An artificial intelligence approach to detection and assessment of concrete cracks based on visual inspection photographs

Mufaro Gomera¹ and Yunus Ballim*¹

¹School of Civil & Environmental Engineering, University of the Witwatersrand, Johannesburg

Abstract. This paper reports on the development of an artificial intelligence system, based on convolutional neural networks and machine learning algorithms to assess photographic images of concrete surfaces for the presence and characteristics of cracks. CNNs are deep learning techniques that are particularly useful for image categorization. An important challenge in the development of the system was to ensure that real cracks could be distinguished from non-crack features or profiles on the concrete surface. After development, the AI system was trained using 1900 images of cracked and non-cracked concrete surfaces. A further 1100 images were then used for validation and testing of the system. The images were segmented or pixelated in order to simplify the representation of the image and make it easier to locate objects and boundaries. The system was further developed to estimate the length and average width of cracks in an image. The testing protocols showed that the AI model was 99.6% accurate in classifying cracked and non-cracked images. Furthermore, the average error for calculation of crack length and crack width was 1.5% and 5% respectively. These results show good promise for development of a fully-fledged AI system to support inspection and maintenance of RC structures.

1 Introduction

Successful and sustainable management of reinforced concrete (RC) infrastructure demands a regular and consistent monitoring and assessment of the integrity of structures and their ability to continue performing their expected functions over time. In this process, often referred to as ‘structural health monitoring’, it is important to assess the continued integrity of the structure in response to the applied loads and to the environmental conditions that may have caused deterioration of the concrete or reinforcing steel, and the state of deterioration of the materials in the structure.

An important feature of structural health monitoring is the need to identify and then to assess the presence and nature of cracking on the surface of RC structures. The form and size of cracks, the general pattern of cracking and the changing characteristics of a particular crack over time, all provide information on the:

- cause or causes of the cracking;
- rate of structural damage and deterioration over time;
- seriousness of the cracking in regard to structural integrity and durability;
- necessary repair and rehabilitation strategies to be considered by the owner of the RC infrastructure.

In situ visual inspection has been the most often used approach to structural health monitoring because it allows for recording and monitoring of cracks on the surface of the concrete. However, the approach is error-prone, labour and time-intensive, is limited to the sections of the structure that are relatively easily accessible and demands a knowledgeable monitor who is able to judge the significance of different forms of cracking and the necessary monitoring responses over time. Furthermore, in many countries, the significant increase in the stock of RC infrastructure has proportionately increased the demand for competent monitors with the necessary resources to undertake proper monitoring. It is becoming increasingly difficult to meet this demand and, while in situ assessment and monitoring by suitably trained people is likely to continue into the foreseeable future, there is an urgent need to improve the efficiency and effectiveness of the inspection process.

In this context, digital image processing technology presents a useful tool for crack identification and diagnosis in ways that may improve the effectiveness of structural health monitoring inspections. Image processing algorithms can be used to identify cracks in images and to estimate the length, width and angle of orientation of those cracks [1, 2]. This information can then be used to evaluate the likely causes of the cracking. Analysis of structural features of photographic images, such as thresholding, are the simplest techniques to detect cracks [3]. Furthermore, general global transforms and edge detection identifiers, such as the fast Haar transform (FHT), fast Fourier transform (FFT), Sobel, and Canny edge detectors, can be used to further increase image processing technology performance [4]. Although image processing techniques (IPTs) are successful at detecting...
some specific features, their robustness is limited because images of cracks obtained from a concrete surface may be influenced by features such as light, shadows, unevenness or profiled surfaces.

Research in this area is therefore turning to machine learning algorithms (MLA) to improve the performance of image-based crack diagnostic methods [5]. The IPTs are used to extract crack features, which are then evaluated to see if the extracted characteristics suggest cracks and their causes [6]. Convolutional neural networks (CNNs) are deep learning techniques based on artificial neural networks (ANNs) and have been used to detect concrete cracks in photographs [7]. CNNs are particularly useful for image categorization and segmentation [8].

The study reported in this paper was aimed at foundational development of a deep convolutional neural network together with image processing techniques to develop an image classifier for concrete crack identification, quantification and segmentation. The primary objective of the project was to detect a crack in a photograph using MLA and IPTs. The system therefore needed to be sufficiently robust to recognise and discount non-crack features that may appear as cracks, such as joints and features in low resolution images. The secondary objective was to determine crack characteristics such as width, length and angle of orientation using IPTs. These characteristics are considered as useful in monitoring and predicting the mode of failure and degree of severity of the crack.

The CNN-based crack detection approach proposed in this paper reduces computational time by eliminating the need for pre-extraction and determination of features [9]. A total of 3000 concrete surface images were collected two-thirds with cracks and one-third without cracks. The CNNs were trained using a part of the dataset that was manually categorised while another part of the dataset was used to validate the model. The data was gathered from concrete surfaces throughout the campus of the University of Witwatersrand and the surrounding Johannesburg area. To process this large dataset, a high-performance computer with a graphics processing unit (GPU) was used to train the neural network.

2 Crack Detection and Classification

Image classification with CNN can be categorized into three types: image patch classification, boundary box regression and semantic segmentation. These classification forms are shown in Figure 1, adapted from Zhang et al. [1]. In image patch classification the image is divided in patches and each patch is labelled with a class (Figure 1a). With boundary box regression, a rectangular area bounds the detected object (a crack, in this case) and indicates its position and boundaries (Figure 1b). These two classification techniques have been extensively used to detect cracks and other defects and have shown promising results [10]. Nevertheless, these techniques are implemented at block level rather than at pixel level. On the other hand, semantic segmentation provides information about the exact location, width or length of any defects or cracks since each pixel is assigned to a class label, as shown in Figure 1c. In recent years, pixel-wise image segmentation has significantly grown in use, in preference to image patch classification and boundary box regression. A review on Deep Learning methods for semantic segmentation applied to various application areas was present by Garcia-Garcia et al. [11].

Fig. 1. Image of a crack on a concrete surface detected with (a) image patch classification, (b) boundary box regression and (c) semantic segmentation. The output of each crack detection technique is denoted by the shaded area.

Recently, Fully Convolutional Networks (FCNs) have been extensively used for semantic segmentation of images [12]. Ding and Anh [13] used FCNs for semantic segmentation of images of concrete cracks by evaluating several pre-trained network architectures serving as the backbone of the FCN encoder. Further developments of the FCN approach to allow for automatic crack segmentation [14] and for improved image processing architecture such as DenseNet-121 [15] have been demonstrated to improve the reliability of image classification and identification of concrete surface defects. In this development, special care was taken to produce a dataset of photographs from asphalt and concrete surfaces with cracks in multi-scale and multi-scene conditions to evaluate the crack detection systems. Hoskere et al [16] implemented a FCN to simultaneously identify material type (eg. concrete, steel, asphalt), as well as structural damage at both the fine level (eg. small cracks or exposed reinforcing steel) and at the coarse level (eg. spalling, corrosion) structural damage.

Although image processing techniques are successful at detecting some specific features, their robustness is limited because images of cracks collected from a concrete building may be influenced by features such as light, shadows, uneven or profiled surfaces. For the project reported in this paper, the CNN model was coupled with improved Otsu image processing [17]. All photographic images were obtained at 0.5 m to 3 m from the concrete surface, in good focus and with the optical axis approximately normal to the concrete surface. To accelerate the data preparation process, a Python code was developed to randomly extract and classify images into two classes: with cracks and without cracks.

A CNN model was trained using an image dataset of cracked and non-cracked concrete surfaces. To detect the crack, a semantic segmentation algorithm was used and this was followed by application of a thinning and
tracking algorithm [18] for image processing to calculate the crack length and width. The CNNs were trained using a part of the dataset that was manually categorised while another part of the dataset was used to validate the model. The data was gathered from concrete surfaces throughout the campus of the University of Witwatersrand and the surrounding Johannesburg area. To process this large dataset, a high-performance computer with a graphics processing unit (GPU) was used to train the neural network.

3 Image acquisition and processing

The databank of images used in this study was collected from concrete elements at the University of the Witwatersrand and the surrounding Johannesburg area. The images collected met the following requirements:

- **Photographing range** – the images were captured within a range of 0.5 m to 3 m from the concrete surface.
- **Resolution** – All images of concrete surfaces included in the databank for this project were obtained using the camera on a hand-held mobile phone with a resolution of 13 Megapixels. This level of resolution was considered as sufficient to depict unique concrete surface details such as aggregate or profiles which could be identified again in images taken later, of the same location but from a different position.
- **Focus** – images were captured in sharp and stable focus using the maximum available image resolution.
- **Illumination** – no attempt was made to control illumination of the concrete surfaces. Images were obtained during the day using the default camera settings under ambient light conditions. Lighting conditions of the images were therefore substantially different and the model was expected to account for these differences.
- **Surface angle** – the optical axis of the camera was held approximately normal to the concrete surface being photographed. This was to keep all parts of the image in focus and to allow reliable estimation of the crack width and length.

3.1 Preparing the databank of images

2000 images of cracked concrete surfaces and 1000 images of un-cracked surfaces were obtained to form the databank of images for the project. These images were grouped into three discreet sets for training (70%), validation (20%) and testing (10%) of the proposed model. The development of a Python code routine referred to above allowed an initial binary classification of the images in those with and without cracks. The code also cropped all images to 227 x 227 pixels. The purpose of cropping the images was to fit them with the CNN architecture used for the model.

The training set was larger than the validation set based on the assumption that more training samples generally lead to a more reliable model, whereas the validation dataset was required to be sufficient in size and variety to meaningfully represent the range of defects. Furthermore, to obtain a CNN classifier with suitable robustness, cropped images included cracks of different character, width and background features. Images with poor lighting conditions and linear features that may look like cracks were also included in the databank to train the neural network model.

3.2 Training the model

The deep neural network model was trained from the beginning using a visual geometry group (VGG) architecture. The neural network consists of multiple stacked layers, each layer consisting of three or four parallel convolution operations with varying filters sizes and one max pooling step. The convolution operation with a filter responds to specific features of interest which are developed throughout the training process. The first stage of the model (binary classifier) had two output nodes – ‘cracked’ and ‘un-cracked’.

Training of the model was conducted on a Windows 10 system using a workstation that was configured with a graphics processing unit. During CNN training, a batch size of 32 images was used in each update or training sequence. Each complete update of a batch size is referred to as an iteration and each complete update using the entire database is referred to as an epoch.

The CNN ‘learns’ features automatically by updating the weights within the model. Each image in the training and validation sets was labelled as cracked or un-cracked, while the images in the testing set was left as unlabelled. Figure 2 shows examples of a cracked and an un-cracked image. Each processing batch used 32 images for training and 28 images for the validation process.

![Fig. 2. Examples of an un-cracked image (a) and a cracked image (b) from the dataset used to develop the CNN model](image)

3.3 Segmenting and characterising cracks

The deep learning model was further developed so that, when a crack was identified in an image, the length, width and orientation of the crack was determined. Images with cracks were segmented at the pixel level to allow boundaries and objects to be located based on local grey-scale values. Images were first improved to remove background noise caused by unevenness of the concrete surface or non-uniform lighting effects. A thresholding algorithm based on Otsu’s method, as proposed by Yu, et
al. [19] and improved by Hoang [20], was used to perform image segmentation. The optimal threshold was selected by the discriminant criterion, which helps to maximize the separability of the resultant classes in grey levels. The Otsu approach consists of returning a single intensity threshold that separates pixels into two classes, foreground, and background. This method looks for the threshold that lowers the intra-class variance, defined as the weighted sum of variances for the two classes. In this study, the MATLAB built-in function called ‘graythresh’ was used to identify the global threshold from a greyscale image. After enhancement, the greyscale image was transformed into a binary image using the MATLAB built-in function ‘im2bw’.

Following enhancement, the crack image was subjected to further noise removal using a technique called ‘closing operation’. This is a standard mathematical morphology operator that relies on morphology transformations such as dilation and erosion. Mathematical morphology is often used for processing geometrical structures based on their typology and random functions. The MATLAB functions ‘bwmorph’ and ‘bwconncomp’ were used to re-connect parts of images that were distorted after dilation and noise removal.

3.4 Determining crack dimensions

Length of crack: The MATLAB function ‘bwboundaries’ was used to determine the exact localisation of the crack. The boundaries of the crack were then identified, illustrated as points A, B, C, and D in Figure 3. The crack length was then determined as the maximum distance between these four points, indicated as AD in Figure 3.

Width of crack: The average crack width was taken as the area of the crack determined from the grey-scale image after segmentation, divided by the length of the crack as described above.

Fig. 3. Crack length, width and orientation as determined in the CNN model.

Orientation of crack: The crack orientation (α) was taken as the absolute angle between the line used to determine the length of the crack and the vertical (for a mainly vertical crack) or the horizontal (for a mainly horizontal crack). This angle is referenced with respect to the edge of the photograph and it was considered as a necessary measurement parameter to allow relative comparison photographs of the same crack obtained in subsequent inspection cycles.

4 Results and discussion

4.1 Image segmentation

The algorithms used for image segmentation performed well in allowing the deep learning CNN to distinguish between cracked and un-cracked images. Figure 4 shows three examples of the segmentation process from the original image on the left to the finally segmented image that allowed decisions on the presence of cracking in the image.

Fig. 4. Examples of the image segmentation process for (a) an image with a crack; (b) an image with non-uniform illumination and linear feature that is not a crack; and (c) an un-cracked image.

4.2 Reliability of crack identification

Accuracy was used as the measure for validating the performance of the CNN in this study. Accuracy is defined as the number of true results (true positive and true negative) as a proportion of the total images judged by the model. The learning rate affects the validation accuracy and convergence speed during training of a CNN [21]. Using the project datasets and CNN parameters, the model converged within 15 epochs as shown in Figure 5. It is also clear from this figure that the computation converges rapidly during the first epoch. Over the range indicated in Figure 5, the model achieved a training accuracy of 99.57%. Given the acceleration capacity of a GPU, the process required 4 hours to train the CNN classifier.
Figure 5. Increasing accuracy of the model with successive epochs

Figure 6 shows the confusion or error matrix that classifies the prediction results of images in the testing dataset. The classification task obtained high accuracy of predictions and low misclassification errors. These results show good promise for development of a fully-fledged AI system to support inspection and maintenance of RC structures.

Figure 7 shows examples of the output from the CNN model on the binary decision regarding the presence of a crack in the images presented for assessment. In general, the model presented false negative decisions when the image showed a crack together with other features such as surface voids or similar features that are not cracks.

The middle image in Figure 7 also shows the model analysis of the crack characteristics, where the average crack width is determined from the crack area. This analysis illustrates the limitation of relying on a notion of the ‘average’ crack width, when the crack is made up a very wide section together with a relatively narrow section. In cases where possible corrosion of reinforcement is a concern, the larger crack zone is of greater interest than the average width. The model clearly needs to be further developed to account for such crack characteristics.

Figure 7. Examples of the model predictions showing a false negative (top); a true positive with crack characteristic determination (middle); and a true negative with linear feature.

4.3 Accuracy of determination of crack characteristics

A small sample of cracks that were photographed were also measured in the field to record the length, average crack width and orientation. Length was measured with a steel measuring tape, crack width was measured with a vernier calliper and orientation was measured using a protractor.

The crack quantification algorithm was verified using this small field test sample and gave an average error of 1.5%, 5%, and 2% for the calculation of the crack length, width, and angle of orientation respectively. Table 1 shows a selected comparison of measured and modelled crack parameters for four dataset images with cracks. This table shows reasonably good estimates from the model algorithm.

This project also highlighted the complexities of analysing images of cracks on concrete surfaces. In images with more complex cracking such as multiple or branching cracks, the CNN model correctly identified the presence of cracks but the algorithm was not able to correctly determine the characteristics of the crack,
particularly the crack width. Figure 8 shows an example where the presence of a dog-leg bend and branched crack resulted in an incorrect model assessment of the length and width of the crack.

Table 1: Comparison between measured crack parameters and the values obtained from the CNN model.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Measured</th>
<th>CNN Model output</th>
<th>% error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Length (cm)</td>
<td>27.9</td>
<td>28.7</td>
<td>2.72</td>
</tr>
<tr>
<td>Width (mm)</td>
<td>10.2</td>
<td>10.0</td>
<td>1.57</td>
</tr>
<tr>
<td>Orientation</td>
<td>38.0°</td>
<td>38.8°</td>
<td>2.08</td>
</tr>
<tr>
<td>Length (cm)</td>
<td>26.0</td>
<td>26.2</td>
<td>0.65</td>
</tr>
<tr>
<td>Width (mm)</td>
<td>1.1</td>
<td>1.0</td>
<td>9.09</td>
</tr>
<tr>
<td>Orientation</td>
<td>9.0°</td>
<td>9.1°</td>
<td>1.22</td>
</tr>
<tr>
<td>Length (cm)</td>
<td>70.0</td>
<td>69.9</td>
<td>0.07</td>
</tr>
<tr>
<td>Width (mm)</td>
<td>2.8</td>
<td>3.0</td>
<td>5.26</td>
</tr>
<tr>
<td>Orientation</td>
<td>14.5°</td>
<td>14.4°</td>
<td>0.61</td>
</tr>
<tr>
<td>Length (cm)</td>
<td>46.7</td>
<td>45.9</td>
<td>2.14</td>
</tr>
<tr>
<td>Width (mm)</td>
<td>2.1</td>
<td>2</td>
<td>4.3</td>
</tr>
<tr>
<td>Orientation</td>
<td>6.0°</td>
<td>5.7°</td>
<td>3.7</td>
</tr>
</tbody>
</table>

Figure 8. An example of incorrect model determination of crack characteristics. The branched and dog-leg crack in the original image (top) was incorrectly assessed for crack length and width by the model.

Figure 9 shows an example of an image where the CNN model correctly identified the presence of cracking and correctly determined the length and width of the large crack in the image. However, the model failed to recognise and determine the characteristics of the smaller, secondary crack in the image.

While the model is very reliable in identifying the presence of cracks on photographic images of concrete surfaces, there is clearly need for further development of its ability to determine the dimensions and characteristics of the cracks. This is an important aspect for further improvement and development of the CNN model to improve its use-value as a support instrument for more effective health monitoring of reinforced concrete structures. Correctly assessing the nature and characteristics of concrete cracking is essential in the diagnosis of the causes of cracking and the possible repair approaches that may be necessary.

5 Conclusions

- This study shows that convolutional neural networks, coupled with improved Otsu image processing can offer a powerful tool for classification, localization, segmentation and quantification of cracking in concrete structural elements, based on photographic images of the concrete surface. Importantly, the model can be developed in accessible platforms such as Python and MATLAB, to improve the effectiveness of structural health monitoring using photographic images.
- Based on the binary classification analysis of ‘cracked’ or ‘un-cracked’ concrete, the model testing accuracy was 99.57%.
- Based on the analysis of crack parameters, the measurement error was 1.5%, 5%, and 2% for determination of the length, width and orientation of cracks respectively.
- The CNN model proposed in this study requires further development to account for the presence and characteristics of more complex crack patterns such as branching of cracks, large variations in the width of cracks and multiple cracks in the same photographic image.
Acknowledgements

The authors are grateful for the support of the School of Computer Science and Applied Mathematics at the University of the Witwatersrand in granting us access to their High Performing Computing workstation. This work would also not have been possible without the kind assistance of friends and students in capturing the many images used in the databank.

References

9. Z. Qu, F. R. Ju, Y. Guo et al., Concrete surface crack detection with the improved pre-extraction and the second percolation processing methods, PloS One, 13(7), Article IDe0201109 (2018).