Automated Road Crack Detection Using Deep Convolutional Neural Networks

Conference Paper · December 2018

DOI: 10.1109/BigData.2018.8622327

Vishal Mandal
WSP USA
9 PUBLICATIONS **30** CITATIONS
SEE PROFILE

Yaw Adu-Gyamfi
University of Missouri
41 PUBLICATIONS **179** CITATIONS
SEE PROFILE

Automated Road Crack Detection Using Deep Convolutional Neural Networks

Vishal Mandal
Department of Civil and Environmental Engineering
University of Missouri-Columbia
Columbia, MO, USA
vmghv@mail.missouri.edu

Lan Uong
Department of Civil and Environmental Engineering
University of Missouri-Columbia
Columbia, MO, USA
lpu5xc@mail.missouri.edu

Yaw Adu-Gyamfi
Department of Civil and Environmental Engineering
University of Missouri-Columbia
Columbia, MO, USA
adugyamfiy@missouri.edu

***Abstract*— Effective and timely identification of cracks on the roads are crucial to propitiously repair and limit any further degradation. Till date, most crack detection methods follow a manual inspection approach as opposed to automatic image-based detection, making the overall procedure expensive and time-consuming. In this study, we propose an automated pavement distress analysis system based on the YOLO v2 deep learning framework. The system is trained using 7,240 images acquired from mobile cameras and tested on 1,813 road images. The detection and classification accuracy of the proposed distress analyzer is measured using the average F1 score obtained from the precision and recall values. Successful application of this study can help identify road anomalies in need of urgent repair, thereby facilitating a much better civil infrastructure monitoring system. The codes associated with the study including the trained model can be found in [11].**

Keywords—Deep Convolutional Neural Network, Road Crack Detection, Artificial Intelligence

I. INTRODUCTION

Maintaining good condition of road is pivotal to safe driving and is one of the major mandates of transportation and regulatory maintenance authorities. One important component of maintaining a safer driving environment is to effectively monitor road surface degradation which is labor intensive and requires domain expertise. Also, the maintenance and rehabilitation of road surfaces necessitates having accurate information about the road crack type. It therefore, becomes extremely difficult to monitor road cracks manually and hence, the need to successfully implement computer vision and machine learning algorithms is of paramount importance. There has been growing interest in this field and recently a number of deep learning algorithms have been applied at automatically studying road surface quality [1-4]. In our study, we focus on detecting road surface cracks using deep learning that would help facilitate road monitoring activity without the aid of human operators. Until now, human visual inspection of road surfaces has been the most usual method of evaluating road quality. The overall procedure is expensive and extremely time-consuming. Also, the results do not exactly conform to the standards set by the regulating bodies as the nature of detections vary from the skill of one operator to another. Therefore, in order to ensure uniformity and reliability, a robust system to automatically predict road irregularities is proposed in this paper.

Several researchers in the past have proposed their work on detecting road cracks using deep learning. Taking advantage of the recent successes on applying deep learning to computer vision problems, [5] used a deep learning based model for road crack detection. In [6], a novel local binary pattern (LBP) based operator for pavement crack detection is proposed whereas a crack detection method using Gabor filter is presented in [2]. An automatic crack detection method referred to as CrackTree is developed in [7] where cracks are detected from pavement images. An important point to consider while applying deep learning techniques for successful crack detection is to focus on discriminative and representative deep features. Zhang et al. [5] developed a crack detection method where the discriminative features are learned from raw image patches using ConvNets. Likewise, [8] proposed a deep architecture of CNNs for detecting concrete cracks. An integrated model for crack detection and characterization is presented in [3] and a complete set of image processing algorithms aimed at detecting and characterizing road surface crack distresses is described in [4]. In the current study, we implement a model trained on YOLO v2 algorithm by using the road crack images obtained from [9].

II. DATA DESCRIPTION

The dataset consisted of 9,053 road images photographed from a vehicle mounted smartphone. Images were captured from 7 local governments across Japan (Ichihara, Chiba, Nagakute, Sumida, Muroran, Adachi and Numazu). Out of the acquired sample of 9,053 images, 7,240 and 1,813 image sets were used as training and testing samples respectively. The road images consist of different crack types. These cracks were divided into eight different categories as shown in Table 1. Each crack type is represented by a class name such as D00, D01, D11, etc.

The number of images corresponding to a certain crack type is shown in Figure 1.

TABLE 1. ROAD CRACK TYPE INCLUDING THE CLASS NAME [12]

Crack Type			Detail	Class name
Crack	Linear Crack	Longitudinal	Wheel mark part	D00
			Construction joint mark	D01
		Lateral	Equal interval	D10
			Construction joint mark	D11
	Alligator Crack		Partial pavement, overall pavement	D20
Other Corruption			Rutting, bump, pothole, separation	D40
			Cross walk blur	D43
			White line blur	D44

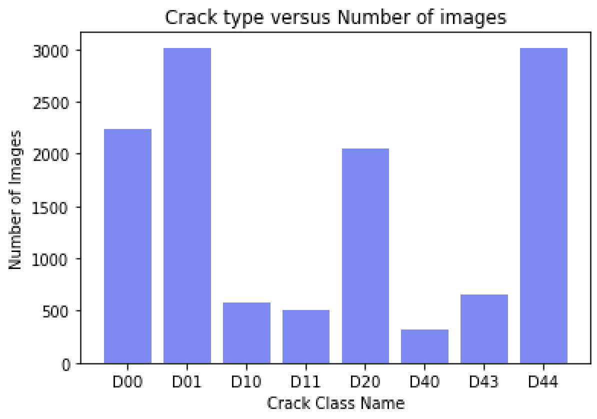

Figure 1. Crack type versus number of images

III. PROPOSED METHODOLOGY

Given a road image, the objective of our method is to automatically detect any crack and assign their corresponding class name. To attain this, we implemented a deep convolutional neural network based on the YOLO v2 framework. NVIDIA GTX 1080Ti GPU was used to effectively handle memory and speed requirement for model training and inferencing. The training time for YOLO v2 was 18.2 hours. The trained model was tested on a selected database of road crack images comprising of 1,813 images to evaluate their performance. Detailed description of YOLO v2 algorithm is provided as follows.

YOLO v2

You look only once (YOLO) is the state of the art object detection algorithm that can attain high mean average precision and speed. Unlike the traditional classifier systems, YOLO looks into the image only once and is able to detect objects in it. Most current day, object detection algorithms repurpose CNN classifiers and conduct detections. In order to perform any object detection, the algorithm uses a classifier for that object and tests it at varied locations and scales in the test image. YOLO reframes object detection; rather than looking at a single image a thousand times, it just looks at the image once and correctly performs object detections. Just by using a single CNN, it can simultaneously predict multiple bounding boxes and class probabilities for the boxes so generated. Due to the presence of this feature, this algorithm is extremely fast and can be easily implemented to different scenarios. The CNN architecture used for training the distress analysis system is shown in Table 2. The model uses standard layer types such as convolutional with a 3 × 3 kernel and max pooling with a 2 × 2 kernel. The last convolutional layer has a 1 × 1 kernel that helps minimize data to the shape 13 × 13 × 125. This 13 × 13 structure is the size of grid where the image gets apportioned. For every grid cell, there are 35 channels predictions. All these grid cells predict 5 bounding boxes and those boxes are described by seven data elements: the values of x, y, width, and height for the bounding box's rectangle; the confidence score; road crack and no-crack probability distribution.

TABLE 2. YOLO v2 MODEL ARCHITECTURE

Layer	Kernel	Stride	Output Shape
Input			[416, 416, 3]
Convolution	3×3	1	[416, 416, 16]
Max Pooling	2×2	2	[208, 208, 16]
Convolution	3×3	1	[208, 208, 32]
Max Pooling	2×2	2	[104, 104, 32]
Convolution	3×3	1	[104, 104, 64]
Max Pooling	2×2	2	[52, 52, 64]
Convolution	3×3	1	[52, 52, 128]
Max Pooling	2×2	2	[26, 26, 128]
Convolution	3×3	1	[26, 26, 256]
Max Pooling	2×2	2	[13, 13, 256]
Convolution	3×3	1	[13, 13, 512]
Max Pooling	2×2	1	[13, 13, 512]
Convolution	3×3	1	[13, 13, 1024]
Convolution	3×3	1	[13, 13, 1024]
Convolution	1×1	1	[13, 13, 35]

In order to speed up training and improve the performance of YOLO, transfer learning was used. Transfer learning has been widely applied in deep learning and computer vision applications. A new task using this technique can benefit extensively from previously well-trained models. The Microsoft COCO dataset contains more than 2 million well-labeled objects of 80 different categories in more than 300,000 images. We used weights pre-trained on COCO dataset as the initialization of this detection task. Multiple transfer learning strategies were tested and eventually we choose to freeze the weights of ResNet-101 and RPN throughout the training and fine tune the location-sensitive score maps and the output layer. In the current study, we observe inference speed for YOLO to be 60 frames per second and the size of the trained model to be 264.4 Megabytes. The codes associated with the study including the trained model is included in [11].

IV. RESULTS

We evaluated the performance of our proposed YOLO v2 model on a set of 1,813 road images. To improve and assess the model's accuracy, YOLO v2 was trained for 40,000 iterations. The learning rate for YOLO v2 was 0.01. It was observed that the model trained for 40,000 iterations performed well and almost identical to the ones trained in between 40,000-60,000 iterations. For evaluating crack detections, a prediction box that captures over 50% Intersection over Union (IoU) in the area with the ground truth box is termed a successful match. In our study, evaluation metric focused on comparing the ground truth Mean F-1 score as shown in equation (1). Here, F-1 score measures accuracy using the statistics precision and recall values. Precision, shown in equation (2) is defined as the ratio of true positives (tp) to all the predicted positives (tp+fp). Similarly, Recall as shown in equation (3) is the ratio of true positives to all the actual positives (tp+fn).

$$F1 = \frac{2*Precision*Recall}{Precision+Recall} \tag{1}$$

$$Precision = \frac{tp}{tp+fp} \tag{2}$$

$$Recall = \frac{tp}{tp+fn} \tag{3}$$

While testing, if prediction bounding box has over 50% IoU with the ground truth box, then that particular crack is classified as a True Positive (tp). Likewise, if any predicted bounding box fails to have a 50% IoU with the ground truth box, then it is classified as false positive (fp). If a detection had 50% overlap with the ground truth but wrong classification, it was counted as a false positive. The false negative category referred to cases where there are no predictions whatsoever. Figure 2 shows samples of detection and classification results obtained using the methodology implemented. The green bounding box corresponds to the ground truth value whereas a blue bounding box represents the predicted value so generated by the model.

although the model failed to detect a crack on the left hand side of the image, it does in fact predict the crack on the right hand side that was left unlabeled. Therefore, images not labeled correctly in some of the cases resulted in misclassification. In Figure 2(c), YOLO was unable to predict any crack due to the presence of over-lapping crack bounding boxes and that confused the model. The Precision, Recall and F1 values obtained from the model is shown in Table 3.

TABLE 3. PRECISION, RECALL AND F1 VALUES OBTAINED FROM YOLO v2

Model	Precision	Recall	F1
YOLO v2	0.8851	0.8710	0.8780

As observed in table 3, YOLO v2 achieved an F1 score of 0.8780 for distress detection. It is important to note that this F1 score was tabulated based on the obtained crack predictions only and did not include any of the crack class type. When predicting cracks alongside its corresponding class type, the overall F1 score drops to 0.7394 which is lower than the value obtained earlier. Table 4 shows detection and classification accuracies for each of the crack type in our image database. The system is able to analyze alligator cracks and potholes (D20 and D40) at a much higher accuracy compared to all other distress types. Transverse cracks (D10 and D11) were the most challenging for the system with F1 scores of 0.7137 and 0.6885. From figure 1, the number of transverse cracks in the training database was relatively lower than other distress types. This could have attributed to the low F1 scores for this category.

Several deep learning frameworks were investigated: Yolo v3, single shot multibox detector (SSD), and region-based convolutional network (faster RCNN). All of these frameworks fell short of Yolo v2 by at least 5% in overall F1-score rate.

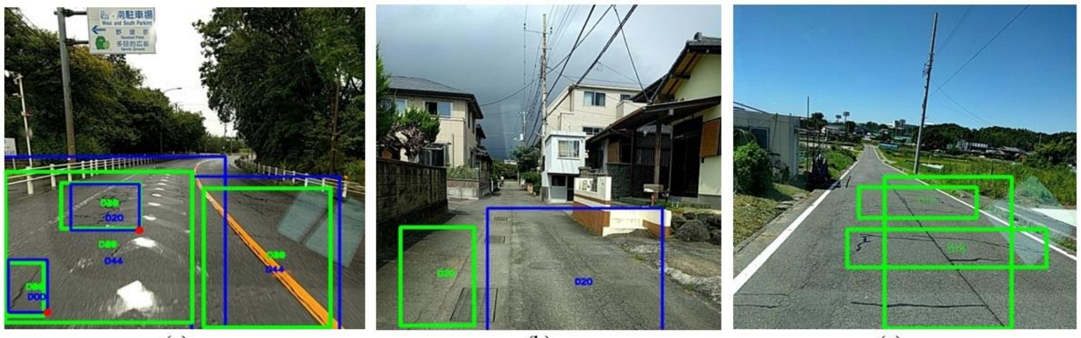

Figure 2. Classification of predicted crack examples: True Positive-(a), False Positive-(b), False Negative-(c) obtained from YOLO v2.

In Figure 2(a) distresses were accurately detected and classified with over 50% IoU with the ground truth box for each crack class and thus, classified as true positive. Similarly, Figure 2(b) is classified as false positive. For Figure 2 (b),

Removing the third convolutional and pooling layer in the original Yolo v2 architecture improved the accuracy of the system by 3%. Varying the batch size and learning rates of the model yielded no significant improvement. The number of iterations used was a key factor. The model accuracy increased on average by 4% per every 5,000 iterations. After 60,000 iterations, the net gain in model accuracy was insignificant.

TABLE 4. DETECTION AND CLASSIFICATION RESULTS FOR EACH CARCK TYPE

Crack Class name	Precision	Recall	F1
D00	0.7664	0.7536	0.7598
D01	0.7801	0.7381	0.7584
D10	0.7818	0.6571	0.7137
D11	0.718	0.6615	0.6885
D20	0.7865	0.7638	0.7749
D40	0.8	0.7535	0.7760
D43	0.7697	0.7697	0.7696
D44	0.749	0.7444	0.7466

V. CONCLUSION

The rapid advancement in the field of machine learning and high-performance computing has highly augmented the range of automatic prediction systems. In the current study, we implemented a deep learning framework to automatically detect and classify different types of cracks. The F1 score for detection without predicting crack class was found out to be 0.8780 and for classification including crack class was 0.7394. It was observed that the distress analyzer so developed is more accurate in evaluating alligator cracks but struggles with transverse cracks. This might be due to the insufficient number of images with transverse cracks in the training database. The disparity in accurately detecting cracks was mainly due to the incorrect labeling of cracks in the image, lower distinction between the crack and background image and shadow formation.

To further improve the training dataset, Google street-view images could be a good source for increasing the database with rich crack information for building much robust models. It provides high resolution images from different camera views (both top-down and side views) of the pavement. Google provides an API that can be used to download these images without cost.

REFERENCES

[1] A. Zhang, K.C. Wang, B. Li, E. Yang, X. Dai, Y. Peng, ... and C. Chen, "Automated pixel-level pavement crack detection on 3D asphalt surfaces using a deep-learning network," Computer-Aided Civil and Infrastructure Engineering, 32(10), pp. 805-819, 2017.

[2] M. Salman, S. Mathavan, K. Kamal, and M. Rahman, "Pavement crack detection using the gabor filter," in Proceedings of IEEE International Conference on Intelligent Transportation Systems, Oct. 2013, pp. 2039–2044.

[3] 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.

[4] H. Oliveira and P.L. Correia, "Crackit-an image processing toolbox for crack detection and characterization," in Proceedings of IEEE International Conference on Image Processing, Oct. 2014, pp. 798–802.

[5] L. Zhang, F. Yang, Y. D. Zhang and Y.J. Zhu, "Road crack detection using deep convolutional neural network," In Image Processing (ICIP), IEEE International Conference, pp. 3708-3712, September 2016.

[6] Y. Hu and C. X. Zhao, "A novel LBP based methods for pavement crack detection," Journal of pattern Recognition research, 5(1), pp. 140-147, 2010.

[7] Q. Zou, Y. Cao, Q. Li, Q. Mao and S. Wang, "CrackTree: Automatic crack detection from pavement images," Pattern Recognition Letters, 33(3), pp. 227-238, 2012.

[8] Y. J. Cha, W. Choi, and O. Büyüköztürk, "Deep learning-based crack damage detection using convolutional neural networks," Computer-Aided Civil and Infrastructure Engineering, 32(5), pp. 361-378, 2017.

[9] H. Maeda, Y. Sekimoto, T. Seto, T. Kashiyama and H. Omata,"Road Damage Detection and Classification Using Deep Neural Networks with Smartphone Images," Computer-Aided Civil and Infrastructure Engineering, June 2018.

[10] J. Redmon and A. Farhadi, "YOLO9000: better, faster, stronger", arXiv preprint, 2017.

[11] https://github.com/TITAN-lab/Road-crack-detection

[12] http://www.mlit.go.jp/road/road_e/pdf/RoadMaintenance.pdf