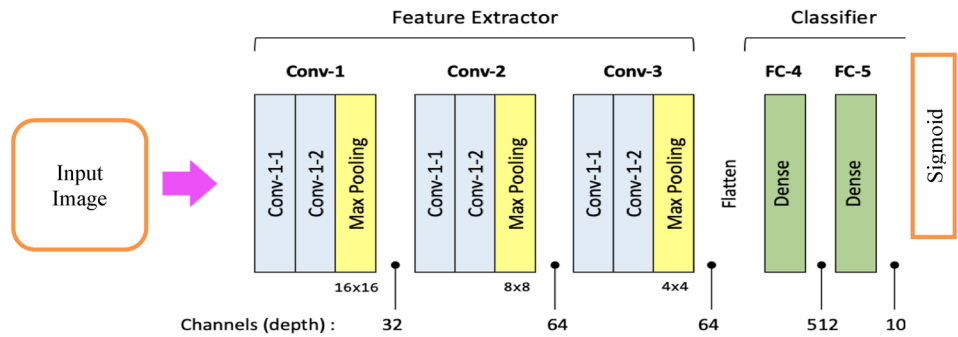
$$G[m, n] = (f * h)[m, n] = \sum_j \sum_k h[j, k] f[m - j, n - k] \quad (7)$$



**Fig. 3** Convolutional neural network



**Fig. 4** Architecture of the model convolutional network



**Fig. 5** Representation of convolutional product

Figure 4 shows the architecture of the convolutional neural network. Representation of the convolutional network as shown in Fig. 5.

Filters are applied to a particular pixel, following every kernel value in pairs is multiplied with matching values in the input picture. In the end, everything is compiled and the outcome is placed in the appropriate location on the output's feature maps.

The output of the kernel by use of a convolution product is also referred to as a filtered image. This can be represented as below:

$$G(x, y) = w \times F(x, y) = \sum_{\delta x=-ki}^{ki} \cdot \sum_{\delta y=-kj}^{kj} \cdot w(\delta x, \delta y) \cdot F(x + \delta x, y + \delta y) \tag{8}$$

Here,  $k$  represents a kernel and

$$-kj \geq \delta y \geq ki, -kj \geq \delta x \geq -ki \tag{9}$$

For the Conv2D layer, we have considered only three arguments, these are filters, activation, and kernel size. The output of the convolution layer can be represented as

$$\text{output} = \frac{\text{input} - \text{kernelsize} + 2 \times \text{padding}}{\text{stride}} + 1 \tag{10}$$

After the above, MaxPooling2D with argument Pool\_size and Global average pooling 2D is used.

### Activation functions

These are used for converting the neuron linear output into non-linear output which helps a neural network to learn the non-linear conducts.

To implement this, Rectified Linear Unit (ReLU) is used, which gives  $x$  for all the positive values of  $x$  and zero and gives 0 for the negative values of  $x$  which is the same as  $\max(x, 0)$ .

The ReLU equation is:

$$f(x) = \max(0, x) \tag{11}$$

Function ReLU (Fig. 6) and its derivative are monotonic and the output range of the function varies from 0 to infinity. It's a default activation function and is widely implemented in neural networks, particularly CNNs.

The sigmoid activation function is differentiable and also bounded. It is also a real function that has a single point of inflection and the derivative at every point is non-negative. It is interpreted for every real value.

It is also defined as a sigmoid curve and is described as a function having a recognizable S-shaped curve. The logistic function is denoted by the following formula:

$$S(x) = 1 - S(-x) = \frac{1}{1 + e^{-x}} = \frac{e^x}{e^x + 1} \tag{12}$$

The sigmoid function (Fig. 7) is also monotonic. One pair of horizontal asymptotes act as a constraint on a sigmoid function as  $X \rightarrow \pm\infty$

In many of the examples, that point is 0, and for values below that point, sigmoid is convex and concave for values above it.

### Compiling and training the model

After the above steps, we compile the model by specifying the optimizer type as Adam. We have specified loss as

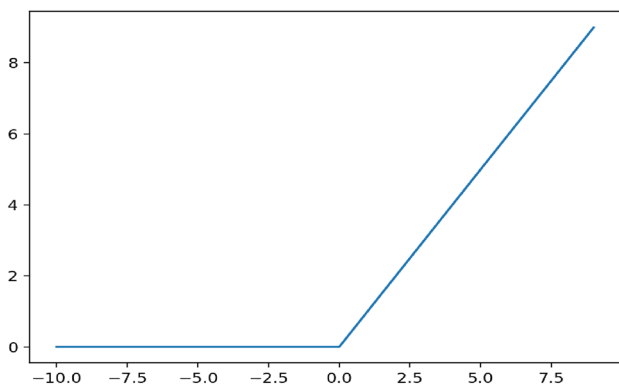


Fig. 6 ReLU activation function

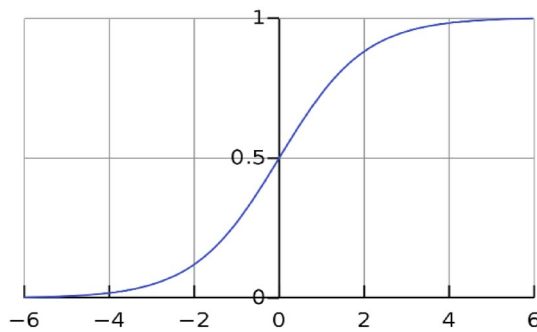


Fig. 7 Sigmoid activation function

binary\_crossentropy. Lastly, we provide accuracy as a further parameter to track throughout training. Although the loss function's value is always recorded by default, you must specify it if you want accuracy. After this, we train and acquire the results required.

### Evaluation metrics

It is important to keep false negatives to a minimum at the expense of raising false positives since the purpose of assessment is to find as many examples from a community as feasible for a screening approach. As a result, false positive rate and true positive rate with accuracy should be considered. The first parameter is referred to as sensitivity (SEN) in medical terminology and is represented by the symbol equation:

$$TPR = SEN = TP/P \tag{13}$$

Here,  $P$  stands for the occurrences of positive events and  $TP$  represents the number of genuine positives. False positive rate estimation, stated as the equation:

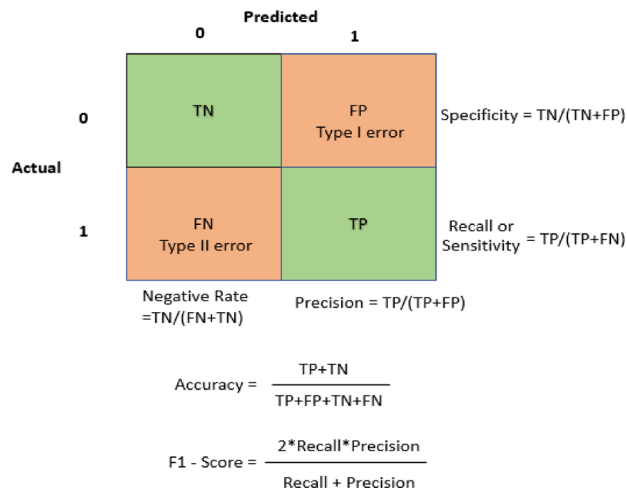


Fig. 8 Evaluation matrix

$$\text{FPR} = \text{FP}/N \quad (14)$$

The fraction of false positives is FP, the Number of total genuine negative samples is  $N$  and cumulative negative events in the population is  $N$ . Evaluation matrix as shown in Fig. 8. On the other side, this statistic is best understood in terms of true negatives and true negatives or specificity (SPEC), which is provided as equation:

$$\text{TNR} = \text{SPEC} = \text{TN}/N = 1 - \text{FPR} \quad (15)$$

Lastly, accuracy establishes the ratio of genuine positives to true negatives. When the number of good and negative events is not equal, this statistic might be very helpful. This can be written as equation:

$$\text{ACC} = (\text{TP} + \text{TN})/(\text{P} + \text{N}) \quad (16)$$

## Result and discussion

As we are in the final step of our proposed work, in this section we discuss the accuracy and losses of the training and testing model that we have implemented with the validations loss over time. We will also have a look at the model classification reports.

We have taken the number of epochs to be 100 to get the maximum accuracy in our training model with each epoch going over 105 cycles. The batch size of the model is 32 which is the standard batch size for any model.

After training the model, we obtained a training accuracy of 98.23%, which is excellent accuracy for any model. The validation of our trained model is 98.21%. The training and validation loss is reduced to a minimum to improve the system's accuracy.

From the plot shown in Fig. 9, we can observe that as the number of epochs increased our validation and test loss over time almost becomes equal (tends to zero) The testing datasets that we used to test the model yields an accuracy of 97.11%, which is excellent for any model. Figure 10 shows the number of epochs used for training and validation.

Test loss	0.10665
Test accuracy	97.11%

The shown graph in Fig. 11 represents the accuracy of the trained datasets and validation datasets. It represents an increase in accuracy as the number of cycles increases. The confusion matrix given below shows the implemented model success rate by giving true positive and true negative numbers in the testing of our model, which indicates the error rate and accuracy of the system model.

```

Epoch 1/100
105/105 [=====] - 35s 327ms/step - loss: 0.6914 - accuracy: 0.5245 - val_loss: 0.6664 - val_accuracy: 0.5214
Epoch 2/100
105/105 [=====] - 20s 191ms/step - loss: 0.6378 - accuracy: 0.6856 - val_loss: 0.5622 - val_accuracy: 0.8107
Epoch 3/100
105/105 [=====] - 20s 192ms/step - loss: 0.5564 - accuracy: 0.7723 - val_loss: 0.5016 - val_accuracy: 0.8179
Epoch 4/100
105/105 [=====] - 20s 187ms/step - loss: 0.4494 - accuracy: 0.8729 - val_loss: 0.3754 - val_accuracy: 0.9048
Epoch 5/100
105/105 [=====] - 20s 194ms/step - loss: 0.3523 - accuracy: 0.9173 - val_loss: 0.3059 - val_accuracy: 0.9060
Epoch 6/100
105/105 [=====] - 20s 190ms/step - loss: 0.2791 - accuracy: 0.9237 - val_loss: 0.3008 - val_accuracy: 0.9452
Epoch 7/100
105/105 [=====] - 20s 186ms/step - loss: 0.2337 - accuracy: 0.9394 - val_loss: 0.2126 - val_accuracy: 0.9583
Epoch 8/100
105/105 [=====] - 20s 189ms/step - loss: 0.1982 - accuracy: 0.9461 - val_loss: 0.1830 - val_accuracy: 0.9595
Epoch 9/100
105/105 [=====] - 20s 188ms/step - loss: 0.1619 - accuracy: 0.9541 - val_loss: 0.1888 - val_accuracy: 0.9464
Epoch 10/100
105/105 [=====] - 20s 186ms/step - loss: 0.1430 - accuracy: 0.9602 - val_loss: 0.1814 - val_accuracy: 0.9655
Epoch 11/100
105/105 [=====] - 20s 188ms/step - loss: 0.1521 - accuracy: 0.9541 - val_loss: 0.1440 - val_accuracy: 0.9702
Epoch 12/100
105/105 [=====] - 20s 186ms/step - loss: 0.1147 - accuracy: 0.9687 - val_loss: 0.1720 - val_accuracy: 0.9702
Epoch 13/100
105/105 [=====] - 20s 190ms/step - loss: 0.0978 - accuracy: 0.9763 - val_loss: 0.1356 - val_accuracy: 0.9690
Epoch 14/100
105/105 [=====] - 20s 190ms/step - loss: 0.1087 - accuracy: 0.9679 - val_loss: 0.1280 - val_accuracy: 0.9762
Epoch 15/100
105/105 [=====] - 20s 186ms/step - loss: 0.0827 - accuracy: 0.9763 - val_loss: 0.1238 - val_accuracy: 0.9810
Epoch 16/100
105/105 [=====] - 20s 189ms/step - loss: 0.0779 - accuracy: 0.9772 - val_loss: 0.1097 - val_accuracy: 0.9798
Epoch 20/100
105/105 [=====] - 20s 187ms/step - loss: 0.0688 - accuracy: 0.9755 - val_loss: 0.1124 - val_accuracy: 0.9833
Epoch 21/100
105/105 [=====] - 20s 189ms/step - loss: 0.0813 - accuracy: 0.9812 - val_loss: 0.1124 - val_accuracy: 0.9821
Epoch 22/100
105/105 [=====] - 20s 189ms/step - loss: 0.0597 - accuracy: 0.9823 - val_loss: 0.1219 - val_accuracy: 0.9821

```

Fig. 9 Training and validation loss



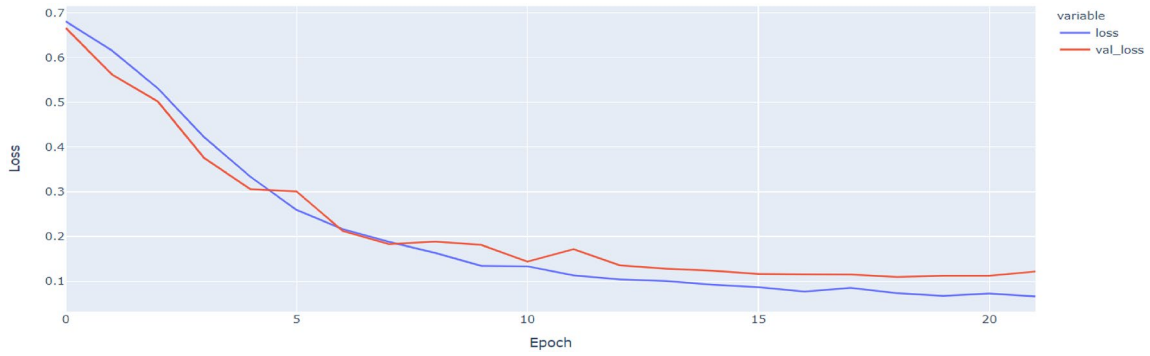


Fig. 10 Epochs processing in the system model

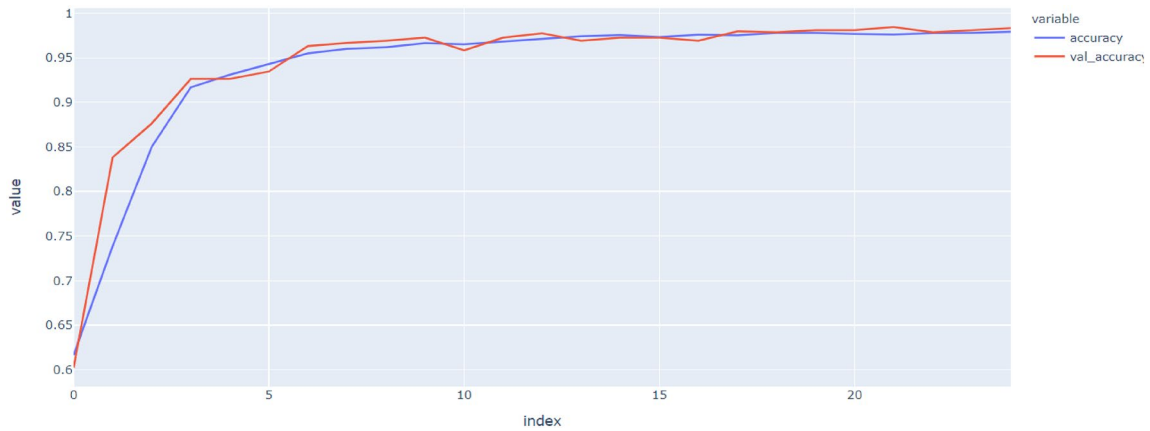


Fig. 11 Accuracy curve

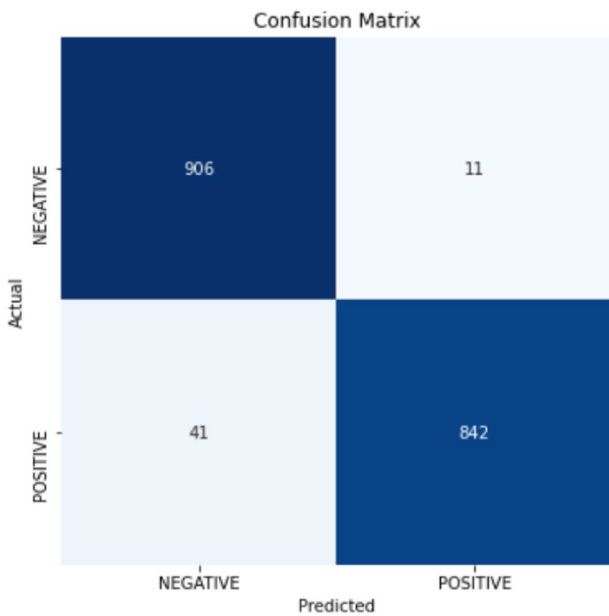


Fig. 12 Designed model confusion matrix

Table 1 Classification report for the designed model

	Precision	Recall	F1-score	Support
Uncracked	0.96	0.99	0.97	917
Cracked	0.99	0.95	0.97	883
Accuracy			0.97	1800
Macro avg	0.97	0.97	0.97	1800
Weighted avg	0.97	0.97	0.97	1800

From the confusion matrix in Fig. 12, analyzing system model performance using classification reports can be done. Table 1 shows the classification report. From the classification report, we can observe that the precision of predicting positive images is correct being almost 99% of the time and the negative images (with no cracks) being 96%. The accuracy of the mode is also well above 97%. Table 1 shows the classification report for the proposed model. Table 2 shows the comparison result of the proposed model with existing model.

**Table 2** Comparison of the proposed work with different methodologies

Method/algorithm	Datasets	Accuracy
GoogleNet CNN, FPN [01]	64,000 uncracked pictures and 64,000 cracked pictures	Recall = 86.09% and The precision of 80.13%
Adaboost, random forest, SVM, STRUM [20]	100 pictures with 1920 × 1280 pixels	Accuracy = 95%
Canny algorithm, decision tree heuristic [21]	320 × 320 pixels 400 pictures	Accuracy = 88% (crack detection)
Deep convolutional encoder-decoder network [22]	527 pictures	Precision = 77.68% and Recall = 71.98%
Morphological operations, NN, SVM [23]	5184 × 3456 pixels and 1910 uncracked and 3961 cracked images	Accuracy of NN = 79.5% and Accuracy of SVM = 78.3%
Recursive tree edge pruning [24]	800 × 600 pixels with 206 pictures	Precision = 79% and Recall = 92%
Proposed method	3000 crack and 3000 noncrack images	Accuracy = 97.11%

## Conclusion and future scope

Here, our work is concluded by explaining the challenges that are present in surface crack detection and how we have tried to overcome them with the future scope of this proposed method.

This work developed a model for image processing to find structural damages and flaws in the structures such as buildings. As digital pictures used for crack investigation provide a variety of challenges for image analysis, some of which are inconsistent lighting, less contrast, and noise. Also the distance of the concrete crack from the camera. The resolution of the camera is also important which might result in blurry photos, poor contrast images, and thin objects that are difficult to identify are the key factors affecting how well the different crack detection systems in concrete operate. The majority of crack detection methods currently in use rely on photographic images, which are ineffective at identifying internal flaws like voids, environmental factors like weather, the color of surfaces, the fog presence along with the appearances of structures, optimizing the minimum value for termination procedures, photos with poor contrast and blurs, all of this lower required accuracy for the cracks identification process. Problems with convolutional neural network approaches include the need for several or large amounts of sample dataset points for fine-tuning a network, also data overfitting, tweaking parameters to boost accuracy, and the necessity for fast processors like graphics processing units are included. To manage the enormous quantity of data collected by such autonomous systems and to find, quantify, and categorize different types of cracks, a system must be developed. It is necessary to undertake a quick examination of fracture detection utilizing reliable decision-making techniques. Another crucial research question is the impact of lighting conditions on the efficacy

and effectiveness of crack-detecting systems. In this paper, we've spoken about utilizing neural networks to solve such issues. We have developed a model that effectively and accurately detects fractures in surface pictures. To prevent data build-up and to improve the system, we employed Keras pre-processing.

The future scope of the proposed work is:

- *Real-world application of the model:* In this study, we developed a model that virtually utilizes the provided datasets. The technology will next be put into practice in areas with real-world applications.
- *Building a more effective model:* The model may be made to accumulate data more quickly, prevent overfitting issues, and process data more quickly for implementations in practical applications.
- *Automated crack inspection:* The process of finding a crack in a structure using any method is called crack detection. The proposed approach makes use of radiometric, geometric, and contextual data that were retrieved from the photos in turn.
- *Railway track damage detection:* A neural network-based method for measuring and detecting railway deterioration. Train accidents are frequently the result of damage to the railroad. This neural network-based measuring system provides a high degree of accuracy and is appropriate for applications requiring online railway damage identification and monitoring.

**Author contributions** All the three authors significantly contributed to all stages of the manuscript.

**Funding** Open access funding provided by HEAL-Link Greece. This research received no external funding.

**Data availability** The data used to support the findings of this study are included within the article.

## Declarations

**Conflict of interest** The authors declare that there is no conflict of interest regarding the publication of this paper.

## References

- Arun, M., & Poobal, S. (2018). Crack detection using image processing: A critical review and analysis. *Alexandria Engineering Journal*, 57, 787–798. <https://doi.org/10.1016/j.aej.2017.01.020>
- Beck, A., & Teboulle, M. (2009). ‘Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems.’ *IEEE Transactions on Image Processing*, 18(11), 2419–2434. <https://doi.org/10.1109/TIP.2009.2028250>
- Cao, W., Liu, Q., & He, Z. (2020). Review of pavement defect detection methods. *IEEE Access*, 8, 14531–14544. <https://doi.org/10.1109/ACCESS.2020.2966881>
- Cha, Y. J., Choi, W., & Büyüköztürk, O. (2017). ‘Deep learning-based crack damage detection using convolutional neural networks.’ *Computer-Aided Civil and Infrastructure Engineering*, 32(5), 361–378. <https://doi.org/10.1111/mice.12263>
- Cho, H., Yoon, H.-J., & Jung, J.-Y. (2018). Image-based crack detection using crack width transform (CWT) algorithm. *IEEE Access*, 6, 60100–60114. <https://doi.org/10.1109/ACCESS.2018.2875889>
- Danajitha, K.K., Sheeba, A., Sophiya, P., & Maha Vishnu, V.C. (2022). ‘Detection of Cracks in High Rise Buildings using Drones,’ International Conference on Electronics and Sustainable Communication Systems (ICESC), pp. 318–324. <https://doi.org/10.1109/ICESC54411.2022.9885251>
- Dong, Y., Wag, J., Wang, Z., Zhang, X., Gao, Y., Sui, Q., & Jiang, P. (2019). A deep-learning-based multiple defect detection method for tunnel lining damages. *IEEE Access*, 7, 182643–182657. <https://doi.org/10.1109/ACCESS.2019.2931074>
- Fan, R., Bocus, M. J., Zhu, Y., Jiao, J., Wang, L., Ma, F., Cheng, S., & Liu, M. (2019). Road crack detection using deep convolutional neural network and adaptive thresholding. *IEEE Intelligent Vehicles Symposium*. <https://doi.org/10.48550/arXiv.1904.08582>
- Gui, Z., & Li, H. (2020). Automated defect detection and visualization for the robotic airport runway inspection. *IEEE Access*, 8, 76100–76107. <https://doi.org/10.1109/ACCESS.2020.2986483>
- Han, D. J., Zhao, Qi., Wang, L., & Yin, K. (2018). Comparison of random forest, artificial neural networks, and support vector machine for intelligent diagnosis of rotating machinery. *Transactions of the Institute of Measurement and Control*, 40, 2681–2693. <https://doi.org/10.1177/0142331217708242>
- Hassene, H., Alavi, A. H., Pengcheng, J., & Nizar, L. (2017). Detection of fatigue cracking in steel bridge girders: A support vector machine approach. *Archives of Civil and Mechanical Engineering*, 17, 609–622. <https://doi.org/10.1016/j.acme.2016.11.005>
- He, K., Zhang, X., Ren, S., & Sun, J. (2015). ‘Spatial pyramid pooling in deep convolutional networks for visual recognition.’ *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(9), 1904–1916. [https://doi.org/10.1007/978-3-319-10578-9\\_23](https://doi.org/10.1007/978-3-319-10578-9_23)
- Kaveh, A. (2017). *Applications of metaheuristic optimization algorithms in civil engineering*. Springer. ISBN: 978-3-319-48012-1.
- Kaveh, A., & Dadras, A. (2018). Structural damage identification using enhanced thermal exchange optimization algorithm. *Engineering Optimization*, 50(3), 430–451. <https://doi.org/10.1080/0305215X.2017.1318872>
- Kaveh, A., & Maniat, M. (2014). Damage detection in skeletal structures based on charged system search optimization using incomplete modal data. *International Journal of Civil Engineering IUST*, 12(2), 291–298.
- Kaveh, A., & Maniat, M. (2015). Damage detection based on MCSS and PSO using modal data. *Smart Structures and Systems*, 15(5), 1253–1270. <https://doi.org/10.12989/SSS.2015.15.5.1253>
- Kaveh, A., & Zolghadr, A. (2012). An improved charged system search for structural damage identification in beams and frames using changes in natural frequencies. *International Journal of Optimization in Civil Engineering*, 2(3), 321–340.
- Kaveh, A., & Zolghadr, A. (2017a). Cyclical parthenogenesis algorithm for guided modal strain energy-based structural damage detection. *Applied Soft Computing*, 57, 250–264. <https://doi.org/10.1016/j.asoc.2017.04.010>
- Kaveh, A., & Zolghadr, A. (2017). Guided modal strain energy-based approach for structural damage identification using tug-of-war optimization algorithm. *Journal of Computing in Civil Engineering*. <https://doi.org/10.1061/%28ASCE%29CP.1943-5487.0000665>
- Lionnie, R., Ramadhan, R. C., Rosyadi, A. S., Jusoh, M., & Alaydrus, M. (2022). Performance analysis of various types of surface crack detection based on image processing. *SINERGI*. <https://doi.org/10.22441/sinergi.2022.1.001>
- Mao, Y., Chen, J., Ping, P., & Chen, H. (2020). Crack detection with multi-task enhanced faster R-CNN model. *IEEE Sixth International Conference on Big Data Computing Service and Applications*. <https://doi.org/10.1109/BigDataService49289.2020.00038>
- Oh, J. K., Jang, G., Oh, S., Lee, J. H., Yi, B. J., Moon, Y. S., Lee, J. S., & Choi, Y. (2009). ‘Bridge inspection robot system with machine vision.’ *Automation in Construction*, 18(7), 929–941. <https://doi.org/10.1016/j.autcon.2009.04.003>
- Qu, Z., Mei, J., Liu, L., & Zhou, D. Y. (2020). ‘Crack detection of concrete pavement with cross-entropy loss function and improved VGG16 network model.’ *IEEE Access*, 8, 54564–54573. <https://doi.org/10.1109/ACCESS.2020.2981561>
- Ren, S., He, K., Girshick, R., & Sun, J. (2017). ‘Faster R-CNN: Towards real-time object detection with region proposal networks.’ *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6), 1137–1149. <https://doi.org/10.1109/TPAMI.2016.2577031>
- Salman, M., Mathavan, S., Kamal, K., & Rahman, M. (2013). ‘Pavement crack detection using the Gabor filter.’ in Proceeding 16th International IEEE Conference on Intelligent Transportation Systems (ITSC), The Hague, The Netherlands, pp. 2039–2044. <https://doi.org/10.1109/ITSC.2013.6728529>
- Shan, Q., & Dewhurst, R. J. (1998). Surface—breaking fatigue crack detection using laser ultrasound. *Applied Physics Letters*, 62, 2649–2651. <https://doi.org/10.1063/1.109274>
- Sizyakin, R., Cornelis, B., Meeus, L., Dubois, H., Martens, M., Voronin, V., & Pižurica, A. (2020). Crack detection in paintings using convolutional neural networks. *IEEE Access*, 8, 74535–74552. <https://doi.org/10.1109/ACCESS.2020.2988856>
- Song, W., Jia, G., Jia, D., & Zhu, H. (2019). ‘Automatic pavement crack detection and classification using multiscale feature attention network.’ *IEEE Access*, 7, 171001–171012. <https://doi.org/10.1109/ACCESS.2019.2956191>
- Sun, Y., Yang, Y., Yao, G., Wei, F., & Wong, M. (2021). Autonomous crack and bughole detection for concrete surface image based on deep learning. *IEEE Access*, 9, 85709–85720.
- Yang, J., Wang, W., Lin, G., Li, Q., Sun, Y., & Sun, Y. (2019). Infrared thermal imaging-based crack detection using deep learning. *IEEE Access*, 7, 182060–182077. <https://doi.org/10.1109/ACCESS.2019.2958264>

- Yao, G., Wei, F., Yang, Y., & Sun, Y. (2019). 'Deep-learning-based bug hole detection for concrete surface image.' *Advances in Civil Engineering*, 2019, 1–12. <https://doi.org/10.1155/2019/8582963>
- Yeum, C. M., & Dyke, S. J. (2015). 'Vision-based automated crack detection for bridge inspection.' *Computer-Aided Civil and Infrastructure Engineering*, 30(10), 759–770. <https://doi.org/10.1111/mice.12141>
- Zhang, L., Zhou, G., Han, Y., Lin, H., & Wu, Y. (2018). 'Application of internet of things technology and convolutional neural network model in bridge crack detection.' *IEEE Access*, 6, 39442–39451. <https://doi.org/10.1109/ACCESS.2018.2855144>
- Zhou, Q., Zhong, Qu., & Cao, C. (2021). Mixed pooling and richer attention feature fusion for crack detection. *Pattern Recognition Letters*, 145, 96–102. <https://doi.org/10.1016/j.patrec.2021.02.005>
- Zinno, R., Haghshenas, S. S., Guido, G., & Vitale, A. (2022). **A27** "artificial intelligence and structural health monitoring of bridges: A review of the state-of-the-art,." *IEEE Access*, 10, 88058–88078. <https://doi.org/10.1109/ACCESS.2022.3199443>
- Zou, Q., Cao, Y., Li, Q., Mao, Q., & Wang, S. (2012). CrackTree: Automatic crack detection from pavement images. *Pattern Recognition Letter*, 33, 227–238. <https://doi.org/10.1016/j.patrec.2011.11.004>

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.