

# Road Crack Detection Using Deep Convolutional Neural Network and Adaptive Thresholding

Rui Fan<sup>1\*</sup>, Mohammad Junaid Bocus<sup>2\*</sup>, Jianhao Jiao<sup>1</sup>, Yilong Zhu<sup>1</sup>, Shanshan Cheng<sup>3</sup>, Ming Liu<sup>1</sup>

**Abstract**—Crack is one of the most common road distresses which may pose road safety hazards. Generally, crack detection is performed by either certified inspectors or structural engineers. This task is, however, time-consuming, subjective and labor-intensive. In this paper, we propose a novel road crack detection algorithm based on deep learning and adaptive image segmentation. Firstly, a deep convolutional neural network is trained to determine whether an image contains cracks or not. The images containing cracks are then smoothed using bilateral filtering, which greatly minimizes the number of noisy pixels. Finally, we utilize an adaptive thresholding method to extract the cracks from road surface. The experimental results illustrate that our network can classify images with an accuracy of 99.92%, and the cracks can be successfully extracted from the images using our proposed segmentation algorithm.

## I. INTRODUCTION

Road damages in the form of cracks may reduce the road performance and pose potential road safety hazards. Every year, government bodies across the globe allocate funds to enhance the quality of their road networks [1]. Road safety should be taken very seriously and authorities are fully aware of the need for suitable road inspection and maintenance techniques [2]. Crack detection is an essential part of road maintenance systems and it has attracted growing interest from researchers in this field over the past few years [3]. Traditional manual road crack detection approaches are known to be very time-consuming, dangerous, labor-intensive and subjective [4], [5]. Therefore, the slow and subjective traditional methods have been gradually replaced by automatic crack detection which provides fast and reliable analysis in intelligent transportation systems (ITS) [6]. Automated crack detection systems can effectively assess the quality of the road surfaces and help governments plan and prioritize the maintenance of the road network, thereby keeping the roads in good condition and extending their service life [7].

With the development of image analysis techniques, road crack detection and recognition have been widely investigated over the past few decades [8], [9], [10], [11]. The traditional framework for crack detection consists of defining a variety of gradient features for each image pixel, and then using a binary classifier to determine whether an image pixel is part of a crack region or not [8]. In early methods

such as [12] and [13], the authors used threshold-based approaches to find crack regions based on the assumption that a pixel lying in a crack area is consistently darker than others [7]. Furthermore, many researchers [10], [11], [14], [15] tried to suppress the inference of noise by considering additional local features, such as the mean and the standard deviation. However, these methods are still very sensitive to noise because only the brightness features are taken into consideration.

In recent years, some novel algorithms, such as minimal path selection (MPS) [16], [17], [18], minimum spanning tree (MST) [19], [20] and crack fundamental element (CFE) [21], [22], have been proposed to improve the existing crack detection approaches. In addition, Hu and Zhao [23] proposed a crack detection algorithm based on local binary patterns (LBP), whereas the authors of [24] utilized Gabor filter for the same purpose. In [19], an automated crack detection algorithm based on a tree structure, referred to as CrackTree, was introduced. Moreover, Oliveira et al. [7], [25] utilized a comprehensive set of image analysis algorithms to detect and characterize cracks from road pavements. Although the above-mentioned algorithms have been widely used in crack detection and they perform well on high-quality datasets [19], [25], [26], it is important to note that these algorithms are not accurate enough to distinguish cracks from the complex background in low-quality images.

Furthermore, some machine learning-based crack detection approaches [27], [28], [29], [30], [31], [32] have been proposed in recent years, and the deep features produced by neural network are very likely to replace the local features utilized in traditional methods [33]. For example, restricted Boltzmann machine (RBM) auto encoder and their variants are capable of detecting cracks when the training samples are limited [34]. In addition, deep convolutional neural networks (DCNNs) are popular for feature-learning and supervised classification [35]. Zhang et al. [33] trained a neural network to determine whether the patches in an images contain cracks or not. Hence in this paper, we build on the recent successful application of deep neural network to image classification and train a convolutional neural network (CNN) to find the images that contain cracks. Then, we present a novel thresholding method to extract cracks from classified color images. Our code is publicly available at: <http://www.ruirangerfan.com>.

The remainder of this paper is structured as follows: Section II introduces the proposed crack detection algorithm. In Section III, we present our experimental results and discuss the performance of the proposed method. Finally, Section IV

<sup>1</sup>R. Fan, J. Jiao, Y. Zhu and M. Liu are with the Robotics and Multi-Perception Laboratory in Robotics Institute at the Hong Kong University of Science and Technology, Hong Kong.

<sup>2</sup>M. Junaid Bocus is with the Visual Information Institute at the University of Bristol, Bristol, United Kingdom.

<sup>3</sup>S. Cheng is with the National Engineering Research Center of Road Maintenance Technologies, Beijing, China.

\*These two authors contributed equally to this work and therefore are joint first authors. Corresponding author: Rui Fan. Email: rui.fan@ieee.org.

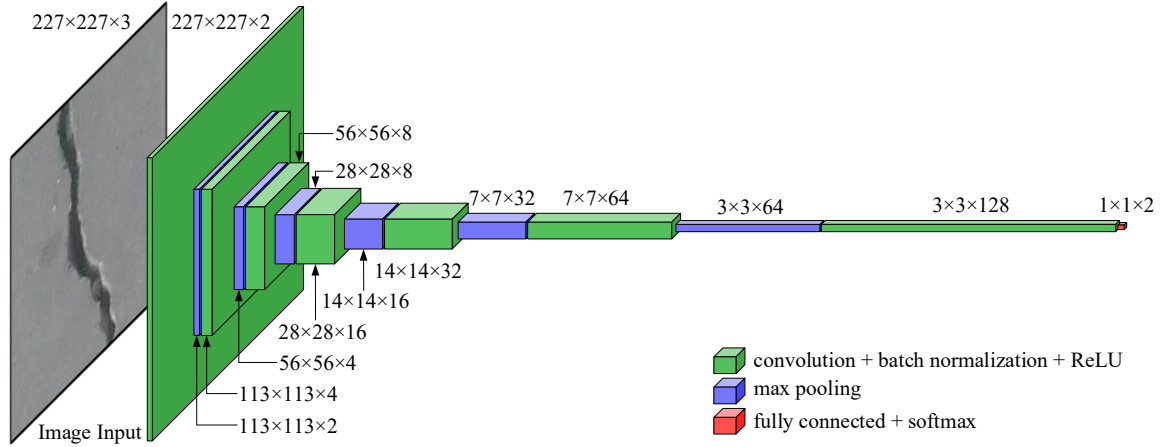


Fig. 1. The structure of the proposed deep neural network for image classification.

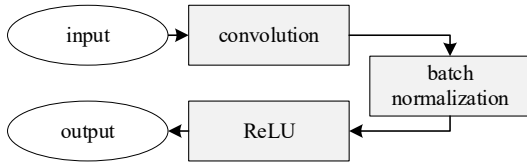


Fig. 2. The structure of the green block in Fig. 1.

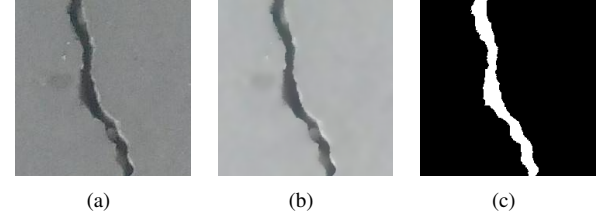


Fig. 3. Bilateral filtering and image segmentation. (a) Original positive image. (b) Filtered positive image. (c) Segmentation result.

concludes the paper and provides some recommendations for future work.

## II. METHODOLOGY

The proposed crack detection method consists of two steps: image classification and image segmentation. For notational convenience, images showing the presence and absence of cracks are referred to as positive and negative images, respectively. Firstly, an image is classified as either positive or negative using a deep convolutional neural network. The positive images are then processed using an adaptive thresholding method. The cracks in the positive images can therefore be extracted. The rest of this section gives a detailed description of these two steps.

### A. Image Classification

The structure of the proposed deep convolutional neural network is shown in Fig. 1, where ReLU represents a rectified linear unit, which is the most popular activation function for deep neural networks due to its better performance than both sigmoid function and hyperbolic tangent function in terms of training and evaluation [35]. A CNN is generally considered as a hierarchical feature extractor [33]. A convolutional layer performs a convolution operation on the image input and passes the extracted features to the next layer [35]. Batch normalization is then performed on the output of the convolutional layer whereby the extracted features are normalized by adjusting and scaling the activations. [36]. The structure of the green block in Fig. 1 are shown in Fig. 2. Max pooling downsamples the input representations

[35], whereas the softmax function translates a vector into a probability distribution. Finally, a fully connected layer computes the score of each class and infers the category of the input image [33]. Therefore, the proposed network is also referred to as a fully connected network (FCN). More details on the training process are provided in Section III.

### B. Image Segmentation

Since the images have already been classified using our proposed deep neural network, only the positive images are considered for processing in this subsection. Before performing image segmentation, we first utilize a bilateral filter [37] to smooth the input images. Bilateral filter outperforms other image filters in terms of edge preservation [38]. A general expression for bilateral filtering is as follows:

$$i_{bf}(u, v) = \frac{\sum_{x=u-\rho}^{u+\rho} \sum_{y=v-\rho}^{v+\rho} \omega_s(x, y) \omega_c(x, y) i(x, y)}{\sum_{x=u-\rho}^{u+\rho} \sum_{y=v-\rho}^{v+\rho} \omega_s(x, y) \omega_c(x, y)}, \quad (1)$$

where

$$\omega_s(x, y) = \exp \left\{ -\frac{(x-u)^2 + (y-v)^2}{\sigma_s^2} \right\}, \quad (2)$$

$$\omega_c(x, y) = \exp \left\{ -\frac{(i(x, y) - i(u, v))^2}{\sigma_c^2} \right\}. \quad (3)$$

$i(x, y)$  represents the intensity of a pixel at  $(x, y)$  in the input image.  $i_{bf}(u, v)$  denotes the intensity of a pixel at  $(u, v)$  in the

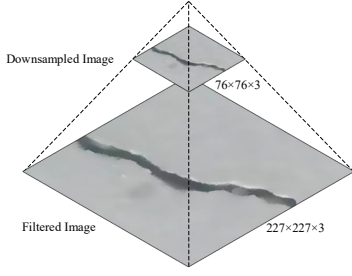


Fig. 4. Image downsampling.

filtered image.  $\omega_s$  and  $\omega_c$  are based on spatial distance and color similarity, respectively. Their values are controlled by two parameters  $\sigma_s$  and  $\sigma_c$ , respectively. In our experiments, the values of  $\sigma_s$  and  $\sigma_c$  are set to 300 and 0.1, respectively.  $\rho$  is set to 5. The filtered image is shown in Fig. 3.

To further reduce noise in the filtered image, the latter is downsampled as shown in Fig. 4. The downsampled image is approximately nine times smaller than the original filtered image and it is utilized as the threshold for image segmentation. It is to be noted that the intensity of a pixel in the downsampled image is normalized.

The proposed thresholding method hypothesizes that the downsampled image is composed of two parts: foreground (cracks) and background (road surface), and they can be separated using one threshold  $\delta$ . Furthermore, we assume that a pixel lying in a crack area is consistently darker than others. To find the best threshold  $\delta$ , we formulate the thresholding problem as a 2D vector quantization problem, where each pixel  $\mathbf{p} = [x, y]^T$  and its neighborhood system  $\mathbf{Q} = [\mathbf{q}_{p_0}, \mathbf{q}_{p_1}, \dots, \mathbf{q}_{p_n}]^T$  provide a vector  $\mathbf{m} = [i_{ds}, i_{nb}]^T$ , where  $i_{ds}$  represents the intensity of  $\mathbf{p} = [x, y]^T$ , and  $i_{nb}$  denotes the mean intensity of  $\mathbf{Q}$ . The vectors are stored in a 2D histogram, as shown in Fig. 5. The relationship between  $i_{ds}$  and  $i_{nb}$  is as follows:

$$i_{nb}(u, v) = \sum_{x=u-\tau}^{u+\tau} \sum_{y=v-\tau}^{v+\tau} i_{ds}(x, y) - i_{nb}(u, v), \quad (4)$$

where  $\tau$  dictates the size of the neighborhood system. The threshold can be determined by partitioning the vectors into two clusters  $\mathcal{S} = \{\mathcal{S}_1, \mathcal{S}_2\}$ , where  $\mathcal{S}_1$  and  $\mathcal{S}_2$  correspond to the foreground and background, respectively.

According to the Markov Random Field [39], the intensity of a pixel which is not located near the boundary between foreground and background is similar to those of its neighbors in all directions. Hence, we search for the threshold along the principal diagonal of the 2D histogram using  $k$ -mean clustering [40]. Given a threshold  $\delta$ , the 2D histogram can be divided into four regions, as shown in Fig. 5. Regions 1 and 2 store the vectors of foreground and background, respectively. On the other hand, regions 3 and 4 store the noisy vectors. In our method, the vectors in regions 3 and 4 are not considered into the clustering process. The best threshold is computed by minimizing the within-cluster sum

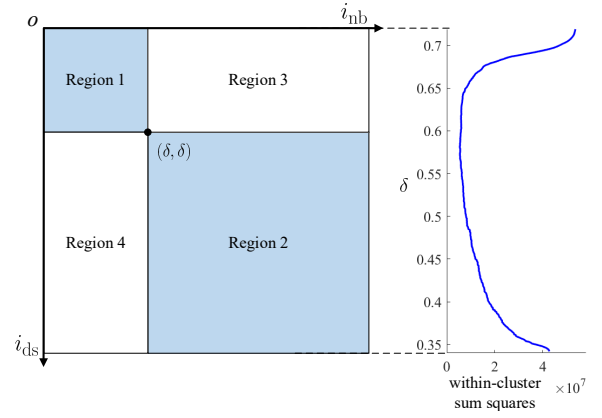


Fig. 5. 2D histogram and the within-cluster sum squares with respect to different  $\delta$ .

squares as follows:

$$\arg \min_{\mathcal{S}} \sum_{i=1}^2 \sum_{\mathbf{m} \in \mathcal{S}_i} \|\mathbf{m} - \boldsymbol{\mu}_i\|^2, \quad (5)$$

where  $\boldsymbol{\mu}_i$  denotes the mean of the vectors in  $\mathcal{S}_i$ . Eq. 5 can be rearranged as follows:

$$\arg \min_{\mathcal{S}} \sum_{i=1}^2 \sum_{\mathbf{m} \in \mathcal{S}_i} \|\mathbf{m}\|^2 - n_1 \|\boldsymbol{\mu}_1\|^2 - n_2 \|\boldsymbol{\mu}_2\|^2, \quad (6)$$

where  $\boldsymbol{\mu}_1$  and  $\boldsymbol{\mu}_2$  represent the means of vectors in regions 1 and 2, respectively.  $n_1$  and  $n_2$  denote the numbers of vectors in regions 1 and 2, respectively. The within-cluster sum squares with respect to different  $\delta$  are shown in Fig. 5, and the corresponding segmentation result is shown in Fig. 3(c), where the crack and road surface are in white and black, respectively. The performance of the crack detection algorithm is evaluated in Section III.

### III. EXPERIMENTAL RESULTS

In our experiments, the proposed deep neural network is trained on an NVIDIA GTX 1080 Ti GPU<sup>1</sup>, which has 3584 CUDA cores and 11 GB GDDR5X memory. The memory bandwidth of the GPU is 484 GB/s. The training is implemented on Matlab R2018b. Our trained neural network is publicly available at: <http://www.ruirangerfan.com>.

The dataset<sup>2</sup> utilized for training the proposed network was created by the researchers from Middle East Technical University. The dataset contains 40000 RGB images (resolution:  $227 \times 227$ ). The number of positive and negative images are both 20000.

In our practical experiments, we randomly select 15000 positive images and 15000 negative images from the dataset, to train the neural network. The rest of the images are utilized to evaluate the performance of the proposed approach. The

<sup>1</sup><https://www.nvidia.com/en-us/geforce/products/10series/geforce-gtx-1080-ti/>

<sup>2</sup><http://dx.doi.org/10.17632/5y9wdsg2zt.1>



Fig. 6. Experimental results of image classification. (a) True positive images. (b) True negative images.



Fig. 7. The failed classification results. (a) False positive images. (b) False negative images.

initial learning rate, the maximum number of epochs and the validation frequency are set to 0.01, 16 and 60, respectively. The stochastic gradient descent with momentum (SGDM) is utilized as the optimizer, and the value of momentum is set to 0.9.

To quantify the accuracy of our proposed image classifier, we compute  $n_{tp}$ ,  $n_{fp}$ ,  $n_{fn}$  and  $n_{tn}$ , which represent the number of testing images that are true positives, false positives, false negatives and true negatives, respectively. The precision, recall, accuracy and  $F_1$ -measure can be computed using the following equation:

$$\text{precision} = \frac{n_{tp}}{n_{tp} + n_{fp}}, \quad (7)$$

$$\text{recall} = \frac{n_{tp}}{n_{tp} + n_{fn}}, \quad (8)$$

$$\text{accuracy} = \frac{n_{tp} + n_{tn}}{n_{tp} + n_{tn} + n_{fp} + n_{fn}}, \quad (9)$$

$$F_1 = 2 \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}. \quad (10)$$

The values of precision, recall, accuracy and  $F_1$ -measure achieved using the proposed network are 99.92% (the number of false positive images and false negative images are both four). The true positive and true negative examples are shown in Fig. 6, while the false positive and false negative results are shown in Fig. 7. The image classification takes

TABLE I  
COMPARISON BETWEEN OUR PROPOSED METHOD AND OTSU'S THRESHOLDING METHOD.

Method	Precision	Recall	Accuracy	$F_1$ -measure
Otsu's thresholding	0.9590	<b>0.9339</b>	0.9848	0.9462
Proposed ( $\tau = 1$ )	0.9774	0.9331	<b>0.9870</b>	<b>0.9548</b>
Proposed ( $\tau = 2$ )	0.9854	0.9246	0.9867	0.9541
Proposed ( $\tau = 3$ )	0.9967	0.9046	0.9848	0.9484
Proposed ( $\tau = 4$ )	<b>0.9955</b>	0.8952	0.9831	0.9427

about 4.8 ms on an Intel Core i7-8700K CPU processed with a single core (3.7 GHz).

Furthermore, we evaluate the performance of the image segmentation at pixel level. Some experimental results of image filtering and segmentation are shown in Fig. 8. Since the dataset we use does not contain pixel-level ground truth, we manually label the crack areas in a set of images and use these ground truth images to quantify the performance of our proposed segmentation method. Moreover, we compare our method with Otsu's thresholding method [41] with respect to pixel-level precision, recall, accuracy and  $F_1$ -measure. The notations  $n_{tp}$ ,  $n_{fp}$ ,  $n_{fn}$  and  $n_{tn}$  in (7), (8) and (9) represent the number of pixels that are true positives, false positives, false negatives and true negatives, respectively. The comparison between these two methods is shown in Table I. It is to be noted here that the crack areas with less than 100 pixels are ignored in our experiments. From Table I, it can be observed that our method outperforms Otsu's thresholding method in terms of precision, accuracy and  $F_1$ -measure, and the proposed segmentation method achieves the best performance when  $\tau$  is set to 1.

#### IV. CONCLUSION AND FUTURE WORK

A novel crack detection approach was proposed in this paper. The main novelties include a fully connected neural network for image classification and a  $k$ -mean clustering based image segmentation algorithm. Firstly, our neural network classified the input images as either positive (crack present) or negative (crack absent). The positive images were then processed using a bilateral filter, which not only minimized the number of noisy pixels but also preserved the edges between the cracks and road surface. Finally, the



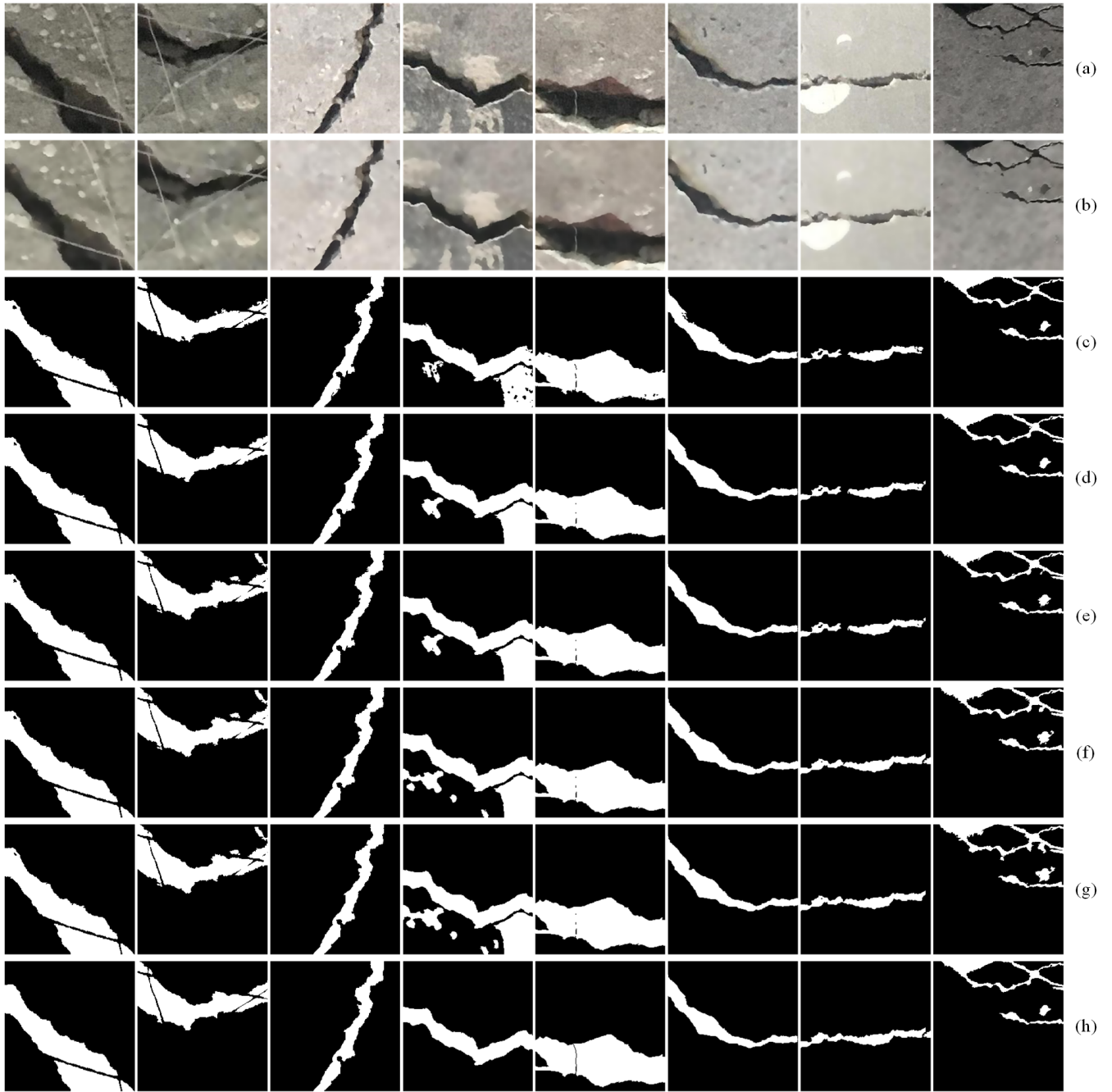


Fig. 8. Experimental results of image segmentation. (a) Input images. (b) Filtered images. (c) Results obtained using Otsu's thresholding method. (d) Results obtained using the proposed method for  $\tau = 1$ . (e)  $\tau = 2$ . (f)  $\tau = 3$ . (g)  $\tau = 4$ . (h) Ground truth.

filtered images were downsampled, and an adaptive threshold was computed by minimizing the within-cluster sum squares. The cracks could therefore be detected by segmenting the filtered images using the adaptively determined threshold. The experimental results showed that the precision of the image classification is 99.92% and the pixel-level segmentation accuracy is around 98.70%.

Although the proposed image segmentation algorithm performs better than Otsu's thresholding method in terms of distinguishing between foreground (cracks) and background

(road surface), some color images with a large number of noisy pixels cannot be properly segmented. Therefore, as a future work, a deep neural network can be trained to segment the positive images into a set of semantically meaningful regions, such as cracks and road surface.

#### ACKNOWLEDGMENT

This work is supported by grants from the Research Grants Council of the Hong Kong SAR Government, China (No. 11210017, No. 16212815, No. 21202816, NSFC U1713211) awarded to Prof. Ming Liu. This work is also supported

by grants from the Shenzhen Science, Technology and Innovation Commission, JCYJ20170818153518789, and National Natural Science Foundation of China (No. 61603376) awarded to Dr. Lujia Wang.

## REFERENCES

- [1] H. Oliveira and P. L. Correia, "Automatic road crack segmentation using entropy and image dynamic thresholding," in *Signal Processing Conference, 2009 17th European*. IEEE, 2009, pp. 622–626.
- [2] R. Fan, X. Ai, and N. Dahnoun, "Road surface 3D reconstruction based on dense subpixel disparity map estimation," vol. 27, no. 6, pp. 3025–3035, Jun. 2018.
- [3] T. S. Nguyen, M. Avila, and S. Begot, "Automatic detection and classification of defect on road pavement using anisotropy measure," in *Signal Processing Conference, 2009 17th European*. IEEE, 2009, pp. 617–621.
- [4] H. Cheng, J.-R. Chen, C. Glazier, and Y. Hu, "Novel approach to pavement cracking detection based on fuzzy set theory," *Journal of Computing in Civil Engineering*, vol. 13, no. 4, pp. 270–280, 1999.
- [5] H. Cheng, J. Wang, Y. Hu, C. Glazier, X. Shi, and X. Chen, "Novel approach to pavement cracking detection based on neural network," *Transportation Research Record: Journal of the Transportation Research Board*, no. 1764, pp. 119–127, 2001.
- [6] Y. Shi, L. Cui, Z. Qi, F. Meng, and Z. Chen, "Automatic road crack detection using random structured forests," vol. 17, no. 12, pp. 3434–3445, Dec. 2016.
- [7] 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.
- [8] H. Oh, N. W. Garrick, and L. E. Achenie, "Segmentation algorithm using iterative clipping for processing noisy pavement images," in *Imaging Technologies: Techniques and Applications in Civil Engineering. Second International Conference Engineering Foundation; and Imaging Technologies Committee of the Technical Council on Computer Practices, American Society of Civil Engineers.*, 1998.
- [9] M. Petrou, J. Kittler, and K. Song, "Automatic surface crack detection on textured materials," *Journal of materials processing technology*, vol. 56, no. 1-4, pp. 158–167, 1996.
- [10] Y. Huang and B. Xu, "Automatic inspection of pavement cracking distress," *Journal of Electronic Imaging*, vol. 15, no. 1, p. 013017, 2006.
- [11] M. Gavilán, D. Balcones, O. Marcos, D. F. Llorca, M. A. Sotelo, I. Parra, M. Ocaña, P. Aliseda, P. Yarza, and A. Amírola, "Adaptive road crack detection system by pavement classification," *Sensors*, vol. 11, no. 10, pp. 9628–9657, 2011.
- [12] M. S. Kaseko and S. G. Ritchie, "A neural network-based methodology for pavement crack detection and classification," *Transportation Research Part C: Emerging Technologies*, vol. 1, no. 4, pp. 275–291, 1993.
- [13] Q. Li and X. Liu, "Novel approach to pavement image segmentation based on neighborhood difference histogram method," in *Image and Signal Processing, 2008. CISP'08. Congress on*, vol. 2. IEEE, 2008, pp. 792–796.
- [14] N. Tanaka and K. Uematsu, "A crack detection method in road surface images using morphology," *MVA*, vol. 98, pp. 17–19, 1998.
- [15] H. Oliveira and P. L. Correia, "Supervised strategies for cracks detection in images of road pavement flexible surfaces," in *Signal Processing Conference, 2008 16th European*. IEEE, 2008, pp. 1–5.
- [16] R. Amhaz, S. Chambon, J. Idier, and V. Baltazart, "A new minimal path selection algorithm for automatic crack detection on pavement images," in *IEEE International Conference on Image Processing (ICIP 2014)*, 2014, pp. 788–792.
- [17] M. Avila, S. Begot, F. Duculty, and T. S. Nguyen, "2d image based road pavement crack detection by calculating minimal paths and dynamic programming," in *Image Processing (ICIP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 783–787.
- [18] R. Amhaz, S. Chambon, J. Idier, and V. Baltazart, "Automatic crack detection on 2d pavement images: An algorithm based on minimal path selection, accepted to ieee trans," *Intell. Transp. Syst.*, 2015.
- [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] K. Fernandes and L. Ciobanu, "Pavement pathologies classification using graph-based features," in *Image Processing (ICIP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 793–797.
- [21] Y.-C. Tsai, C. Jiang, and Y. Huang, "Multiscale crack fundamental element model for real-world pavement crack classification," *Journal of Computing in Civil Engineering*, vol. 28, no. 4, p. 04014012, 2012.
- [22] Y. J. Tsai, C. Jiang, and Z. Wang, "Implementation of automatic crack evaluation using crack fundamental element," in *Image Processing (ICIP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 773–777.
- [23] Y. Hu and C.-x. Zhao, "A novel lbp based methods for pavement crack detection," *Journal of pattern Recognition research*, vol. 5, no. 1, pp. 140–147, 2010.
- [24] M. Salman, S. Mathavan, K. Kamal, and M. Rahman, "Pavement crack detection using the gabor filter," in *16th international IEEE conference on intelligent transportation systems (ITSC 2013)*. IEEE, 2013, pp. 2039–2044.
- [25] H. Oliveira and P. L. Correia, "Crackit—an image processing toolbox for crack detection and characterization," in *Image Processing (ICIP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 798–802.
- [26] S. Varadharajan, S. Jose, K. Sharma, L. Wander, and C. Mertz, "Vision for road inspection," in *Applications of Computer Vision (WACV), 2014 IEEE Winter Conference on*. IEEE, 2014, pp. 115–122.
- [27] H. R. Roth, L. Lu, J. Liu, J. Yao, A. Seff, K. Cherry, L. Kim, and R. M. Summers, "Improving computer-aided detection using convolutional neural networks and random view aggregation," *IEEE transactions on medical imaging*, vol. 35, no. 5, pp. 1170–1181, 2016.
- [28] D. Cireşan, A. Giusti, L. M. Gambardella, and J. Schmidhuber, "Deep neural networks segment neuronal membranes in electron microscopy images," in *Advances in neural information processing systems*, 2012, pp. 2843–2851.
- [29] D. C. Cireşan, A. Giusti, L. M. Gambardella, and J. Schmidhuber, "Mitosis detection in breast cancer histology images with deep neural networks," in *International Conference on Medical Image Computing and Computer-assisted Intervention*. Springer, 2013, pp. 411–418.
- [30] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.
- [31] Y. Zhang, K. Sohn, R. Villegas, G. Pan, and H. Lee, "Improving object detection with deep convolutional networks via bayesian optimization and structured prediction," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 249–258.
- [32] J. Kivinen, C. Williams, and N. Heess, "Visual boundary prediction: A deep neural prediction network and quality dissection," in *Artificial Intelligence and Statistics*, 2014, pp. 512–521.
- [33] L. Zhang, F. Yang, Y. D. Zhang, and Y. J. Zhu, "Road crack detection using deep convolutional neural network," in *Image Processing (ICIP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 3708–3712.
- [34] Y. Xu, S. Li, D. Zhang, Y. Jin, F. Zhang, N. Li, and H. Li, "Identification framework for cracks on a steel structure surface by a restricted boltzmann machines algorithm based on consumer-grade camera images," *Structural Control and Health Monitoring*, vol. 25, no. 2, p. e2075, 2018.
- [35] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, p. 436, 2015.
- [36] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.
- [37] S. Paris, P. Kornprobst, J. Tumblin, F. Durand, B. Curless, L. Van Gool, and R. Szeliski, "Bilateral filtering: Theory and applications: Series: Foundations and trends® in computer graphics and vision," 2009.
- [38] R. Fan and N. Dahnoun, "Real-time stereo vision-based lane detection system," *Measurement Science and Technology*, vol. 29, no. 7, p. 074005, 2018.
- [39] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," vol. 23, no. 11, pp. 1222–1239, Nov. 2001.
- [40] A. Ahmad and L. Dey, "A k-mean clustering algorithm for mixed numeric and categorical data," *Data & Knowledge Engineering*, vol. 63, no. 2, pp. 503–527, 2007.
- [41] N. Otsu, "A threshold selection method from gray-level histograms," and *Cybernetics IEEE Transactions on Systems, Man*, vol. 9, no. 1, pp. 62–66, Jan. 1979.