applied sciences

MDPI

Article

Automated Vision-Based Crack Detection on Concrete Surfaces Using Deep Learning

Rajagopalan-Sam Rajadurai and Su-Tae Kang *

Department of Civil Engineering, Daegu University, Gyeongsan 38453, Korea; samraj9593@gmail.com
* Correspondence: stkang@daegu.ac.kr; Tel.: +82-53-850-6528

Abstract: Cracking in concrete structures affects performance and is a major durability problem. Cracks must be detected and repaired in time in order to maintain the reliability and performance of the structure. This study focuses on vision-based crack detection algorithms, based on deep convolutional neural networks that detect and classify cracks with higher classification rates by using transfer learning. The image dataset, consisting of two subsequent image classes (no-cracks and cracks), was trained by the AlexNet model. Transfer learning was applied to the AlexNet, including fine-tuning the weights of the architecture, replacing the classification layer for two output classes (no-cracks and cracks), and augmenting image datasets with random rotation angles. The fine-tuned AlexNet model was trained by stochastic gradient descent with momentum optimizer. The precision, recall, accuracy, and F_1 metrics were used to evaluate the performance of the trained AlexNet model. The accuracy and loss obtained through the training process were 99.9% and 0.1% at the learning rate of 0.0001 and 6 epochs. The trained AlexNet model accurately predicted 1998/2000 and 3998/4000 validation and test images, which demonstrated the prediction accuracy of 99.9%. The trained model also achieved precision, recall, accuracy, and F_1 scores of 0.99, respectively.

Keywords: crack detection; deep learning; convolutional neural networks; image processing; AlexNet network

check for updates

Citation: Rajadurai, R.-S.; Kang, S.-T. Automated Vision-Based Crack Detection on Concrete Surfaces Using Deep Learning. *Appl. Sci.* **2021**, *11*, 5229. https://doi.org/10.3390/app11115229

Academic Editors: Yun-Kyu An and Soojin Cho

Received: 26 April 2021
Accepted: 2 June 2021
Published: 4 June 2021

1. Introduction

Many of the existing concrete structures built during the 1960–70s are rapidly nearing the end of their service life [1]. It is estimated that nearly 10% of bridges built during this time have been repaired in the United States [2]. In Korea, the number of buildings over 30 years old was evaluated as 3.8% in 2014, reaching 13.8% by 2024, and 33.7% by 2029 [3,4]. Likewise, concrete structures are often exposed to aggressive environments, fatigue stresses, and cyclic loading that initiate cracks on the surfaces [5,6]. The cracks in structures have a significant impact on durability and make it easy for external aggressive substances to reach the reinforcement bars and cause corrosion [7,8]. In addition, cracks in the structures also reduce the local stiffness and cause material discontinuities [9,10]. Therefore, cracks must be detected and repaired in time in order to maintain the reliability and performance of the structure. Generally, crack detections were performed by non-destructive and destructive tests [11]. Visual inspections combined with surveying equipment were manually performed to detect cracks in the structures [12]. A PZT-based electro-mechanical admittance method combined with FEM analysis was enacted to quantitatively identify the damage caused by concrete cracking and steel yielding of flexural beams subjected to monotonic and cyclic loading [13]. Chalioris et al. [14] developed a wireless impedance/admittance monitoring system to identify the incipient damages caused by concrete cracking. In addition, non-destructive testing techniques such as infrared, thermal, ultrasonic, laser, and radiographic tests were also used to detect and analyze the crack development in concrete structures [15]. Although the above methods provide reliable crack detection results, they are difficult and time-consuming to perform because they require large instrumentation, and are expensive and labor intensive [16]. To overcome the shortcomings of the manual methods, several image processing methods were developed to provide automated crack detection and visualization in concrete structures. Most of the image processing methods used filtering, thresholding, and feature extraction techniques to identify and localize the cracks [17–22]. Furthermore, the crack regions were separated by fuzzy transforms and segmentation algorithms [21]. Although image processing methods were effective in detecting cracks, the real-time applicability in structures was limited due to the variations in external environmental factors such as light, shadows, and rough surfaces. To improve the performance of image processing techniques, machine learning algorithms were developed through pattern recognition and extraction [23]. Machine learning algorithms such as support vector machine (SVM) and artificial neural network (ANN) have also been explored to detect cracks in the concrete structures [24–27]. A local entropy-based thresholding algorithm was proposed that automatically detects spalled regions on the surface of the reinforced concrete columns [28]. In addition, the length and width of cracks were also measured using a local binary pattern (LBP) algorithm [29]. Machine learning algorithms consisting of feature extraction and classification were used to extract relevant crack features. The machine learning algorithm extracts only a few layers of features, and the algorithm might not provide accurate crack detection results if the extracted features do not reflect the cracks.

Deep learning algorithms such as convolutional neural networks (CNNs) have been used in many studies for crack detection and classification to improve the feature extraction process. CNN models can extract relevant features from the input data through multilayer neural networks, which are more advantageous than the existing limitations of image processing and machine learning methods. CNN-based crack detection was performed for the safety diagnosis and localization of damages in concrete structures in [30]. Similarly, Bayesian algorithms were used to identify cracks in nuclear power plants, and deep learning segmentation algorithms were used to identify cracks in the tunnels [31]. Furthermore, deep convolutional neural networks (DCNNs) were recently explored for crack detection and classification [32,33]. Most of the DCNNs focused on pixel-wise crack classification through semantic segmentation by associating each pixel [32,34]. Deep learning networks require a large amount of training data and time. These can be minimized by fine-tuned pretrained DCNNs that use small amounts of data and provide reliable results in minimal time. Fine-tuned pretrained DCNNs such as AlexNet, GoogleNet, ResNet, SqueezeNet, and VGGNet have recently been used to detect and classify cracks in concrete structures. A VGG19 pretrained model was applied to create pixel-level crack maps on concrete pavements and walls [35]. Crack segmentations were performed using SegNet, U-Net, and ResNet models [36–39]. A DenseNet-121-based fully convolutional network was studied to provide the pixel-level detection of multiple damages including cracks, spalling, efflorescence, and holes in concrete structures [40]. Furthermore, DCNNs based on VGG16 were also used for crack segmentation on the concrete surfaces [41].

All aforementioned deep learning approaches have shown promising performance in the crack detection of structures. Since the performance of DCNNs depends on various factors, such as data, filters, the number of layers, the number of epochs, and the network depth, it is difficult to select an appropriate pre-trained DCNN for crack detection with high precision and accuracy. The advantage of selecting an appropriate DCNNs is that it ensures better generalization and prevents overfitting. AlexNet, with many pre-trained DCNNs, is the most influential CNN widely applied to image classification and won the ImageNet LSVRC-2012 competition with a minor error rate of 15.3% [42]. The highlights of AlexNet are listed as follows: there are more filters in each layer; each convolutional layer is followed by a pooling layer; it uses ReLU instead of tanh, arctan, and logistic to add non-linearity that increases speed by up to 6x with the same accuracy; it uses a dropout layer instead of regularization to deal with overfitting; and it makes use of an overlap pooling layer to reduce the size of the network [43,44]. These characteristics motivated the utilization of AlexNet in this study for crack detection and classification.

This study utilized AlexNet, a pre-trained deep convolutional neural network, for the automated vision-based crack detection and classification. The proposed method consists of three steps: (1) collecting a large number of images from an open-source image dataset with subsequent categorization of two classes (no-crack and crack images); (2) developing a DCNN model, transferring the learning and augmentation process; and (3) automatically detecting and classifying the images using the trained deep learning model. Additionally, a cross-dataset study was performed to verify the ability of the trained AlexNet model. The precision, recall, accuracy, and F_1 metrics were used to evaluate the performance of the trained AlexNet model. The accuracy of the trained AlexNet model was further compared to other pretrained DCNNs such as GoogleNet, ResNet101, InceptionResNetv2, and VGG19.

2. Methodology

2.1. Scheme of the CNN Model

In this study, a pre-trained DCNN was used for automated crack detection and classification. Pre-trained DCNNs consist of convolutional layers for extracting features and classifying images. Pre-trained DCNNs have been widely used in many applications to classify images, and there are many pre-trained DCNNs available (e.g., AlexNet, GoogleNet, ResNet, SqueezeNet, and VGGNet). This study used the AlexNet pre-trained model to detect and classify images in three stages: image database acquisition, CNN model and transfer learning process, and classification. Figure 1 shows the scheme of crack detection and classification model. First, an image database consisting of thousands of images was acquitted for two classes: crack images and no-crack images. Second, a CNN classifier model was developed to detect and classify images using AlexNet. Third, the trained DCNNs detected cracks and classify the set of validation and test images. Then, the cross-dataset was used to verify the ability of the trained model.

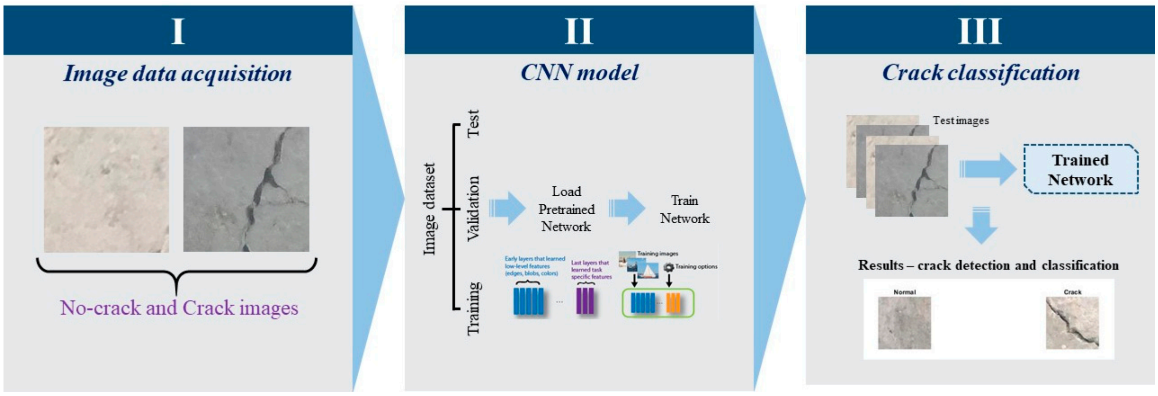


Figure 1. Scheme of the crack detection model.

2.2. Image Database Acquisition

An open-source dataset of concrete crack images was used for detection and classification [45]. The image dataset consists of 20,000 images, evenly divided into crack and no-crack classes, with an input image size of 227 × 227 × 3 pixels. The image dataset was divided into 70% for training, 10% for validation, and 20% for testing. The image dataset details are shown in Table 1. For training, classification, and testing, the images were divided into 14,000, 2000, and 4000 images, respectively. This study also used an image dataset consisting of crack and no-crack images to perform a cross-dataset study [46]. The dataset consists of 16,285 images taken on bridge-decks, walls, and pavements. The details of the cross-image datasets are shown in Table 2.

Table 1. Concrete crack image datasets.

Total Number of Images	20,000
Crack Images	10,000
No-crack Images	10,000
Size (Pixels)	227 × 227
Number of Color Channels	3
Training Data	14,000
Validation Data	2000
Test Data	4000

Table 2. Cross-image datasets.

Total Number of Images	16,285
Bridge-Decks	4086
Pavements	7300
Walls	4899

2.3. AlexNet CNN Model

A CNN consists of several hidden layers as well as input and output layers. The layers of a CNN generally consist of convolutional, ReLU, pooling, fully connected, and normalization layers. This study analyzed the image database by applying a DCNN classifier to classify the input images into two categories: no-crack and crack. The CNN network was designed based on AlexNet for image classification. Figure 2 shows an overview of the CNN classifier based on AlexNet. AlexNet, a large neural network with 60 million parameters and 650,000 neurons, consists of 5 convolutional layers followed by max-pooling layers, 3 fully connected layers, and a final 1000-way SoftMax layer. AlexNet has been widely trained on more than a million images and can classify images into 1000 classes. Since the number of image classes in this study was two (no-crack and crack), the output number of classes was changed to two.

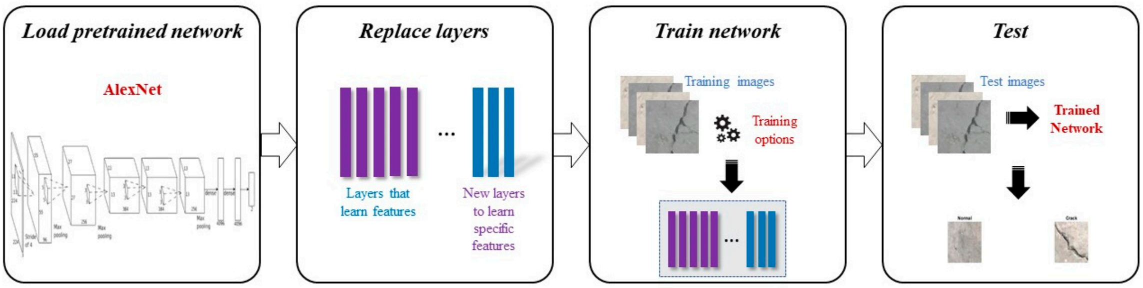


Figure 2. AlexNet CNN classifier.

A new activation function was used in the AlexNet neural networks to provide nonlinearity. Several traditional activation functions, including logistic function, tanh function, and arctan function, tend to cause gradient vanishing problems. To overcome this, a new activation function was used, the rectified linear unit (ReLU), and its definition is shown in Equation (1).

$$\mathrm{ReLU}(x) = \max(x, 0) \tag{1}$$

Deep neural networks with ReLU as the activation function converge faster than those with tanh units. Dropout was employed in fully connected layers to avoid overfitting that trains only a portion of the neurons in each iteration. The dropout reduces joint adaptation between neurons and improves generalization and robustness. Convolution was employed for automatic feature extraction and defined as in Equation (2).

$$C(m,n) = (M.w)(m,n) = \sum_k \sum_l M(m-k, n-l)w(k,l) \quad (2)$$

where w is the convolution kernel. Pooling was employed for automatic feature reduction, which considered a group of neighboring pixels in the feature map and generated a representation value. Cross-channel normalization was used to improve the generalization. In addition, fully connected layers were used for classification in which the neurons in fully connected layers were directly linked. The SoftMax activation function was expressed as in Equation (3). The SoftMax activates neurons by constraining the output in the range of (0, 1).

$$\text{SoftMax } (x)_i = \frac{\exp(x_i)}{\sum_{j=1}^{n} \exp(x_j)} \quad (3)$$

2.3.1. Transfer Learning

The network analyzer was applied to display interactive visualizations of network architectures and detailed information about network layers. The network architecture is shown in Table 3. The first layer was an image input layer with an input image size of $227 \times 227 \times 3$, where the 3 is the number of color channels. Additionally, the CNN consisted of convolution layers, pooling layers, fully connected layers, and the SoftMax layer. It also included other operations such as ReLU, cross-channel normalization, and dropout layers. The last three layers, fully connected, SoftMax, and the classification output layer of the pretrained network, were configured for 1000 classes. These three layers were fine-tuned by the transfer learning for the two classes (no-crack and crack) as shown in Table 4.

Table 3. Network architecture of AlexNet.

No.	Layer	Output
1	Image Input	227 × 227 × 3
2	Convolution	55 × 55 × 96
3	ReLU	55 × 55 × 96
4	Cross Channel Normalization	55 × 55 × 96
5	Max Pooling	27 × 27 × 96
6	Grouped Convolution	27 × 27 × 256
7	ReLU	27 × 27 × 256
8	Cross Channel Normalization	27 × 27 × 256
9	Max Pooling	13 × 13 × 256
10	Convolution	13 × 13 × 384
11	ReLU	13 × 13 × 384
12	Grouped Convolution	13 × 13 × 384
13	ReLU	13 × 13 × 384
14	Grouped Convolution	13 × 13 × 256
15	ReLU	13 × 13 × 256
16	Max Pooling	6 × 6 × 256
17	Fully Connected	1 × 1 × 4096
18	ReLU	1 × 1 × 4096
19	Dropout	1 × 1 × 4096
20	Fully Connected	1 × 1 × 4096
21	ReLU	1 × 1 × 4096
22	Dropout	1 × 1 × 4096
23	Fully Connected	1 × 1 × 1000
24	SoftMax	1 × 1 × 1000
25	Classification Output	1000

Table 4. Transfer learning.

AlexNet Layers			Transfer Learning		
No.	**Layers**	**Output**	**No.**	**Layers**	**Output**
23	Fully Connected	$1 \times 1 \times 1000$	23	Fully Connected	$1 \times 1 \times 2$
24	SoftMax	$1 \times 1 \times 1000$	24	SoftMax	$1 \times 1 \times 2$
25	Classification Output	1000 classes	25	Classification Output	2 classes

2.3.2. Augmentation Process

The AlexNet network requires input images of size $227 \times 227 \times 3$. Therefore, image augmentations were used to automatically resize the training images as the image size in the datastore may differ. Additional augmentation operations to perform on the training images were also specified and included randomly flipping the training images along the vertical axis and randomly converting up to 30 pixels horizontally and vertically. The data augmentation prevented the network from overfitting and memorizing the exact details of the training images.

2.4. Training and Classification by AlexNet CNN Model

Matlab R2020b was used for image processing and data analysis. The fine-tuned AlexNet model was trained by stochastic gradient descent with momentum (SGDM) optimizer. The initial learning rate was set as 0.001 and 0.0001, the minibatch size was set as 15, and the max epoch was set as 6. After the training, the validation images and test images were classified using the fine-tuned network, and the images were displayed with their predicted labels. To quantify the accuracy of the trained model, the precision, recall, accuracy, and F_1 scores were computed using Equations (4)–(7).

$$\text{Precision} = \frac{TP}{TP + FP} \tag{4}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{5}$$

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{6}$$

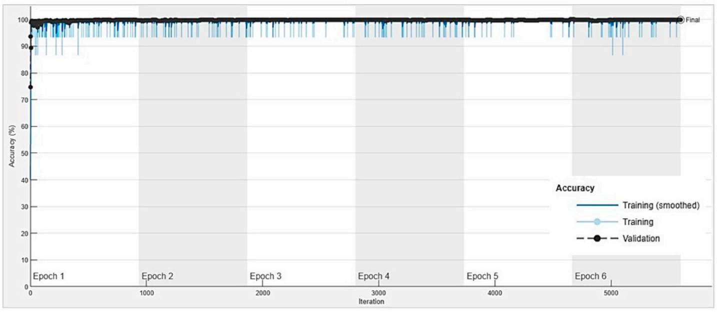
$$F_1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{7}$$

where TP, FP, FN, and TN represent true positive, false positive, false negative, and true negative, respectively.

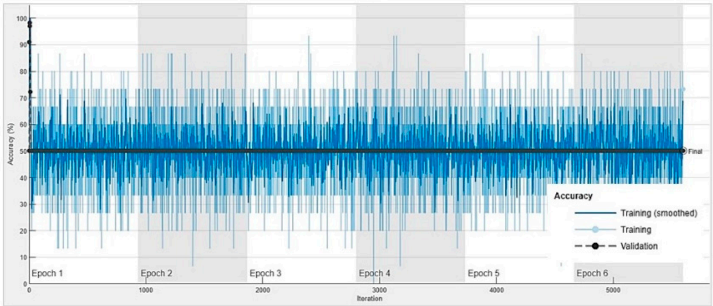
3. Results and Discussion

3.1. Performance of the Trained Network

The training progress included the accuracy and cross-entropy loss for each epoch of training and validation. To determine the appropriate learning rate, neural networks with different learning rates over 6 epochs were trained. The training progress with learning rates of 0.0001 and 0.001 were trained and compared. During the training process, the maximum number of iterations was 5598, with 933 iterations per epoch. The training progress plot of accuracy (%) with different learning rates is shown in Figure 3. The accuracy obtained from the 0.0001 learning rate and 6 epochs was 99.9%. The change in accuracy at the 0.0001 learning rate was minimal after 1 epoch. With the 0.001 learning rate, the obtained accuracy was 50%. At two different learning rates, a better performance was achieved with the learning rate of 0.0001 after 6 epochs. In the same way, the training progress plot of loss (%) with different learning rates is shown in Figure 4. The loss obtained from the 0.0001 learning rate was 0.1%. At the 0.001 learning rate, the acquired loss of training and validation was 50%. In the two different learning rates, the loss was least at the 0.0001 learning rate. Considering the above results, the learning rate of 0.0001 and 6 epochs was fixed and trained in this study.

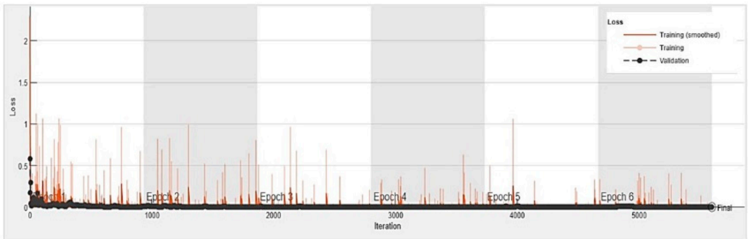


(**a**)

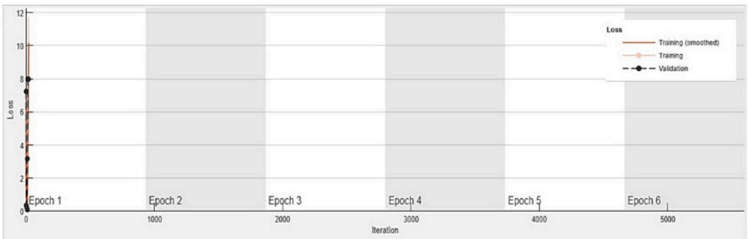


(**b**)

Figure 3. Training progress plot of accuracy (%) with different learning rates: (**a**) 0.0001; (**b**) 0.001.



(**a**)



(**b**)

Figure 4. Training progress plot of loss (%) with different learning rates: (**a**) 0.0001; (**b**) 0.001.

3.2. Classification Using the Trained Network

The pre-trained DCNN was trained with the training set of 14,000 images using the AlexNet model, which obtained 99.9% accuracy during the training process. After training, the trained model was validated before the test. The validation image dataset consisted of 10% of the total image dataset, and had 2000 images. During the validation process, the trained model predicted the images into two classes: crack and no-crack. Sample images of predicted crack and no-crack classes in the validation images are shown in

Figure 5. The confusion matrix for the validation images is shown in Figure 6. From the set of 2000 images, 1000 images had cracks and 1000 images had no-cracks. An amount of 1998 of the 2000 images were accurately predicted, representing a 99.9% accuracy. In the crack image dataset, 1000 images were accurately predicted. Similarly, 998 no-crack images were accurately predicted. The prediction accuracies of crack and no-crack images were both 99.9%. In the validation images, 99.9% accuracy and 0.1% loss were obtained. The performance metrics were computed, and are shown in Table 5. The precision, recall, and F_1 scores obtained from the confusion matrix were 1, 0.99, and 0.99. The prediction accuracy was 0.99.

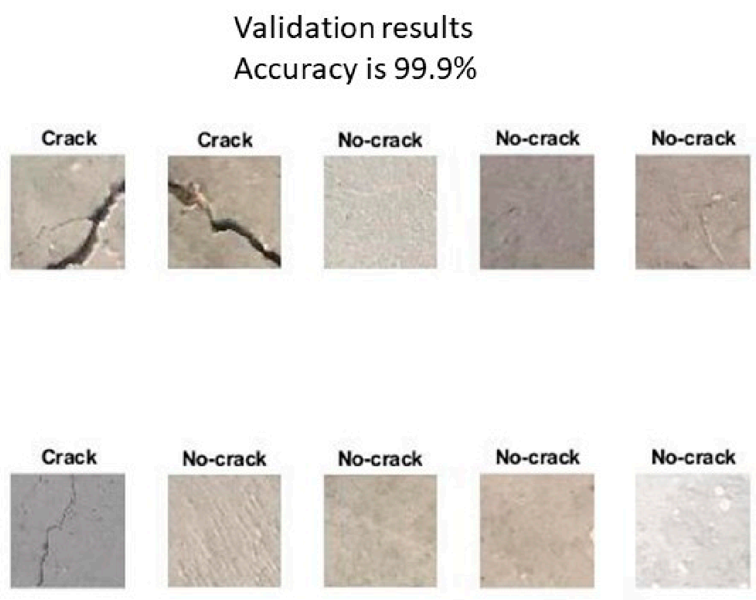


Figure 5. Classification and labelled images with the validation data.

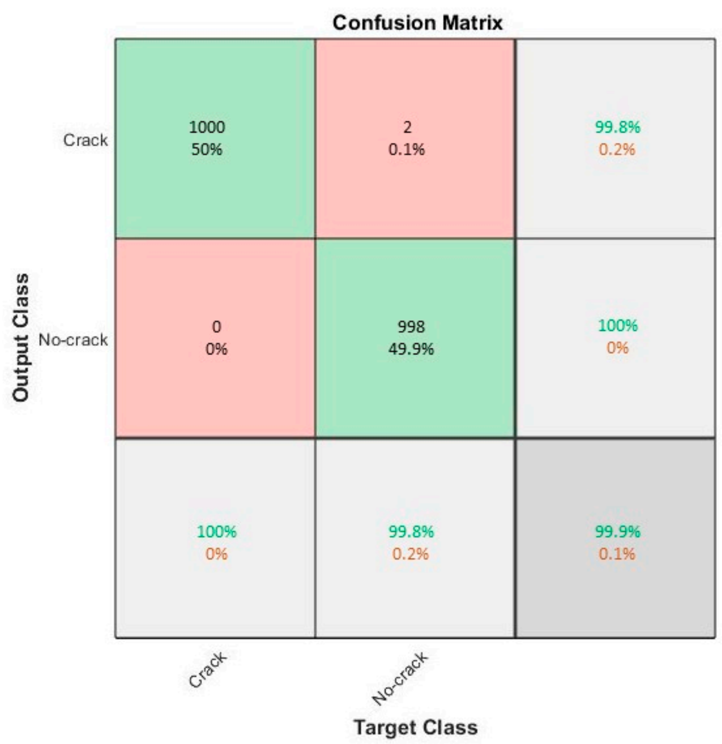


Figure 6. Confusion matrix of validation images.

Table 5. Performance metrics of the validation images.

Precision	Recall	F_1 Score	Accuracy
1	0.99	0.99	0.99

After the validation of the trained model, the test images were classified using the trained network. The test image dataset consists of 20% of the images from the original dataset, comprising 4000 crack and no-crack images. The predicted crack and no-crack classes of the test images are shown in Figure 7. In the test image dataset, there were 2000 images of each class (crack and no-crack). The confusion matrix of the test images is shown in Figure 8. The model trained in the test images accurately predicted 3998 from a total of 4000 images. Only two images were left unpredicted. A total of 1999 crack images and 1999 non-crack images were accurately predicted, representing a 99.99% accuracy. Considering the total test images, the prediction accuracy was 99.9% with a 0.1% loss. The computed performance metrics are shown in Table 6. The precision, recall, and F_1 scores obtained from the confusion matrix were all 0.99. In addition, the accuracy of the prediction in the test images was 0.99.

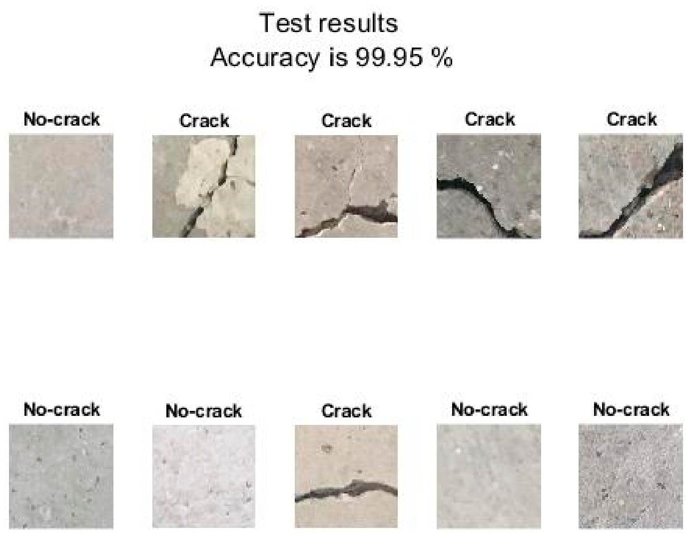


Figure 7. Classification and labelled images with the test data.

Table 6. Performance metrics of the test results.

Precision	Recall	F_1 Score	Accuracy
0.99	0.99	0.99	0.99

The accuracy of the trained model was further compared to the other pretrained models and is shown in Table 7. The GoogleNet, ResNet101, InceptionResNetv2, and VGG19 DCNNs were compared to the trained AlexNet DCNN model. The AlexNet, GoogleNet, and VGG19 obtained accuracies of 0.99. In addition, the ResNet101 and InceptionResNetv2 models obtained accuracies of 0.9833 and 0.95, respectively. While the other DCNN models also obtained high accuracies, the AlexNet has fewer layers compared to other DCNNs, and can be trained in less time. The other DCNNs have more layers for feature extraction, which requires more time for training. Therefore, AlexNet was superior to other pretrained DCNNs for crack detection and classification.

3.3. Cross-Dataset Study of the Trained Network

To validate the ability of the trained AlexNet model, a cross-dataset was tested using different images that were not used for training. The dataset consists of crack and no-crack images taken on bridge-decks, walls, and pavements. Examples of the cross-image dataset are shown in Figure 9.

Table 7. Accuracies of pretrained DCNNs.

No.	Pretrained DCNN	Accuracy
1	AlexNet	0.99
2	GoogleNet	0.99
3	ResNet101	0.9833
4	InceptionResNetV2	0.95
5	VGG19	0.99

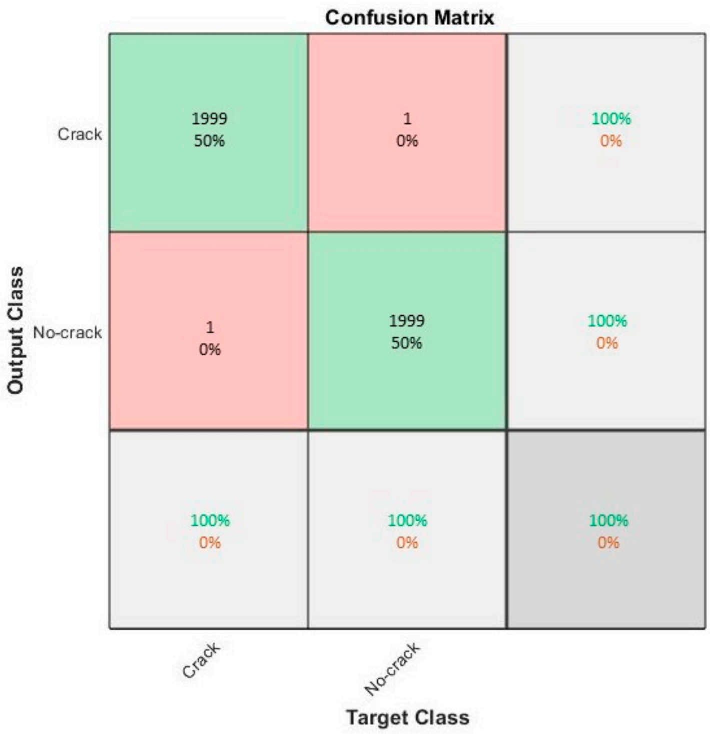
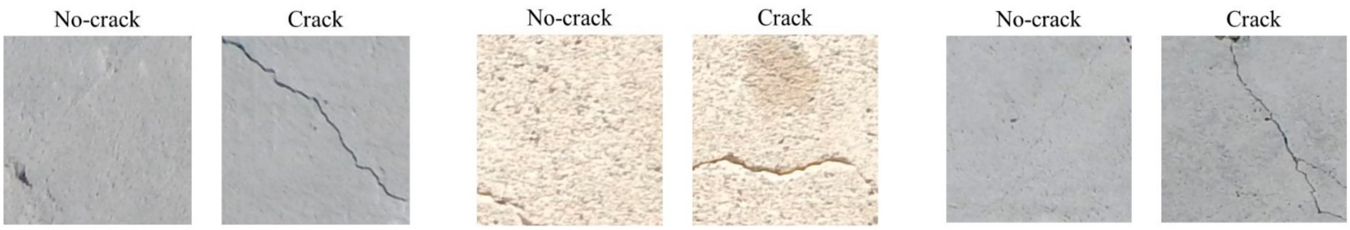


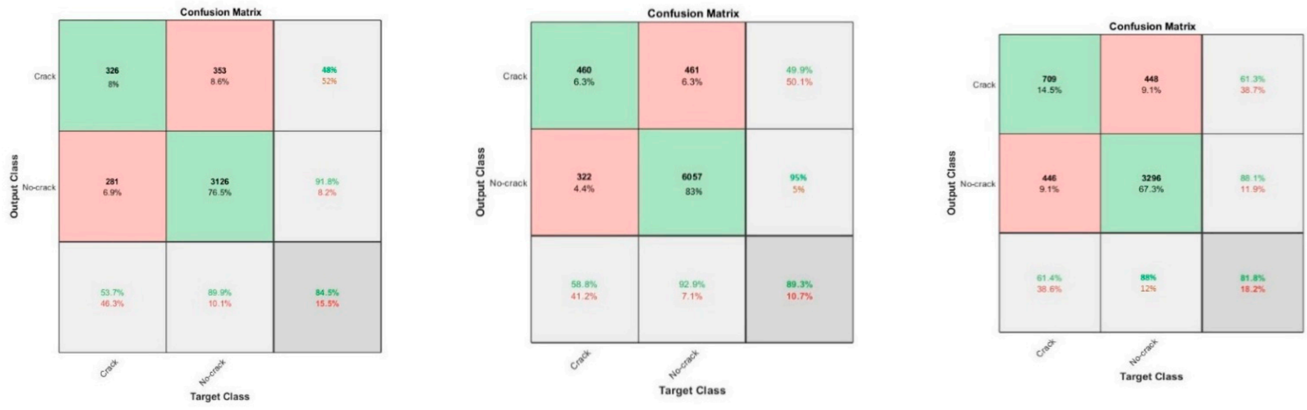
Figure 8. Confusion matrix of test images.



(**a**) (**b**) (**c**)

Figure 9. Examples of cross-dataset images: (**a**) bridge-decks; (**b**) pavements; (**c**) walls.

The trained AlexNet model was saved, and a cross-image dataset was tested using the trained model. The trained model predicted the images taken on bridge-decks with an accuracy of 84.5%. The trained model also predicted the images taken on pavements and walls with 89.3% and 81.9% accuracy, respectively. The loss obtained from the images taken on bridge-decks, pavements, and walls was 15.5%, 10.7%, and 18.2%, respectively. The confusion matrix of the cross-dataset test images is shown in Figure 10. To quantify the trained model, the precision, recall, accuracy, and F_1 scores were computed and are presented in Table 8. For the bridge-deck images, the obtained precision, recall, and F_1 scores were 0.89, 0.91, and 0.90. The precision, recall, and F_1 scores obtained from the images taken on pavements were all 0.92. Similarly, the precision, recall, and F_1 scores obtained from the images taken on walls were 0.88, 0.82, and 0.85, respectively. In addition, the prediction accuracies for the three categories were 0.84, 0.89, and 0.81, respectively. The prediction accuracy obtained from the original dataset was 0.99, while the accuracy of the cross-dataset were decreased to 0.84, 0.89, and 0.81, respectively. These decreases in the prediction accuracy were due to the presence of a variety of obstructions including shadows, surface roughness, scaling, edges, holes, and background debris in the images [46]. These obstructions resulted in the loss of accurate prediction of the images.



(**a**) (**b**) (**c**)

Figure 10. Confusion matrices of the cross-dataset images: (**a**) bridge-decks; (**b**) pavements; (**c**) walls.

Table 8. Performance metrics of cross-dataset test results.

Type of Dataset	Precision	Recall	F_1 Score	Accuracy
Bridge-decks	0.89	0.91	0.90	0.84
Pavements	0.92	0.92	0.92	0.89
Walls	0.88	0.82	0.85	0.81

4. Conclusions

This study investigated automated crack detection based on a CNN. An open-source image dataset with two subsequent classes (no-crack and crack) was used. The image datasets were divided into 70%, 10%, and 20% for training, validation, and testing, respectively. The CNN model was designed based on AlexNet for image classification. AlexNet consists of convolution layers, pooling layers, fully connected layers, and SoftMax layers, as well as other operations such as ReLU, cross-channel normalization, and dropout layers. The last three layers (fully connected, SoftMax, and classification output layer) of the pretrained network were fine-tuned by transfer learning for the two classes (no-crack and crack). Image augmentations were then used to automatically resize the training images. The fine-tuned AlexNet model was trained by stochastic gradient descent with momentum (SGDM) optimizer. After training, the validation images and test images were classified using the fine-tuned network.

The fine-tuned AlexNet model was trained, and the training progress evaluated the accuracy and cross-entropy loss for each epoch. The accuracy obtained at the 0.0001 learning rate and epoch 6 was 99.9%, and the validation loss was 0.1%. The trained model was validated, and it accurately predicted 1998 from 2000 images. The accuracy obtained during the validation was 99%, and the loss of accurate prediction was 0.1%. After the validation, the test images were classified using the trained network. In the test images, the trained model accurately predicted 3998 from the total 4000 images. Considering the total test images, the prediction accuracy was 99.9% with 0.1% loss. This study confirmed that the CNN-based method demonstrates a high level of applicability to detect cracks, with a 99.9% accuracy. The performance of the trained model was quantified by the precision, recall, accuracy, and F_1 metrics, which were all equal to 0.99. Furthermore, the accuracies were compared to other pretrained DCNNs. AlexNet showed an accuracy of 0.99, which is beneficial for detecting and classifying cracks with high precision. The trained AlexNet model was further tested with different cross-dataset images which consisted of several obstructions, including shadows, surface roughness, scaling, edges, holes, and background debris. The existence of these obstructions resulted in a nominal loss of accurate predictions, with an accuracy of around 0.81–0.89%.

Author Contributions: Conceptualization, methodology, S.-T.K. and R.-S.R.; software, R.-S.R.; validation, formal analysis, S.-T.K.; writing—original draft preparation, review, and editing, S.-T.K. and R.-S.R. Both authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: This study did not report any data.

Acknowledgments: This research was supported by the Daegu University Research Grant (No. 20200222).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Kim, H.; Ahn, E.; Shin, M.; Sim, S.H. Crack and non-crack classification from concrete surface images using machine learning. *Struct. Health Monit.* **2018**, *18*, 725–738. [CrossRef]
2. ASCE's 2017 Infrastructure Report Card | GPA: D+. Available online: https://infrastructurereportcard.org/tag/2020/ (accessed on 1 March 2021).
3. Park, C.-H.; Lee, H.-I. *Future Trend of Capital Investment for Korean Transportation Infrastructure*; Construction and Economy Research Institute of Korea: Seoul, Korea, 2016.
4. Kim, B.; Cho, S. Automated multiple concrete damage detection using instance segmentation deep learning model. *Appl. Sci.* **2020**, *10*, 8008. [CrossRef]
5. Huang, X.; Yang, M.; Feng, L.; Gu, H.; Su, H.; Cui, X.; Cao, W. Crack detection study for hydraulic concrete using PPP-BOTDA. *Smart Struct. Syst.* **2017**, *20*, 75–83.
6. Kim, H.; Ahn, E.; Cho, S.; Shin, M.; Sim, S.-H. Comparative analysis of image binarization methods for crack identification in concrete structures. *Cem. Concr. Res.* **2017**, *99*, 53–61. [CrossRef]
7. Song, J.; Kim, S.; Liu, Z. A real time nondestructive crack detection system for the automotive stamping process. *IEEE Trans. Instru. Meas.* **2016**, *65*, 2434–2441. [CrossRef]
8. Bas, P.Y.L.; Anderson, B.E.; Remillieux, M. Elasticity nonlinear diagnostic method for crack detection and depth estimation. *J. Acoust. Soc. Am.* **2015**, *138*, 1836. [CrossRef]
9. Budiansky, B.; O'connell, R.J. Elastic moduli of a cracked solid. *Int. J. Sol. Struct.* **1976**, *12*, 81–97. [CrossRef]
10. Aboudi, J. Stiffness reduction of cracked solids. *Eng. Fract. Mech.* **1987**, *26*, 637–650. [CrossRef]
11. Dhital, D.; Lee, J.R. A fully non-contact ultrasonic propagation imaging system for closed surface crack evaluation. *Exp. Mech.* **2012**, *52*, 1111–1122. [CrossRef]
12. Oliveira, H.; Correia, P.L. Automatic road crack detection and characterization. *IEEE Trans. Intell. Transp. Syst.* **2013**, *14*, 155–168. [CrossRef]
13. Chalioris, C.E.; Kytinou, V.K.; Voutetaki, M.E.; Karayannis, C.G. Flexural damage diagnosis in reinforced concrete beams using a wireless admittance monitoring system—Tests and finite element analysis. *Sensors* **2021**, *21*, 679. [CrossRef]
14. Chalioris, C.E.; Maristella, E.V.; Angelos, A.L. Structural health monitoring of seismically vulnerable RC frames under lateral cyclic loading. *Earthq. Struct.* **2020**, *19*, 29–44.
15. Jahanshahi, M.R.; Jazizadeh, F.; Masri, S.F.; Becerik-Gerber, B. Unsupervised approach for autonomous pavement-defect detection and quantification using an inexpensive depth sensor. *J. Comput. Civ. Eng.* **2012**, *27*, 743–754. [CrossRef]
16. Fujita, Y.; Hamamoto, Y. A robust automatic crack detection method from noisy concrete surfaces. *Mach. Vis. Appl.* **2011**, *22*, 245–254. [CrossRef]

17. Yiyang, Z. The design of glass crack detection system based on image preprocessing technology. In Proceedings of the IEEE 7th Joint International Information Technology and Artificial Intelligence Conference, Chongqing, China, 20–21 December 2014; pp. 39–42.
18. Broberg, P. Surface crack detection in welds using thermography. *NDT E Int.* **2013**, *57*, 69–73. [CrossRef]
19. Wang, P.; Huang, H. Comparison analysis on present image-based crack detection methods in concrete structures. In Proceedings of the 3rd International Congress on Image and Signal Processing, Yantai, China, 16–18 October 2010; pp. 2530–2533.
20. Cheng, H.D.; Shi, X.J.; Glazier, C. Real-time image thresholding based on sample space reduction and interpolation approach. *J. Comput. Civ. Eng.* **2003**, *17*, 264–272. [CrossRef]
21. Yamaguchi, T.; Nakamura, S.; Saegusa, R.; Hashimoto, S. Image-based crack detection for real concrete surfaces. *IEEJ Trans. Electr. Electron. Eng.* **2008**, *3*, 128–135. [CrossRef]
22. Yeum, C.M.; Dyke, S.J. Vision-based automated crack detection for bridge inspection. *Comput. Civ. Infrastruct. Eng.* **2015**, *30*, 759–770. [CrossRef]
23. Zhang, W.; Zhang, Z.; Qi, D.; Liu, Y. Automatic crack detection and classification method for subway tunnel safety monitoring. *Sensors* **2014**, *14*, 19307–19328. [CrossRef] [PubMed]
24. Moussa, G.; Hussain, K. A new technique for automatic detection and parameters estimation of pavement crack. In Proceedings of the 4th International Multi-Conference on Engineering Technology Innovation, Orlando, FL, USA, 19–22 July 2011.
25. Gavilan, M.; Balcones, D.; Marcos, O.; Llorca, D.F.; Sotelo, M.A.; Parra, I.; Ocana, M.; Aliseda, P.; Yarza, P.; Amirola, A. Adaptive road crack detection system by pavement classification. *Sensors* **2011**, *11*, 9628–9657. [CrossRef] [PubMed]
26. Lee, B.J.; Lee, H.D. Position-invariant neural network for digital pavement crack analysis. *Comput. Aided Civ. Infrastruct. Eng.* **2004**, *19*, 105–118. [CrossRef]
27. Saar, T.; Talvik, O. Automatic asphalt pavement crack detection and classification using neural networks. In Proceedings of the 12th Biennial Baltic Electronics Conference, Tallinn, Estonia, 4–6 October 2010; pp. 345–348.
28. Stephanie, G.; Ioannis, B.; Reginald, D. Rapid entropy-based detection and properties measurement of concrete spalling with machine vision for post-earthquake safety assessments. *Adv. Eng. Inf.* **2012**, *26*, 846–858.
29. Song, K.; Yan, Y. A noise robust method based on completed local binary patterns for hot-rolled steel strip surface defects. *Appl. Surf. Sci.* **2013**, *285*, 858–864. [CrossRef]
30. Prasanna, P.; Dana, K.J.; Gucunski, N.; Basily, B.B.; Hung, M.L.; Lim, R.S.; Parvardeh, H. Automated crack detection on concrete bridges. *IEEE Trans. Autom. Sci. Eng.* **2016**, *13*, 591–599. [CrossRef]
31. Chen, F.C.; Jahanshahi, M.R. Nb-cnn: Deep learning-based crack detection using convolutional neural network and naive bayes data fusion. *IEEE Trans. Ind. Electron.* **2018**, *65*, 4392–4400. [CrossRef]
32. Zhang, L.; Yang, F.; Zhang, Y.D.; Zhu, Y.J. Road crack detection using deep convolutional neural network. In Proceedings of the 2016 IEEE International Conference on Image Processing, Phoenix, AZ, USA, 25–28 September 2016; pp. 3708–3712.
33. Makantasis, K.; Protopapadakis, E.; Doulamis, A.; Doulamis, N.; Loupos, C. Deep convolutional neural networks for efficient vision based tunnel inspection. In Proceedings of the 2015 IEEE International Conference on Intelligent Computer Communication and Processing, Cluj-Napoca, Romania, 3–5 September 2015.
34. Cha, Y.J.; Choi, W.; Buyukozturk, O. Deep learning-based crack damage detection using convolutional neural networks. *Comput. Aided Civ. Infrastruct. Eng.* **2017**, *32*, 361–378. [CrossRef]
35. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
36. Badrinarayanan, V.; Handa, A.; Cipolla, R. Segnet: A deep convolutional encoder decoder architecture for robust semantic pixel-wise labelling. *arXiv* **2015**, arXiv:1505.07293.
37. Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. In Proceedings of the International Conference on Medical Image Computing and Computer Assisted Intervention, Munich, Germany, 5–9 October 2015; pp. 234–241.
38. Bang, S.; Park, S.; Kim, H.; Kim, H. Encoder-decoder network for pixel-level road crack detection in black-box images. *Comput.-Aided Civ. Infrastruct. Eng.* **2019**, *34*, 713–727. [CrossRef]
39. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the 30th IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 27–30 June 2016; pp. 770–778.
40. Li, S.; Zhao, X.; Zhou, G. Automatic pixel-level multiple damage detection of concrete structure using fully convolutional network. *Comput. Aided Civ. Inf.* **2019**, *34*, 616–634. [CrossRef]
41. Dung, C.V. Autonomous concrete crack detection using deep fully convolutional neural network. *Autom. Constr.* **2019**, *99*, 52–58. [CrossRef]
42. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.* **2012**, *445*, 1097–1105. [CrossRef]
43. Zhang, J.; Meng, Y.; Wu, J.; Qin, J.; Yao, T.; Yu, S. Monitoring sugar crystallization with deep neural networks. *J. Food Eng.* **2020**, *280*, 109965. [CrossRef]
44. Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 248–255.
45. Özgenel; Fırat, C. Concrete Crack Images for Classification. *Mendeley Data* **2019**, *V2*. [CrossRef]
46. Maguire, M.; Dorafshan, S.; Thomas, R.J. SDNET2018: A concrete crack image dataset for machine learning applications. *Utah State Univ.* **2018**. [CrossRef]