INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
Online ISSN 1841-9844, ISSN-L 1841-9836, Volume: 18, Issue: 1, Month: Feburary, Year: 2023
Article Number: 4977, https://doi.org/10.15837/ijccc.2023.1.4977

CCC Publications

Deep Learning for Assessing Severity of Cracks in Concrete Structures

A. BaniMustafa, R. AbdelHalim, O. Bulkrock, A. Al-Hmouz

Ahmed BaniMustafa*
Data Science and Artificial Intelligence Department
Isra University, Amman, Jordan
*Corresponding author: a.banimustafa@iu.edu.jo

Rozan AbdelHalim
Computer Science Department
Middle East University, Amman, Jordan

Olla Bulkrock
Data Science Department
Princess Sumaya University for Technology, Amman, Jordan

Ahmad Al-Hmouz
Computer Science Department Middle East University, Amman, Jordan

Abstract

Most concrete structures suffer from degradation, where cracks are the most obvious visual sign. Concrete structures must be continuously monitored and assessed to avoid further deterioration, which may lead to a partial or total collapse. This is particularly important when constructing large structures such as towers, bridges, tunnels, and dams. This work aims to demonstrate and evaluate several deep learning approaches that can be used for monitoring and assessing the level of concrete degradation based on the cracks' visual signs, which can then be embedded in Health Monitoring Systems (SHM). The experimental work in this study involves creating three models: Two were built using ResNet-50 and Xception transfer learning networks. In contrast, the third was built using a customized Sequential Convolutional Neural Network (SCNN) architecture. The dataset comprises 2,000 image samples sampled from a larger dataset that contains 56,000 images and which belong to four severity classes: minor, moderate, and severe, in addition to a normal class for no crack signs. The SCNN model achieved an accuracy of 90.2%, while the Xception and ResNet-50 models scored an accuracy of 86.3% and 70%, respectively.

Keywords: Machine Learning, Deep Learning, Structural Health monitoring (SHM), Concrete Structure, Civil Engineering

1 Introduction

Concrete is the dominant material in civil engineering and is used to construct houses, buildings, towers, bridges, tunnels, dams, and other structures. However, concrete structures may deteriorate during or after the plastic construction phase. Visual signs such as cracks are among the most significant indications of degradation, which may lead to the partial or total failure of the structure if not assessed and maintained carefully. Cracks may appear due to external and internal causes. Internal causes happen during the plastic phase and hardening, while external causes are associated with the environmental conditions after hardening [31]. Ignoring cracks may lead to a high repair cost or expose the structure to partial or complete collapse that may endanger lives and incur huge financial losses [22, 26].

Deep learning is a back-propagation neural network with multiple hidden layers [20]. Deep learning has achieved great success in several scientific and engineering applications, reported in several recent studies [1, 5, 6, 7, 8, 9]. Convolutional Neural Networks (CNN) is one of the most popular deep learning networks, particularly in image classification and object recognition. CNN is based on a feed-forward approach that extracts filtered features as the image is passed through the network convolutions [17].

This work investigates and evaluates the use of three deep learning architectures for assessing the severity of concrete structure degradation based on a dataset of 2,000 images sampled from a larger dataset containing 56,000 images [14]. The constructed deep learning models can be embedded in a structural health monitoring (SHM), which could provide an automated solution to real-time detection of cracks and assess the severity level of their severity [16, 25].

The proposed approach involves constructing three deep learning models: the first two were built using a pre-trained version of the ResNet-50 [18], and Xception [12] neural network architectures. In contrast, the third was built using customized sequential convolutional neural network(CNN) architecture. All the created models are trained and tested using two separate datasets: 1,400 images are used for training, while another 600 images are used for testing. The accuracy [19, 21] and loss of both the training and validation phases are used for model comparison and evaluation. The performance of the champion model is then examined in comparison with other models reported in the literature. The paper is organised in a standard fashion. Section two reviews seven related works, while section three presents the dataset and its relevant processing. The fourth section provides an overview of the proposed research methodology and evaluation mechanisms, while section five presents the results. The seventh section discusses the deep learning results. In contrast, the final section concludes the paper with a summary of its findings. It discusses the prospect of its utilization and comments on the limitations and future work of the research.

Research Hypothesis: Deep learning can predict the levels of concrete structure degradation.

Research Question: Can deep learning assess the severity of concrete structure damage based on crack images?

Research Aim: Creating and evaluating a deep learning model which is used for detecting, assessing and monitoring the progression of concrete structure degradation.

2 Related Work

This section reviews the recent and popular work published in the field of automatic assessment of concrete structures and provides a critical analysis of the datasets used; techniques applied; number of labels; and performance achieved in each study. Table 1 provides a summary for review.

Yamaguchi et al. proposed an approach for identifying cracks in concrete images that depends on a percolation model, a scalable local processing method. The proposed method depends on extracting the texture of images by measuring the brightness and shape of cracks in images. The authors reported a precision of 70

Dung and Anh reported using a fully convolutional network to classify cracks using a dataset that consisted of 40,000 images that had a dimension of (227 X 227) pixels. The authors reported using Inception V3, VGG16, and ResNet algorithms, achieving an average precision of 90% [15].

Author	Dataset	Techniques Applied	Performance	Classes
Yamaguchi et. al. (2008)	N/A	Percolation Model	70% Precision, 90% Recall%	Binary
Dung and Anh (2019)	40,000	VGG16, ResNet	90% Precision	Binary
da Silva and de Lucena (2018)	3,500	VGG16	92.27% CA	Binary
Cha et. al. (2017)	332	CNN	98% CA	Binary
Arbaoui et. al. (2021)	2,000	AlexNet,ResNet-50	90% CA	Binary
Yamane et. al. (2020)	100	Semantic Segmentation and Deep Learning	79% Precision, 99%CA	Binary
Billah et. al. (2019)	33	Encoder-Decoder	15% Precision, 56% Recall, 98.5% CA	Binary

Table 1: A comparison of the related work that is based on the number of predicted classes, size of the dataset, techniques applied and results obtained

Da Silva and Lucena reported applying the VGG16 architecture [24] to a set of 3,500 images divided into cracked and non-cracked classes. The authors reported a classification accuracy of 92.27% when the dataset was split into two subsets. The first contained 80% of the dataset and was used for model training, while the other contained 20% of the samples and was used for model testing [23].

Cha et al. reported using a convolutional neural network to detect cracks in 332 images, which have a dimension of (256 X 256) pixels. The authors reported a 98% accuracy rate in recognising cracks in the concrete images. The authors reported success in distinguishing crack from non-crack classes [11].

Arbaui et al. reported using AlexNet and ResNet-50 for classifying cracked and non-cracked concrete images. The authors used a dataset that consisted of 1000 images. The authors reported a classification accuracy of 88.5% using AlexNet and 91% using ResNet-50, and a precision of 89.2% using AlexNet and 91.8% and using ResNet-50 [2].

Yamane et al. reported a study originally published in Japanese and later translated into English. The authors proposed a method that depends on semantic segmentation through deep learning. The researchers reported a sensitivity of 78.8% and classification accuracy of 99% using a dataset of 100 images. The study classified the images into two classes: cracked and non-cracked samples [30].

Billah et al. reported using an encoder-decoder deep learning-based algorithm for classifying cracked and non-cracked concrete samples. The researcher reported an accuracy of 98.5%, a precision of 15.3%, and a recall of 56.1%. The dataset consisted of 33 large images with a resolution of (3304 x 3456) pixels, which were used for generating a dataset of 6000 images with a resolution of (512 x 512) pixels. The validation in this study was conducted using 200 images with a size of (1024 X 1024) pixels [10].

The reported performance of the models in these studies varies, as do the applied evaluation metrics. The classification accuracy of five of the seven reviewed studies was quite impressive, as it exceeded a classification accuracy of 98%. However, most of the models reported in these studies need more robustness, which is quite apparent in the gap between precision and classification accuracy reported in these studies, such as [10, 29, 30]. However, most other models lack robustness, which is quite apparent in the gap between precision and classification accuracy results [10, 29, 30].

Most of the reported techniques in the reviewed studies depended on using known deep learning architectures such as CNN [11], VGG16[15, 23], ResNet-15 [2, 15], AlexNet [2]. In contrast, others applied other techniques such as percolation model[30], semantic segmentation [30], and encoder-decoder [10] algorithm.

In addition, although some researchers tried to rely on a large dataset consisting of tens of thousands of images, most of the reported studies use samples that vary between tens, hundreds, and a few thousand images. On the other hand, a few researchers, such as [10] tried to use another approach where the large dataset of images was generated from a few tens of high-resolution images.

The analysis of the related works investigated shows that most of these studies aim to classify samples into cracked and non-cracked classes. This is a modest aim given what deep learning achieves using its current cutting-edge technology. This identified gap is intended to be addressed in this study by considering four severity classes. The study also aims to improve the performance and robustness of the constructed model by using a considerable number of images (2,000 images) distributed equally in each class.

3 Materials

The dataset used in this study consists of 56,000 images that represent various types of concrete structures, such as buildings, walls, tunnels, dams, and bridges [14]. A sample of 2,000 images was then selected to represent three classes of concrete degradation severity and a normal class. The domain expert labelled the images in each class according to the crack's width, depth, shape, orientation, and count in each sample. The dataset was split into two parts: the first part consists of 1,400 images, which are used for model training, while the second part consists of 600 images, which are used for model testing. Figure 1 demonstrates a sample of crack images for each class. Images augmentation was applied to the dataset images using Keras to increase the performance of the model and also to avoid over-fitting. Figure 2 demonstrates a variety of image orientations of an image in the sample dataset.

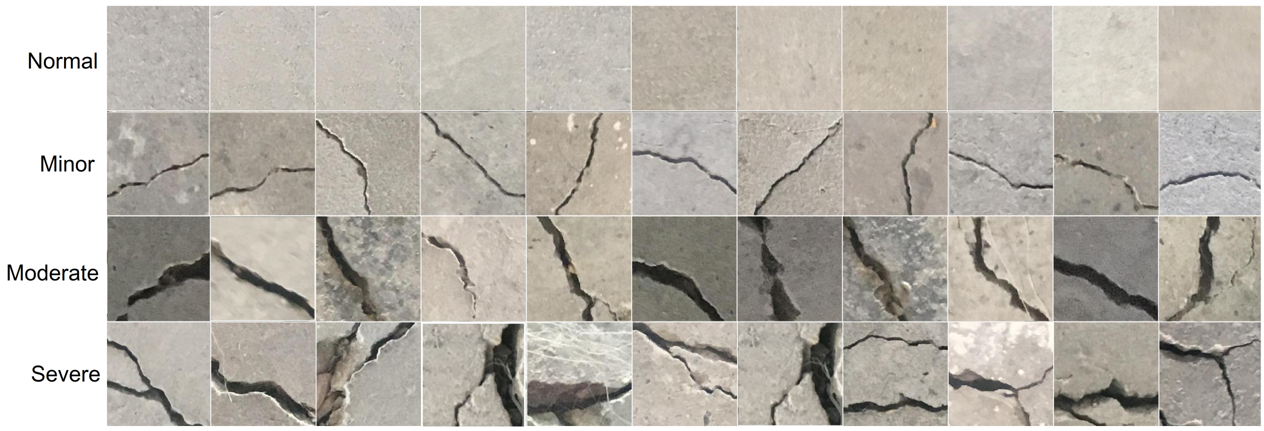

Figure 1: Sample of image for each class

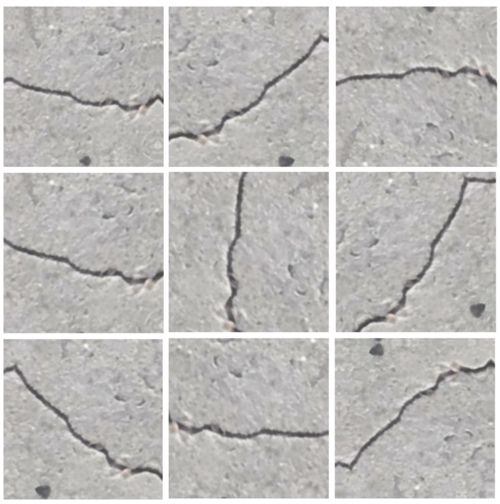

Figure 2: A sample crack image at different orientation

4 Methods

The proposed method involves constructing three deep learning models based on convolutional neural network architectures: the first two are constructed using ResNet-50 and Xception pre-trained models. At the same time, the third is built using a customized sequential convolutional neural network architecture (SCNN). The work of these three network architectures is similar to filtering, where each convolution in the network layers filters some of the image features and then passes it for filtering by the next layer.

ResNet-50

ResNet is a deep residual neural network architecture that was first introduced in [18]. It uses a residual function to reference its layers' inputs. The architecture was designed particularly for image recognition and classification. ResNet consists of a very deep hidden layer that can reach eight-fold the depth of typical convolutional neural networks such as VGG16 [24]. ResNet-50 is a popular version of the architecture, which consists of 50 layers. The size of its inputs is 224-by-224. ResNet-50 can be used as a pre-trained model which uses its assigned weights based on ImageNet [13]. Fig. 3 illustrates the architecture of the ResNet-50 network.

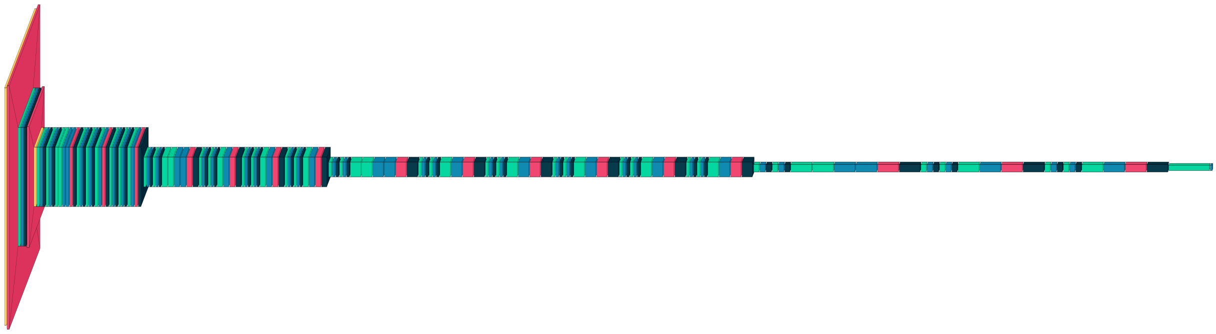

Figure 3: An illustration of the Residual Neural Network architecture (ResNet-50)

Xception

Xception is a modified deep neural network architecture based on Inception V3 [27, 28]. The architecture organizes the network into a number of depth-wise separable convolutions where the standard inception modules are placed as an intermediate step in the middle [12]. Xception achieved slightly better performance than its predecessor architecture (Inception V3), particularly using the ImageNet database, and outperformed the Inception V3 architecture significantly when applied to a larger database consisting of 350 million images and more labels which reached a total number of 17,000 classes. Fig. 4 illustrates the architecture of the Xception network.

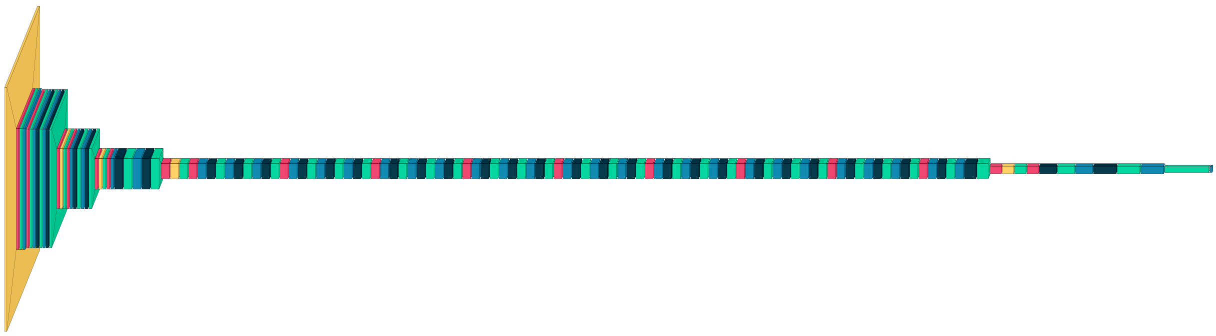

Figure 4: An illustration of the Xception network architecture

Sequential convolutional neural networks (SCNN)

A customised version of sequential convolutional neural networks (SCNN) [3] is used in this study. The network was built in Python using Keras and TensorFlow [4], which were directly applied to extract image features and classify them using several predefined epochs [4] implementations in Python. SCNN involves extracting the image features and classifying them directly into four predefined classes. Each corresponds to a level of concrete crack severity. Fig. 5 illustrates the architecture of a typical sequential convolutional neural network (SCNN), while Fig. 6 shows the constructed SCNN sequential model architecture and configuration.

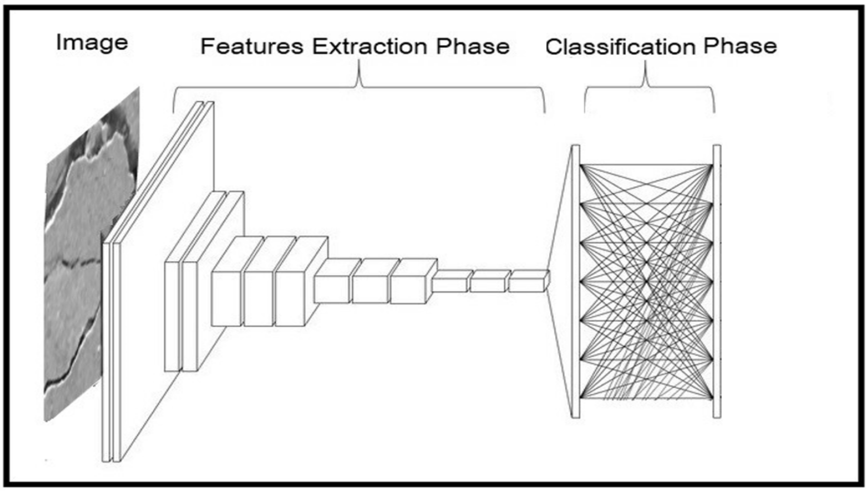

Figure 5: Sequential convolutional neural network (SCNN) architecture

Model: "sequential_2"

Layer (type)	Output Shape	Param #
sequential_1 (Sequential)	(None, 224, 224, 3)	0
rescaling_2 (Rescaling)	(None, 224, 224, 3)	0
conv2d_3 (Conv2D)	(None, 224, 224, 16)	448
max_pooling2d_3 (MaxPooling2	(None, 112, 112, 16)	0
conv2d_4 (Conv2D)	(None, 112, 112, 32)	4640
max_pooling2d_4 (MaxPooling2	(None, 56, 56, 32)	0
conv2d_5 (Conv2D)	(None, 56, 56, 64)	18496
max_pooling2d_5 (MaxPooling2	(None, 28, 28, 64)	0
dropout (Dropout)	(None, 28, 28, 64)	0
flatten_1 (Flatten)	(None, 50176)	0
dense_2 (Dense)	(None, 128)	6422656
dense_3 (Dense)	(None, 4)	516

Total params: 6,446,756
Trainable params: 6,446,756
Non-trainable params: 0

Figure 6: SCNN sequential model Architecture

Performance Evaluation Method

Classification Accuracy (CA) [19, 21] is used as the study's major performance evaluation metric. The equation of the classification accuracy metric is shown in Equation 1. The model training and validation performance is scored [3] and then plotted to show the progress of model training and validation for each epoch. All the constructed models are trained using 75% of the images, while 25% are used for model testing.

$$CA = (TruePositive + TrueNegative)/(ActualPositive + ActualNegative) \tag{1}$$

5 Results

All three constructed models successfully classified the crack images into their correct severity levels with an excellent classification performance. Figure 7 illustrates the results of image filtering as it passes through the convolutional neural network layers. The performance results of each model are reported and explained in their respective subsections.

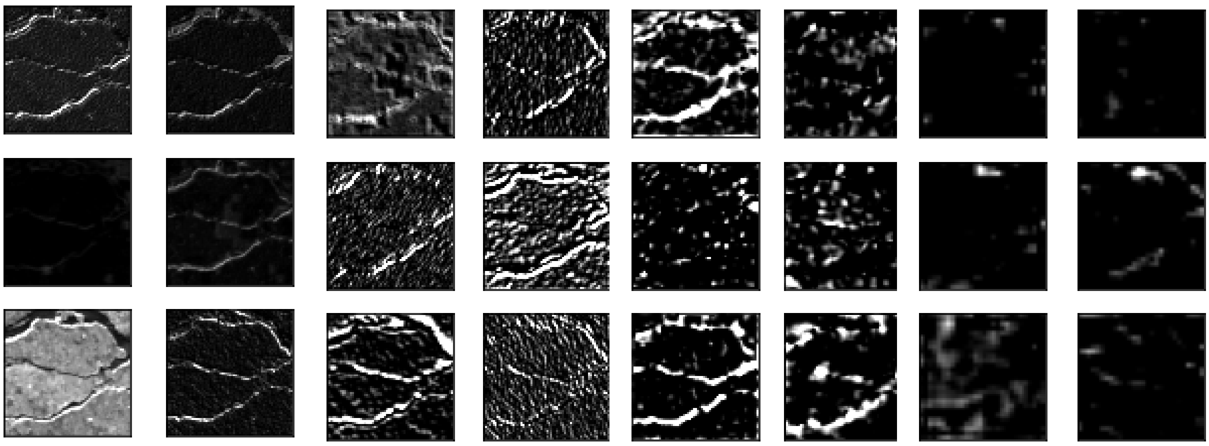

Figure 7: an example image as passing through the CNN convolutions

ResNet-50 Model

The performance of the ResNet-50 model was evaluated using both training and validation classification accuracy based on 50 epochs and 44 batch sizes.

The model scored 85.2% in training and 70.0% in validation classification accuracy. The results were produced using Keras and TensorFlow implementations in Python.

The model was built and trained using a separate random training dataset, which makes 75% of the total number of images, and then tested using an independent testing sample that makes 25% of the images in the dataset. The training samples were placed in a separate folder and then loaded using their appropriate generator in the Keras preprocessor at a learning rate of 0.001.

Table 2 shows the training and validation accuracy for each epoch. In contrast, the curve in Fig. 8 shows the progression of classification accuracy and loss rate for each epoch.

Xception Model

Xception model performance was assessed using training and validation datasets based on the model classification performance. The epoch size in each iteration was set to 50, while the size of each batch was set to 88.

Like ResNet-50, the Python Keras and TensorFlow implementations were used to build, train, and test the model using the typical parameter settings reported in similar applications in the deep learning literature.

Epoch.	Training CA	Validation CA	Epoch	Training CA	Validation CA
1	25.5%	40.0%	**26**	78.6%	71.5%
2	52.7%	62.3%	**27**	83.5%	73.0%
3	65.4%	63.8%	**28**	82.3%	65.7%
4	64.3%	66.2%	**29**	82.7%	71.7%
5	68.0%	73.8%	**30**	86.7%	73.8%
6	68.0%	67.0%	**31**	85.8%	68.5%
7	73.7%	69.8%	**32**	77.8%	71.0%
8	71.5%	56.8%	**33**	85.3%	62.2%
9	69.0%	68.0%	**34**	81.0%	74.5%
10	74.0%	62.0%	**35**	87.0%	73.2%
11	69.2%	58.8%	**36**	84.3%	68.7%
12	73.6%	60.3%	**37**	85.7%	71.8%
13	70.0%	73.3%	**38**	81.8%	73.5%
14	73.8%	70.2%	**39**	87.7%	65.5%
15	77.4%	70.2%	**40**	78.5%	66.8%
16	77.4%	69.2%	**41**	78.5%	67.7%
17	79.4%	70.3%	**42**	82.4%	69.3%
18	78.5%	61.7%	**43**	79.9%	66.0%
19	76.4%	68.3%	**44**	83.3%	72.2%
20	83.6%	70.8%	**45**	84.5%	69.9%
21	78.6%	68.2%	**46**	84.5%	68.7%
22	83.5%	70.2%	**47**	78.4%	72.5%
23	82.3%	75.2%	**48**	78.1%	72.0%
24	82.7%	71.2%	**49**	82.1%	75.7%
25	86.7%	67.5%	**50**	85.2%	70.0%

Table 2: The ResNet-50 model performance over training and validation accuracy based on 50 epochs

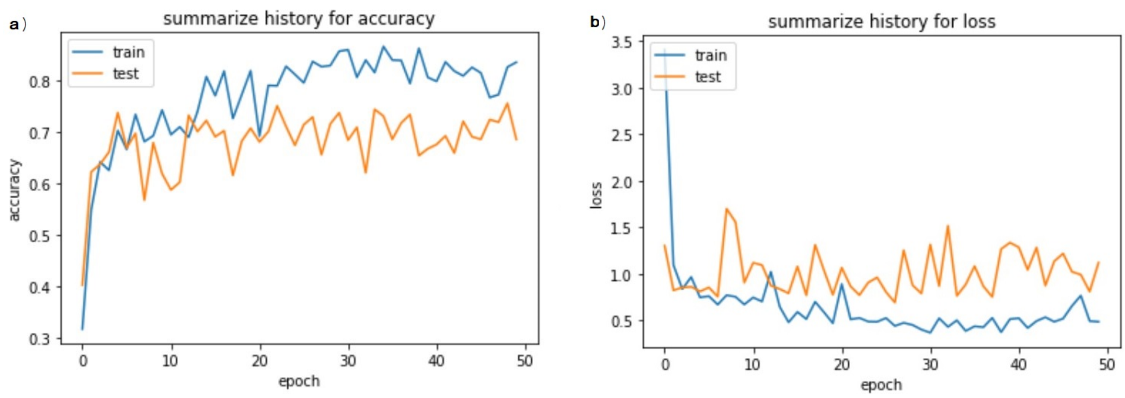

Figure 8: The ResNet-50 model performance: a) Training and validation classification accuracy curve b) Training and validation loss curve

Likewise, the ResNet-50, the Xception model, was also built and trained using a separate and random training dataset, which makes 75% of the total number of images, and then tested using an independent testing sample that makes 25% of the images in the dataset.

The model scored 97.7% in training and 86.3% in validation classification accuracy. The training samples were placed in separate folders and then loaded using their appropriate generator in the Keras preprocessor at a learning rate of 0.001.

Table 3 shows the training and validation accuracy for each epoch, while the curve in Fig. 9 shows the progression of classification accuracy and loss rate for each epoch.

Epoch.	Training CA	Validation CA	Epoch	Training CA	Validation CA
1	80.5%	75.3%	**26**	96.8%	80.7%
2	87.8%	85.1%	**27**	95.9%	82.8%
3	89.9%	84.8%	**28**	96.6%	82.5%
4	93.2%	81.0%	**29**	96.6%	82.3%
5	91.4%	81.3%	**30**	96.9%	84.2%
6	92.8%	82.0%	**31**	96.9%	80.7%
7	94.2%	82.1%	**32**	96.9%	83.5%
8	94.6%	83.0%	**33**	97.9%	84.2%
9	94.6%	81.7%	**34**	96.8%	83.0%
10	95.8%	83.7%	**35**	97.5%	81.5%
11	95.2%	84.0%	**36**	96.9%	83.3%
12	95.2%	83.3%	**37**	98.1%	86.3%
13	96.2%	83.0%	**38**	97.5%	82.8%
14	95.2%	83.1%	**39**	98.1%	82.7%
15	96.2%	78.8%	**40**	98.0%	82.2%
16	95.5%	84.3%	**41**	97.6%	83.8%
17	95.4%	83.5%	**42**	97.7%	85.2%
18	97.8%	84.2%	**43**	97.3%	83.5%
19	96.7%	84.0%	**44**	97.9%	81.8%
20	97.1%	85.2%	**45**	97.4%	84.8%
21	96.3%	83.3%	**46**	97.3%	83.8%
22	95.7%	84.2%	**47**	97.6%	81.7%
23	96.5%	81.8%	**48**	97.3%	83.8%
24	97.3%	84.8%	**49**	97.6%	86.0%
25	97.4%	87.0%	**50**	97.7%	86.33%

Table 3: The Xception model performance over training and validation accuracy based on 50 epochs

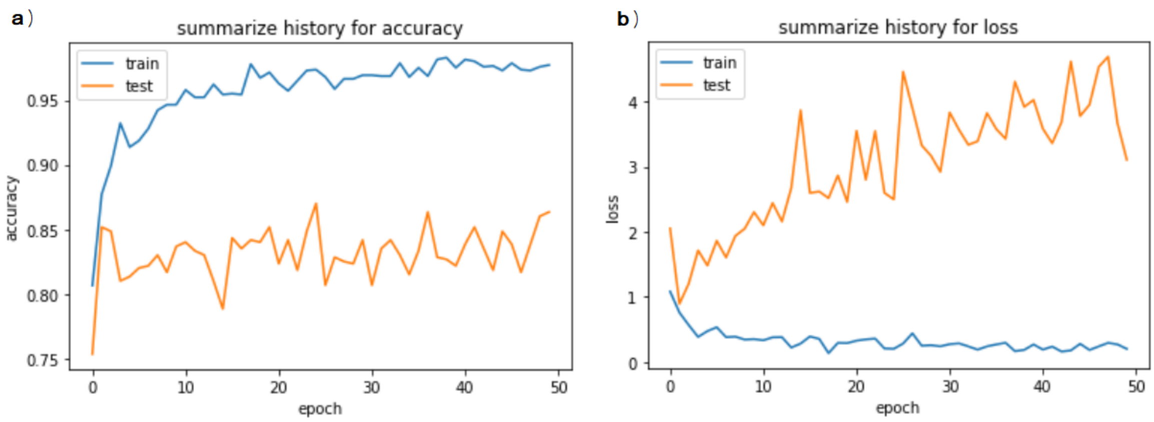

Figure 9: The Xception model performance: a) Training and validation classification accuracy curve b) Training and validation loss curve

SCNN Model

The SCNN model was constructed using three activation functions: ReLU, Sigmoid, and Tanh. However, the results obtained using the ReLU function were far better than the other two; therefore, their results have been discarded. The activation functions were applied using a configuration of (16, 3), (32, 3), and (64, 3) 2D convolutions.

The flat dropout penalty was assigned to 0.2. The results of applying the Sequential Convolutional Neural Networks (SCNN) -implemented in Keras and TensorFlow- were quite acceptable, with 90.2% validation accuracy when the model was trained and evaluated using 20 epochs.

Table 4 shows the progression of the model training and validation performance for each epoch, while Table 5 shows the model confusion matrix. Fig. 10 shows the model accuracy and loss progression through the 20 epochs.

Epoch	Training CA	Validation CA
1	26.5%	29.8%
2	51.3%	62.2%
3	73.2%	81.4%
4	77.0%	79.4%
5	80.6%	81.4%
6	81.7%	84.4%
7	84.0%	84.2%
8	84.6%	86.0%
9	86.4%	86.8%
10	86.5%	84.2%
11	82.5%	80.6%
12	80.9%	87.6%
13	87.3%	88.2%
14	88.7%	87.0%
15	87.0%	88.2%
16	88.8%	86.6%
17	90.4%	91.0%
18	90.8%	89.8%
19	90.3%	86.6%
20	90.0%	90.2%

Table 4: The SCNN model performance over training and validation accuracy based on 20 epochs

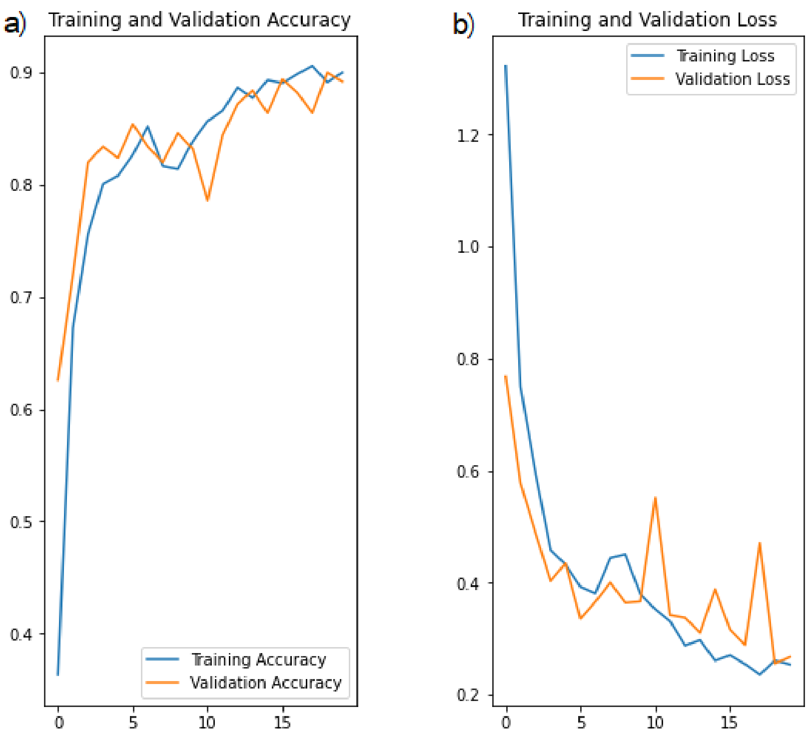

Figure 10: SCNN model training and validation performance: a) accuracy curve b) loss curve

Table 5: The Confusion matrix for the CNN model

	Normal	**Minor**	**Moderate**	**Severe**
Normal	99	1	0	0
Minor	0	89	1	10
Moderate	0	0	94	6
Severe	0	1	22	77

6 Discussion

This work successfully applied, evaluated, and compared three deep learning models. All the constructed models successfully assessed the severity of the degradation of concrete structures. The first two were built using the pre-trained Xception and ResNet-50 architectures, while the third was built as a customized version of the sequential convolutional neural network (SCNN).

The results confirmed the validity of the study's hypothesis. They provided a positive answer to the research question, as all three models successfully predicted and assessed the severity of damage in concrete structures. The constructed models achieved the study's aims as they can all be deployed and embedded in an SHM system to help detect, assess, and monitor the progression of concrete structure degradation.

The SCNN model achieved the best classification accuracy performance of 90.25%, followed by the Xception model, which scored 86.3%, and the ResNet-50 model, which scored 70%. These results are quite acceptable compared to those reported in the literature, particularly when considering the dataset's size and the consistency of the model's performance. Moreover, this study classifies samples into four severity classes, while all the other surveyed studies only classified the samples into two binary classes.

The SCNN model was selected as the champion model as it scored higher than the other two models and has also shown a stable progression across all epochs. The confusion matrix of the CNN model shows a robust performance across all classes. The model successfully predicted the correct classes for 99% of the samples in the normal class, 94% of the samples in the moderate class, and 89% of the minor class. However, the model's classification accuracy dropped to 77% as it missed classifying 22 samples as being in the moderate class while they belonged to the severe class. Nevertheless, after carefully examining the misclassified images, it was noticed that the images were on the borderline between moderate and severe classes, and it was difficult for the human expert to draw a sharp line between them.

On the other hand, the Xception model performed significantly better than the ResNet-50 as it scored a validation accuracy of 86.3% compared to 70%. After analysing the validation and loss curves of the ResNet-50 model, it was noticed that its validation performance is quite fluctuating, and it fails to improve as it moves over the 50 epochs. However, the loss curve shows a tendency towards increasing instead of declining. In addition, the gap between the validation and training accuracy is quite large, affecting the model's robustness when deployed to classify unseen instances. On the other hand, and despite the modest performance of the ResNet-50 model, the gap between its validation and training performance was found narrow. However, despite the fluctuation of the model validation accuracy, the ResNet-50 model performance improved over the 50 epochs.

The dataset size was quite acceptable compared to those used in similar studies. The dataset consisted of 2,000 images sampled from a larger dataset that included 56,000 images. The sample consists of 500 images in each class, representing various crack types, directions, and concrete colors. The resolution and dimension of the images are compatible with images that can be captured by ordinary cameras, unlike some of the studies reported in the literature that used only a few high-resolution pictures that were acquired by a high-tech camera, which may increase the costs and constrain the potential deployment or embedding in an SHM system. On the other hand, some of the related studies that have been investigated in this research use only very small datasets of very high-resolution images [10, 30], which limits the reproducibility of the results and casts doubt on the validity of the results as they depend on generated rather than original images.

Furthermore, all the related work investigated only identified the cracks, classifying the images into two classes: cracked and non-cracked samples. This work moves further by assessing the levels of concrete degradation into three categories: minor, moderate, and severe, in addition to the fourth normal level. This contribution is important for providing continuous and real-time monitoring of concrete health, particularly in strategic structures such as dams, tunnels, bridges, walls, and towers, which help avoid the catastrophic partial or total collapse of these structures, which may endanger lives and cause huge financial damages.

7 Conclusion

In this work, we have successfully created three deep learning models that assess the severity of concrete structure degradation and its severity level using deep learning. The results have demonstrated that the SCNN model achieved an excellent performance in validation classification accuracy with a score of 90.2% based on 20 epochs. In contrast, the Xception and ResNet-50 models scored an accuracy of 86.3% and 70%, respectively, based on 50 epochs.

The models created in this work can be integrated into an SHM system to automate the assessment of concrete cracks and diagnose the degradation of concrete structures based on the crack's visual signs, which would help to stop or create remedies for the concrete degradation and would also help to provide a faster and better response in case of emergency.

Future works that extend this study may involve using more images and innovative algorithms to achieve even better results. It also enhances the results by investigating the influence of image resolution, dimensionality, colouring, shadowing, and other image properties. We also recommend conducting further research that may involve deploying the models as part of an Internet of Things (IoT) system that can collect and process the crack images in real-time. Similar studies can also be conducted to assess cracks in asphalt and road payments as part of road engineering and maintenance.

References

[1] Alpaydin, E., 2021. Machine learning. MIT Press. doi:10.7551/mitpress/13811.001.0001.

[2] Arbaoui, A., Ouahabi, A., Jacques, S., Hamiane, M., 2021. Concrete cracks detection and monitoring using deep learning-based multiresolution analysis. Electronics 10, 1772. doi:10.3390/electronics10151772.

[3] Baccouche, M., Mamalet, F., Wolf, C., Garcia, C., Baskurt, A., . Sequential deep learning for human action recognition, in: International workshop on human behavior understanding, Springer. pp. 29–39. doi:10.1007/978-3-642-25446-8_4.

[4] Ballard, W., 2018. Hands-on Deep Learning for Images with TensorFlow: Build Intelligent Computer Vision Applications Using TensorFlow and Keras. Packt Publishing Ltd.

[5] Bani Mustafa, A.M., 2012. A knowledge discovery and data mining process model for metabolomics. Thesis. Computer Science Department.

[6] BaniMustafa, A., . Predicting software effort estimation using machine learning techniques, in: 2018 8th International Conference on Computer Science and Information Technology (CSIT), IEEE. pp. 249–256. doi:10.1109/csit.2018.8486222.

[7] BaniMustafa, A., 2019. Enhancing learning from imbalanced classes via data preprocessing: A data-driven application in metabolomics data mining. The ISC International Journal of Information Security 11, 79–89. doi:10.22042/isecure.2019.11.0.11.

[8] Banimustafa, A., Hardy, N., 2020. A scientific knowledge discovery and data mining process model for metabolomics. IEEE Access 8, 209964–210005. doi:10.1109/access.2020.3039064.

[9] BaniMustafa, A.H., Hardy, N.W., 2012. A strategy for selecting data mining techniques in metabolomics. Springer. book section 18. pp. 317–333. doi:10.1007/978-1-61779-594-7_18.

[10] Billah, U.H., Tavakkoli, A., La, H.M., . Concrete crack pixel classification using an encoder decoder based deep learning architecture, in: International Symposium on Visual Computing, Springer. pp. 593–604. doi:10.1007/978-3-030-33720-9_46.

[11] Cha, Y.J., Choi, W., Büyüköztürk, O., 2017. Deep learning-based crack damage detection using convolutional neural networks. Computer-Aided Civil and Infrastructure Engineering 32, 361–378. doi:10.1111/mice.12263.

[12] Chollet, F., . Xception: Deep learning with depthwise separable convolutions, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1251–1258. doi:10.1109/cvpr.2017.195.

[13] Deng, J., Dong, W., Socher, R., Li, L.J., Kai, L., Li, F.F., . Imagenet: A large-scale hierarchical image database, in: IEEE Conference on Computer Vision and Pattern Recognition, pp. 248–255. doi:10.1109/CVPR.2009.5206848.

[14] Dorafshan, S., Thomas, R.J., Maguire, M., 2018. Comparison of deep convolutional neural networks and edge detectors for image-based crack detection in concrete. Construction Building Materials 186, 1031–1045. doi:10.1016/j.conbuildmat.2018.08.011.

[15] Dung, C.V., 2019. Autonomous concrete crack detection using deep fully convolutional neural network. Automation in Construction 99, 52–58. doi:10.1016/j.autcon.2018.11.028.

[16] Farrar, C.R., Worden, K., 2007. An introduction to structural health monitoring. Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences 365, 303–315. doi:10.1098/rsta.2006.1928.

[17] Goodfellow, I., Bengio, Y., Courville, A., 2016. Deep learning. MIT press.

[18] He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770–778. doi:10.1109/CVPR.2016.90.

[19] Jin, H., Ling, C.X., 2005. Using auc and accuracy in evaluating learning algorithms. IEEE Transactions on Knowledge and Data Engineering 17, 299–310. doi:10.1109/TKDE.2005.50.

[20] LeCun, Y., Bengio, Y., Hinton, G., 2015. Deep learning. nature 521, 436–444. doi:10.1038/nature14539.

[21] Ling, C.X., Huang, J., Zhang, H., . Auc: A better measure than accuracy in comparing learning algorithms, in: Xiang, Y., Chaib-draa, B. (Eds.), Advances in Artificial Intelligence, Springer Berlin Heidelberg. pp. 329–341. doi:10.1007/3-540-44886-1_25.

[22] Park, R., Paulay, T., 1991. Reinforced concrete structures. John Wiley & Sons.

[23] da Silva, W.R.L., de Lucena, D.S., . Concrete cracks detection based on deep learning image classification, in: Multidisciplinary digital publishing institute proceedings, p. 489. doi:10.3390/icem18-05387.

[24] Simonyan, K., Zisserman, A., 2014. Very deep convolutional networks for large-scale image recognition. arXiv preprint URL: https://arxiv.org/abs/1409.1556.

[25] Speckmann, H., Henrich, R., . Structural health monitoring (shm)–overview on technologies under development, in: Proceedings of the 16th World Conference on NDT, pp. 1–8. URL: https://www.ndt.net/abstract/wcndt2004/563.htm.

[26] Sumitro, S., Hida, K., Diouron, T.L., . Structural health monitoring paradigm for concrete structures, in: 28th Conference on Our World In Concrete & Structures, Singapore, 27-29 August 2003, Conference Documentation Volume XXII, Citeseer. pp. 525–532. URL: https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.502.2512.

[27] Szegedy, C., Ioffe, S., Vanhoucke, V., Alemi, A.A., a. Inception-v4, inception-resnet and the impact of residual connections on learning, in: Thirty-first AAAI conference on artificial intelligence, pp. 1–12. doi:10.1609/aaai.v31i1.11231.

[28] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z., b. Rethinking the inception architecture for computer vision, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 2818–2826. doi:10.1109/cvpr.2016.308.

[29] Yamaguchi, T., Nakamura, S., Saegusa, R., Hashimoto, S., 2008. Image-based crack detection for real concrete surfaces. IEEJ Transactions on Electrical and Electronic Engineering 3, 128–135. doi:10.1002/tee.20244.

[30] Yamane, T., Chun, P.j., 2020. Crack detection from a concrete surface image based on semantic segmentation using deep learning. Journal of Advanced Concrete Technology 18, 493–504. doi:10.3151/jact.18.493.

[31] Yuen, K.V., Kuok, S.C., 2010. Modeling of environmental influence in structural health assessment for reinforced concrete buildings. Earthquake Engineering Engineering Vibration 9, 295–306. doi:10.1007/s11803-010-0014-4.

Cite this paper as:
BaniMustafa, A.; AbdelHalim, R., O. Bulkrock, Al-Hmouz, A. (2023). Deep Learning for Assessing Severity of Concrete Structures Cracks, *International Journal of Computers Communications & Control*, 18(1), 4977, 2023. https://doi.org/10.15837/ijccc.2023.1.4977