

Research Article

Exploring the Detection Accuracy of Concrete Cracks Using Various CNN Models

Mohammed Ameen Mohammed ¹, Zheng Han ¹ and Yange Li ^{1,2}

¹School of Civil Engineering, Central South University, Changsha 410075, China

²Key Laboratory of Engineering Structures of Heavy-Haul Railway, Ministry of Education, Changsha 410075, China

Correspondence should be addressed to Zheng Han; zheng_han@csu.edu.cn

Received 16 March 2021; Revised 18 August 2021; Accepted 1 September 2021; Published 9 September 2021

Academic Editor: Paweł Kłosowski

Copyright © 2021 Mohammed Ameen Mohammed et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Automatic crack detection with the least amount of workforce has become a crucial task in the inspection and evaluation of the performances of concrete structure in civil engineering. Recently, although many concrete crack detection models based on convolutional neural networks (CNNs) have been developed, the accuracy of the proposed models varies. Up-to-date, the issue regarding the convolutional neural network architecture with best performance for detecting concrete cracks is still debated in many previous studies. In this paper, we choose three established open-source CNN models (Model1, Model2, and Model3) which have been well-illustrated and verified in previous studies and test them for the purpose of crack detection of concrete structures. The chosen three models are trained using a concrete crack dataset containing 40,000 images those with 227×227 -pixel in size. The performance of three different convolutional neural network (CNN) models was then evaluated. The comprehensive comparison result indicates that Model2 which used batch normalization is capable of the best performance amongst the three models as selected for concrete cracks detection, with recording the highest classification accuracy and low loss. In a conclusion, we recommend Model2 for a concrete crack detection task.

1. Introduction

A significant number of civil infrastructures have progressively approached their life span expectancy; thus, the integrity of the structural system needs to be checked. At the same time, how to constantly and automatically check the structure with even less workforce has become a vital research path with the aging of the population and the rise in labor costs [1–4].

Visually observed cracks are often a concern for inspection engineers because cracks not only provide access to dangerous and corrosive chemicals within the concrete but often result in water and deicing salts infiltrate concrete that damage the integrity of structures [5, 6]. For instance, more than 100,000 bridges in the United States have early age deck cracks [7]. Besides, among the 570,000 bridges in the USA, 40% were listed as deficient, requiring repair or reconstruction according to the

requirements of the Federal Highway Administration (FHWA), with an expected cost of 50 billion dollars [8, 9]. More than 510,000 new bridges have been constructed in China over the last 40 years, 15% of which have reached their service life earlier due to cracks. Because cracks will effectively function as a significant predictor in assessing structural damage, crack identification would have elevated functional implementation values in activating early bridge repair alerts, safety assurance, and loss mitigation [10].

Human-based visual inspection is a familiar method for inspecting and evaluating the health of concrete structures. However, the human-based visual inspection and evaluation are time-consuming and subjective [11]. The accuracy of damage diagnosis depends mainly on the skill level and experience of the inspectors. Therefore, automatic damage detection is a crucial task for achieving objectivity and efficiency of damage assessment [12].

Digital technologies have developed computer models to process images automatically gathered from structures to detect structural damage and cracks. In this context, many previous studies have developed models for defect detection, and most of these models have focused on the detection of cracks in civil structures such as techniques based on image processing, clustering, edge detection, filter-based models and edge detection, and image filtering-based machine learning (ML) [13–25]. Most of these remarkable works used computer vision-based methods to improve the performance of crack detection. Computer vision-based methods used the image process techniques (IPTs) to extract crack detection and evaluate the extracted features. However, the results by this kind of method have been influenced by the extraction of the false feature using IPTs because the performance of these methods depends on the extracted crack features [26, 27]. To overcome these challenges, a convolutional neural network (CNN) is used to detect concrete crack without the extracting process of crack features [28, 29]. Yu et al. [30] proposed a deep convolutional neural network (DCNN) capable of automatically extracting high-level features from low-level features and optimally selecting the combination of extracted features to satisfy any damage recognition objective.

Recently, remarkable progress has been achieved in crack detection procedures using convolutional neural networks (CNNs) [26, 29, 31–35]. For example, research studies regarding the application of CNNs to concrete crack detection include, using a four-layer CNN [36], a deep fully convolutional network (FCN) [29], a fully convolutional neural network (FCN) named Ci-Net [37], and a deep convolutional neural network (DCNN) architecture with eight layers [38]. The accuracy of these proposed models is, respectively, 87%, 90%, 93.6%, and 98%. Li et al. [39] proposed an approach for detecting safety helmets on construction sites in real-time using a convolutional neural network-based SSD-MobileNet algorithm. The recorded training model precision was 95%.

Researchers have also proposed a crack detection method using pretrained CNN models such as VGG16, AlexNet, Resnet50, and InceptionV3 from scratch for classification [40–43]. Deep CNN pretrained on the original datasets (e.g., ImageNet) has proven beneficial for several computer vision problems, including new problems involving different classes [36].

Han et al. [44] proposed a solution that integrates Monte Carlo simulations into a local thresholding processing method to address the fundamental problems of traditional methods in terms of complex background, spatially varying illumination, and block size uncertainties. Kang et al. [45] proposed an automated crack detection, localization, and quantification approach that detects crack regions by integrating a faster region proposal convolutional neural network (Faster R-CNN) algorithm. The Faster R-CNN-based crack damage detection obtained results with an average precision of 95%. Choi and Cha designed the SDDNet semantic segmentation network to detect cracks, with a processing speed of 36 FPS (frames per second) for images with a resolution of 1025×512 pixels [46].

The models of CNN-based crack detection have the outstanding advantage that it avoids multifarious work from features pre-extraction and computation compared with traditional methods. In addition, CNN does not need to convert the format of the input images but learns the crack features from the image automatically, thus reducing the workload of crack detection [47]. Moreover, CNN is particularly powerful for detecting thin cracks under illumination conditions that are difficult to detect when using conventional methods [29]. Recently, although many concrete crack detection models based on convolutional neural networks (CNNs) have been developed, the accuracy of the proposed models varies. Up-to-date, the issue regarding the convolutional neural network architecture with best performance for detecting concrete cracks is still debated in many previous studies.

In this paper, we choose three established open-source CNN models (Model1, Model2, and Model3) which have been well-illustrated and verified in previous studies and test them for the purpose of crack detection of concrete structures. The chosen three models are trained using a concrete crack dataset containing 40,000 images those with 227×227 -pixel in size. The performance of three different convolutional neural network (CNN) models was then evaluated. Comprehensive comparison result indicates that Model2 is capable of best performance for concrete cracks detection task.

This paper is divided into six sections. Section 1 provides a brief review of research on concrete crack detection, while Section 2 describes the construction of the dataset. Similarly, Section 3 introduces the research methodology while Section 4 concerns the presentation and comparison of the experimental results. Section 5 discusses the result of the study. Finally, Section 6 presents the conclusions of the study.

2. Methodology

The convolutional neural network (CNN) is a type of deep learning method for image classification and recognition tasks. The CNN architecture is comprised of multilayers neural network, including an input layer, convolutional layer, activation function, fully connected layers, and output layers. Besides, batch normalization (BN) and dropout layers which are provided according to the purposes of use are additional layers of CNN.

In this paper, a CNN is built by modifying three models taken from Kaggle. Those models have been trained based on image data of natural scenes around the world, and the output number of their image classes is 6, whereas the output number of image classes in this paper is 2 (cracked and noncracked).

An attempt was made in order to modify and adjust the models' structures thereby making an additional dropout layer to the first and third models to avoid overfitting. Also, the number of epochs for all the three models was changed from 30, 30, and 50 to a unified number of 10 epochs for all and hence yielded robust models which allow for obtaining better results in terms of automatic crack detection.

The CNN architectures of the three models are shown in Figures 1–3. The input images are 3-channel RGB images, with a size of 227×227 pixels. The detailed operations and dimensions of each layer for the three models are shown in Tables 1–3.

2.1. Data Preprocessing. OpenCV2 to load images from a file is used in Model1. At the same time, the Image-DataGenerator class provided by TensorFlow in Keras is used in both the Model2 and Model3. It reads the image from the disk and preprocesses it into the appropriate tensor. It helps prevent overfitting and helps the model to generalize better.

2.2. Convolution Layer. The filter size was 3×3 and used the ReLU activation function of each model. ReLU activation function is superior to tanh and sigmoid functions in almost all applications [51]. Besides, ReLU is used as an activation function in neural networks because the convergence rate of ReLU is faster during training than networks using sigmoid functions.

2.3. Pooling Layer. The main function of the pooling layer is to perform downsampling, progressively reduces the size of the image, preserves the distinctive features of the image, and be useful for dimensionality reduction role. A pooling layer is usually added after the convolutional layer to decrease the amount of parameters, filter redundant information, and avoid overfitting when training the network. In this study, pool sizes of 4×4 , 2×2 , 2×2 , for Model1, Model2, and Model3, are used, respectively.

2.4. Padding. Padding is one of the parameters that control the spatial size of the output volume. It is useful when the output size is required to be the same as the input size. In Model2, padding = "SAME" because the output and input sizes are the same, and the padding type is called SAME.

2.5. Dropout. Dropout is used to avoid the issue of overfitting in the neural networks and reducing the co-adaptation between neurons to generate more effective training examples. Dropout layers were used in the three models. Moreover, we have added the dropout layer after the third layer of the convolutional in Model1 and the dropout layer after the second layer of the convolutional in Model3.

2.6. Batch Normalization (BN). Batch normalization (BN) can improve neural network speed and efficiency in addition to creating more reliable models [29]. Batch normalization can also reduce the epoch number to train a neural network. Batch normalization, when applied to network architectures, avoids overfitting and achieves greatly improved classification accuracy in fewer epochs. In Model2, batch normalization (BN) is used before each pooling layer. The mathematical expression of batch normalization is as follows:

$$\begin{aligned}\mu_B &= \frac{1}{m} \sum_{i=1}^m x_i, \\ \sigma_B^2 &= \frac{1}{m} \sum_{i=1}^m (x_i - \mu_B)^2, \\ \hat{x}_i &= \frac{x_i - \mu_B}{\sqrt{\sigma_{B+\epsilon}^2}}, \\ y_i &= \gamma \hat{x}_i + \beta,\end{aligned}\tag{1}$$

where μ_B and σ_B^2 are the mean and variance of the current training batch, respectively; x_i represents the normalized output after subtracting the mean and variance of the current training batch; and y_i represents the result of translation and scaling, where parameters that need to be learned are the scaling factor γ and the translation coefficient β .

2.7. Hyperparameters. An Adam optimizer was used in the three models. The dropout rate at the dropout layer located before the ReLU is 0.5. The models are trained for 10 epochs with a batch size of 64. Model2 sets a lower bound on the learning rate on $1e-8$.

3. Concrete Crack Image Dataset

This paper used a publicly available concrete crack images dataset [52]. The dataset of concrete crack images is split into cracked and noncracked images for image classification. The dataset contains a total of 40,000 images with 227×227 pixels and an RGB channel, which contains 20,000 positive crack images and 20,000 negative images of concrete structures. The datasets were generated from 458 high-resolution images (4032×3024 pixel) with the method proposed by Zhang et al. [36]. The preparation of these datasets is as follows. Training dataset: it used 70% of images randomly selected from the base dataset. The dataset for training consists of 28 K images. Validation dataset: it used 10% of images randomly selected from the base dataset. The dataset for validation consists of 4 K images. Throughout the training, the validation dataset is used to monitor the network's learning curve. Testing dataset: it used 20% of images that are randomly chosen from the base dataset. The testing dataset consists of 8 K images. Some images of the dataset are shown in Figure 4.

4. Results

In this study, the performances of three CNN models have been compared using the same dataset for concrete crack detection. The models were trained for 10 epochs on Google Colab, which is a cloud computing platform for machine learning where accuracy and the categorical cross-entropy loss function are used to measure the performance of classification models. Figure 5 shows the accuracy of the training and validation of the models. Model2 achieved superior performance, as shown in Table 4. The recorded training accuracies of the models were 93.23%, 99.85%, and

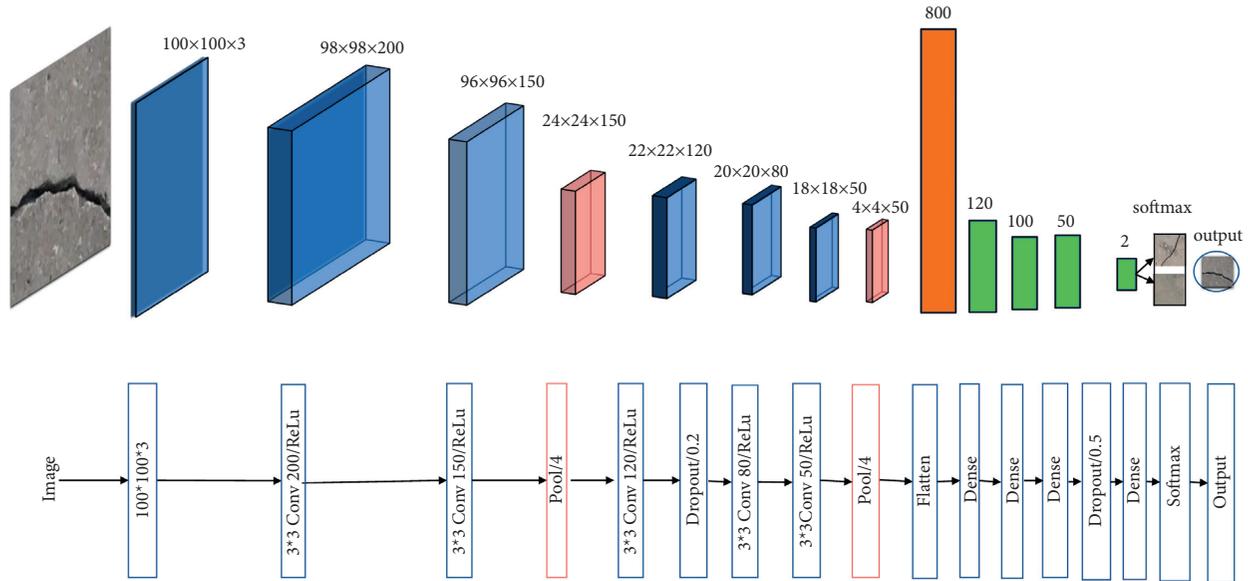


FIGURE 1: The architecture of Model1.

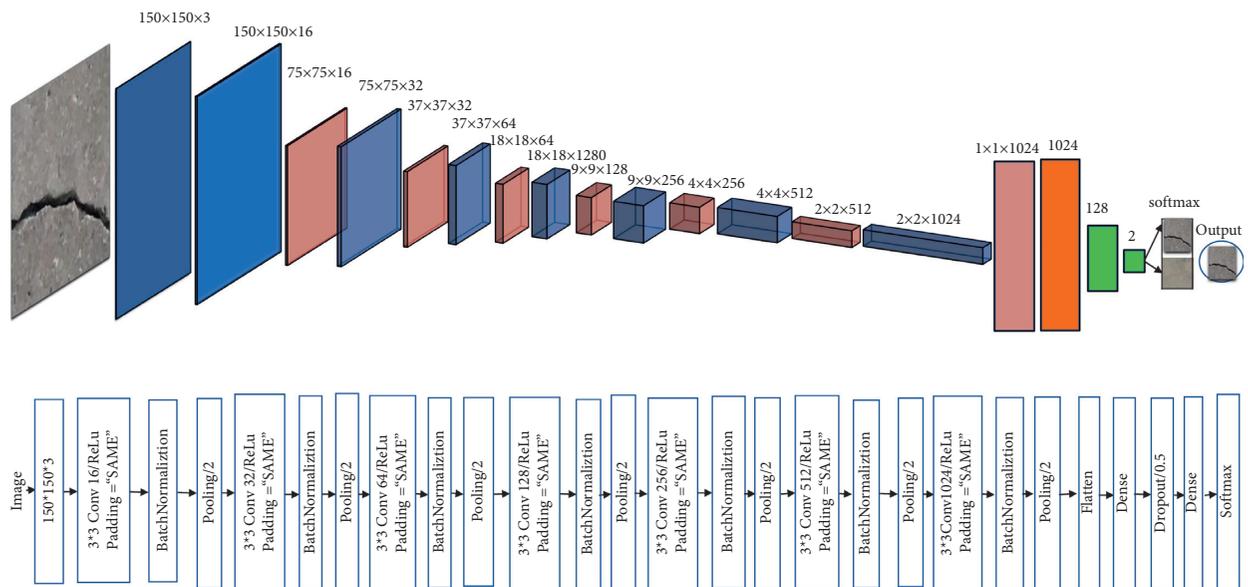


FIGURE 2: The architecture of Model2.

99.44% for Model1, Model2, and Model3, respectively. Similarly, the recorded testing accuracies of the models were 98.04%, 99.89%, and 99.69% for Model1, Model2, and Model3, respectively. Moreover, another metric used to evaluate network performance is the low loss value. The loss function is used to optimize the values of the parameters in the neural network architecture. Whereas the categorical cross-entropy loss function is used for measuring the performance of a classification model with probability values between 0 and 1. Therefore, the low loss values obtained using the categorical cross-entropy approach indicate that the models were not affected by the overfitting problems. For instance, Model1 has a loss value of 18.45% and 7.16% for

training and testing, respectively. Furthermore, the loss value of Model3 has a lower value than that of Model1 as the loss values were 2.12% and 1.34% for training and testing, respectively. At the same time, Model2 achieves a lower loss than the other two models, with values of 0.42% and 0.38% for training and testing, respectively. Comparison of training time of models is shown in Table 5. The incorrectly predicted images were verified and printed a total error in the batch selected. Model2 prediction was best where there was no error prediction. Model3 came next best, where there were few false predictions. Model1 was incorrectly predicted for more wrong prediction. Some images of the wrong prediction are shown in Figure 6.

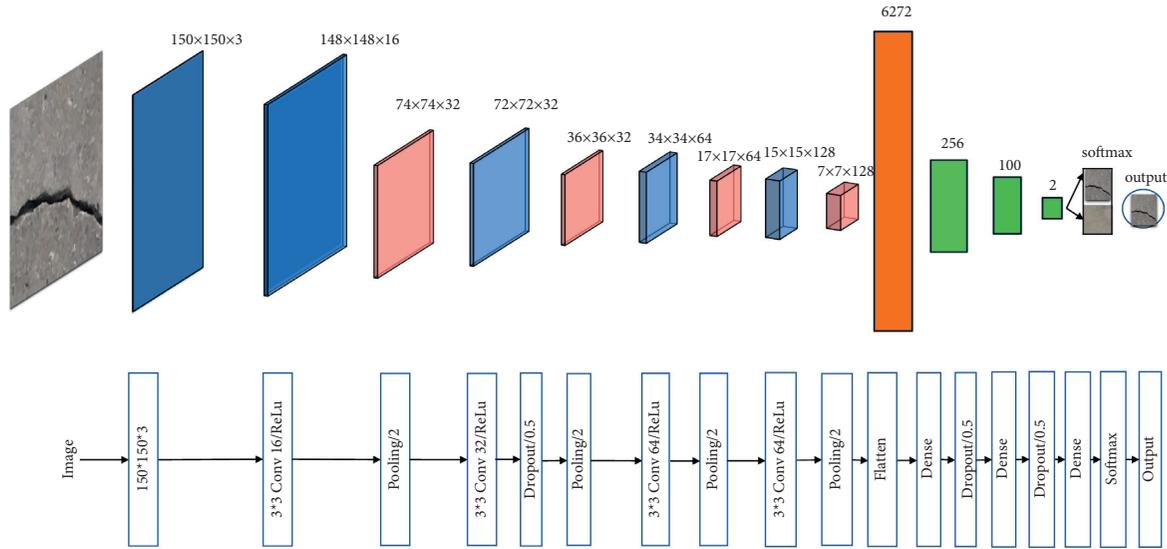


FIGURE 3: The architecture of Model3.

TABLE 1: Layer dimensions and operations.

Model	Layer	Height	Width
Model1 [48]	Conv1	98	98
	Conv2	98	98
	Pool1	24	24
	Conv3	22	22
	Dropout1	22	22
	Conv4	20	20
	Conv5	18	18
	Pool 2	4	4
	Flatten	1	1
	Dense 1	1	1
	Dense 2	1	1
	Dense 3	1	1
	Dropout2	1	1
	Dense 4	1	1
	Softmax	1	1

TABLE 2: Layer dimensions and operations.

Model	Layer	Height	Width
Model2 [49]	Conv1	150	150
	BN	150	150
	Pool1	75	75
	Conv2	75	75
	BN	75	75
	Pool 2	37	37
	Conv3	37	37
	BN	37	37
	Pool 3	18	18
	Conv4	18	18
	BN	18	18
	Pool 4	9	9
	Conv5	9	9
	BN	9	9
Pool 5	4	4	
Conv6	4	4	
BN	4	4	
Pool 6	2	2	
Conv7	2	2	
BN	2	2	
Pool 7	1	1	
Flatten	1	1	
Dense 1	1	1	
Dropout	1	1	
Dense 2	1	1	
Softmax	1	1	

5. Discussion

5.1. *The Effect of the Total Number of Parameters and Network Complexity in terms of the Number of Convolution Network.* It can be seen that Model1 has more layers than Model3 and fewer total parameters than the other models, but it starts from a deep convolutional network with decreasing depth, while Model3 started from a shallow convolutional network and gets progressively deeper. Model2 has more layers and more of the number of parameters than the other models and also starts from a shallow convolutional network and gets deeper. Similarly, the last four convolutional layers of the network in Model2 are deeper than the other models. Although the total number of parameters in Model2 is higher than the other models, the computation time required to train 28 K data sets per period for all networks is close to that of Model3. It seems that increasing the number of convolutional layers started from a shallow convolutional network and gets progressively deeper, and more of the number of parameters helps to

improve the performance of the network in the crack detection task compared with the other models. This proves that the CNN structure of Model2 showed good performance compared with the other two. The test results of this study are shown in Table 2. The accuracy was 99.85% and 99.89 for training and testing, respectively. At the same time, the loss values were lower than the other two models with 0.42% and 0.38% loss for training and testing, respectively. Furthermore, Model2 prediction was best where there was

TABLE 3: Layer dimensions and operations.

Model	Layer	Height	Width
Model3 [50]	Conv1	148	148
	Pool1	74	74
	Conv2	72	72
	Pool 2	36	36
	Dropout1	36	36
	Conv3	34	34
	Pool 3	17	17
	Conv4	15	15
	Pool 4	7	7
	Flatten	1	1
	Dense 1	1	1
	Dropout2	1	1
	Dense 1	1	1
	Dropout2	1	1
	Dense 2	1	1
	Softmax	1	1

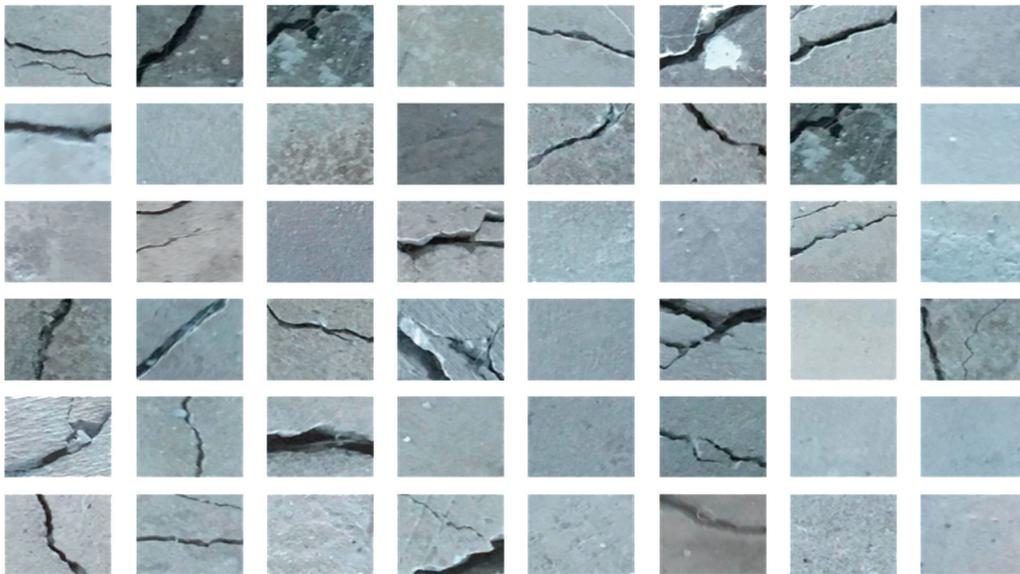


FIGURE 4: Image dataset [51].

no error; some images of the prediction are shown in Figure 7.

5.2. The Effect of Batch Normalization (BN) on the Training Speed and Accuracy in Concrete Crack Detection Models. The results show that the type of network used has an impact on the speed and performance of the model, as well as on the accuracy obtained in training and testing. Although the total training parameters in Model2 are the largest compared with the other models, using batch normalization (BN) improved the performance and speed of the neural network, as well as producing more reliable models [53]. Batch normalization can also reduce the epochs number to train a neural network. Batch normalization, when applied to network architectures, avoids overfitting and can achieve a significant improvement in classification accuracy in fewer epochs than other neural

network models that do not use batch normalization. The second model achieves more than 99% training accuracy in the second epoch, while the first model achieves more than 99% training accuracy after eight epochs, and the third model achieves more than 99% training accuracy after the fifth epoch. Hence, BN has performed well in improving the accuracy of CNN network models for detecting concrete cracks [4, 9, 54].

5.3. Limitations. However, this study has some limitations. For example, the study is applied only to the most commonly used dataset for training and testing of computer vision-based crack detection [51]. Similarly, physical parameters, such as depth, width, and length of the concrete crack and also damage size, and minimum and maximum crack width, were not studied. In this study, three CNN models were

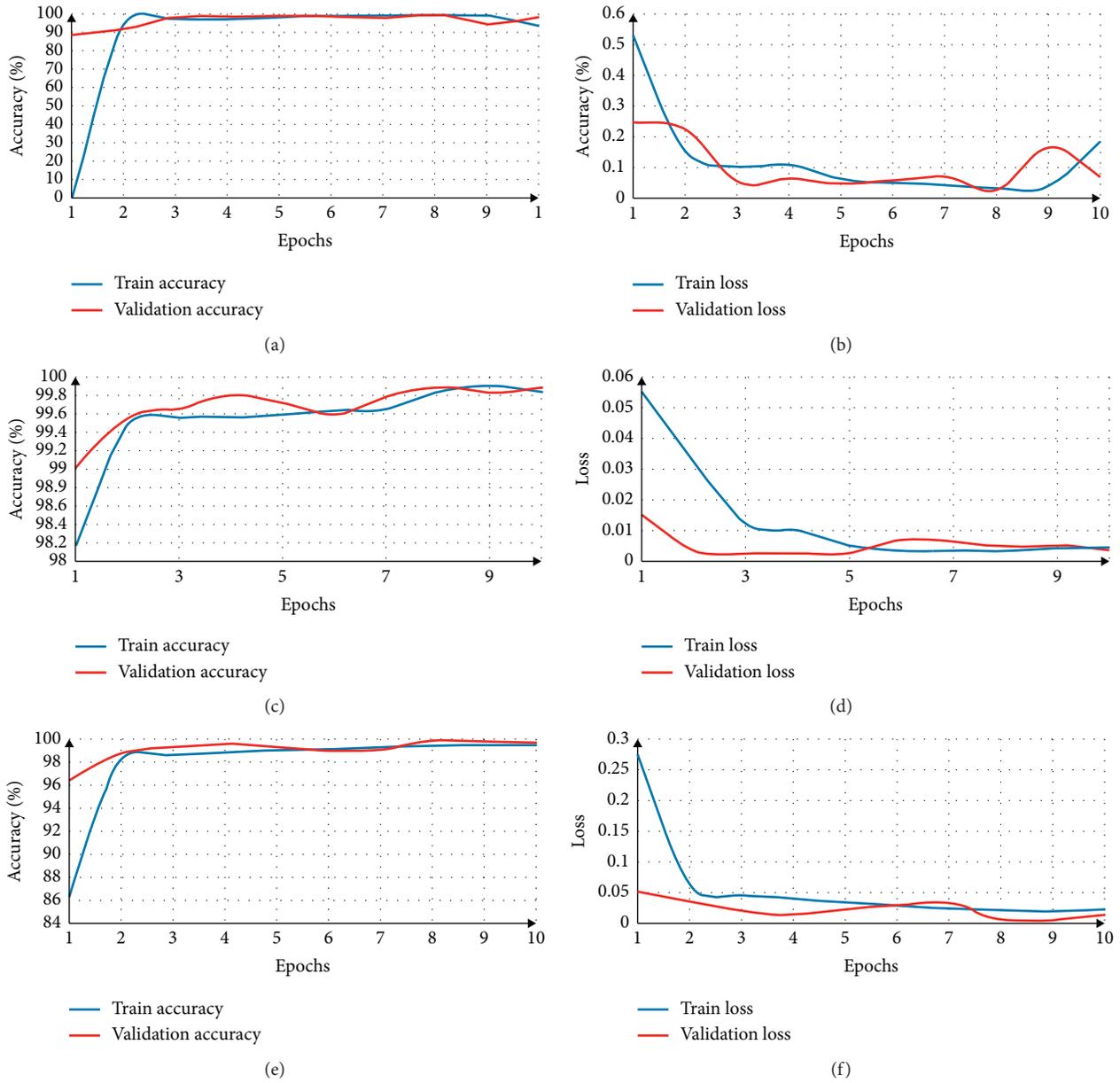


FIGURE 5: Accuracy and loss values of the three models: (a) Model1 accuracy; (b) Model1 loss; (c) Model2 accuracy; (d) Model2 loss; (e) Model3 accuracy; (f) Model3 loss.

TABLE 4: The performance of models for 10 epochs.

Model	Training accuracy (%)	Testing accuracy (%)	Training loss	Testing loss
Model1	93.23	98.04	0.1845	0.0716
Model2	99.85	99.89	0.0042	0.0038
Model3	99.44	99.69	0.0212	0.0134

TABLE 5: Comparison of training time of models.

Model	Epochs	Running time
Model1	10	14 m 30 s
Model2	10	26 m 54 s
Model3	10	29 m 47 s

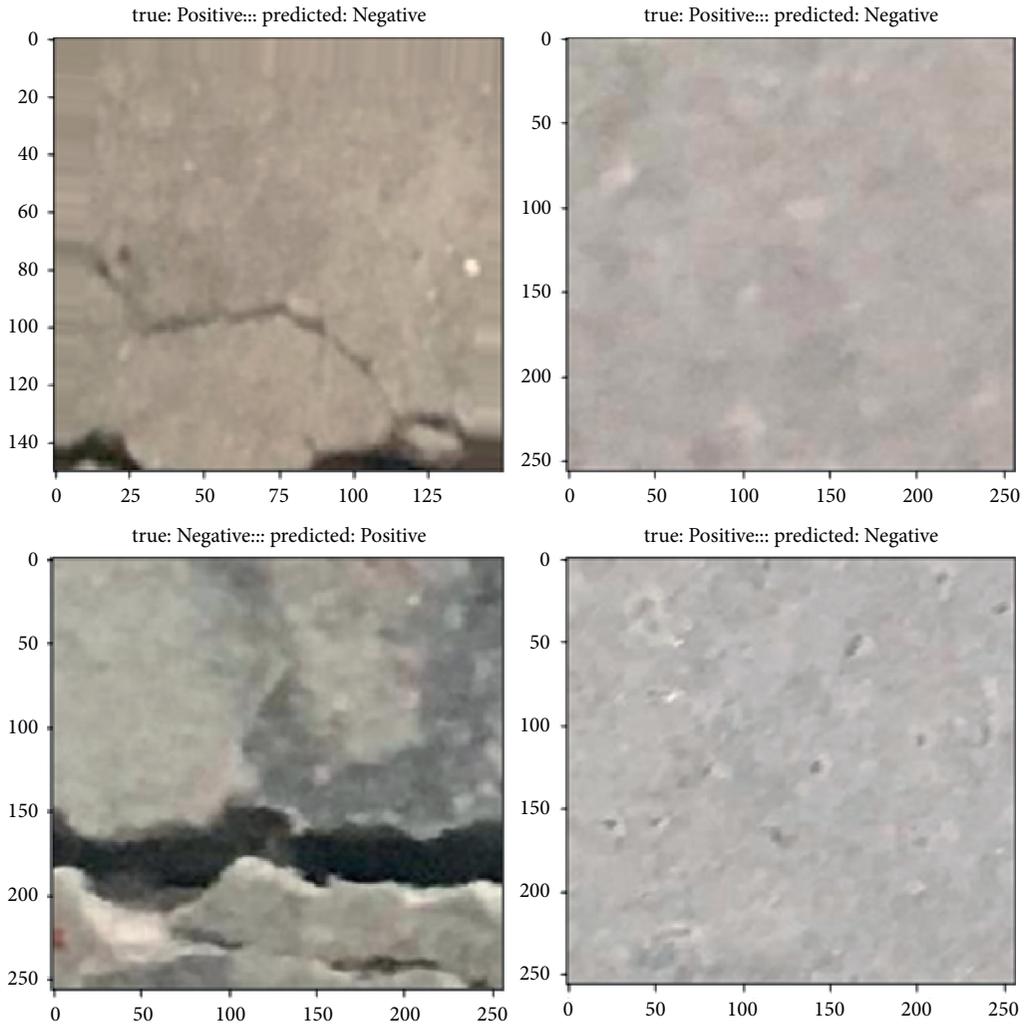


FIGURE 6: Wrong prediction of Model1 and Model2.

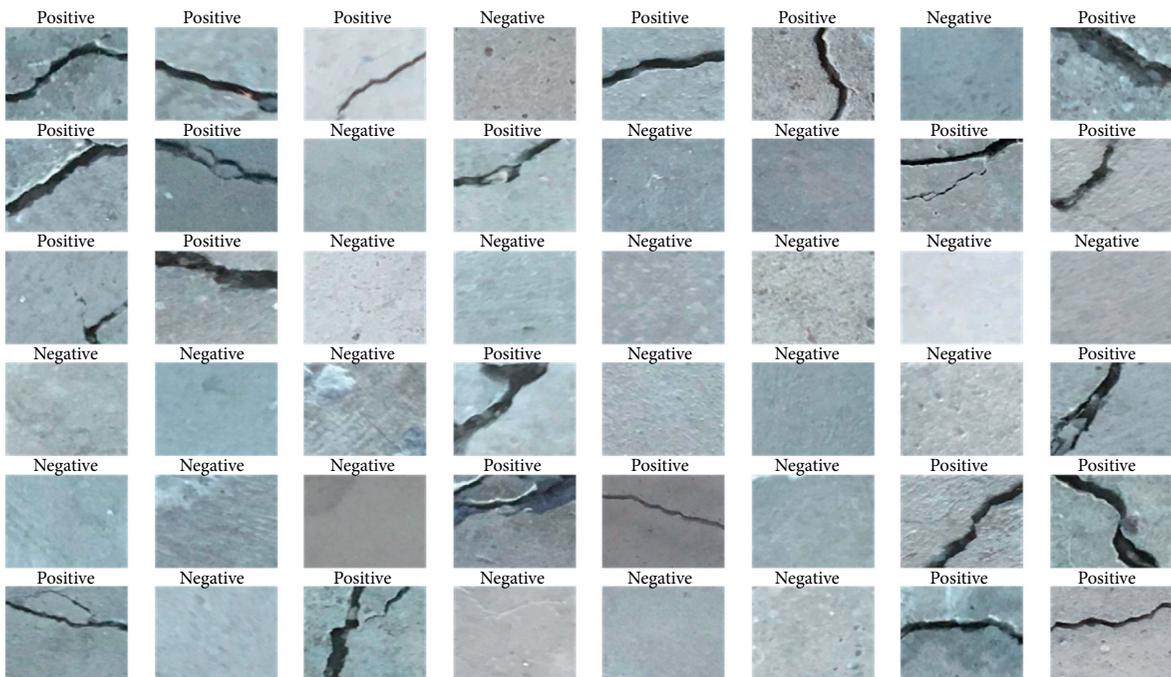


FIGURE 7: Validation results.

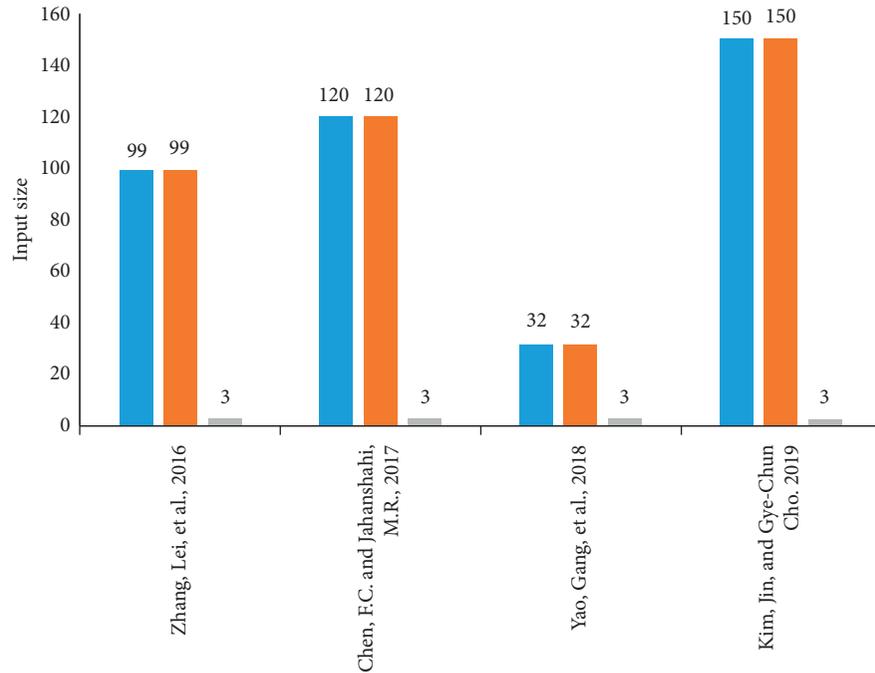


FIGURE 8: The input size of some studies to monitor infrastructures [56–59].

compared using approximately average input sizes. Mishkina et al. [55] showed that using 128×128 pixel images is sufficient to make qualitative conclusions about optimal network structure and is an order of magnitude faster than using standard 224-pixel images. Because the accuracy depends on image size linearly, the needed computations grow quadratically, so it is a very difficult way to perform again. The input size of some studies to monitor infrastructures is shown in Figure 8.

6. Conclusions

This study compared the performance of three CNN models on the same concrete crack dataset, which is a database of 40,000 images with a resolution of 227×227 pixels. The models were trained on Google Colab, which is a cloud computing platform for machine learning. The dataset is split into training, validation, and testing in a 70:10:20 ratio. We used open-source models, which have been trained on other image data, and then used them on the concrete crack image domain for comparative analysis. Based on the results of the study, the following conclusions can be drawn:

- (1) For preprocessing, ImageDataGenerator class provided by TensorFlow in Keras was used in Model2 and Model3 and showed good performance. It reads the image from the disk and preprocesses it into suitable tensors. It helps prevent overfitting and helps the model to generalize better.
- (2) Based on comparative analysis, Model2 achieved superior performance; the recorded accuracy was 99.85% and 99.89% for training and testing, respectively.

- (3) Based on comparative analysis, Model2 achieves a lower loss than the other two models, with loss values of 0.42% and 0.38% for training and testing, respectively.
- (4) It is better to use ReLU activation, padding, batch normalization, and pooling layers for the neural network architecture in concrete crack detection.
- (5) The number of neural network layers in Model2 was 7 convolutional layers, started from a shallow convolutional network and getting deeper. Similarly, the last 4 convolutional layers of the network are deeper than the convolutional layers in the other models; this indicates that increasing the number of convolutional layers and going deeper gradually help to improve the performance of the network in the crack detection task.

In future studies, we will focus on improving the accuracy of models to detect various physical parameters such as depth, width, length, corrosion, spalling, and voids in concrete structures. However, the development of deep learning-based algorithms for detecting physical parameters of concrete cracks and autonomous crack size estimation is still problematic in these studies, mainly when the test image contains many noisy crack-like features. Therefore, future research should focus on improving the proposed methods to make autonomous crack density estimations.

Data Availability

The Concrete Crack Images data supporting this study are from previously reported studies and datasets, which have been cited. The processed data are available at <https://data.mendeley.com/datasets/5y9wdsg2zt/2>.

Conflicts of Interest

The authors declare that they have no conflicts of interest regarding the publication of this paper.

Acknowledgments

This study was financially supported by the National Key R&D Program of China (grant no. 2018YFD1100401); the National Natural Science Foundation of China (grant no. 52078493); the Natural Science Foundation for Excellent Young Scholars of Hunan (grant no. 2021JJ20057); and the Innovation Driven Program of Central South University (grant no. 2019CX011). These financial supports are gratefully acknowledged.

References

- [1] I. Abdel-Qader, S. Pashaie-Rad, O. Abudayyeh, and S. Yehia, "PCA-based algorithm for unsupervised bridge crack detection," *Advances in Engineering Software*, vol. 37, no. 12, pp. 771–778, 2006.
- [2] O. Abudayyeh, M. A. Bataineh, and I. A. Qader, "An imaging data model for concrete bridge inspection," *Advances in Engineering Software*, vol. 35, no. 8, pp. 473–480, 2004.
- [3] Z. Liu, Y. Cao, Y. Wang, and W. Wang, "Computer vision-based concrete crack detection using u-net fully convolutional networks," *Automation in Construction*, vol. 104, pp. 129–139, 2019.
- [4] Y. S. Yang, C. M. Yang, and C. W. Huang, "Thin crack observation in a reinforced concrete bridge pier test using image processing and analysis," *Advances in Engineering Software*, vol. 83, pp. 99–108, 2015.
- [5] P. D. Krauss and E. A. Rogalla, "Transverse cracking in newly constructed bridge decks," 1996.
- [6] R. S. Adhikari, O. Moselhi, and A. Bagchi, "Image-based retrieval of concrete crack properties for bridge inspection," *Automation in Construction*, vol. 39, pp. 180–194, 2014.
- [7] P. Prasanna, K. J. Dana, N. Gucunski et al., "Automated crack detection on concrete bridges," *IEEE Transactions on Automation Science and Engineering*, vol. 13, no. 2, pp. 591–599, 2016.
- [8] A. S. Nowak, *Bridge Evaluation, Repair And Rehabilitation*, Springer Science & Business Media, New York, NY, USA, 2012.
- [9] M. Flah, A. R. Suleiman, and M. L. Nehdi, "Classification and quantification of cracks in concrete structures using deep learning image-based techniques," *Cement and Concrete Composites*, vol. 114, Article ID 103781, 2020.
- [10] Y. Pan, G. Zhang, and L. Zhang, "A spatial-channel hierarchical deep learning network for pixel-level automated crack detection," *Automation in Construction*, vol. 119, Article ID 103357, 2020.
- [11] S. Li and X. Zhao, "Automatic crack detection and measurement of concrete structure using convolutional encoder-decoder network," *IEEE Access*, vol. 8, pp. 602–618, 2020.
- [12] S. Li, X. Zhao, and G. Zhou, "Automatic pixel-level multiple damage detection of concrete structure using fully convolutional network," *Computer-Aided Civil and Infrastructure Engineering*, vol. 34, no. 7, pp. 616–634, 2019.
- [13] I. Abdel-Qader, O. Abudayyeh, and M. E. Kelly, "Analysis of edge-detection techniques for crack identification in bridges," *Journal of Computing in Civil Engineering*, vol. 17, no. 4, pp. 255–263, 2003.
- [14] K. C. P. Wang, Q. Li, and W. Gong, "Wavelet-based pavement distress image edge detection with \hat{A} trous algorithm," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2024, no. 1, pp. 73–81, 2007.
- [15] H. Oliveira and P. L. Correia, "Automatic road crack segmentation using entropy and image dynamic thresholding," in *Proceedings of the 2009 17th European Signal Processing Conference*, pp. 622–626, Glasgow, Scotland, August 2009.
- [16] Y.-C. Tsai, V. Kaul, and R. M. Mersereau, "Critical assessment of pavement distress segmentation methods," *Journal of Transportation Engineering*, vol. 136, no. 1, pp. 11–19, 2010.
- [17] L. Ying and E. Salari, "Beamlet transform-based technique for pavement crack detection and classification," *Computer-Aided Civil and Infrastructure Engineering*, vol. 25, no. 8, pp. 572–580, 2010.
- [18] H. Oliveira and P. L. Correia, "Automatic road crack detection and characterization," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 1, pp. 155–168, 2013.
- [19] Q. Zou, Y. Cao, Q. Li, Q. Mao, and S. Wang, "CrackTree: automatic crack detection from pavement images," *Pattern Recognition Letters*, vol. 33, no. 3, pp. 227–238, 2012.
- [20] D. Lattanzi and G. R. Miller, "Robust automated concrete damage detection algorithms for field applications," *Journal of Computing in Civil Engineering*, vol. 28, no. 2, pp. 253–262, 2014.
- [21] E. Zalama, J. G. G. Bermejo, R. Medina, and J. Llamas, "Road crack detection using visual features extracted by gabor filters," *Computer-Aided Civil and Infrastructure Engineering*, vol. 29, no. 5, pp. 342–358, 2014.
- [22] C. Koch, K. Georgieva, V. Kasireddy, B. Akinci, and P. Fieguth, "A review on computer vision based defect detection and condition assessment of concrete and asphalt civil infrastructure," *Advanced Engineering Informatics*, vol. 29, no. 2, pp. 196–210, 2015.
- [23] C. M. Yeum and S. J. Dyke, "Vision-based automated crack detection for bridge inspection," *Computer-Aided Civil and Infrastructure Engineering*, vol. 30, no. 10, pp. 759–770, 2015.
- [24] H. Kim, E. Ahn, M. Shin, and S. H. Sim, "Crack and noncrack classification from concrete images using machine learning," *Structural Control and Monitoring*, vol. 18, 2018.
- [25] A. Mohan and S. Poobal, "Crack detection using image processing: a critical review and analysis," *Alexandria Engineering Journal*, vol. 57, no. 2, pp. 787–798, 2018.
- [26] Y. Fujita, Y. Mitani, and Y. Hamamoto, "A method for crack detection on a concrete structure," in *Proceedings of the 18th International Conference on Pattern Recognition (ICPR'06)*, vol. 3, pp. 901–904, Hong Kong, China, August 2006.
- [27] T. Nishikawa, J. Yoshida, T. Sugiyama, and Y. Fujino, "Concrete crack detection by multiple sequential image filtering," *Computer-Aided Civil and Infrastructure Engineering*, vol. 27, no. 1, pp. 29–47, 2012.
- [28] E. Protopapadakis, K. Makantasis, G. Kopsiaftis, N. Doulamis, and A. Amditis, "Crack identification via user feedback, convolutional neural networks and laser scanners for tunnel infrastructures," in *Proceedings of the Special Session on RGB and Spectral Imaging for Civil/Survey Engineering, Cultural, Environmental, Industrial Applications*, pp. 725–734, Rome, Italy, January 2016.
- [29] Y.-J. Cha, W. Choi, and O. Büyüköztürk, "Deep learning-based crack damage detection using convolutional neural networks," *Computer-Aided Civil and Infrastructure Engineering*, vol. 32, no. 5, pp. 361–378, 2017.

- [30] Y. Yu, C. Wang, X. Gu, and J. Li, "A novel deep learning-based method for damage identification of smart building structures," *Structural Health Monitoring*, vol. 18, no. 1, pp. 143–163, 2019.
- [31] Y.-J. Cha and W. Choi, "Vision-based concrete crack detection using a convolutional neural network," in *Proceedings of the Society for Experimental Mechanics Series*, pp. 71–73, Springer, Indianapolis, IN, USA, June 2017.
- [32] Y.-J. Cha, W. Choi, G. Suh, S. Mahmoudkhani, and O. Büyüköztürk, "Autonomous structural visual inspection using region-based deep learning for detecting multiple damage types," *Computer-Aided Civil and Infrastructure Engineering*, vol. 33, no. 9, pp. 731–747, 2018.
- [33] W. R. L. d. Silva and D. S. d. Lucena, "Concrete cracks detection based on deep learning image classification," *Proceedings*, vol. 2, no. 8, 2018.
- [34] K. Zhang, H. D. Cheng, and B. Zhang, "Unified approach to pavement crack and sealed crack detection using pre-classification based on transfer learning," *Journal of Computing in Civil Engineering*, vol. 32, no. 2, Article ID 4018001, 2018.
- [35] M. Alipour, D. K. Harris, and G. R. Miller, "Robust pixel-level crack detection using deep fully convolutional neural networks," *Journal of Computing in Civil Engineering*, vol. 33, no. 6, Article ID 4019040, 2019.
- [36] L. Zhang, F. Yang, Y. Daniel Zhang, and Y. J. Zhu, "Road crack detection using deep convolutional neural network," in *Proceedings of the 2016 IEEE International Conference on Image Processing (ICIP)*, pp. 3708–3712, Phoenix, AR, USA, September 2016.
- [37] C. V. Dung and L. D. Anh, "Autonomous concrete crack detection using deep fully convolutional neural network," *Automation in Construction*, vol. 99, pp. 52–58, 2019.
- [38] Y. Xu, S. Wei, Y. Bao, and H. Li, "Automatic seismic damage identification of reinforced concrete columns from images by a region-based deep convolutional neural network," *Structural Control and Health Monitoring*, vol. 26, no. 3, Article ID e2313, 2019.
- [39] Y. Li, H. Wei, Z. Han, J. Huang, and W. Wang, "Deep learning-based safety helmet detection in engineering management based on convolutional neural networks," *Advances in Civil Engineering*, vol. 2020, Article ID 9703560, 10 pages, 2020.
- [40] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems*, vol. 25, pp. 1097–1105, 2012.
- [41] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, <https://arxiv.org/abs/1409.1556>.
- [42] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, Las Vegas, NV, USA, June 2016.
- [43] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2818–2826, Las Vegas, NV, USA, June 2016.
- [44] Z. Han, B. Su, Y.-g. Li, Y.-f. Ma, W.-d. Wang, and G.-q. Chen, "An enhanced image binarization method incorporating with monte-carlo simulation," *Journal of Central South University*, vol. 26, no. 6, pp. 1661–1671, 2019.
- [45] D. Kang, S. S. Benipal, D. L. Gopal, and Y.-J. Cha, "Hybrid pixel-level concrete crack segmentation and quantification across complex backgrounds using deep learning," *Automation in Construction*, vol. 118, Article ID 103291, 2020.
- [46] W. Choi, Y. J. Cha, SDDNet: real-time crack segmentation," *IEEE Transactions on Industrial Electronics*, vol. 67, no. 9, pp. 8016–8025, 2019.
- [47] S. Li and X. Zhao, "Image-based concrete crack detection using convolutional neural network and exhaustive search technique," *Advances in Civil Engineering*, vol. 2019, Article ID 6520620, 12 pages, 2019.
- [48] H. Asem, "Image classification by CNN," 2019, <https://www.kaggle.com/heshamasem/image-classification-by-cnn>.
- [49] S. Gupta, "Data classification," 2019, <https://www.kaggle.com/samg2003/90-3-accuracy-seg-test-intel-dataclassification/notebook>.
- [50] vijay chauhan, "Image classification->90% intel image classification," 2019, <https://www.kaggle.com/vchauhanusf/notebooks>.
- [51] R. H. R. Hahnloser, H. S. Seung, and J.-J. Slotine, "Permitted and forbidden sets in symmetric threshold-linear networks," *Neural Computation*, vol. 15, no. 3, pp. 621–638, 2003.
- [52] Ç. F. Özgenel, "Concrete crack images for classification," *Mendeley Data*, vol. 1, 2018.
- [53] S. Ioffe and C. Szegedy, "Batch normalization: accelerating deep network training by reducing internal covariate shift," in *Proceedings of the 32nd International Conference on Machine Learning, Ser. Proceedings of Machine Learning Research*, vol. 37, pp. 448–456, Lille, France, July 2015.
- [54] K. Chen, A. Yadav, A. Khan, Y. Meng, and K. Zhu, "Improved crack detection and recognition based on convolutional neural network," *Modelling and Simulation in Engineering*, vol. 2019, Article ID 8796743, 8 pages, 2019.
- [55] D. Mishkina, N. Sergievskiy, and J. Matasa, "Systematic evaluation of CNN advances on the imagenet," *Center for Machine Perception, Faculty of Electrical Engineering*, vol. 53, 2016.
- [56] M. Eisenbach, R. Stricker, D. Seichter et al., "How to get pavement distress detection ready for deep learning? a systematic approach," in *Proceedings of the 2017 International Joint Conference on Neural Networks (IJCNN)*, pp. 2039–2047, IEEE, Anchorage, AL, USA, May 2017.
- [57] F. C. Chen and M. R. Jahanshahi, "Nb-CNN: Deep learning-based crack detection using convolutional neural network and Naïve Bayes data fusion," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 5, pp. 4392–4400, 2017.
- [58] G. Yao, F. J. Wei, J. Y. Qian, and Z. G. Wu, "Crack detection of concrete surface based on newline convolutional neural networks," in *Proceedings of the 2018 International Conference on Machine Learning and Cybernetics (ICMLC)*, pp. 246–250, IEEE, Chengdu, China, 2018, July.
- [59] J. Kim and G. C. Cho, "Image-based crack detecting using convolution neural network (CNN) with less data," in *Proceedings of the ASEM19. International Association of Structural Engineering and Mechanics*, Jeju Island, Korea, September 2019.