Proceedings of TENCON 2018 - 2018 IEEE Region 10 Conference (Jeju, Korea, 28-31 October 2018)

Pixel-level Crack Detection using U-Net

Jierong Cheng
Institute for Infocomm Research
Agency for Science, Technology and Research
Singapore
chengjr@i2r.a-star.edu.sg

Wei Xiong
Institute for Infocomm Research
Agency for Science, Technology and Research
Singapore
wxiong@i2r.a-star.edu.sg

Wenyu Chen
Institute for Infocomm Research
Agency for Science, Technology and Research
Singapore
chenw@i2r.a-star.edu.sg

Ying Gu
Institute for Infocomm Research
Agency for Science, Technology and Research
Singapore
guy@i2r.a-star.edu.sg

Yusha Li
Institute for Infocomm Research
Agency for Science, Technology and Research
Singapore
li_yusha@i2r.a-star.edu.sg

***Abstract*—In this paper, we proposed an automatic crack detection method based on deep convolutional neural network –U-Net [4]. Unlike existing machine learning based crack detection methods, we can process an image as a whole without patchifying, thanks to the encoder-decoder structure of U-Net. The segmentation result is output from the network as a whole, instead of aggregation from neighborhood patches. In addition, a new cost function based on distance transform is introduced to assign pixel-level weight according to the minimal distance to the ground truth segmentation. In experiments, we test the proposed method on two datasets of road crack images. The pixel-level segmentation accuracy is above 92% which outperforms other state-of-the-art methods significantly.**

***Index Terms*—U-Net, convolutional neural network, crack detection, crack segmentation**

I. Introduction

Crack detection is a routine and essential task in road maintenance and building inspection. Automatic detection and segmentation of cracks is challenging due to the complex appearance of cracks and various illumination conditions and texture on background surface. Early study on automatic crack detection adopted conventional image processing methods to extract features and thresholding or binary classifiers are applied to segment cracks from background.

Recently, more and more sophisticated methods emerged for crack detection, especially machine-learning based methods. In [10], cracks in pavement images are detected by selecting dark pixels as endpoints and finding minimal paths between endpoint pairs. This method relies on the photometry only and is performed in a fully unsupervised way. Random structured forest is used in [1] for the detection of road cracks. After the structured learning procedure, structured labels are used to characterize and classify the defect from the statistical histograms. The biggest limitation of random structured forest is that any prediction must have been observed during training: novel labels are unable to be synthesized [3]. In practice, this shortcoming is ameliorated with custom ensemble models. For instance in edge detection, structured predictions for each image patch are obtained independently and overlapping predictions are averaged. It results in a blurred edge response and needs to be thresholded to generate crack segmentation [1].

Due to the tremendous advancement of deep convolutional neural network (CNN) in the field of image processing and pattern recognition, a few CNN-based methods have been proposed for crack detection. Zhang et al. used trained CNN to classify each image patch into crack and non-crack in pavement images. To detect the cracks from patch level to pixel level, a threshold maximizing the F-score was applied on the probability map and the result often overestimate the crack width. Schmugge et al. adapted SegNet [6] for crack segmentation in video frames [5]. The crack probability for each pixels are computed from SegNet and the pixels are classified by aggregating the probabilities of the same physical location from multiple views. In these works, deep features learned by deep neural networks showed their distinct advantages over conventional hand-crafted features. However, a common issue in all above-mentioned methods is that the training and testing of crack images are performed both in patch-level and the final segmentation result is an aggregation from the prediction of a neighborhood of pixels.

In this work, we proposed an pixel-level crack detection method using a CNN called U-Net. The input and output of the CNN are whole images and no neighborhood aggregation is required in post-processing. The contributions of this work are two-folds: 1) apply CNN on pavement crack detection and segmentation and achieve outstanding pixel-level accuracy; 2)

978-1-5386-5457-6/18/$31.00 ©2018 IEEE

propose a lost function using distance transform and prove its efficiency.

II. Method

A. Network Architecture

The architecture of U-Net is shown in Fig. 1. On the left side of the network, the image is contracted by two 3×3 convolution and 2×2 max pooling each for four times, each time doubling the number of feature channels. On the right side of the network, the image is expanded by 2×2 up-convolution and two 3×3 convolution each for four times, each time halving the number of feature channels. Every convolution is activated by a rectified linear unit (ReLU). A 1×1 convolution layer is added last to map each feature vector to class labels using sigmoid function.

The left side of the U shape can be considered as an encoder and the right side a decoder. The encoder gradually reduces the spatial dimension of pooling layers and the decoder gradually recovers the spatial dimension and object details. The direct copy and crop operations between the encoder and decoder (gray arrows in Fig. 1) helps the recovery of details in the target.

The input data for U-Net is a whole image which might contain one or more cracks. The output is of the same size of input image which each pixel is given a probability for each class label. For our two-class problem, the output segmentation map is a grayscale image with crack pixel in black and non-crack pixel in white.

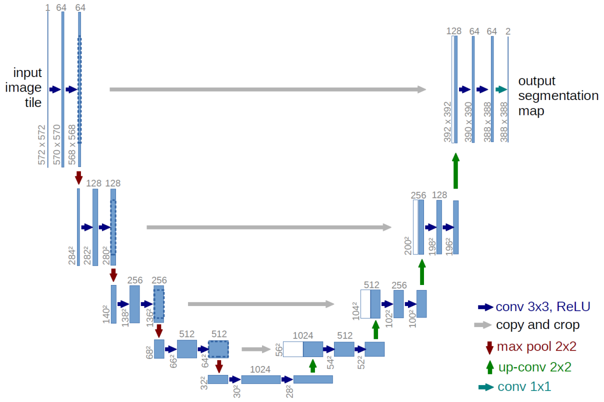


Fig. 1. Architecture of U-Net [4].

B. Lost Function

Cross-entropy is commonly used in the loss function for training and testing of CNNs. It measures the performance of a classifier whose output is a probability value between 0 and 1. Let y_i be a binary indicator if the true label is class label i and $\hat{y}_i$ be the predicted probability of class i. At each pixel, cross entropy loss is defined by

$$L = -\sum_{i=1}^{K} (y_i \log \hat{y}_i + (1 - y_i) \log(1 - \hat{y}_i)) \tag{1}$$

where K is the number of classes. The loss increases as the predicted probability deviates from the actual label.

In [4], U-Net was applied to the problem of segmenting HeLa cells in microscopy images. To emphasize the network on predicting the narrow boundaries between touching cells, the authors introduced a weight map based on the distance to the boundary of the nearest cells. In such a way, pixels located on the border of two cells are assigned with higher weight than the rest of pixels. The value of weight and coefficients are empirically set.

For crack segmentation, we face another challenge as the cracks occupy only a small portion of pixels on the image. To handle the imbalance between crack pixels and non-crack pixels, we can use class balanced cross entropy

$$L = -\sum_{i=1}^{K} (\beta_i y_i \log \hat{y}_i + (1 - \beta_i)(1 - y_i) \log(1 - \hat{y}_i)) \tag{2}$$

where β_i is the ratio of samples in class i.

Chamfer distance [12] is commonly employed in shape-based object detection and template matching with binary images. Let T be a binary template image denoting ground true and S be a segmentation image or prediction map. The chamfer distance between T and S is given by the average of distances from each non-zero pixel $x \in S$ to the closest non-zero pixels in the T,

$$d_{CD}(T, S) = \frac{1}{|S|} \sum_{x \in S, x > 0} \min_{t \in T, t > 0} d(t, x), \tag{3}$$

where $d(t, x)$ is Euclidean distance between x and t.

Based on chamfer distance, we propose to use a pixel-wise weighted cross entropy as lost function

$$\begin{aligned} L(x) = -\sum_{i=1}^{K} \Big(& d_i^I(x) \beta_i y_i(x) \log \hat{y}_i(x) \\ & + d_i^O(x)(1 - \beta_i)(1 - y_i(x)) \log(1 - \hat{y}_i(x)) \Big), \end{aligned} \tag{4}$$

where $d^I(x)$ and $d^O(x)$ represent the inner and outer Euclidean distance transform map of T respectively

$$\begin{aligned} d^I(x) &= \min_{t \in T, t = 0} d(t, x), \\ d^O(x) &= \min_{t \in T, t > 0} d(t, x). \end{aligned} \tag{5}$$

The lost function can be interpreted in this way:

a) $y_i(x) = 0$ and $\hat{y}_i(x) > 0$: The image is over-segmented for class i at pixel x. The lost should be higher when x is further away from the template boundary, i.e. higher $d^I(x)$ produces higher weight on this pixel.

b) $y_i(x) = 1$ and $\hat{y}_i(x) < 1$: The image is under-segmented for class i at pixel x. The lost should be higher when x is further away from the template boundary, i.e. higher $d^O(x)$ produces higher weight on this pixel.

III. Experiments

The network is implemented in Python with Tensorflow and Keras. All experiments are performed using an Intel® Xeon® E5-2630 2.20GHz CPU with 64GB RAM and two Nvidia Titan Xp GPU. The batch size for each iteration is 4 and the number of epochs is 10 for both datasets.

Two public datasets of crack images and manually labeled ground truth are downloaded to evaluate the performance of our crack detection algorithm. The parameter setting in data augmentation is: rotation_range(degree)=0.2, width_shift_range(fraction of total width)=0.05, height_shift_range(fraction of total width)=0.05, shear_range=0.05, zoom_range=0.05, horizontal_flip=True. We use 3-fold cross validation to split the dataset into training and testing images. For each training image, 30 new images will be generated randomly by data augmentation and added to training images. The details on the training and testing images are given in Table I.

TABLE I
Training and testing dataset in 3-fold cross validation .

Dataset	CFD	AigleRN
Image dimensions	480×320	991×462 311×462
Total number of images	118	38
Training images	78~79	25~26
Training images after augmentation	2415~2448	773~804
Testing images	39~40	12~13

We compare our result from 3-fold cross validation with those from the state-of-the-art road crack detection methods (CrackTree [7], CrackIT [8], Minimal Path Selection (MPS) [10], and CrackForest [1]). We use Precision, Recall and F_1 Score (F_1) to measure accuracy of crack segmentation, which are defined in (6)-(8). TP, FP, and FN are the numbers of true positive, false positive, and false negative pixels respectively. Detected pixels which are no more than five pixels away from the manually labeled pixels are considered true positive pixels (we use the same setting as the experiments in [1]).

$$Precision = \frac{TP}{TP + FP} \tag{6}$$

$$Recall = \frac{TP}{TP + FN} \tag{7}$$

$$F_1 = \frac{2 * TP}{2 * TP + FP + FN} \tag{8}$$

A. CrackForest Dataset (CFD)

The CrackForest dataset consists of 118 images of cracks on urban road surface in Beijing, each of size 480×320 pixels. Images are resized to 512×512 before input to U-Net and the prediction map output by U-Net are resized to 480×320 before evaluation of segmentation accuracy. To generate the final binary segmentation, the threshold on the prediction map is determined so that the F_1 Score is maximized. For this dataset, the threshold is 0.63.

Examples of original image and segmentation map are displayed in Fig. 2. It can be observed that cracks with complex morphology can be detected with details preserved and false detections are very few. The comparison result in Table II shows that the proposed method outperforms other methods by 6-9% in accuracy at least.

TABLE II
Pixel level crack segmentation accuracy on CFD.

Method	Precision	Recall	F_1
CrackTree	73.22%	76.45%	74.80%
CrackIT	67.23%	76.69%	71.64%
CrackForest (SVM)	82.28%	89.44%	85.71%
U-Net	92.12%	95.70%	93.88%

B. AigleRN Dataset

The AigleRN dataset comprises of 38 images of French pavement surface [11]. Half of them are 991×462 pixels in dimension and half of them are 311×462. For this dataset, the threshold on the prediction map is 0.55.

Examples of original image and segmentation map are displayed in Fig. 3. The segmentation problem is more challenging in this dataset because the background is roughly textured with the presence of dirt and oil stain. The segmentation result indicates that our method has high tolerance for noise and we achieve the highest accuracy among all methods as shown in Table III.

TABLE III
Pixel level crack segmentation accuracy on AigleRN.

Method	Precision	Recall	F_1
CrackIT	76.84%	74.32%	76.56%
MPS	86.66%	90.06%	88.33%
CrackForest (SVM)	90.28%	86.58%	88.39%
U-Net	92.02%	93.21%	92.61%

C. Multiscale Test

In this test, we halve the length and width of the four images displayed in Figs. 2(a)-2(d) and piece together into a single mosaic image Fig. 4(a). Similar, we randomly selected sixteen images from CFD, resize them in quarter and piece them together into Fig. 4(b). The accuracy of the two test images are shown in Table IV. The segmentation results show that U-Net can tolerant a certain level of scale change. When the test images are resized in half, the F_1 score of U-Net (86.09%) is still better than other segmentation methods. When the test images are resized in quarter, the accuracy deteriorates drastically due to loss of details in the output.

TABLE IV
Multiscale test on CFD.

Test image	Precision	Recall	F_1
Fig. 4(a)	88.80%	83.54%	86.09%
Fig. 4(b)	49.06%	65.09%	55.95%

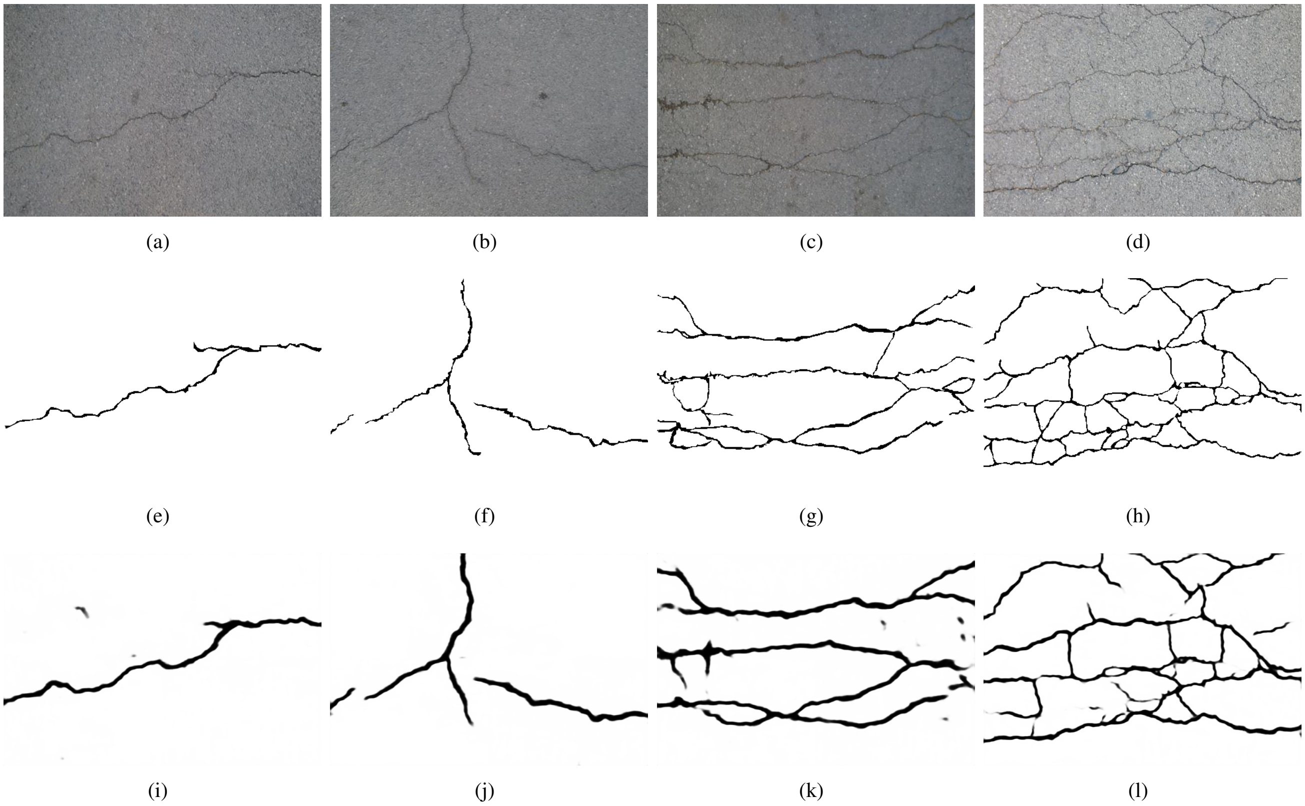


Fig. 2. (a)-(d) Original image of road surface cracks in CFD. (e)-(h) Ground truth. (i)-(l) Segmented images.

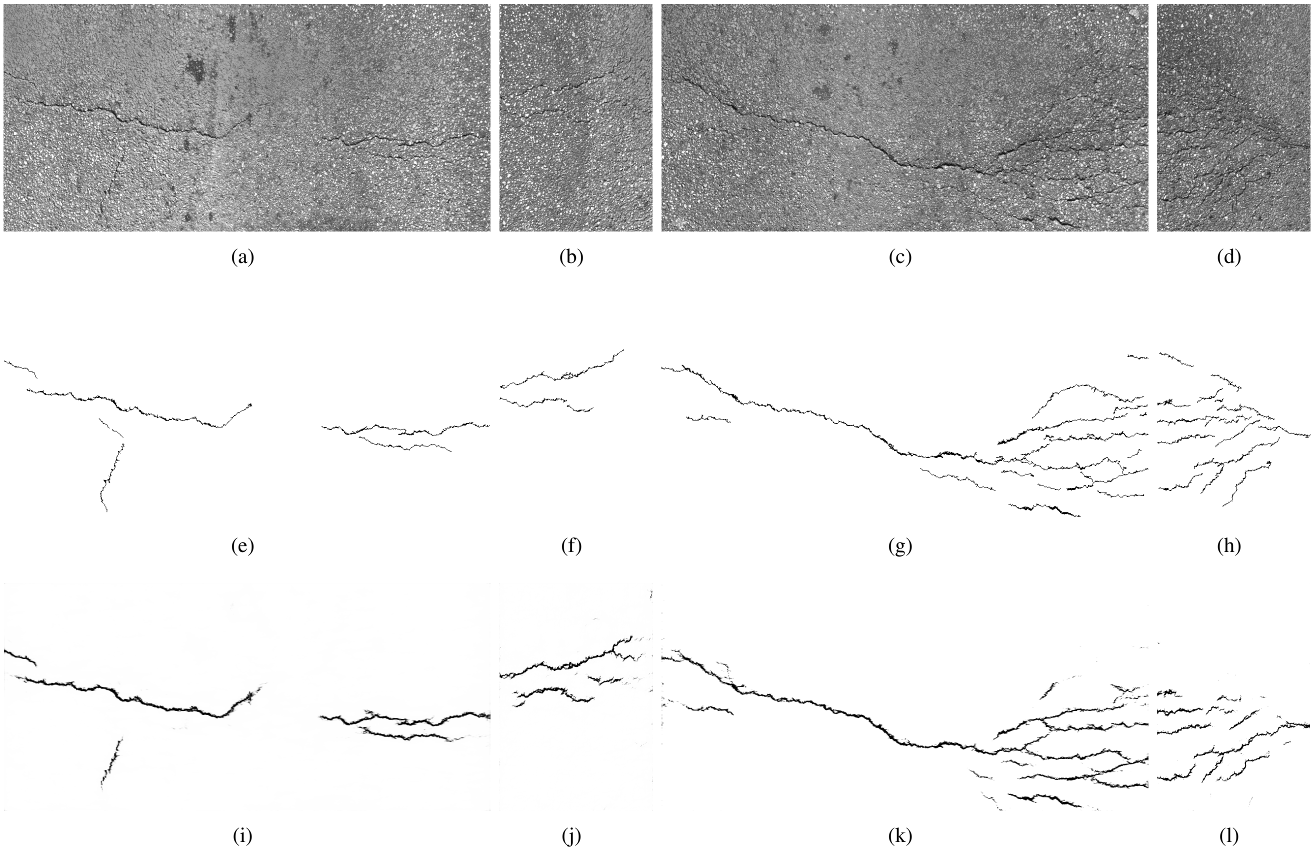
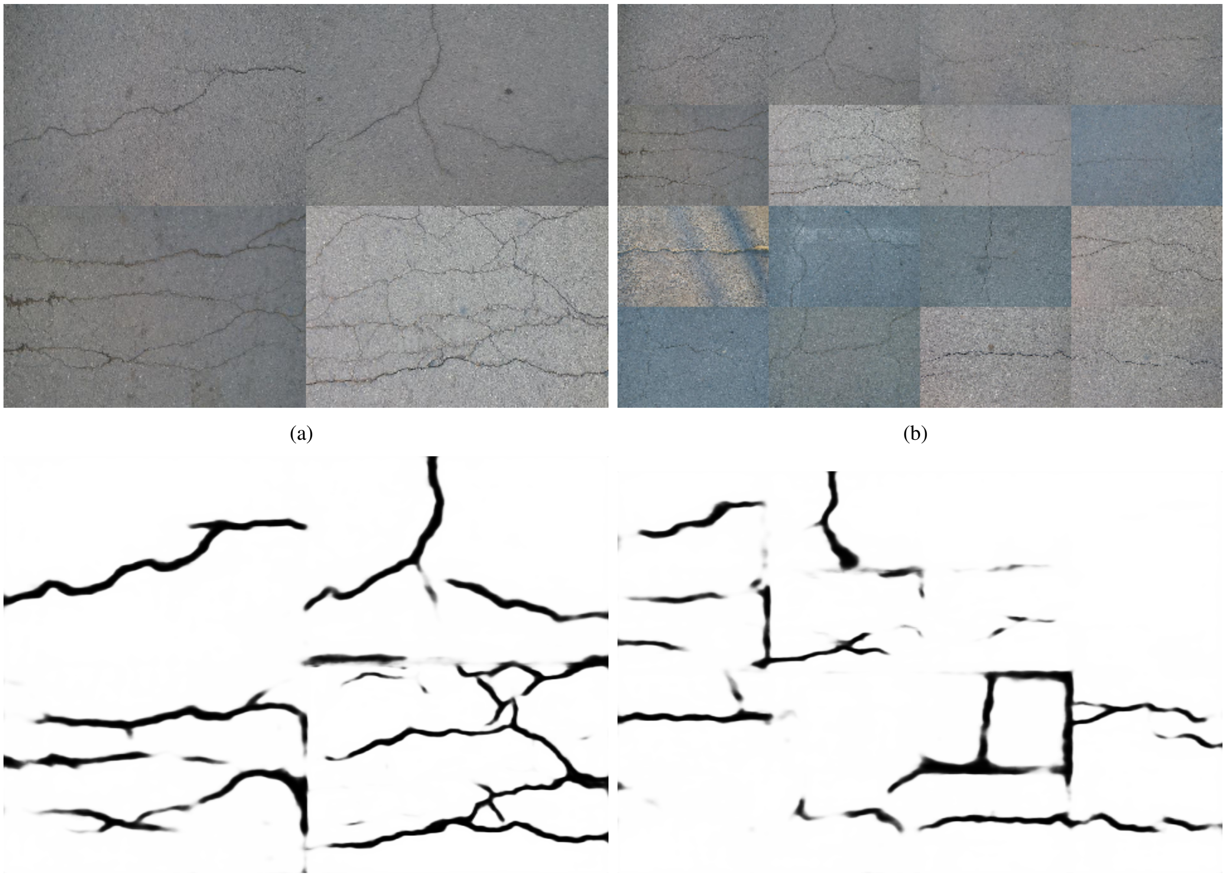


Fig. 3. (a)-(d) Original image of road surface cracks in AigleRN. (e)-(h) Ground truth. (i)-(l) Segmented images.



(a)

(b)

(c)

(d)

Fig. 4. (a) Mosaic of four half scaled images. (b) Mosaic of sixteen quarter scaled images. (c) Segmentation result of (a). (d) Segmentation result of (b).

IV. Conclusion

We proposed an automatic crack detection method based on deep convolutional neural network –U-Net. To the best of our knowledge, this is the first study that uses deep learning based method to process an image as a whole and generate a crack segmentation directly from the neural network without patchifying it. A new lost function based on distance transform is proposed and achieves outstanding pixel-level accuracy.

References

[1] Y. Shi, L. Cui, Z. Qi, F. Meng, and Z. Chen, "Automatic Road Crack Detection Using Random Structured Forests," IEEE Transactions on Intelligent Transportation Systems, Vol. 17, No. 12, December 2016.

[2] https://github.com/cuilimeng/CrackForest-dataset.

[3] P. Dollr and L. Zitnick, Fast Edge Detection Using Structured Forests, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 37, No. 8, December 2015.

[4] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," Medical Image Computing and Computer-Assisted Intervention (MICCAI), Springer, LNCS, Vol.9351: 234–241, 2015.

[5] S. J. Schmugge, L. Rice, J. Lindberg, R. Grizzi, C. Joffe, and M. C. Shin, "Crack Segmentation by Leveraging Multiple Frames of Varying Illumination," IEEE Winter Conference on Applications of Computer Vision (WACV), March 2017.

[6] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 39, No. 12, December 2017.

[7] Q. Zou, Y. Cao, Q. Li, Q.Mao, and S.Wang, "CrackTree: Automatic crack detection from pavement images," Pattern Recognit. Lett., vol. 33, no. 3, pp. 227238, Feb. 2012.

[8] H. Oliveira and P. L. Correia, "CrackIT−An image processing toolbox for crack detection and characterization," IEEE International Conference on Image Processing, Oct. 2014.

[9] L. Zhang, F. Yang, Y. Zhang, Y. Zhu, "Road crack detection using deep convolutional neural network," IEEE International Conference on Image Processing, Sept. 2016.

[10] R. Amhaz, S. Chambon, J. Idier, and V. Baltazart, "Automatic crack detection on 2D pavement images: An algorithm based on minimal path selection," IEEE Transactions on Intelligent Transportation Systems, Vol. 17, No. 10, Oct. 2016.

[11] S. Chambon, AigleRN. [Online]. Available: http://www.irit.fr/ Sylvie.Chambon/Crack_Detection_Database.html.

[12] H. Barrow, J. Tenenbaum, R. Bolles, and H. Wolf "Parametric correspondence and chamfer matching: two new techniques for image matching", 5th International Joint Conference on Articial Intelligence, vol. 2, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1977), pp. 659-663.