Journal of Integrated and Advanced Engineering (JIAE)
Vol. 2, No. 2, September 2022: 123-134
http://asasijournal.id/index.php/jiae
http://doi.org/10.51662/jiae.v2i2.82

Detection of road cracks using Convolutional Neural Networks and Threshold Segmentation

Arselan Ashraf[1], Ali Sophian[2*], Amir Akramin Shafie[3], Teddy Surya Gunawan[1], Norfarah Nadia Ismail[3], Ali Aryo Bawono[4]

[1]Department of Electrical and Computer Engineering, International Islamic University Malaysia, Malaysia
[2]Department of Mechatronics Engineering, International Islamic University Malaysia, Malaysia
[3]School of Civil Engineering, College of Engineering, Universiti Teknologi MARA, Malaysia
[4]Faculty of Rail, Transport, and Logistics, Technical University of Munich Asia, Singapore

Abstract
Automatic road crack detection is a vital transportation maintenance responsibility for ensuring driving comfort and safety. However, manual inspection is considered risky because it is time-consuming, costly, and dangerous for inspectors. Automated road crack detecting techniques have been extensively researched and developed in order to overcome this issue. Despite the difficulties, most proposed methodologies and solutions involve machine vision and machine learning, which have recently acquired traction largely due to the increasingly more affordable processing power. Nonetheless, it remains a difficult task due to the inhomogeneity of crack intensity and the intricacy of the background. In this paper, a convolutional neural network-based method for crack detection is proposed. Recent advancements inspire the method of machine learning to computer vision. The primary goal of this work is to employ convolutional neural networks to detect road cracks. Data in the form of images has been used as input, preprocessing, and threshold segmentation are applied to the input data. The processed output is fed to CNN for feature extraction and classification. The training accuracy was found to be 96.20 %, the validation accuracy to be 96.50 %, and the testing accuracy to be 94.5 %.

Keywords:
Crack Detection;
Computer Vision;
Convolutional Neural Networks;
Machine Learning.

Article History:
Received: August 22, 2022
Revised: September 16, 2022
Accepted: September 18, 2022
Published: September 25, 2022

Corresponding Author:
Ali Sophian,
Department of Mechatronics Engineering, International Islamic University Malaysia, Malaysia
Email: ali_sophian@iium.edu.my

INTRODUCTION

Pavement performance can be negatively impacted by various factors [1]. The road deteriorates from elements like the sun and rain over time. Human mistakes, moving vehicles, poor pavement materials, poor construction, and negligent maintenance are also major contributors to pavement deterioration. Damaged roads that are not promptly repaired can lower road lifespan, degrade road quality, and possibly lead to accidents [2]. Existing methods of road crack detection rely mostly on manual inspection, which can be time-consuming and labor-intensive, disruptive to traffic flow, dangerous for inspectors, and prone to human error. As a roadway expands rapidly, it becomes increasingly challenging to meet the detection criteria of such a massive infrastructure project, and current methods fall far short of the mark [3].

Therefore, crack detection must be automated to replace manual defect inspection methods for rapid, effective, and reliable damage assessment. Crack detection has recently been automated using several testing methods, including laser, infrared, thermal, radiographic, and thermal testing methodologies [4, 5, 6]. Lately, there has been a rise in image-based approaches for crack detection. Images of the concerned portion are taken and then analyzed algorithmically for cracks. This approach is quick, cheap, and reliable. It is possible to classify the techniques as either image processing or machine learning. Filters, morphological analysis, statistical approaches, and percolation techniques are used in the image processing methods for crack detection [7][8], and no model training process is necessary. However, with machine learning, a dataset of images is gathered and fed into the machine learning model of choice during the training phase. While preprocessing and noise removal in image processing may be required for such approaches, the actual detection of cracks is left to a trained learning algorithm [9].

An image processing-based technique for crack detection is depicted in Figure 1. The first step is to take pictures of the intended part using a camera or other imaging device. After that, filters, segmentation, and other methods are used to clean up the photos by removing unwanted details like noise and blemishes. Converting the image to grayscale or binary form may be necessary depending on the approach to detect cracks. The resulting image is fed into the crack detection algorithm, which employs several image processing methods (such as edge detection) to isolate the damaged area of the picture. Such evaluations are useful in determining the extent of a crack. Figure 2 shows the fundamental procedures to construct an ML model for crack detection. It is necessary to initially collect a dataset including examples of surface cracks for the machine learning model to analyze. To improve the quality, the images undergo a preprocessing phase in which image processing techniques are used to eliminate shadows, crop the images, and enhance brightness and contrast. Next, the photos are labelled, with the cracks being annotated. A manual or labelling tool can be used to carry out this procedure. Next, a machine learning model for crack detection should be chosen. Support vector machines (SVM), convolutional neural networks (CNN), and decision trees are some of the machine learning models that have been employed in past studies for crack detection [10]. Next, researchers formulate an optimization function to reduce the loss or training cost [11, 12, 13, 14]. The dataset contains a collection of annotated images that will be used to train the specified model. After the model has been trained, it will be applied to a new batch of images to determine how well it can distinguish between different types of cracks.

Most of the earliest approaches [15, 16, 17] relied on threshold processing algorithms that assumed the crack pixel was darker than its neighbors. The threshold segmentation approach was frequently utilized in early image segmentation techniques because of its ease of use and speed. Early researchers proposed a number of threshold-based automatic detection methods from multiple perspectives. For example, automatic crack detection was proposed by [16], who suggested using a threshold technique based on the neighboring difference histogram. Experimental findings are improved above those obtained with the standard threshold method because this method maximizes the difference between crack and non crack pixels.

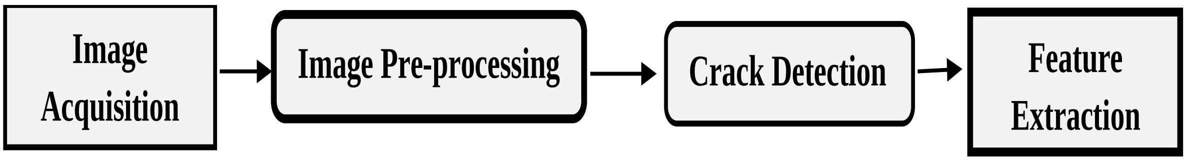


Figure 1. Image processing-based technique for crack detection

Figure 2. ML-based technique for crack detection

In order to detect cracks in the pavement, [17] used a linked domain technique (directed segmentation expansion algorithm) to produce a binary image, which was then processed. The threshold segmentation approach is typically used in conjunction with other algorithms to improve segmentation accuracy due to its reliance on gray-level features and its susceptibility to noise. According to [18], they discovered that false positives could be caused by the presence of non crack features in the image. It is hypothesized that the non crack image's crack area can be obtained by averaging the pixels' grey levels along the linear object's inner and outer boundaries. First, the image is split into many overlapping sub image regions by [19]. Sub images were then segmented, and the cracks within them fused using the neighborhood difference histogram. The final step is to extract the picture region containing the crack. This approach works well for a limited number of complex cracks but fails across a wide range of complex backgrounds.

In order to detect cracks, the CNN model is often used. This model uses a convolutional layer, a pooling layer, and a fully connected layer. The convolutional layer learns to differentiate between crack and non-crack images by extracting characteristics from them. Adjusting the image's dimensions on the pooling layer allows for down sampling, which reduces the file size. The final level of a CNN model is a fully connected layer, which receives the previous layer's output as input and maps it to an output label. A crack net is a convolutional neural network-based automatic pavement detecting system proposed by [20]. This technique has successfully implemented automatic detection of pavement cracks at the pixel level. The crack net served as the basis for a deep network suggested by [21] for automatic pixel-level crack detection in a 3D image of an asphalt road. This network was given the name crack net-v. The crack net-v improved the accuracy and efficiency of the calculations due to its more complex structure and fewer parameters. To facilitate supervised learning at the pixel level, Crack net-v utilized a uniform space size across all layers. In order to automate the detection of cracks, the researchers in [22] created a trainable deep convolution neural network deep crack, which was able to learn the more complex aspects of crack representation. Features learned at various scales by successive convolution layers are combined to create a linear representation. Using this technique, they obtained image features with better representation characteristics in both large and small-scale feature maps.

Using convolutional neural networks, [23] developed an identifier that can automatically detect cracks in images of solid surfaces. However, the detection rate was low for the discolored concrete. Cracks and other auxiliary damages such as voids, spalling, and cement corrosion have been identified from images using CNN classifiers that are less impacted by noise introduced by illumination, shadow, and projection [24]. In [25], they used deeper networks to detect pavement cracks, demonstrating the promise of deep learning. One of the difficulties in the literature is determining which holes are actual cracks and which are simply sealed over. Crack identification on road images with complex textures was the research subject of [26]. Specifically, they sought a solution to the difficulty of telling the difference between cracks and sealed cracks of the same breadth and brightness. A convolutional neural network (CNN) model is trained to distinguish between cracks, sealed cracks, and backgrounds in pavement images [27]. To perform pixel-based segmentation of cracks and sealed cracks, a thresholding procedure is used for the final image. The crack or sealed crack is extracted using a curve identification method based on tensor voting. A total of 800 photos are used in the system’s testing procedure. The system performed admirably with a recall of 0.951 and a precision of 0.847.

Inspired from machine learning techniques especially convolutional neural networks, this research has been conducted to detect pavement cracks using the CNN algorithm. The paper is organized as follows. In Section 2, the proposed methodology is presented. After that, in Section 3, the experimental results are shown. Finally, this paper is concluded with a discussion presented in Section 4.

METHOD

Figure 3 depicts the architecture of the proposed road crack detecting model. Using the image data, road crack detection is executed. The employed training algorithm is convolutional neural networks (CNN). The input images undergo preprocessing, and the output from this stage is fed into the CNN.

Data Preparation

The dataset used for this study is Concrete Crack Images for Classification [28]. This dataset includes pictures of cracked concrete. The data is gathered from METU Campus. In order to facilitate image classification, the dataset is split into negative and positive crack images. Images are 227 x 227 pixels in size, with RGB channels for a total of 40000 images. There is a lack of consistency in high-resolution photographs concerning surface polish and lighting. In other words, no data enhancements such as random flipping or rotation are used.

For this study, we selected 1500 images each from a positive and negative folder for the model training. Therefore, a total of 3000 images are used for training. For validation, the number of images selected is 600, out of which 300 are from a positive folder and 300 from a negative folder. Finally, 400 images are chosen for testing, with 200 from a positive and 200 from a negative folder. Hence, the total number of images selected for this study is 4000. Figure 4 and Figure 5 show the sample of images from a positive and negative folder.

Threshold Segmentation

Image segmentation is a technique used in computer vision and image processing to partition a digital image into smaller parts called segments, regions, or objects (sets of pixels). The purpose of segmentation is to transform an image's representation into one that is simpler, more informative, and more accessible for analysis. Objects and boundaries (lines, curves, etc.) can be detected in an image with the help of image segmentation. When separating parts of an image, thresholding is the quickest and most straightforward technique.

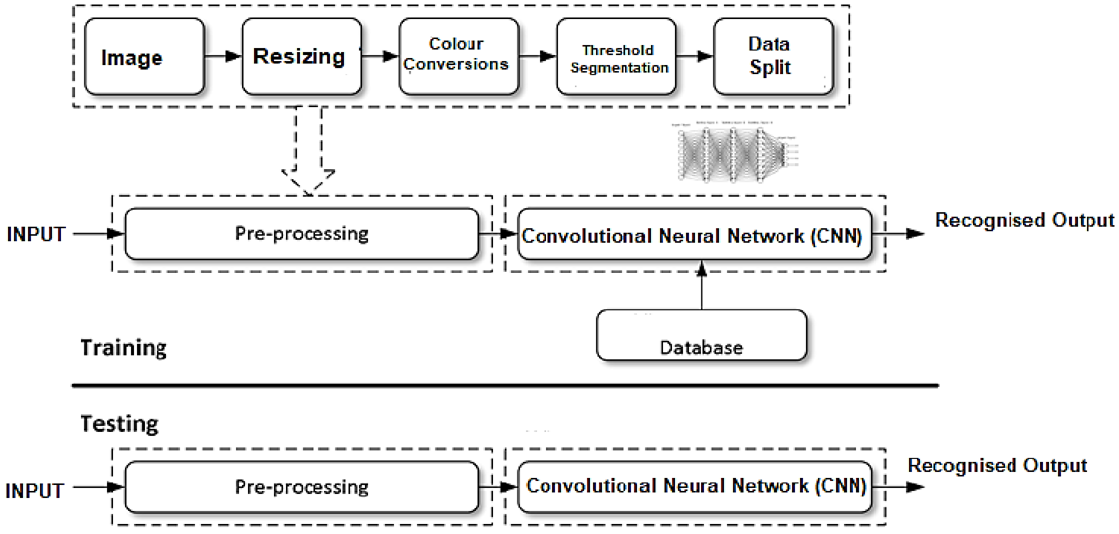


Figure 3. Proposed Workflow Diagram

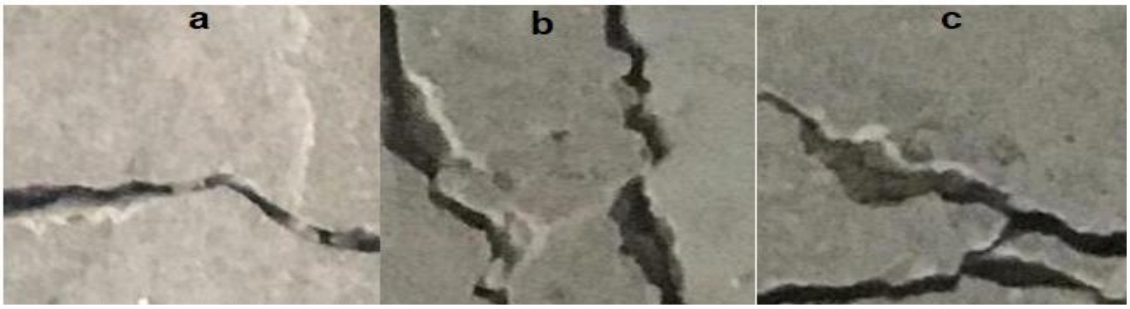


Figure 4. Image samples from positive folder

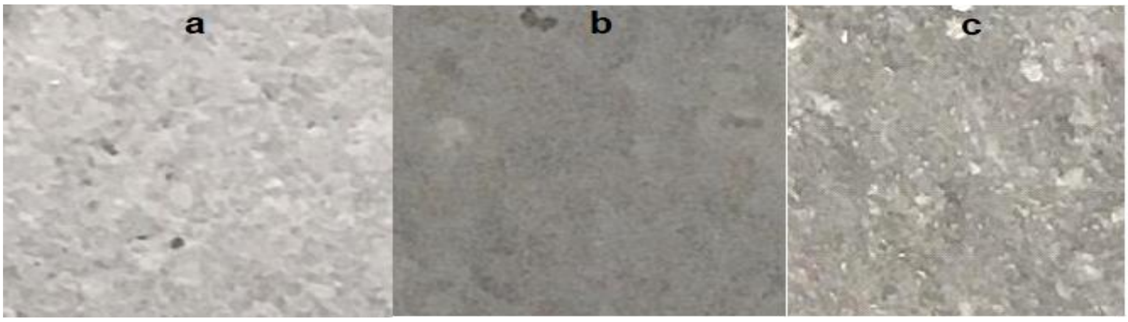


Figure 5. Image samples from negative folder

This non-linear procedure takes a grayscale image and assigns one of two levels to each pixel based on whether or not it is below or above the threshold value. To rephrase, if the pixel value is larger than some threshold, it will be given one value (perhaps white). Otherwise, it will be given another value (maybe black).

This study uses image thresholding over cracked and non-cracked road images. Thresholding separates an object from its background by labelling each pixel as either an object points or a background point based on its intensity relative to a predetermined threshold value. Figure 6 shows the original image and the threshold segmented image.

In general, $T = T[x, y, p(x, y), f(x, y)]$.

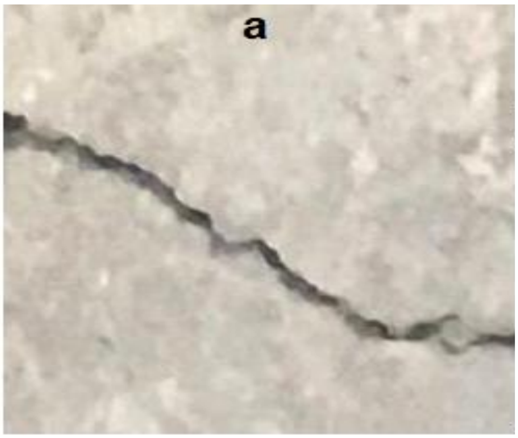
If T is just a function of $f(x, y)$, then global thresholding applies.

If T depends on both the global features $f(x, y)$ and the local features $p(x, y)$, then we have local thresholding.

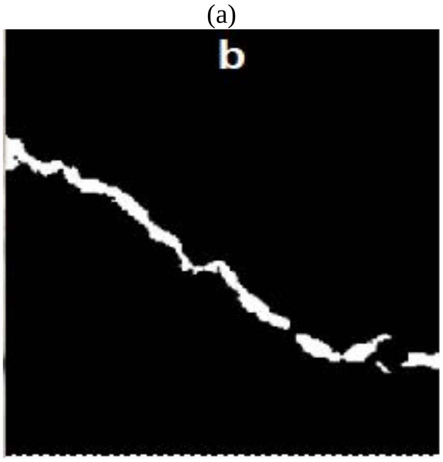
If T changes as a function of coordinates (x, y), we call this dynamic or adaptive thresholding.

CNN Architecture

Convolutional neural networks are a type of Deep Learning algorithm that can take in an input image, extract features (learnable weights and biases), and then classify the objects inside the image. When compared to other classification methods, CNN requires significantly less pre-processing time. While filters in primitive methods are usually hand-engineered, CNNs can be trained to learn the appropriate filters and other properties. Temporal and spatial dependencies in an image can be captured by CNN using appropriate filters. The architecture better fits the image dataset using fewer parameters and reusing weights. The CNN architecture used in this study is shown in Figure 7.



(a)



(b)

Figure 6. Original Image Vs Thresholded Image

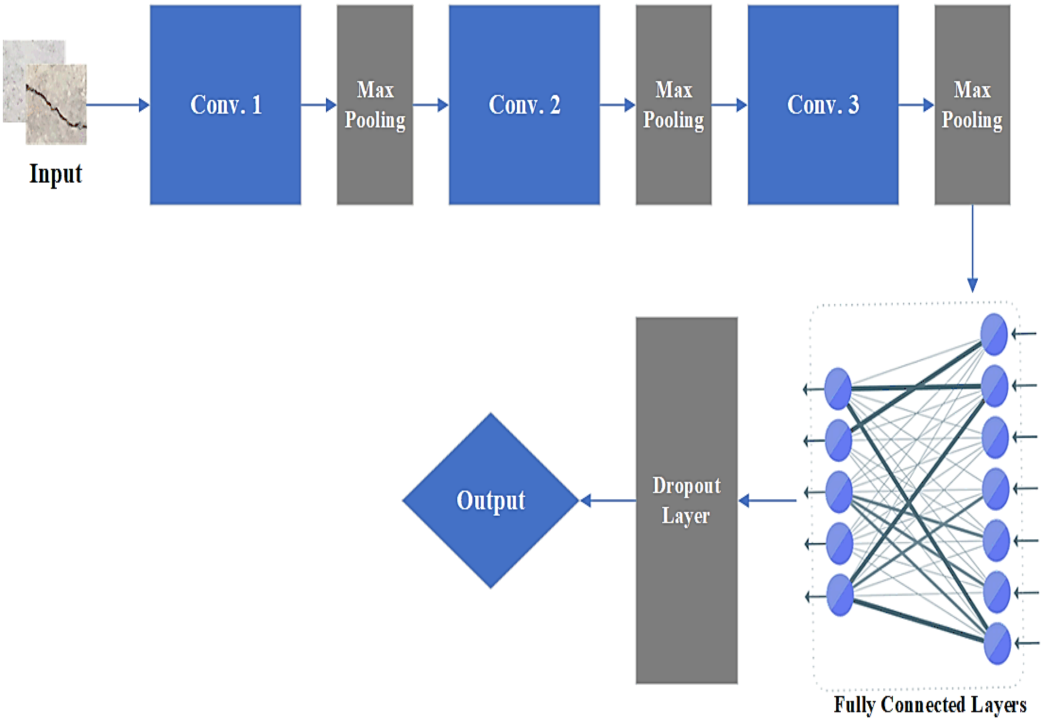


Figure 7. CNN Architecture

The network's input is a face image extracted from the input dataset. Three convolutional layers follow, each with polling applied. Filter sizes 3×16, 3×32, and 3×64 make up the three layers, respectively. In order to give the model more depth and the ability to identify more complex features, second and third convolution layers have been added. A max-pooling layer is of size 2×2. Pooling layers consolidate the features learned by convolutional layers in CNNs.

In order to limit the number of parameters and computation required in the network, it gradually reduces the spatial size of the representation. A batch normalization layer follows the convolution layer, and then two fully connected layers have 348 and 348 neurons, respectively. Each mini-batch is normalized by batch normalization, a method for training neural networks. This settles the learning process and significantly reduces the number of training epochs needed to create neural networks. The significance of using such a configuration allowed this research to train the model at a smaller number of epochs per iteration. A dropout layer follows two fully connected layers with a dropout ratio of 50% to overcome overfitting. If the dropout layer is not there, the first batch of training data disproportionately greatly influences the learning process compared to subsequent batches. This, in turn, would prevent the learning of traits that only exist in later samples or batches from occurring.

CNN Training

The CNN is trained using input images of road cracks by adjusting several hyperparameters, including momentum, learning rate, dropout, batch size, and epochs. The hyperparameter data that was employed for the training of the CNN model presented in this study can be found in Table 1.

RESULTS AND DISCUSSION

Table 2 presents the hardware components used for this project. All the calculations and computations, including training and testing of the network, were performed on the laptop using Jupyter as IDE with Python language.

Crack Image Labelling

Labeling of crack images is an absolute necessity if the accurate classification of cracks is to be accomplished. The labelling has been determined based on whether or not there is a crack present. Each image has been given a label, which can be seen below in Table 3, which was utilized for classification.

Experimental Outcomes

Accuracy has been the primary metric by which we have evaluated the system's effectiveness. Seventy percent of the dataset is used to train the network, while the remaining 30 percent is used for testing. The batch size is set at 150 iterations, and the evaluation runs for 30 epochs. The classification outcomes are dependent on whether or not a crack is present. Table 4 presents the training and validation performance parameters.

The validation accuracy of this model came out to be 96.50%, which indicates that the predictions of this model are promising. In addition, the accuracy of the test data was encouraging, at 94.5%. The elapsed time for the model was also less, which contributed to its swiftness. Figure 8 shows the accuracy of test data. Also, Figure 9 represents the graphical visualization of performance parameters.

Table 1. CNN Training Options

No.	Hyperparameters	Assigned Value
1	Momentum	0.5
2	Learning Rate	0.01
3	Dropout	0.5
4	Batch Size	150
5	Epochs	30

Table 2. Hardware Specifications

Computer	HP PAVILION 15-BC408TX
CPU	Intel Core i7-8750H (8th Gen)
RAM	8 GB DDR4 RAM
HDD	1TB
GPU	NVIDIA GeForce GTX 1050
Graphics Memory	4GB

Table 3. Data Labels

Type	Training Files	Labels
No Crack	1500	0
Crack	1500	1

Table 4. Training and Validation parameters

No.	Epoch	Elapsed Time	Accuracy	Loss	Validation Loss	Validation Accuracy
1	1/30	11s 4ms	0.8750	0.7662	0.5864	0.9633
2	2/30	8s 3ms	0.9420	0.8692	0.7664	0.7567
3	3/30	8s 3ms	0.9090	0.9478	0.5822	0.9553
4	4/30	8s 3ms	0.9673	0.5207	0.5054	0.9653
5	5/30	8s 3ms	0.9410	0.8936	0.7868	0.9463
6	6/30	8s 3ms	0.9420	0.9162	0.7144	0.9557
7	7/30	8s 3ms	0.9637	0.5642	0.5320	0.9657
8	8/30	8s 3ms	0.9667	0.5257	0.5728	0.9663
9	9/30	8s 3ms	0.9530	0.7312	0.8739	0.9447
10	10/30	8s 3ms	0.9643	0.5566	0.5586	0.9550
11	11/30	8s 3ms	0.9633	0.5746	0.5589	0.9560
12	12/30	8s 3ms	0.9613	0.6008	0.8028	0.9653
13	13/30	8s 3ms	0.9603	0.5300	0.5722	0.9663
14	14/30	8s 3ms	0.9680	0.6122	0.6878	0.9440
15	15/30	8s 3ms	0.9630	0.4734	0.5532	0.9450
16	16/30	8s 3ms	0.9660	0.6131	0.5320	0.9467
17	17/30	8s 3ms	0.9667	0.5089	0.5054	0.9653
18	18/30	8s 3ms	0.9647	0.5771	0.5054	0.9553
19	19/30	8s 3ms	0.9667	0.5361	0.5058	0.9543
20	20/30	8s 3ms	0.9657	0.5246	0.6578	0.9653
21	21/30	8s 3ms	0.9670	0.5563	0.5586	0.9760
22	22/30	8s 3ms	0.9557	0.5236	0.5995	0.9557
23	23/30	8s 3ms	0.9640	0.5412	0.5633	0.9253
24	24/30	8s 3ms	0.9436	0.5265	0.5583	0.9660
25	25/30	8s 3ms	0.9654	0.6895	0.5195	0.9447
26	26/30	8s 3ms	0.9654	0.5660	0.8362	0.967
27	27/30	8s 3ms	0.9640	0.5277	0.5252	0.9667
28	28/30	8s 3ms	0.9670	0.5627	0.5347	0.9650
29	29/30	8s 3ms	0.9647	0.6039	0.5434	0.9650
30	30/30	8s 3ms	0.9620	0.6036	0.5592	0.9650

```
400/400 [==============================] - 1s 3ms/step
Final Accuracy :  94.5 %
Final Loss :  0.8803455146919461
```

Figure 8. Overall test accuracy

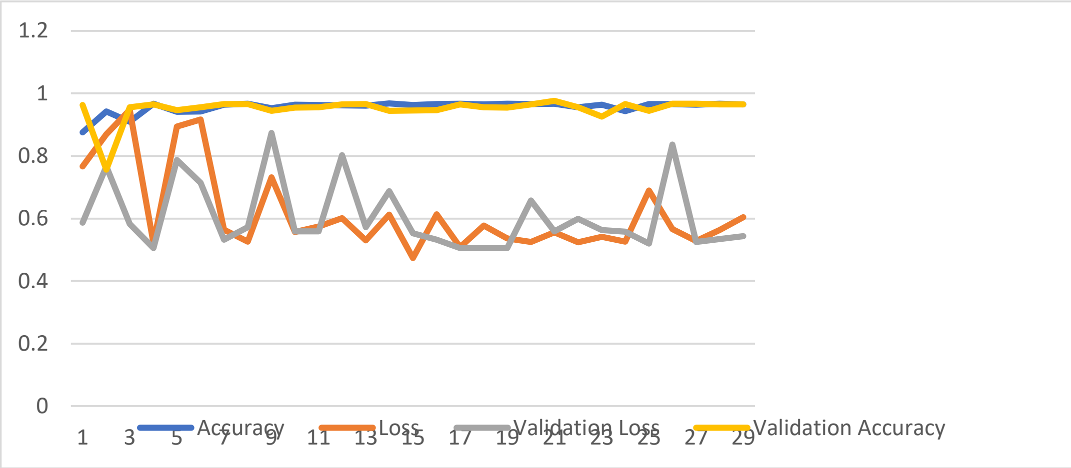


Figure 9. Graphical Visualization of Performance Parameters

Figure 10 and Figure 11 show the predicted result from the crack and no crack image testing data, respectively. The image predictor functions have been constructed in such a way that if the prediction label is less than 0.8, then the image is predicted as having no crack. If it is greater than 0.8, the image is anticipated to have a crack.

```
In [40]:  predict_image2('Positive',4000)
          Working On Predictable Data : Positive
          Images Processed from 04000 to 04001
          Raw Predicted Label(Numeric): 1.0
          Predicted Label : Crack
Out[40]:  <function __main__.predict_image2(type_, num)>
```

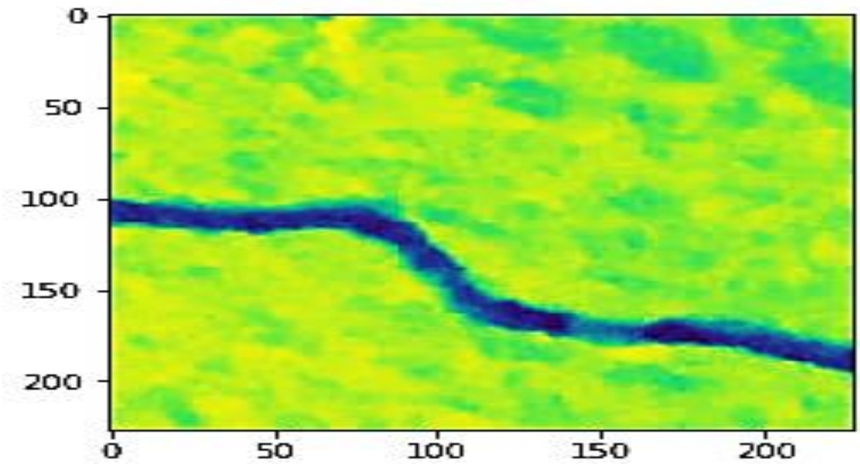


Figure 10. Predicted “Crack” Image

```
In [34]: predict_image2('Negative',4000)
         Working On Predictable Data : Negative
         Images Processed from 04000 to 04001
         Raw Predicted Label(Numeric): 0.0
         Predicted Label : No Crack
Out[34]: <function __main__.predict_image2(type_, num)>
```

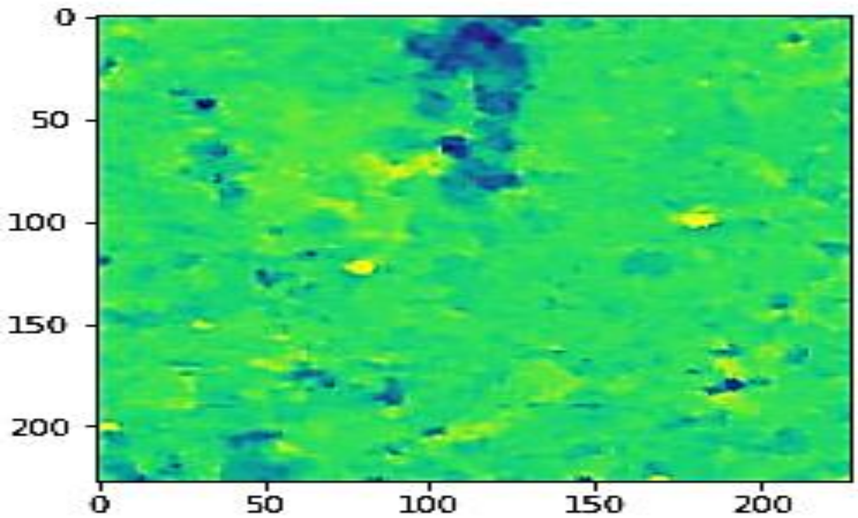


Figure 11. Predicted "No Crack" Image

CONCLUSION

This study aims to evaluate the effectiveness of applying machine learning to detect road cracks. The crack detection process was carried out in accordance with whether or not a crack was present in the input data. Concrete Crack Images for Classification was the database that was employed for this study. Determining the road crack through convolutional neural networks is the primary focus of this work. The input data in the form of images were preprocessed, and then threshold segmentation was performed on them. The output was fed to CNN, and it was there that the process of feature extraction and categorization took place. Considerable attention has been taken towards the configuration as well as the operation of the CNN model. The findings that were obtained showed tremendous potential. The accuracy of the training was determined to be 96.20 %, the accuracy of the validation was determined to be 96.50 %, and the accuracy of the testing was determined to be 94.5 %.

This research will benefit significantly from future enhancements in the sphere of road crack detection. The presentation limits can be improved by utilizing deep learning procedures with various architectures like ResNet, U-Net, etc. Once identified, cracks can be further characterized based on the type of crack and the size of the crack. It is indeed conceivable that this will help determine how severe the crack is. Additionally, deep learning-based segmentation techniques can be employed for detailed characterization alongside the use of multiple datasets.

ACKNOWLEDGMENT

The authors would like to express their gratitude to the Malaysian Ministry of Higher Education (MOHE) through the Fundamental Research Grant Scheme, FRGS21-241-0850, which has provided research funding and facilities. In addition, the author would also like to express utmost gratitude to the Kulliyyah of Engineering, International Islamic University

Malaysia, for providing the KOE Postgraduate Tuition Fee Waiver Scheme and a facilitated platform for conducting the research work.

REFERENCES

[1] V. P. Tran, T. S. Tran, H. J. Lee, K. D. Kim, J. Baek, and T. T. Nguyen, “One stage detector (RetinaNet)-based crack detection for asphalt pavements considering pavement distresses and surface objects,” *Journal of Civil Structural Health Monitoring*, vol. 11, no. 1, pp. 205–222, 2020, doi: 10.1007/s13349-020-00447-8

[2] X. Z. Xiang, Y. Q. Zhang, and A. El Saddik, “Pavement crack detection network based on pyramid structure and attention mechanism,” *IET Image Processing*, vol. 14, no. 8, pp. 1580–1586, 2020, doi: 10.1049/iet-ipr.2019.0973

[3] R. Lionnie, R. C. Ramadhan, A. S. Rosyadi, M. Jusoh & M. Alaydrus,“ Performance analysis of various types of surface crack detection based on image processing,“ SINERGI, vol. 26, no. 1, pp. 1-6, 2022, doi: 10.22441/sinergi.2022.1.001

[4] N. Safei, O. Smadi, B. Safei & A. Masoud,“ Efficient Road Crack Detection Based on an Adaptive Pixel-Level Segmentation Algorithm,“ *Transportation Research Record: Journal of the Transportation Research Board*, vol. 26759, vol. 9, 2021, doi: 10.1177/03611981211002203

[5] Q. Zou, Z. Zhang, Q. Li, X. Qi, Q. Wang and S. Wang, "DeepCrack: Learning Hierarchical Convolutional Features for Crack Detection," in *IEEE Transactions on Image Processing*, vol. 28, no. 3, pp. 1498-1512, March 2019, doi: 10.1109/TIP.2018.2878966

[6] Y. G. Mustapha, A. M. Halliru, F. A. Hobon, A. Aibinu, S. Thomas and N. S. Malami, "Crack Detection on Surfaces Using Digital Image Processing," *2021 1st International Conference on Multidisciplinary Engineering and Applied Science (ICMEAS)*, 2021, pp. 1-6, doi: 10.1109/ICMEAS52683.2021.9692367

[7] R. Desai, *Structural Optimization Using Genetic Algorithm*, 1st Ed., Florida: US, CRC Press, 2021

[8] K. Koc, Ö. Ekmekcioğlu & A. P. Gurguna,“ Integrating feature engineering, genetic algorithm and tree-based machine learning methods to predict the post-accident disability status of construction workers,“ *Automation in Construction*, vol. 131, 2021, doi: 10.1016/j.autcon.2021.103896

[9] T. T. Nguyen, S. C. Calvert, H. L. Vu and H. van Lint, "An Automated Detection Framework for Multiple Highway Bottleneck Activations," in *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 6, pp. 5678-5692, June 2022, doi: 10.1109/TITS.2021.3055640

[10] H. Suliman et al.,“ Image-Based Crack Detection Methods: A Review,“ *Infrastructures*, vol. 6, no. 8, pp. 115, 2021, doi: 10.3390/infrastructures6080115

[11] R. G. Lins and S. N. Givigi, "Automatic Crack Detection and Measurement Based on Image Analysis," in *IEEE Transactions on Instrumentation and Measurement*, vol. 65, no. 3, pp. 583-590, March 2016, doi: 10.1109/TIM.2015.2509278

[12] Y. Shi, L. Cui, Z. Qi, F. Meng and Z. Chen, "Automatic Road Crack Detection Using Random Structured Forests," in *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 12, pp. 3434-3445, Dec. 2016, doi: 10.1109/TITS.2016.2552248

[13] L. Wu, S. Mokhtari, A. Nazef, B. Nam & H. B. Yun,“H.B. Improvement of crack detection accuracy using a novel crack defragmentation technique in image-based road assessment,“ *Journal of Computing in Civil Engineering*, vol. 30, ID: 04014118, 2016, doi: 10.1061/%28ASCE%29CP.1943-5487.0000451

[14] F. -C. Chen and M. R. Jahanshahi, "NB-CNN: Deep Learning-Based Crack Detection Using Convolutional Neural Network and Naïve Bayes Data Fusion," in *IEEE Transactions on Industrial Electronics*, vol. 65, no. 5, pp. 4392-4400, May 2018, doi: 10.1109/TIE.2017.2764844

[15] X. Liang, X. Yu, C. Chen, Y. Jin and J. Huang, "Automatic Classification of Pavement Distress Using 3D Ground-Penetrating Radar and Deep Convolutional Neural Network," in *IEEE Transactions on Intelligent Transportation Systems*, 2022, doi: 10.1109/TITS.2022.3197712

[16] Y. Li, J. Ma, Z. Zhao & G. Shi,“ A Novel Approach for UAV Image Crack Detection,“ *Sensors*, vol. 22, no. 9, pp. 3305, 2022, doi: 10.3390/s22093305

[17] N. Hoang, T. Huynh, X. Tran and V. Tran,“ A Novel Approach for Detection of Pavement Crack and Sealed Crack Using Image Processing and Salp Swarm Algorithm Optimized Machine Learning,“ *Intelligent Techniques for Structural Health Monitoring of Civil Engineering Structures*, vol. 2022, ID: 9193511, 2022, doi: 10.1155/2022/9193511

[18] M. A, S. K. T. V. S, M. M, S. G. Ch and R. S, "Classification of Road Cracks using Deep Neural Networks," *2021 2nd International Conference on Smart Electronics and Communication (ICOSEC)*, 2021, pp. 1-7, doi: 10.1109/ICOSEC51865.2021.9591783

[19] C. Chu, L. Wang & H. Xiong,“ A review on pavement distress and structural defects detection and quantification technologies using imaging approaches,“ *Journal of Traffic and Transportation Engineering*, vol. 9, no. 2, pp. 135-150, 2022, doi: 10.1016/j.jtte.2021.04.007
[20] A. Zhang, K. C. P. Wang, B. Li et al., “Automated pixel-level pavement crack detection on 3D asphalt surfaces using a deep-learning network,” *Computer-Aided Civil and Infrastructure Engineering*, vol. 32, no. 10, pp. 805–819, 2017, doi: 10.1111/mice.12297
[21] Y. Fei, K. C. P. Wang, A. Zhang et al., "Pixel-Level Cracking Detection on 3D Asphalt Pavement Images Through Deep-Learning- Based CrackNet-V," in *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 1, pp. 273-284, Jan. 2020, doi: 10.1109/TITS.2019.2891167
[22] Q. Zou, Z. Zhang, Q. Li, X. Qi, Q. Wang and S. Wang, "DeepCrack: Learning Hierarchical Convolutional Features for Crack Detection," in *IEEE Transactions on Image Processing*, vol. 28, no. 3, pp. 1498-1512, March 2019, doi: 10.1109/TIP.2018.2878966
[23] S. Yokoyama and T. Matsumoto, "Development of an Automatic Detector of Cracks in Concrete Using Machine Learning," *Procedia Engineering*, vol. 171, pp. 1250-1255, 2017, doi: 10.1016/j.proeng.2017.01.418
[24] Y. Cha, W. Choi and O. Büyüköztürk,"Deep Learning-Based Crack Damage Detection Using Convolutional Neural Networks," *Computer-Aided Civil and Infrastructure Engineering Engineering*, vol. 32, no. 5, pp. 361-378, 2017, doi: 10.1111/mice.12263
[25] L. Pauly, H. Peel, S. Luo, D. Hogg and R. Fuentes, "Deeper Networks for Pavement Crack Detection," in *34th International Symposium on Automation and Robotics in Construction (ISARC)*, Taipei, Taiwan, 2017, doi: 10.22260/ISARC2017/0066
[26] K. Zhang, H. D. Cheng and B. Zhang, “A unified approach to pavement crack and sealed crack detection using preclassification based on transfer learning,“ *Journal of Computing in Civil Engineering*, vol. 32, no. 2, ID: 04018001, 2018, doi: 10.1061/(ASCE)CP.1943-5487.0000736
[27] Y. Jiang, H. Wang, S. Chen & G. Tian,“ Visual quantitative detection of rail surface crack based on laser ultrasonic technology,“ *Optik*, vol. 237, ID: 166732, 2021, doi: 10.1016/j.ijleo.2021.166732
[28] Ç. Özgenel, “Concrete Crack Images for Classification”, *Mendeley Data*, V2, 2019, doi: 10.17632/5y9wdsg2zt.2, 2019