ICIC Express Letters
Part B: Applications
Volume **11**, Number **4**, April **2020**

ICIC International ©2020 ISSN 2185-2766
pp. 373–379

PRACTICAL METHODS OF IMAGE DATA PREPROCESSING FOR ENHANCING THE PERFORMANCE OF DEEP LEARNING BASED ROAD CRACK DETECTION

Yunjung Shin[1], Minsoo Kim[2], Kyong-Won Pak[3] and Dongsoo Kim[1,*]

[1]Department of Industrial and Information Systems Engineering
Soongsil University
369 Sangdo-ro, Dongjak-gu, Seoul 06978, Korea
tlsdbswnd@soongsil.ac.kr; *Corresponding author: dskim@ssu.ac.kr

[2]Division of Systems Management and Engineering
Pukyong National University
45 Yongso-ro, Nam-gu, Busan 48513, Korea
minsky@pknu.ac.kr

[3]Korea Pavement Technology
66 Chungmin-ro, Songpa-gu, Seoul 05838, Korea
pkw4425@naver.com

Received October 2019; accepted January 2020

Abstract. *This paper proposes practical preprocessing methods for effective training of the deep learning model, taking into consideration the characteristics of image data used to detect cracks on the road surface. Convolution deep neural network extracts and recognizes features of a region included in an image, and classifies the image based on the features. However, collected raw image data do not guarantee the effective training of the deep learning model because the characteristics of the crack region and the non-crack region are not clearly distinguished and irregular noises are included in the raw image data. Therefore, we propose the preprocessing methods that can improve the performance of deep learning model in order to solve these problems.*
Keywords: Road crack, Crack detection, Image processing, Deep learning, CNN

1. **Introduction.** Recently, due to the remarkable development of image processing and machine learning technologies, deep neural networks using image data in various fields are being introduced. The technologies not only categorize labels according to objects included in image data, but also detect and recognize the objects by the segmentation of the pixels [1].

This paper deals with practical preprocessing methods to improve the performance of the deep neural network model for detecting road cracks. Specifically, we propose preprocessing methods to effectively extract features to be used in machine learning considering characteristics of road images. In the images, crack and non-crack regions are indistinguishable, as well as the noise is severe due to the impurities contained in the asphalt components. Also, because each image has high resolution, memory usage is too high to be processed in model [2]. With these backgrounds, the purpose of this study is to propose useful methods of preprocessing the image data set for effective training of the deep learning model.

The remainder of this paper is organized as follows. Section 2 presents a preliminary study for road crack detection and a case study of applying road image to CNN classifying architecture. Section 3 introduces the proposed preprocessing methods. Section 4 presents

DOI: 10.24507/icicelb.11.04.373

the overall system framework and the implementation results of the preprocessing module. Finally, Section 5 concludes the paper and suggests future research directions.

2. **Related Work.** This section deals with previous studies of techniques used in the field of automatic crack detection. A variety of studies from the filtering-based method to the deep-running model are described. And we discuss the means of collecting image data as well as the proposed model dedicated for the crack detection. Researches on the automatic crack detection have been going on before the deep learning model has been introduced in image processing field. Zalama et al. proposed a method of recognizing the visual patterns of cracks based on Garbor filters and classifying the crack images [3]. Oliveira and Correia have attempted to differentiate preprocessing methods for cracks and non-crack regions operating blockwise pre-labeling, and proposed a method for effectively applying pixel-based and block-based processing [4]. Eisenbach et al. mentioned the necessity of automatic road crack detection in a view of the amount of data to be processed in the German road maintenance environment, and suggested a deep learning model as an alternative [5]. Deep learning architectures such as ResNet and vgg16 have been studied for image classification, and Zhang et al. proposed CrackNet, a dedicated neural network structure for crack detection [6]. In addition, studies are being actively carried out to apply the deep learning model to image classification, and the studies are being conducted to apply the deep learning technique to road crack as well as surface crack detection of other industries [7]. In addition to the models used for the detection, study has also been conducted on the cost of collection of road image data. Maeda et al. used low-cost images collected on smartphones [8].

We studied preprocessing methods to train deep learning models using images collected using dedicated road scanners [2]. The data were taken from the actual images collected for road crack detection. However, due to the characteristics of these images, there are limitations that they cannot be directly used for deep learning model training like the existing studies. Therefore, this study proposes appropriate image processing methods as effective preprocessing to overcome the limitations.

3. **Methods of Image Processing for DNN.** The cracks in roads exist in a too small region compared to the size of the image, also the difference with the non-crack region is little. Therefore, preprocessing for making the crack area sensible is necessary. In addition, since the asphalt, which occupies most of the road surface, contains a lot of impurities, a precise preprocessing is required to remove the noise without deforming the crack region. For this purpose, we propose four preprocessing methods: white noise reduction, vertical normalization, image decimation, and intensity transformation.

3.1. **White noise reduction.** First of all, white noise reduction is conducted to remove asphalt impurities and white lanes. The object of this study is to investigate the existence of cracks on road surface. For example, white lanes or impurities must be removed in advance, because they deviate from the region of interest and adversely affect crack feature extraction.

Figure 1 is the process of eliminating the white noise by applying the Otsu binarization method [9]. The Otsu method divides the original image (a) into two opposite binary masks (b, e). The first mask (b) is used to extract the region of interest (d) from the raw image (c), and the opposite mask (e) is used to repair the blank region that was deleted in the previous masking process. After generating a fake image (f) using a normal distribution following the mean and deviation of the image (d), the repair mask (e) is applied to generate a patch (g), and the results of the two masks; (d) and (g) are combined to form a complete image (h).

Figures 2(a) and 2(b) show the histogram of the intensity of the image shown in Figure 1(a) and 1(h), respectively. The vertical line in Figure 2(a) is the threshold by the Otsu

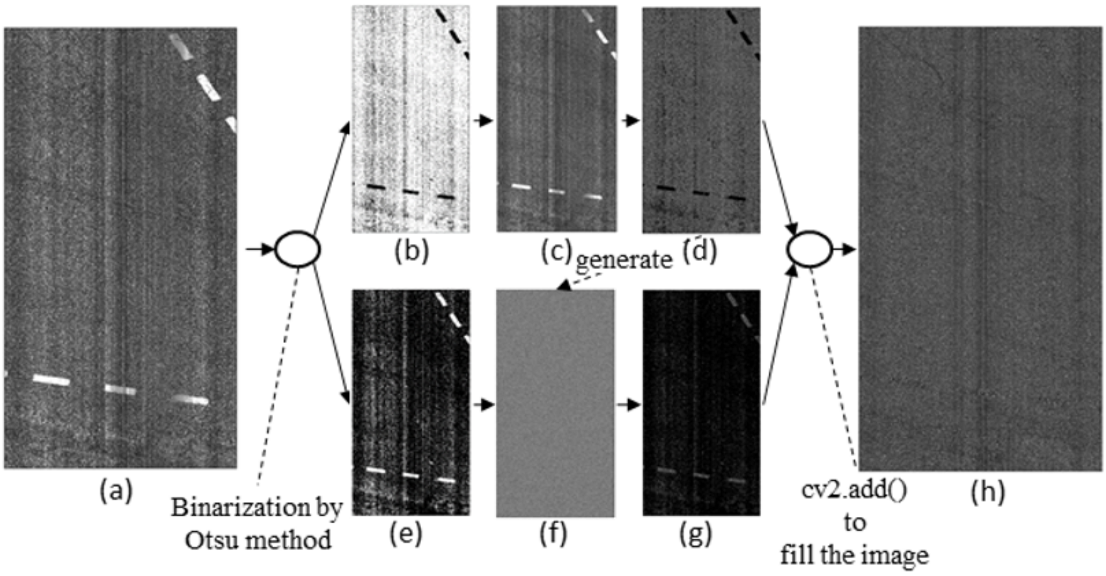

FIGURE 1. Process of white noise reduction

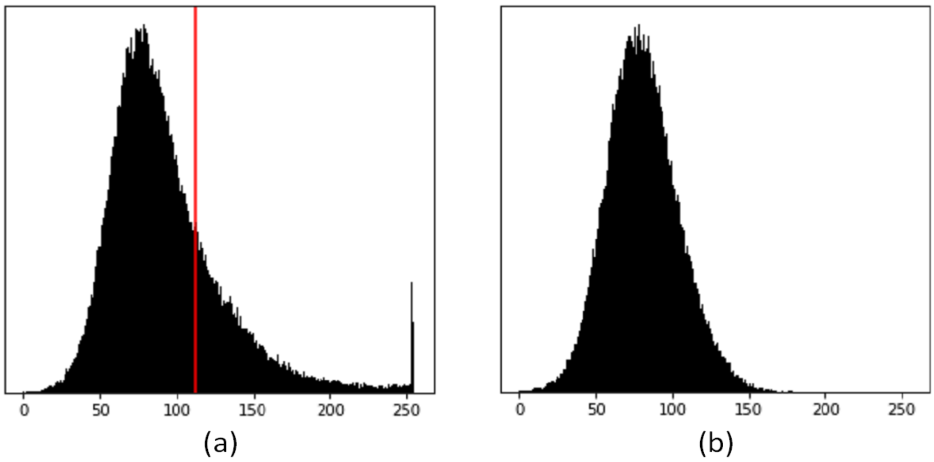

FIGURE 2. Histogram of raw image (a) and white noise reduction image (b)

method that distinguishes the white noise and the preserved region. Unlike 2(a), the histogram 2(b) shows that the tail portion of the histogram that represents bright pixels have been successfully removed.

3.2. **Vertical normalization.** The following is a method for eliminating the afterimage in the vertical direction appearing in the image. In Figure 1(h), although the white noise is removed in the level of pixel, a dark afterimage remains in the vertical direction on the image. This dark afterimage is an obstacle to DNN for featuring the dark brightness as a crack based on the intensity value. In addition, considering that the difference in intensity between the crack and non-crack region is insignificant, the relatively dark afterimage may be misclassified as a crack.

In the equation below, I_{ij} denotes the intensity value of each pixels. m_j and M denote the vertical mean and overall mean of each image respectively. Equation below is a method of applying a correction value to each pixel of an image for vertical normalization, and the correction value is calculated by the difference between the vertical average of m_j and the global average M. In Figure 3(a), there is a large difference from 60 to 120 in the average according to the column index of the raw image. Because the white noise, such

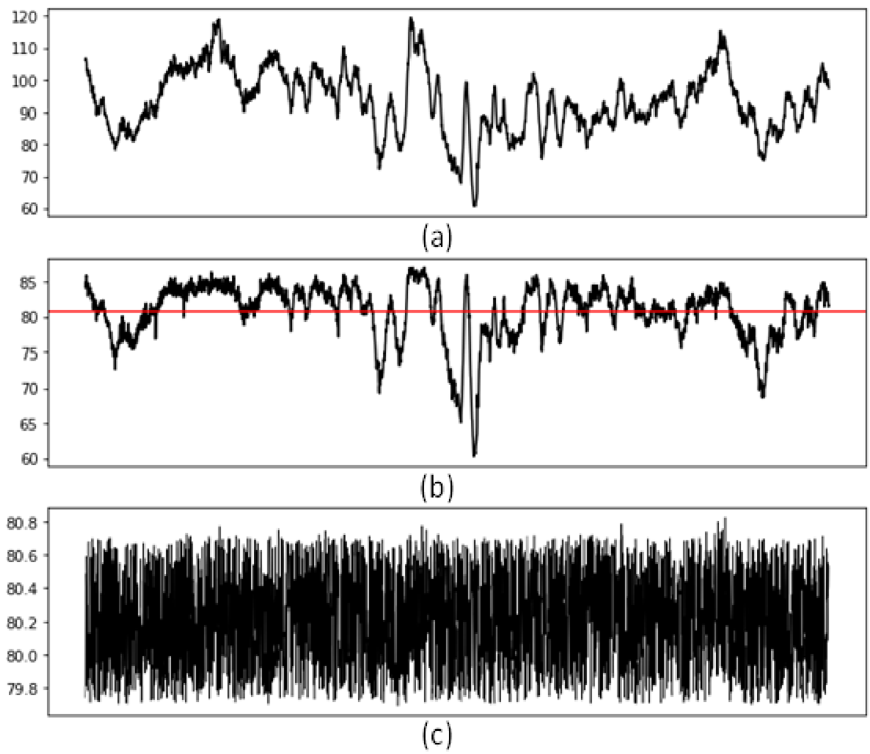

FIGURE 3. Vertical mean of raw (a), white noise reduced (b) and vertically normalized (c) image

as white lane or asphalt impurity, may affect the vertical mean, the process in 3.1 has to be performed as Figure 3(b) in advance. As shown in 3(c), the vertical mean of noise reduced image, has been preprocessed to have less than the error range 1 by reducing the average difference in each column from 79.8 to 80.8.

$$
\begin{aligned}
I_{ij} &:= I_{ij} - (M - m_j) \quad \text{for } i = 1, 2, \ldots, m \text{ and } j = 1, 2, \ldots, n \\
m_j &= \frac{\sum_i^m I_{ij}}{m} \quad \text{for } j = 1, 2, \ldots, n \\
M &= \frac{\sum_i^m \sum_j^n I_{ij}}{m \times n}
\end{aligned}
$$

3.3. **Image decimation.** Next, preprocessing to resize the image is required. The raw image collected has a resolution of 3739×10000 being about 38 million pixel values. Considering the parameter size of DNN, it is necessary to reduce the size as well as the memory usage efficiency to improve the processing speed.

When decimating the image, care must be taken not to smooth the intensity values representing the crack region. Figure 4 shows a decimated image of a crack region obtained by applying different interpolation to an image. Each interpolation is contained in the python library OpenCV's built-in functions. The interpolation methods used in Figure 4(a) are performed by the mean of the window to be downsized, resulting in smoothing of the crack region. On the other hand, 4(b) is performed by resampling the pixel values existing in the window.

3.4. **Intensity transformation.** Lastly, it is need to transform the intensity for more distinctive difference between the crack and non-crack region. The intensity stretching method converts the lower values to lower and the higher values to higher, centered by the mean of the image, as shown in Figure 5(a). By expanding the intensity of the narrow

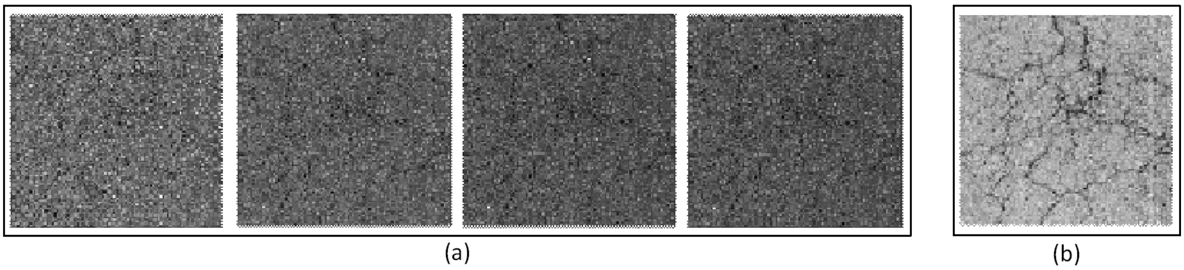

FIGURE 4. Results from the interpolation methods used for image decimation (from left to right Nearest, Linear, Cubic, Lanzcos and Area interpolation are applied respectively)

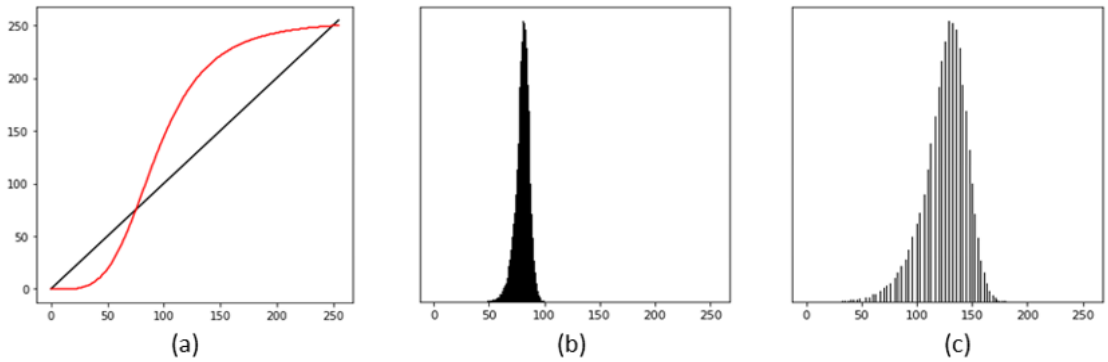

FIGURE 5. Histogram of pixel values due to intensity stretching

range of 5(b), the contrast difference can be extended as shown in 5(c), so the difference between cracks and non-cracks can be clear.

4. **Preprocessing Module.** The preprocessing methods proposed in Section 3 are performed in the order of white noise reduction, vertical normalization, image decimation, and intensity stretching. White noise has to be the first to be removed because it can affect subsequent processes. Vertical normalization is performed before the decimation to avoid losing the dark intensity pixels representing cracks in the process of resampling. Because the range of pixel values must be performed with all noise removed in order to effectively expand, the intensity transformation is performed last.

4.1. **Framework of system.** After the preprocessing, the image data set trains the deep learning model to classify whether or not each image includes any types of crack. The system is shown in Figure 6.

4.2. **Implementation results.** Each image in Figure 7 is a step-by-step result of preprocessing. From left to right, white noise reduction, vertical normalization, image decimation, and intensity transformation are there. The pixel values of the crack region are not lost, the noise is minimized, and the crack and non-crack region are made a clear difference. These results can be seen from the image obtained by extracting the crack region of the image in Figure 8.

5. **Conclusions.** This paper deals with preprocessing for effective training of the deep learning model and proposed noise reduction, image decimation, and intensity transformation as key processes. The preprocessed image is expected to remove the noise and make the feature of the crack to be more distinguishable, and to perform the feature extraction process of the deep learning model effectively. Also crack-preserved and decimated images can improve the memory efficiency of the deep learning model operation.

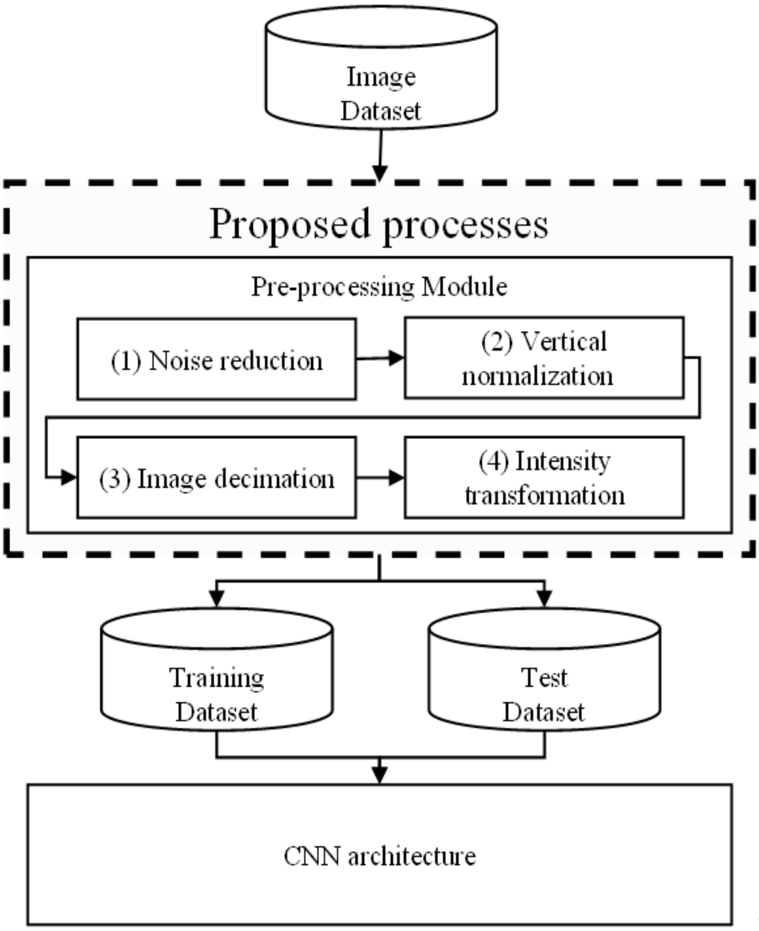

Figure 6. Overview of system framework

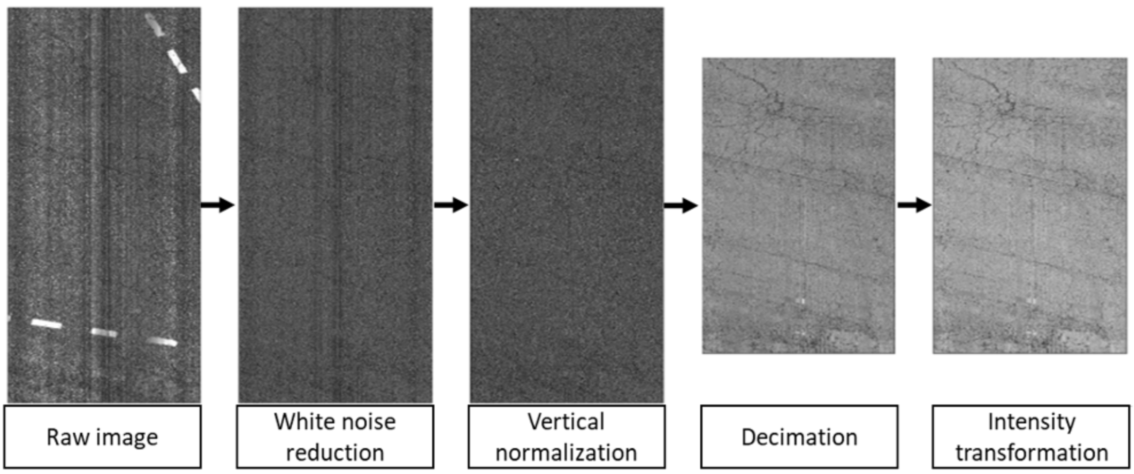

Figure 7. Result by preprocessing step

In the future, we will train the classifier of the deep learning model of various architectures with the preprocessed images. At first, binary classification model for the presence of cracks will be performed. And then multi-classification by the crack types and an object detection model for the location of cracks will be performed. As a model for this, we will use memory compression models such as SqueezeNet, and MobileNet. And in

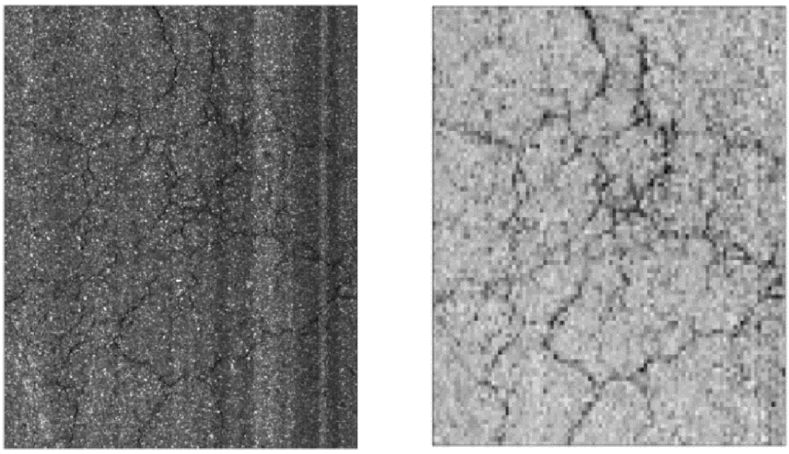

Figure 8. Crack region of the image before and after preprocessing

the labeling stage of the road crack type, it is necessary to study mutually non-exclusive multi-label because an image may contain various types of cracks. Ultimately, we construct a complete system to segment the cracks, not just labeling according to the type of cracks, and generate crack information data that can be immediately used in the road maintenance policy.

Acknowledgment. This work was supported by the Basic Science Research Program with the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2017R1D1A1B05029080).

REFERENCES

[1] K. He et al., Mask R-CNN, *Proc. of the IEEE International Conference on Computer Vision*, pp.2961-2969, 2017.

[2] Y. Shin and D. Kim, Improvement of road crack detection processes based on image processing technologies in Korea, *ICIC Express Letters, Part B: Applications*, vol.10, no.5, pp.419-426, 2019.

[3] E. Zalama et al., Road crack detection using visual features extracted by Gabor filters, *Computer-Aided Civil and Infrastructure Engineering*, vol.29, no.5, pp.342-358, 2014.

[4] 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.

[5] M. Eisenbach et al., How to get pavement distress detection ready for deep learning? A systematic approach, *International Joint Conference on Neural Networks (IJCNN)*, pp.2039-2047, 2017.

[6] A. Zhang 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.

[7] 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, 2018.

[8] H. Maeda et al., Road damage detection using deep neural networks with images captured through a smartphone, arXiv preprint, *arXiv:1801.09454*, 2018.

[9] H. Otsu, A threshold selection method from gray-level histogram, *IEEE Transactions on Systems, Man, and Cybernetics*, vol.9, no.1, pp.62-66, 1979.