Predicting characteristics of cracks in concrete structure using convolutional neural network and image processing

Waqas Qayyum[1], Rana Ehtisham[1], Alireza Bahrami[2]*, Junaid Mir[3], Qaiser Uz Zaman Khan[1], Afaq Ahmad[1] and Yasin Onuralp Özkılıç[4]

[1]Civil Engineering Department, University of Engineering and Technology, Taxila, Pakistan, [2]Department of Building Engineering, Energy Systems and Sustainability Science, Faculty of Engineering and Sustainable Development, University of Gävle, Gävle, Sweden, [3]Electrical Engineering Department, University of Engineering and Technology, Taxila, Pakistan, [4]Department of Civil Engineering, Faculty of Engineering, Necmettin Erbakan University, Konya, Turkey

The degradation of infrastructures such as bridges, highways, buildings, and dams has been accelerated due to environmental and loading consequences. The most popular method for inspecting existing concrete structures has been visual inspection. Inspectors assess defects visually based on their engineering expertise, competence, and experience. This method, however, is subjective, tiresome, inefficient, and constrained by the requirement for access to multiple components of complex structures. The angle, width, and length of the crack allow us to figure out the cause of the propagation and extent of the damage, and rehabilitation can be suggested based on them. This research proposes an algorithm based on a pre-trained convolutional neural network (CNN) and image processing (IP) to obtain the crack angle, width, endpoint length, and actual path length in a concrete structure. The results show low relative errors of 2.19%, 14.88%, and 1.11%, respectively for the crack angle, width, and endpoint length from the CNN and IP methods developed in this research. The actual path length is found to be 14.69% greater than the crack endpoint length. When calculating the crack length, it is crucial to consider its irregular shape and the likelihood that its actual path length will be greater than the direct distance between the endpoints. This study suggests measurement methods that precisely consider the crack shape to estimate its actual path length.

KEYWORDS

concrete, convolutional neural network, image processing, crack angle, crack width, crack length

1 Introduction

Infrastructures such as bridges, highways, buildings, and dams have been degraded increasingly owing to environmental and loading consequences. Many bridges in Japan and the United States have been used for over 50 years (Dung and Anh, 2019). The level of urbanization in a nation is influenced by the growth of civil structures. As the world's population raises, so does the application for higher infrastructures. Nevertheless, the failure of even a single bridge might have a devastating effect on the entire transportation network and cause various fatalities. Moreover, the influence of natural catastrophes like earthquakes,

hurricanes, and floods on infrastructures leads to shoddy building designs that pose a serious threat to civilization. Skyscrapers are also vulnerable to structural damage brought on by wind, which over time could jeopardize the integrity of the structures. Infrastructures must be subjected to thorough and frequent inspections in order to periodically assess their conditions. It is customary to have a group of experts physically check structures before compiling a report on their findings. However, owing to the intricate nature of structural design and heights, it becomes unsafe for human workers/inspectors to carry out structural health inspections when sophisticated structures are developed. Because of this, an autonomous solution may be essential in fixing this issue by performing structural health monitoring (SHM) more precisely and consistently while avoiding endangering human examiners.

SHM is the practice of continually monitoring the state of a structure, such as a building or bridge, with sensors and other measurement equipment to identify and diagnose any damage or degradation that may occur over time. The purpose of SHM is to detect and diagnose issues before they become serious, preventing failure and ensuring the structure's and its users' safety. SHM employs various methods, including vibration analysis, strain and deformation measurement, and non-destructive testing. With the help of implanted piezoceramic transducers, Song et al. (2007) detected deterioration to a 6.1 m long reinforced concrete bridge bent cap. Their research confirmed that cracks might be detected and assessed using a damage index derived from wavelet packet investigation in conjunction with piezoceramic transducers. Antunes et al. (2012) showed that optical technology, optical fiber Bragg grating accelerometers, may be utilized for dynamic SHM of tall slender buildings. The findings of active monitoring of an elevated water reservoir were used to determine its longitudinal and transverse dynamic parameters. While taking optical sensor readings of the reservoir, they also employed a seismograph to determine the reservoir's natural frequencies with an error of not more than 0.06%.

Yu et al. (2007) developed a system to objectively quantify and inspect cracks in concrete structures for safety evaluations. The setup comprised of a mobile robot and a crack recognition system. The mobile robot had been engineered to keep the same distance from the walls, while a CCD camera captured images. Through image processing (IP) techniques, the crack detection system extracted information related to cracks that were specific to each image. Akbar et al. (2019) explored an autonomous SHM system that used unmanned aerial vehicles (UAVs) to capture images of the structure. UAV captured images of the entire structure, which were then merged to create a comprehensive view. The process of stitching the images together utilized a widely used technique called speeded-up robust features (SURF), which identifies key features in the images.

Digital technology has proven to be helpful in facilitating effective organization of monitoring and inspection activities for owners. Convolutional neural networks (CNNs) are a form of deep learning (DL) neural networks that have gained widespread adoption for their effectiveness in recognizing images and videos. Thanks to their capabilities, CNNs have found extensive usage in detecting and classifying cracks. One major advantage of CNNs is their ability to automatically learn spatial hierarchies of features

from input images in an adaptive manner. CNNs are commonly utilized in various applications, including but not limited to image sorting, semantic segmentation, object recognition, image generation and restoration, and face recognition. Researchers have been inspired by these benefits and have employed CNNs to detect and classify cracks in various studies.

Abdel-Qader et al. (2003) used four methods, including Canny, Fast Haar Transform (FHT), Sobel, and Fast Fourier Transform (FFT), to identify cracks and found that FFT performed the best compared with the other methods. Prasanna et al. (2012) employed the Support Vector Machine technique with a linear kernel function to identify cracks, achieving a 76% accuracy in classifying 118 images as either cracked or uncracked. Munawar et al. (2021) presented an innovative approach named CycleGAN, which employed a cycle generative adversarial network along with sixteen convolutional layers to identify cracks in buildings in Sydney, Australia. The method used UAVs and publicly available images from the year 2000. The suggested CNN architecture is heavily reliant on the integration of guided image filtering and discriminative random fields to enhance its accuracy. A real-world damage dataset from buildings in Sydney was utilized to evaluate the proposed framework, and the outcomes demonstrated that the deep hierarchical CNN architecture achieved the highest global accuracy (99.9%) in comparison with the other models, including Guided Filtering (GF), Baseline (BN), Deep-Crack GF, and SegNet. Furthermore, the proposed architecture yielded an average accuracy of 93.9% for each class, with mean intersection of all union classes (IoU) at 88.1%, precision at 89.5%, recall at 85.7%, and F1-score at 85.7%.

For the crack detection, Liu et al. (2019) conducted a comparative study between U-Net and Deep CNN (DCNN). Their findings revealed that U-Net was a more advanced and effective approach than DCNN, resulting in superior accuracy and performance. Ali et al. (2022) employed the CNN architecture to find and categorize building cracks. Superior fine crack segmentation may be achieved using encoder and decoder designs like fully convolutional network, SegNet, and U-Net. U-Net and DeepLabV3+ enabled by the vision transformer (ViT) were used by Shamsabadi et al. (2022) to create a method for identifying fractures in concrete and asphalt surfaces. The identification of concrete cracks is only one use of DL. Yin et al. (2020) utilized a CNN-based object identification method called YOLO-V3 to identify drainage system faults such as holes, cracks, roots, and deposits. For this model's training, a total of 4,056 images were used. The overall accuracy of the framework was 85.37%. The investigation resulted in closed-circuit television (CCTV) footage with fault types and descriptions listed in each frame. There was a proposal for a CNN-based fault classification system by Hassan et al. (2019). The main goal was to include the CCTV footage in the categorization process. Out of the CCTV footage, they could compile a dataset of 4,702 images representing six types of damages: debris silty, longitudinal, joint defective, joint open, lateral, and damaged surface. A maximum of 96.33% accuracy was measured. Gehri et al. (2022) proposed certain modifications for an automatic crack identification and measurement method called digital image correlation (DIC)

that enables an evaluation of crack behavior in extensive studies, including intricate patterns. The improvements made include a canny-based edge-crack indicator and advancements in crack kinematic measurement. The improved technique was validated and proved to be a valuable tool in providing an understanding of the mechanical behavior of structural concrete. Thériault et al. (2022) introduced an affordable approach for conducting periodic visual inspections with quantifiable data using the DIC method. This method enables monitoring of the fracture development and deformations over time. The study validated the effectiveness of the approach by examining the shear fracture kinematics of seven specimens tested in a laboratory setting. Despite minor imperfections caused by the camera movement, 98% of the 462 measurements taken during testing were within the tolerance limit of 0.1 mm compared with the measurements taken using a stationary camera with a long focal length lens as a reference. These findings indicated the potential of the DIC method for improving the infrastructures management through structural inspections. Ji et al. (2020) developed vision-based techniques for monitoring deformations in reinforced concrete structures and defining cracks. They measured the coordinates of objects of interest using a target tracking method and estimated deformation components using geometry analysis. In the cyclic testing of reinforced concrete wall specimens, the methodologies were verified and provided more efficiency and meaningful information than the conventional measuring approaches.

The concept of transfer learning as a strategy for moving previously acquired expertise from one domain to another has recently been developed in computer science. The domains of the training and test data may differ, which is not considered by typical machine learning or semi-supervised techniques (Lu et al., 2015). Several pre-trained models are available that are effectively utilized in transfer learning. Examples of pre-trained neural networks are Inception-V3 (Szegedy et al., 2016), VGG-16,19 (Russakovsky et al., 2015; Simonyan and Zisserman, 1409), Dense-Net (Huang et al., 2017), ResNet (Zoph et al., 2018), Inception-ResNet (Szegedy et al., 2017), Darknet (Redmon), Xception (Chollet, 2017), EfficientNet (Tan and Le, 2019), ShuffleNet (Zhang et al., 2018), and SqueezeNet (Iandola et al., 2017). Based on a comparison of four pre-trained models for the crack recognition and direction across four classes, Ehtisham et al. (2022) discovered that the ResNet50 model achieved an accuracy of 86.22%. In a study conducted by Ahmed et al. (2022), the ResNet50 CNN model was employed to detect pavement cracks with a remarkable accuracy. The proposed method obtained an accuracy of 99.8% and a precision rate of 100%. Qayyum et al. (2022) utilized three distinct CNN models including GoogleNet, MobileNet-V2, and Inception-V3 for identifying cracks in concrete images captured in various orientations of diagonal, horizontal, and vertical directions. The findings illustrated that the Inception-V3 model outperformed the other two models, achieving an accuracy of 97.2% for detecting cracked and uncracked concrete images, and 92%, 95%, and 96% for detecting diagonal, horizontal, and vertical cracks images, respectively. These results highlighted the potential of deep learning-based techniques for the precise and efficient identification of cracks in concrete structures. Özgenel and Sorguç (2018) conducted an

evaluation of the performance of pre-trained neural networks consisting of VGG19, AlexNet, ResNet101, VGG16, GoogleNet, ResNet50, and ResNet152. They analyzed the depth of networks, the size of training dataset, the number of training epochs, and their ability to be applied to different types of building materials.

Only a few researchers have focused on identifying the characteristic of cracks, including the crack angle, width, and length. The combination of these three factors can give a good indication of the severity of the crack and the potential for failure. Engineers use this information to determine the appropriate action, such as repairing or replacing the damaged structure. Flah et al. (2020) created algorithms that combined the Keras classifier with IP to assess the crack length, width, and angle with quantification errors of 1.5%, 5%, and 2%, respectively. Qayyum et al. (2023) assessed seven pre-trained neural networks, including GoogLeNet, MobileNet-V2, Inception-V3, ResNet-18, ResNet-50, ResNet-101, and ShuffleNet for the crack detection and classification. In terms of accuracy, Inception-V3 outscored all the other models.

This research has used a pre-trained CNN model and IP to obtain the width, angle, endpoint length, and actual path length of cracks in a concrete structure. The CNN model was utilized to classify the cracks images based on their orientation. Inception-V3 was employed for classification purposes, and the dataset for training, validation, and testing remained the same, as suggested by Qayyum et al. (2023). After the classification step, image segmentation techniques were used to extract cracks from the images and achieve their characteristics.

The characteristics of cracks in concrete structures are critical for ensuring the safety, durability, performance, and cost-effectiveness of critical infrastructure assets, including buildings, dams, and bridges. Most of the studies have been based on the classification of cracks images. Flah et al. (2020) worked on obtaining cracks length, width, and angle. The actual path length of cracks was not measured. The current study has classified the concrete images into four categories, i.e., horizontal, vertical, and diagonal cracks (HC, VC, and DC, respectively), and un-cracked (UC) using CNN. Also, this research has proposed a new procedure based on CNN and IP to achieve the cracks angle, width, and endpoint length which is the displacement between the ending tips of the cracks. Moreover, this investigation has proposed a new method to determine the actual path length of the cracks.

2 Methodology

The methodology of this research is demonstrated in Figure 1, which displays the overall process. The first step was to gather a substantial dataset of representative images for the classes that needed classification. This dataset was considered in three sets: training, validation, and testing. Transfer learning, a well-known technique in deep learning, was utilized to start training a new model using a pre-trained model. In this case, the pre-trained Inception-V3 model was employed as the starting point to train a new model on the crack images dataset. The model's performance was assessed by evaluating it in the testing sets, utilizing accuracy, precision, recall, and F1-score as the evaluation metrics. The

subsequent stage involved implementing the IP techniques to identify cracks in the image. Otsu's method was employed, a prevalent IP method that uses thresholding to divide the image's foreground and background by detecting a threshold value. In the final step, various array operations were utilized on the binary image to determine the crack angle, width, endpoint length, and actual path length.

2.1 Image dataset

In total, 32,000 images have been taken from online sources, such as SDNET (Maguire et al., 2018) and collected from the site. These site images include images of concrete buildings at the University of Engineering and Technology (UET) and in the Taxila region. Images were put into one of four groups of DC, HC, VC, and UC based on what the category of cracks was. In Figure 2, each category is represented by a single image. The

training and validation of the models used 8,000 images for each class, with a 70:30 ratio for the training and validation, respectively. There were 100 images in each category in a separate image database utilized for testing the model, which was not part of the training or validation datasets.

2.2 Pre-trained CNN model

The pre-trained CNN model employed to classify the crack image was Inception-V3. It is a CNN architecture for image classification developed by Google. It was published in 2015 as part of the Inception series of models, including Inception-V1 and Inception-V2. The key feature of Inception-V3 is the use of "Inception modules", which are blocks of layers that perform multiple types of convolutions and pooling in parallel. This enables the network to learn local and global information from the input image, hence enhancing its accuracy. ImageNet
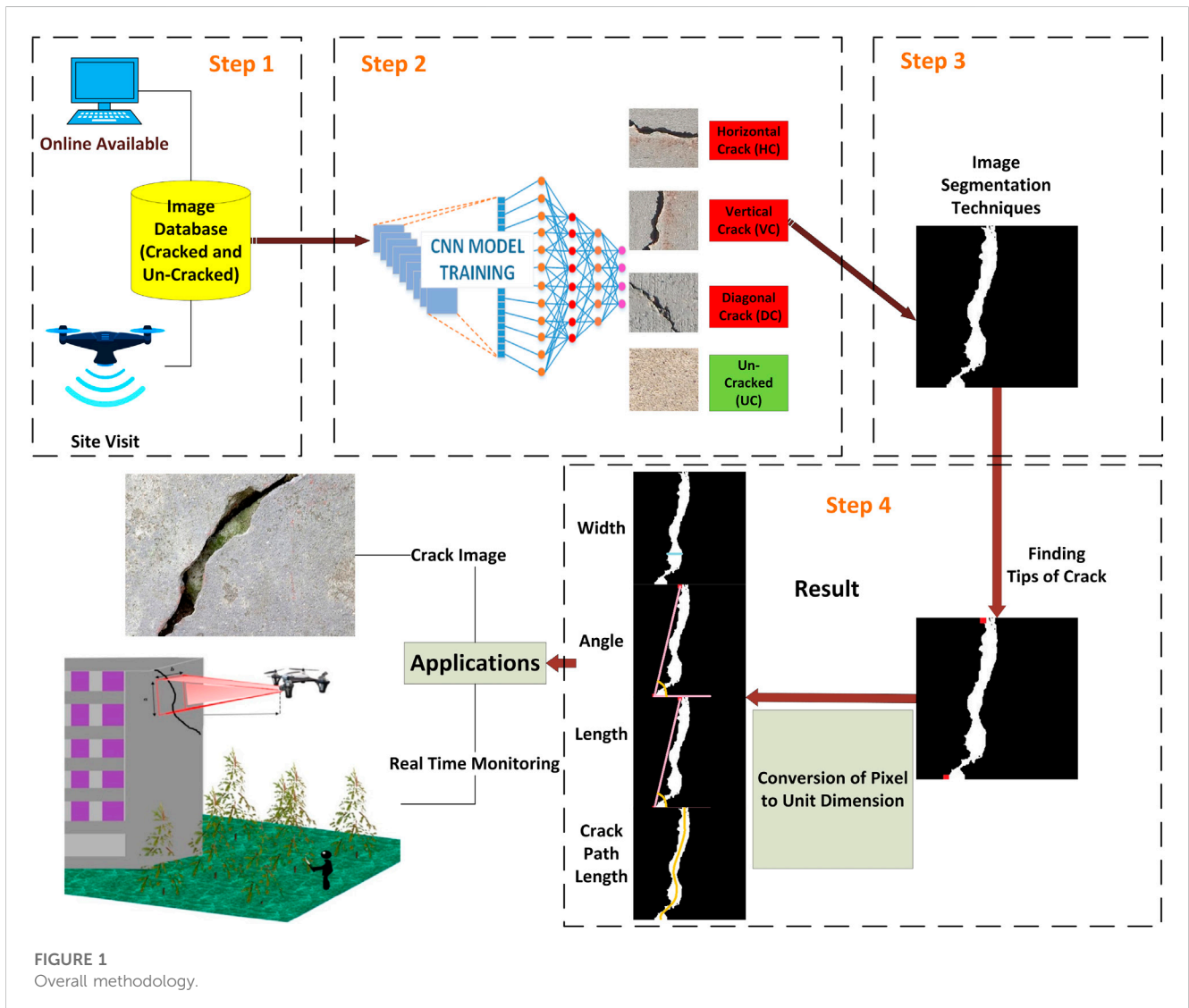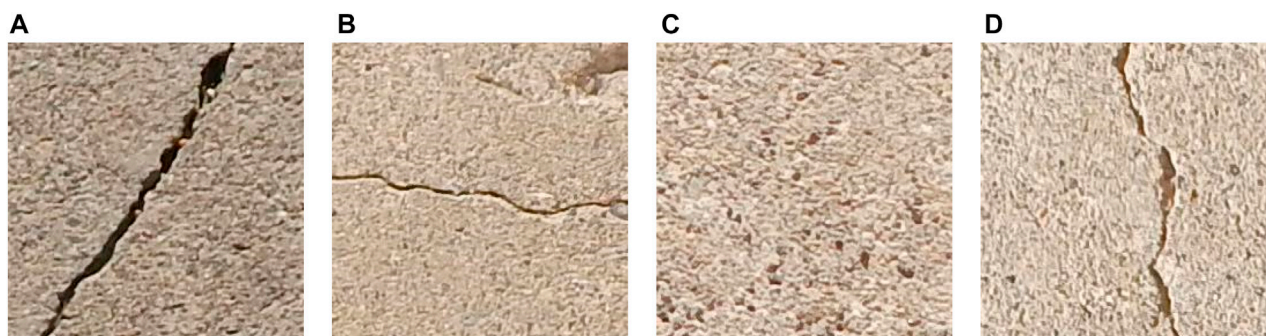

FIGURE 1
Overall methodology.

FIGURE 2
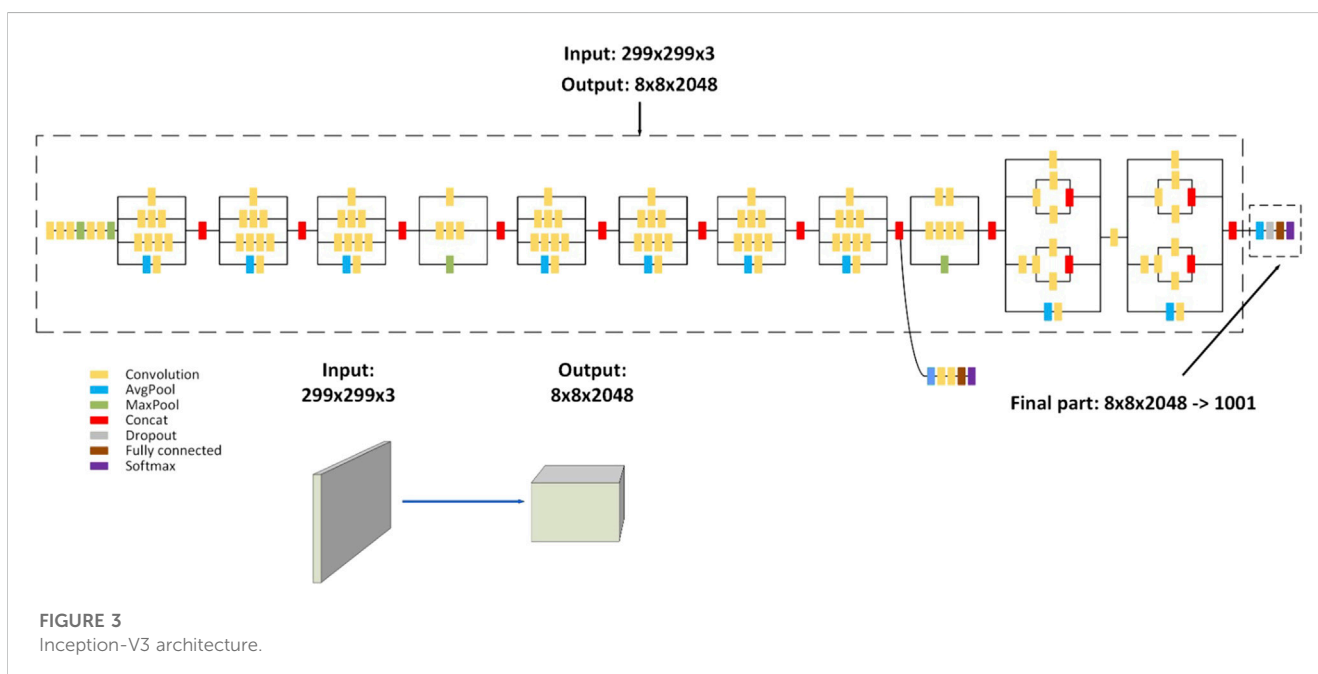Images (Qayyum et al., 2023): **(A)** DC, **(B)** HC, **(C)** UC, **(D)** VC.

FIGURE 3
Inception-V3 architecture.

was used to train Inception-V3, which at the time of its release reached state-of-the-art performance. Figure 3 depicts the Inception-V3 architecture.

2.3 Different techniques of IP

After the classification of the image, if the image were from class DC, VC, and HC, it would be forwarded for the implementation of the IP techniques. These functions used for IP are summarized in Table 1.

rgb2gray (RGB): With this function, RGB images were converted to grayscale. It removes information about shades and saturation while preserving brightness. Graythresh: Using the Otsu's method, graythresh was used to calculate a global threshold from a grayscale image. The Otsu's method thresholding technique reduces the intraclass alteration of black and white pixels. Imbinarize: This function was utilized to convert the grayscale image into the binary form. All the values above the threshold value received from the graythresh function are replaced by 0's, and the values below are replaced by 1's. Imfill: This function was employed with a parameter of "holes", which fills the holes in the binary image. Bwconncomp: This function was applied to count the number of connected components. The structure contains parameters including Connectivity, ImagSize, NumObjects, and PixelIdxList. Connectivity by default is equal to 8 for 2D image and 26 for 3D image, ImageSize receives the size of the image in pixel, NumObjects receives a total number of connected components, and PixelIdxList has the vector containing the linear indices of the object. NumObjects and PixelIdxList were used to obtain the size of

TABLE 1 Sequence of IP techniques used for extracting cracks.

No.	IP functions
1	rgb2gray
2	graythresh
3	imbinarize
4	imfill
5	bwconncomp
6	bwareaopen

the connected object in the binary image. Bwareaopen: This function was utilized to eliminate all the objects except one with the largest size. Figure 4 indicates the crack image before and after IP.

2.4 Array operation on binary image

Binary images are digital images containing only two values, typically black and white, representing a foreground object and the background. To perform operations on the binary image, the image is usually represented as a 2D array where each element corresponds to a pixel in the image, and its value represents the pixel's intensity.

3 Features of concrete cracks

3.1 Finding tips of cracks

The position of the cracks tips was located to obtain the endpoint length and angle. Two types of searches were used.

3.1.1 Horizontal search

Figure 5 shows how horizontal searching works. The search for the upper tip begins from the first row and ends at the last row, with the first white pixel being searched in each column within each row. However, for the lower tip, the search begins from the last row and ends at the first row, with the first white pixel being searched in each column within each row.

3.1.2 Vertical search

Figure 6 displays the process of vertical searching. To detect the left tip, the search begins from the first column and ends at the last column. Within each column, the algorithm looks for the first white pixel in every row. To identify the right tip, the search starts from the last column and ends at the first column, and for each column, the algorithm searches for the first white pixel in every row.

Horizontal searching was used for the VC images, and vertical searching was employed for the HC images. For the DC images, either of two searches can be utilized. For this research, horizontal searching was applied for the DC images. The positions of the first tip, i.e., i1 (row), j1 (column), and second tip, i.e., i2, j2, were used to obtain the angle and endpoint length.

3.2 Angle of cracks

Eqs. 1 and 2 were utilized to achieve the angle of the cracks.

$$Slope = \frac{j2 - j1}{i2 - i1} \tag{1}$$

$$Angle = tan^{-1}(Slope) \tag{2}$$

3.3 Endpoint length of cracks

Eq. 3 was employed to determine the length between the points.
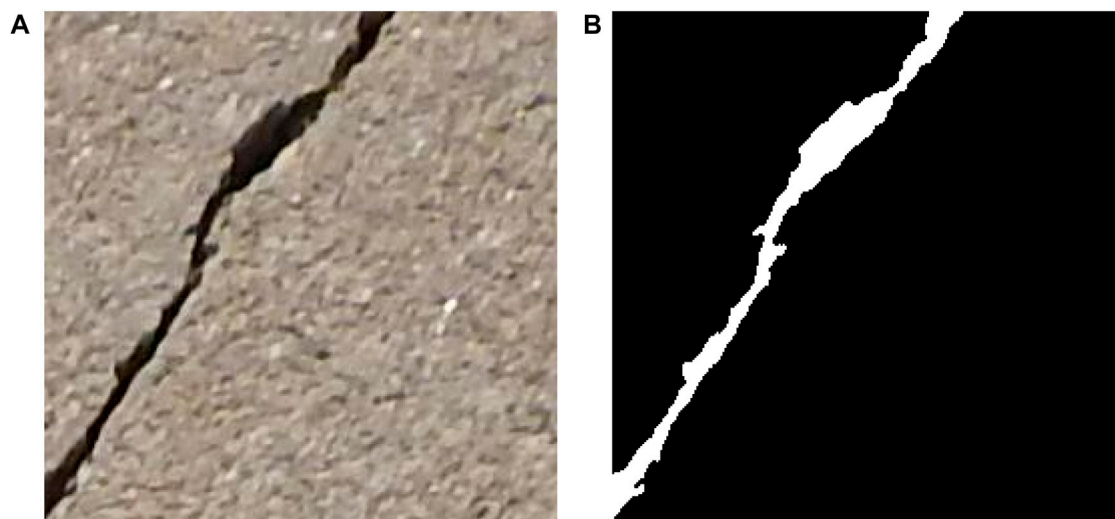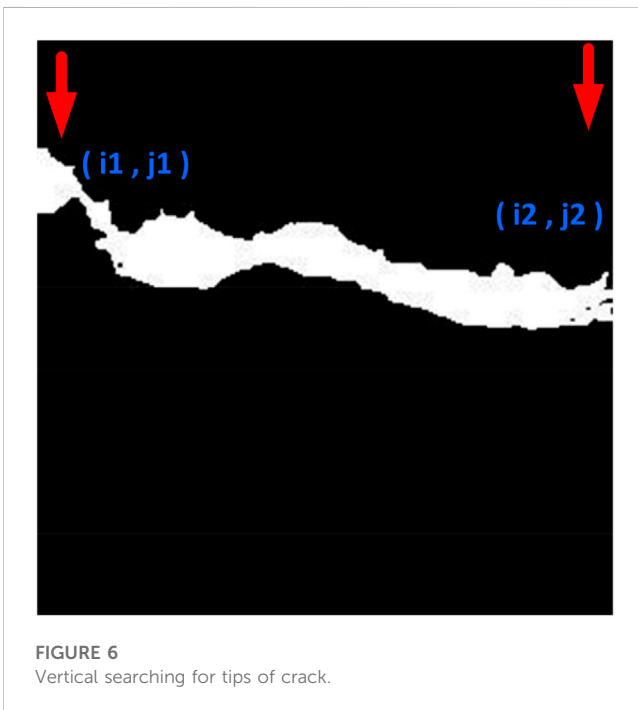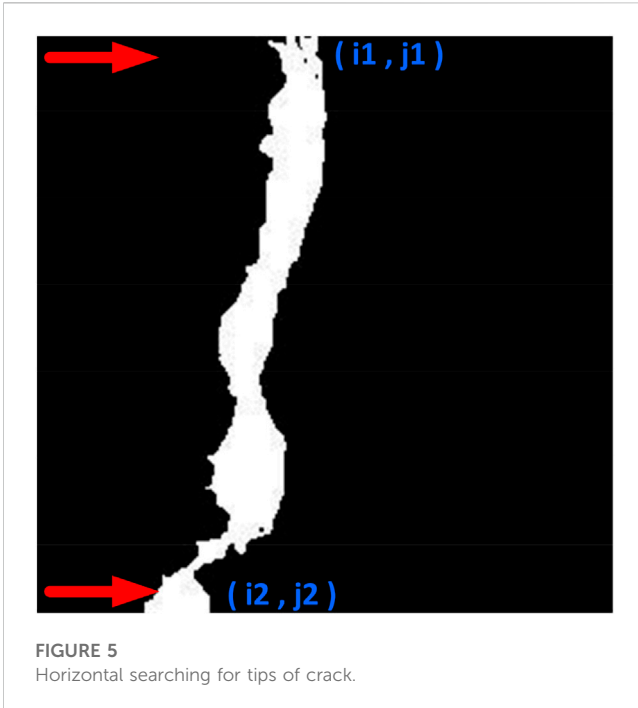

FIGURE 4
Crack image: **(A)** before IP, **(B)** after IP.

FIGURE 5
Horizontal searching for tips of crack.

FIGURE 7
Crack width.

FIGURE 6
Vertical searching for tips of crack.

$$Endpoint\ length = \sqrt{(j2 - j1)^2 + (i2 - i1)^2} \qquad (3)$$

3.4 Width of cracks

To measure the width of the cracks, the number of white pixels in each row of the VC and DC binary images was counted. Similarly, for HC, the width was measured by counting the number of white pixels in

each column. These values were then stored in an array, and the highest value in the array was considered as the width of the crack. Figure 7 depicts the width of the crack obtained using this method.

3.5 Actual path length of cracks

It is important to accurately determine the length of a concrete crack as it can impact the strength and stability of the structure. Engineers can determine the appropriate repair method based on the crack length, severity, and location. For VC and DC, the procedure of averaging the position of the white pixels in each row of a binary image matrix was adopted, while the HC procedure of averaging the position of the white pixels in each column of a binary image matrix was used. For VC and DC, these steps were done: 1) iteration through each row in the binary image matrix, 2) for each row, a list to store the column numbers of the white pixels was created, 3) once all rows were processed, the mean of the column numbers in the list was calculated and the value was rounded off, 4) row matrix with corresponding mean column number to represent central points in a 2D array was combined, 5) "circshift" function was employed to achieve the distance between those points, and added up to calculate the actual path length in pixels. Figure 8 demonstrates the procedure adopted to obtain the actual path length of VC.

A similar procedure was taken for HC, but in this case, for each column, a list was created to store the row numbers of the white pixels of the cracks.

3.6 Conversion of pixels to units

The values of the width, endpoint length, and actual path length were measured in pixels and converted to units using Eq. 4.
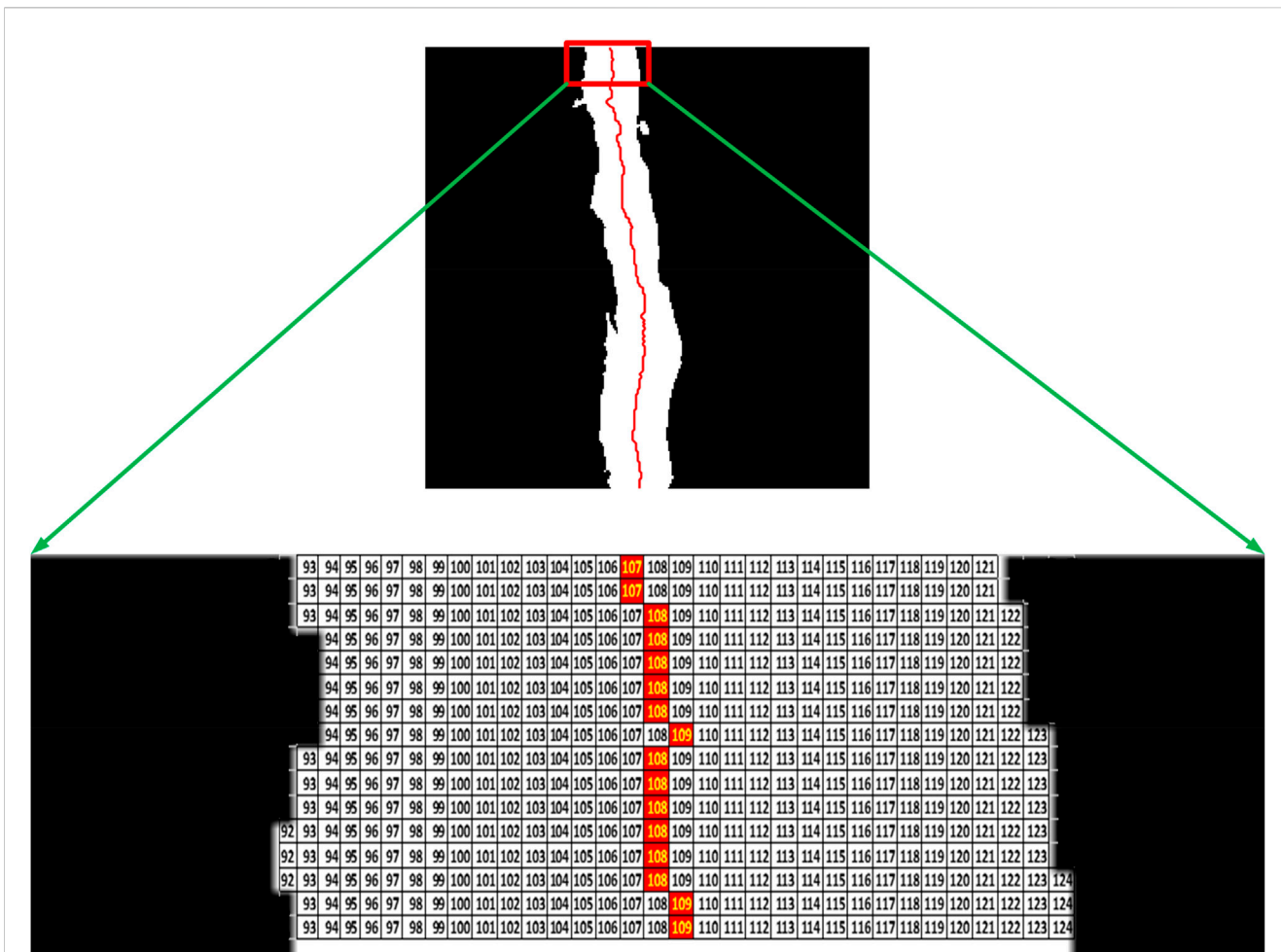
FIGURE 8
Array operation used to obtain actual path length of cracks.

$$D\left(units\right) = D\left(pixels\right) \times \sqrt{\frac{A}{n \times m}} \qquad (4)$$

A is the area of the original image resulted by the multiplication of length and width in the units. *n* is the length, and *m* is the width in the pixel form. *D* (pixels) is the distance in terms of pixels measured by the CNN and IP techniques for the width, endpoint length, and actual path length.

4 Results and discussion

The developed algorithms were tested on images. Cracks images were taken to measure the cracks endpoint length, width, and angle. Then, those measurements were compared with the results produced by this research's algorithms. The first step was to classify the image using a trained CNN model. Figure 9 indicates the confusion matrix plot for Inceprion-V3. The overall accuracy attained by the model was 88.5%.

Figure 10 shows some typical crack images taken from the site with their measurements. The proposed algorithm was
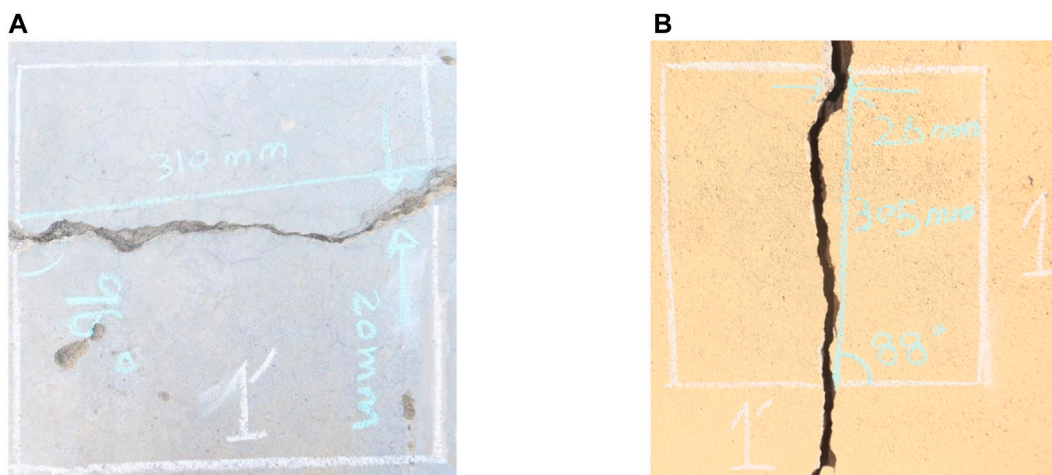

FIGURE 9
Confusion matrix plot for Inception-V3.

FIGURE 10
Crack images taken from site with measurements: **(A)** Image 1, **(B)** Image 2, **(C)** Image 3.

tested on these examples to ensure its efficiency. A box of 1 foot × 1 foot was drawn to denote the region to be captured and pre-processed before continuing with the algorithm. Image 1 was taken from a concrete pavement, while images 2 and 3 were taken from a wall.

Figures 11–13 illustrate the process in which the images were taken from the site, and then, using the CNN and IP methods, the cracks were extracted, and the angle, width, endpoint length, and actual path length of the cracks were measured.

The measurements taken from the site were compared with the values generated by the algorithms, as summarized in

Table 2. Obtaining the actual path length of the cracks was impossible, but it was achieved by algorithms (Table 2).

Small-scale field testing confirmed the accuracy of the crack characteristics algorithm, which gave relative errors of 2.19%, 14.88%, and 1.11% for the angle, width, and endpoint length of the cracks, respectively, as provided in Table 3.

The algorithm tracked the actual path of the cracks and found their actual path lengths, in which their mean value was 14.69% greater than the cracks endpoint length. Table 4 displays the comparison between the endpoint length and the actual path length of the cracks.
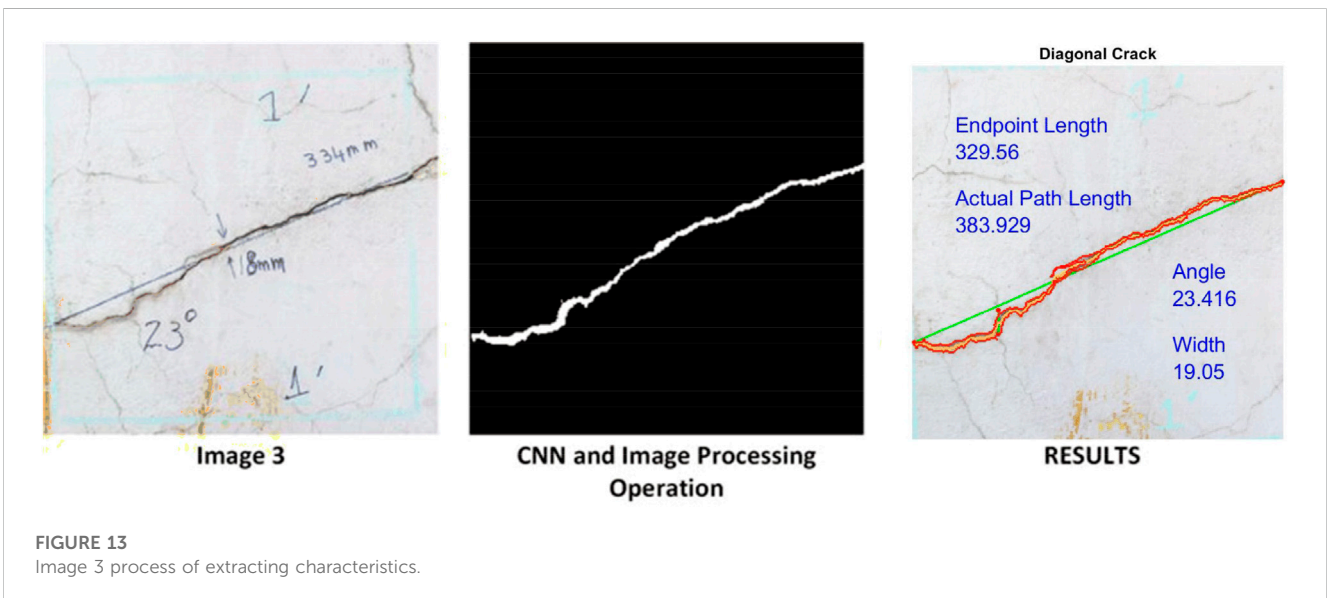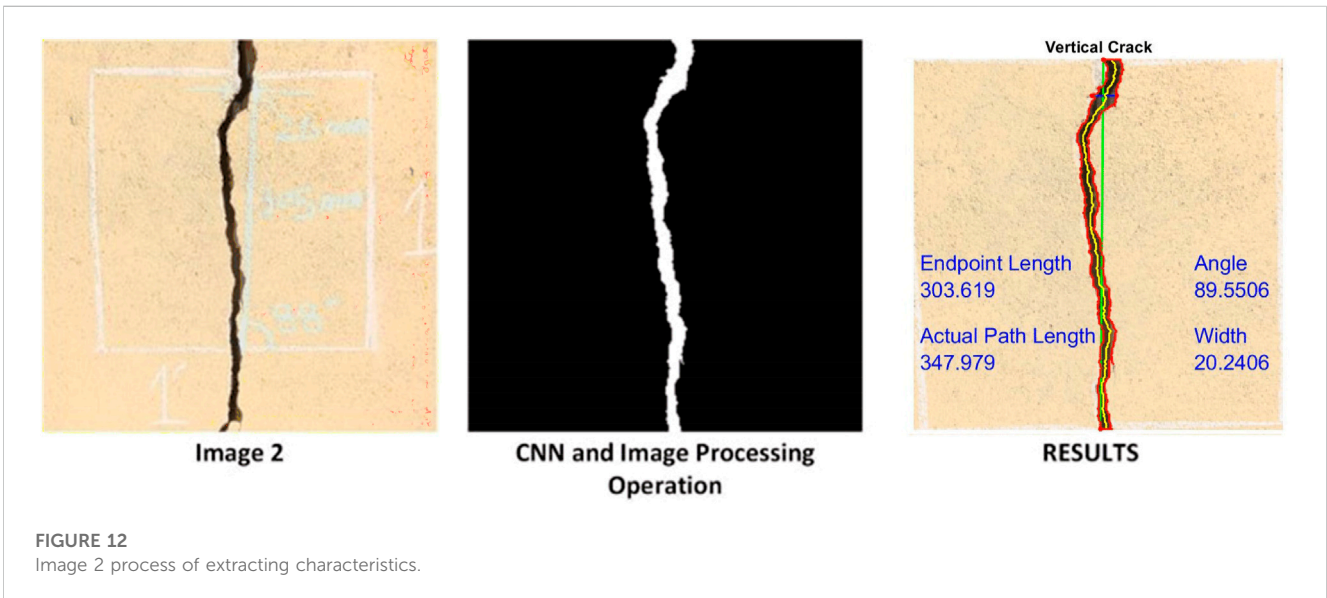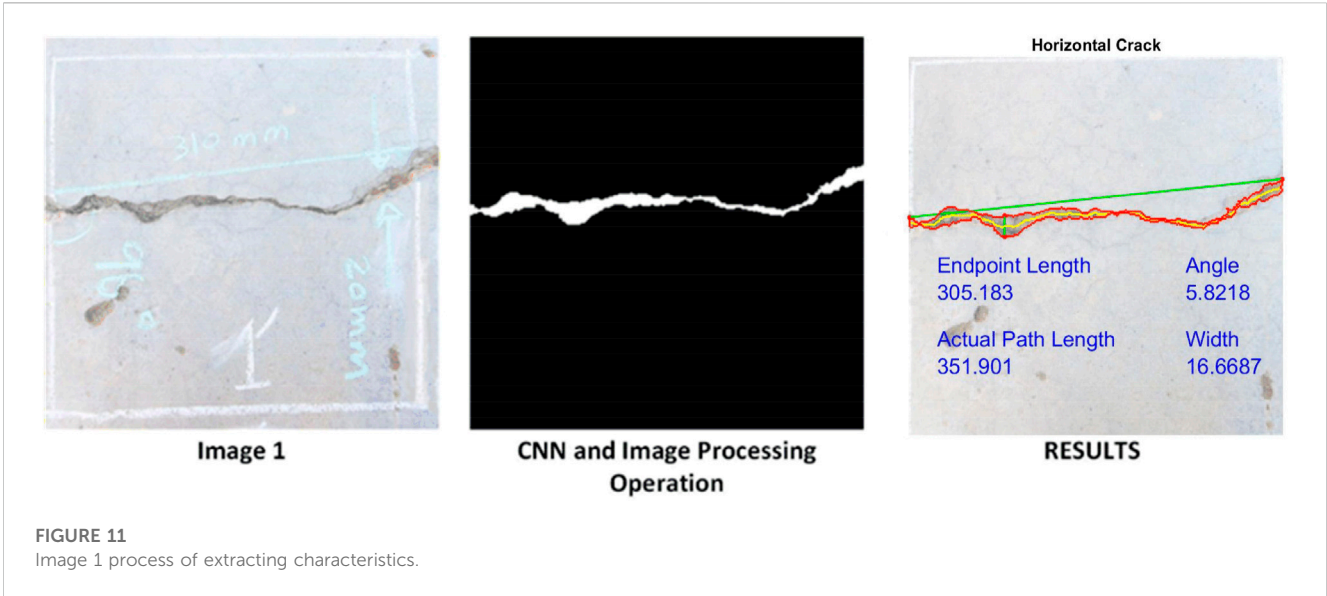
FIGURE 11
Image 1 process of extracting characteristics.

FIGURE 12
Image 2 process of extracting characteristics.

FIGURE 13
Image 3 process of extracting characteristics.

TABLE 2 Site measurement vs. measurement generated by CNN and IP.

S. No.	Measurement taken from site			Measurement using CNN and IP			Measurement of crack path
	Angle (degree)	Width (mm)	Length (mm)	Angle (degree)	Width (mm)	Length (mm)	Length (mm)
Image 1	6	20	310	5.82	16.67	305.183	351.9
Image 2	88	26	305	89.55	20.24	303.619	347.98
Image 3	23	18	334	23.416	19.05	329.56	376.16

TABLE 3 Errors in angle, width, and endpoint length of cracks.

S. No.	Relative error of angle (%)	Relative error of width (%)	Relative error of endpoint length (%)
Image 1	3	16.65	1.55
Image 2	1.76	22.15	0.45
Image 3	1.81	5.83	1.33
Mean	2.19	14.88	1.11

TABLE 4 Comparison between endpoint length and actual path length of cracks.

S. No.	Endpoint length (mm)	Crack path length (mm)	Increased length (%)
Image 1	305.183	351.9	15.31
Image 2	303.619	347.98	14.61
Image 3	329.56	376.16	14.14
Mean increased length		14.69%	

5 Conclusion

This study proposed an automated inspection system for obtaining the cracks angle, width, endpoint length, and actual path length in a concrete structure. A CNN model, Inception-V3, was trained to classify concrete images in four classes, i.e., DC, HC, VC, and UC. 32,000 images were used to train the CNN model, which was tested with 400 images. The accuracy of this model was 88.5%. Different IP techniques were utilized to extract cracks from the concrete images. Then, using array operations, the cracks angle, width, and endpoint length were measured with relative errors of 2.19%, 14.88%, and 1.11%, respectively. The algorithm tracked the actual path of the cracks and achieved the actual path length of the cracks with the mean value of 14.69% greater than the cracks endpoint length.

In reality, cracks are not straight lines and typically have a complex, meandering path that follows the contours of the surface. This increases the length of the crack path compared with the straight-line distance between the endpoints, making the actual path length of the cracks greater than the direct distance between the endpoints. Therefore, when measuring the length of a crack, it is important to consider its irregular shape and the fact that its actual path length is likely longer than the direct distance between the endpoints. Our research proposed a crack measurement technique that considered the crack shape to estimate its path length accurately.

With further development, these algorithms can be embedded in UAVs to provide a fully automated inspection system.

Data availability statement

The original contributions presented in the study are included in the article/supplementary material, further inquiries can be directed to the corresponding author.

Author contributions

WQ: investigation; writing—original draft. RE: investigation; writing—original draft. AB: conceptualization; methodology; investigation; validation; formal analysis; resources; writing—original draft; writing—review and editing; project administration. JM: investigation; validation. QUZK: investigation; formal analysis. AA: conceptualization; methodology; investigation; validation; formal analysis; resources; project administration. YÖ: validation; formal analysis. All authors contributed to the article and approved the submitted version.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

Abdel-Qader, I., Abudayyeh, O., and Kelly, M. E. (2003). Analysis of edge-detection techniques for crack identification in bridges. *J. Comput. Civ. Eng.* 17 (4), 255–263. doi:10.1061/(asce)0887-3801(2003)17:4(255)

Ahmed, C. F., Cheema, A., Qayyum, W., and Ehtisham, R. (2022). "Detection of pavement cracks of UET Taxila using pre-trained model Resnet50 of CNN," in Proceedings of the 1st International Conference on Advances in Civil and Environmental Engineering, Taxila, Pakistan, February 22–23, (IEEE).

Akbar, M. A., Qidwai, U., and Jahanshahi, M. R. (2019). An evaluation of image-based structural health monitoring using integrated unmanned aerial vehicle platform. *Struct. Control Health Monit.* 26 (1), e2276. doi:10.1002/stc.2276

Ali, R., Chuah, J. H., Talip, M. S., Mokhtar, N., and Shoaib, M. A. (2022). Structural crack detection using deep convolutional neural networks. *Automation Constr.* 133, 103989. doi:10.1016/j.autcon.2021.103989

Antunes, P., Travanca, R., Rodrigues, H., Melo, J., Jara, J., Varum, H., et al. (2012). Dynamic structural health monitoring of slender structures using optical sensors. *Sensors* 12 (5), 6629–6644. doi:10.3390/s120506629

Chollet, F. (2017). "Xception: Deep learning with depthwise separable convolutions," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, USA, 17-19, June 1997 (IEEE), 1251–1258.

Dung, C. V., and Anh, L. D. (2019). Autonomous concrete crack detection using deep fully convolutional neural network. *Automation Constr.* 99, 52–58. doi:10.1016/j.autcon.2018.11.028

Ehtisham, R., Camp, C., Mir, J., Chairman, N., and Ahmad, A. (2022). "Evaluation of pre-trained ResNet and MobileNetV2 CNN models for the concrete crack detection and crack orientation classification," in Proceedings of the 1st International Conference on Advances in Civil and Environmental Engineering, Taxila, Pakistan, February 22–23, (IEEE).

Flah, M., Suleiman, A. R., and Nehdi, M. L. (2020). Classification and quantification of cracks in concrete structures using deep learning image-based techniques. *Cem. Concr. Compos.* 114, 103781. doi:10.1016/j.cemconcomp.2020.103781

Gehri, N., Mata-Falcón, J., and Kaufmann, W. (2022). Refined extraction of crack characteristics in large-scale concrete experiments based on digital image correlation. *Eng. Struct.* 251, 113486. doi:10.1016/j.engstruct.2021.113486

Hassan, S. I., Dang, L. M., Mehmood, I., Im, S., Choi, C., Kang, J., et al. (2019). Underground sewer pipe condition assessment based on convolutional neural networks. *Automation Constr.* 106, 102849. doi:10.1016/j.autcon.2019.102849

Huang, G., Liu, Z., Van Der Maaten, L., and Weinberger, K. Q. (2017). "Densely connected convolutional networks," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, (Honolulu, CVF (Computer Vision Foundation)) 4700–4708.

Iandola, F. N., Han, S., Moskewicz, M. W., Ashraf, K., Dally, W. J., and Keutzer, K. (2017). SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0. *5MB Model size* 1–13. doi:10.48550/arXiv.1602.07360

Ji, X., Miao, Z., and Kromanis, R. (2020). Vision-based measurements of deformations and cracks for RC structure tests. *Eng. Struct.* 212, 110508. doi:10.1016/j.engstruct.2020.110508

Liu, Z., Cao, Y., Wang, Y., and Wang, W. (2019). Computer vision-based concrete crack detection using U-net fully convolutional networks. *Automation Constr.* 104, 129–139. doi:10.1016/j.autcon.2019.04.005

Lu, J., Behbood, V., Hao, P., Zuo, H., Xue, S., and Zhang, G. (2015). Transfer learning using computational intelligence: A survey. *Knowledge-Based Syst.* 80, 14–23. doi:10.1016/j.knosys.2015.01.010

Maguire, M., Dorafshan, S., and Thomas, R. J. (2018). *SDNET2018: A concrete crack image dataset for machine learning applications.* USA: Utah State University. Available at; https://digitalcommons.usu.edu/all_datasets/48/.

Munawar, H. S., Ullah, F., Heravi, A., Thaheem, M. J., and Maqsoom, A. (2021). Inspecting buildings using drones and computer vision: A machine learning approach to detect cracks and damages. *Drones* 6 (1), 5. doi:10.3390/drones6010005

Özgenel, Ç. F., and Sorguç, A. G. (2018). "Performance comparison of pretrained convolutional neural networks on crack detection in buildings," in Proceedings of the 35th International Symposium on Automation and Robotics in Construction, Berlin, Germany, 693–700.

Prasanna, P., Dana, K., Gucunski, N., and Basily, B. (2012). "Computer-vision based crack detection and analysis," in Proceedings of Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems, San Diego, CA, March 11–15, 8345, 1143–1148. doi:10.1117/12.915384

Qayyum, W., Ahmad, A., Chairman, N., and Aljuhni, A. (2022). "Evaluation of GoogLenet, Mobilenetv2, and Inceptionv3, pre-trained convolutional neural networks for detection and classification of concrete crack images," in Proceedings of the 1st International Conference on Advances in Civil and Environmental Engineering, Taxila, Pakistan, February 22–23, (IEEE).

Qayyum, W., Ehtisham, R., Bahrami, A., Camp, C., Mir, J., and Ahmad, A. (2023). Assessment of convolutional neural network pre-trained models for detection and orientation of cracks. *Mater.* 16 (2), 826. doi:10.3390/ma16020826

Redmon, Joseph. Darknet: Open source neural networks in C. Available at: https://pjreddie.com/darknet.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., et al. (2015). Imagenet large scale visual recognition challenge. *Int. J. Comput. Vis.* 115, 211–252. doi:10.1007/s11263-015-0816-y

Shamsabadi, E. A., Xu, C., Rao, A. S., Nguyen, T., Ngo, T., and Dias-da-Costa, D. (2022). Vision transformer-based autonomous crack detection on asphalt and concrete surfaces. *Automation Constr.* 140, 104316. doi:10.1016/j.autcon.2022.104316

Simonyan, K., and Zisserman, A. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 2014 Sep 4.

Song, G., Gu, H., Mo, Y. L., Hsu, T. T., and Dhonde, H. (2007). Concrete structural health monitoring using embedded piezoceramic transducers. *Smart Mater. Struct.* 16 (4), 959–968. doi:10.1088/0964-1726/16/4/003

Szegedy, C., Ioffe, S., Vanhoucke, V., and Alemi, A. (2017). "Inception-v4, inception-resnet and the impact of residual connections on learning," in Proceedings of the AAAI Conference on Artificial Intelligence, 31 (No. 1). doi:10.1609/aaai.v31i1.11231

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. "Rethinking the inception architecture for computer vision," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, USA, June 27, 2016 (IEEE), 2818–2826.

Tan, M., and Le, Q. V. EfficientNet: Rethinking model scaling for convolutional neural networks. arXiv.org. 2019. Available at: https://arxiv.org/abs/1905.11946.

Thériault, F., Noël, M., and Sanchez, L. (2022). Simplified approach for quantitative inspections of concrete structures using digital image correlation. *Eng. Struct.* 252, 113725. doi:10.1016/j.engstruct.2021.113725

Yin, X., Chen, Y., Bouferguene, A., Zaman, H., Al-Hussein, M., and Kurach, L. (2020). A deep learning-based framework for an automated defect detection system for sewer pipes. *Automation Constr.* 109, 102967. doi:10.1016/j.autcon.2019.102967

Yu, S. N., Jang, J. H., and Han, C. S. (2007). Auto inspection system using a mobile robot for detecting concrete cracks in a tunnel. *Automation Constr.* 16 (3), 255–261. doi:10.1016/j.autcon.2006.05.003

Zhang, X., Zhou, X., Lin, M., and Sun, J. (2018). "Shufflenet: An extremely efficient convolutional neural network for mobile devices," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 6848–6856.

Zoph, B., Vasudevan, V., Shlens, J., and Le, Q. V. (2018). "Learning transferable architectures for scalable image recognition," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 8697–8710.