

Crack Detection using Faster R-CNN and Point Feature Matching



Yahya Mohammed M¹, Nasim Uddin², Chenjun Tan³ and Zhenhua Shi³

¹Department of Civil Construction and Environmental Engineering, University of Alabama at Birmingham, USA

²Department of Civil Construction and Environmental Engineering, University of Alabama at Birmingham, USA

³Department of Civil Construction and Environmental Engineering, University of Alabama at Birmingham, USA

Submission: July 27, 2020; **Published:** August 14, 2020

***Corresponding author:** Yahya Mohammed M, Department of Civil Construction and Environmental Engineering, University of Alabama at Birmingham, USA

Abstract

The detection of cracks on concrete surfaces is the most important step during the inspection of concrete structures. Inspection technology utilizing drones with image processing technique has recently been applied to crack assessment to overcome the drawbacks of visual inspection. However, identification of crack location requires watching the inspection videos to exactly locate cracks. This paper proposes a fast and easy method for cracks detection which provides the inspector with photos for the inspected structure that show the crack locations, and the inspector does not need to watch any inspection video. In the beginning, the drone will take a photo for the inspection element or region (Target Image). Then the drone will start searching for cracks using Faster Region- Convolution Natural network (Faster R-CNN) algorithm which allow for real time crack detection. When drone catches crack it will take a photo for this crack which will be called "reference image". Point Feature matching algorithm is applied to locate the crack image "reference image" on the element image "target image". This is achieved by matching the features points of the crack photo (reference image) with the feature's points of the element photo (Target Image).

Keywords: R-CN; Crack, Detection; Point Feature Matching; Image Processing

Introduction

The detection of cracks on concrete surfaces is the most important step during the inspection, diagnosis, and maintenance. In the past, crack detection was performed manually by experienced human inspectors which called visual inspection [1-2]; as a result, the detection methods are expensive and subjective. Monitoring using sensors require a continuous power supply, which is not available in all structures like bridges. Because automation of visual inspections can address the limitations of the ordinary human-oriented visual approach, researchers have been attracted to the development of computer vision-based methods for structural damage detection. Various image processing techniques have been used over the past few decades for crack detection until it reaches real time detection techniques. Edge detection techniques were applied to provide a boundary between a crack and the image background. Edge detection techniques include gradient algorithm, Laplacian algorithm, and Canny algorithm [3-6]. However, these methods were sensitive to the environment noise and cannot remove the spots and noises [7]. Marques [8] used the average grayscale of the image

for the threshold instead of using the same fixed value of all pavement images, but this method has the poor adaptability to the environment. These image-processing techniques (IPTs) are applied to detect different type of damage, such as concrete cracks [3 & 9] pavement cracks [10-11], steel cracks [12], loosened bolts [13-14]. These methods still require the pre and post processing techniques, which are time consuming and can detect only one damage type.

Recently, Convolutional Neural Networks (CNNs) have been proven to achieve a state-of-the-art performance in many computer vision tasks such as image classification [15-16], object detection [17] or semantic segmentation [18]. One of the every important and highly successful frameworks for generic object detection is driven by the success of region proposal methods (RPN) [19] and region-based convolutional neural networks (R-CNNs) [20], which is a kind of CNN extension for solving the object detection tasks. Since then, great improvements to R-CNN have been released, both in terms of accuracy and speed [21]. Ren et al. introduced Faster R-CNN [17], by combining RPN and

Fast R-CNN which introduced by Girshick [22]. In Faster R-CNN, the RPN shares feature with the object detection network in [22] to simultaneously learn prominent object proposals and their associated class probabilities. Cha [23] used the Faster R-CNN to provide quasi real- time simultaneous detection of multiple types of damages, which shows less influenced by the noise caused by lighting, shadow casting, blur, and so on.

In this article, the Faster R-CNN algorithm will be used for real time crack detection. This algorithm will allow the drone to take photos whenever crack is detected and store these photos (references images). The drone will take another photo that shows the whole inspected element (Target Image). Then the Point Feature Matching technique will be used to locate the references images in the target image. His will be achieved by matching the features between each reference image and the target image. The final product from this study will be one photo for each element that shows cracks locations if found. This article is organized as the follows. In Section 2, an overview of the Faster R-CNN algorithm which will be used for crack detection. In Section 3, the procedure for Point Feature Matching technique will be discussed. In Section 4, the results of the proposed method. Finally, the article ends with the conclusion in Section 5.

Faster R-CNN

Convolutional Neural Network (CNN)

Recently, algorithms based on convolutional neural network (CNN) have led to dramatic advance in state-of-the-art for fundamental problems in computer vision, such as object classification, object detection, object localization, semantic segmentation, and object instance segmentation [17,22,24-25]. This has led to increased interest in the applicability of CNN based method for solving the problem in civil engineering. As a typical machine/deep learning method, CNN is specific to image

recognition. By convolution, it is possible to extract any features in an area rather than at a single point. (Figure 1) shows the main architectures of CNN, which mainly includes convolution layer, activate the layer, pooling layer, fully connected layer and softmax layer. The convolution layer is to perform the convolution of the feature. The pattern in the images can be detected by the convolution of the feature, where is automatically acquired by training/learning. The output of convolution layer is called feature map, which means a mapping of features from input space to a reproducing a specific observations space (kernel Hilbert space), where usually it is a very high dimension, or even infinite dimension. Rectified linear unit (RELU) allows for faster and more effective detection by mapping negative values to zero and maintaining positive values. This is sometimes referred to as activation, because only the activated features are carried forward into the next layer. The pooling layers serve to progressively reduce the spatial size of the representation, to reduce the number of parameters and amount of computation in the network, and hence to also control overfitting. The max pooling process will discard some information as to enhance the response of the filter at any position of the image and implement the universality of response to the features appearing in the image. In this process, a raster scanning \times pixel region is placed on the \times input image, and the maximum value from the area will be extracted by max pooling. (Figure 2) shows an example of using a scanning 2×2 -pixel region to be taken a 4×4 -pixel input image by max pooling. (Figure 2). Pooling layer. The next-to-last layer is a Fully Connected (FC) layer that outputs a vector of K dimensions where K is the number of classes that the network will be able to predict. This vector contains the probabilities for each class of any image being classified. The final layer of the CNN architecture uses a classification layer such as softmax to predict the class with the highest probability as the classification results.

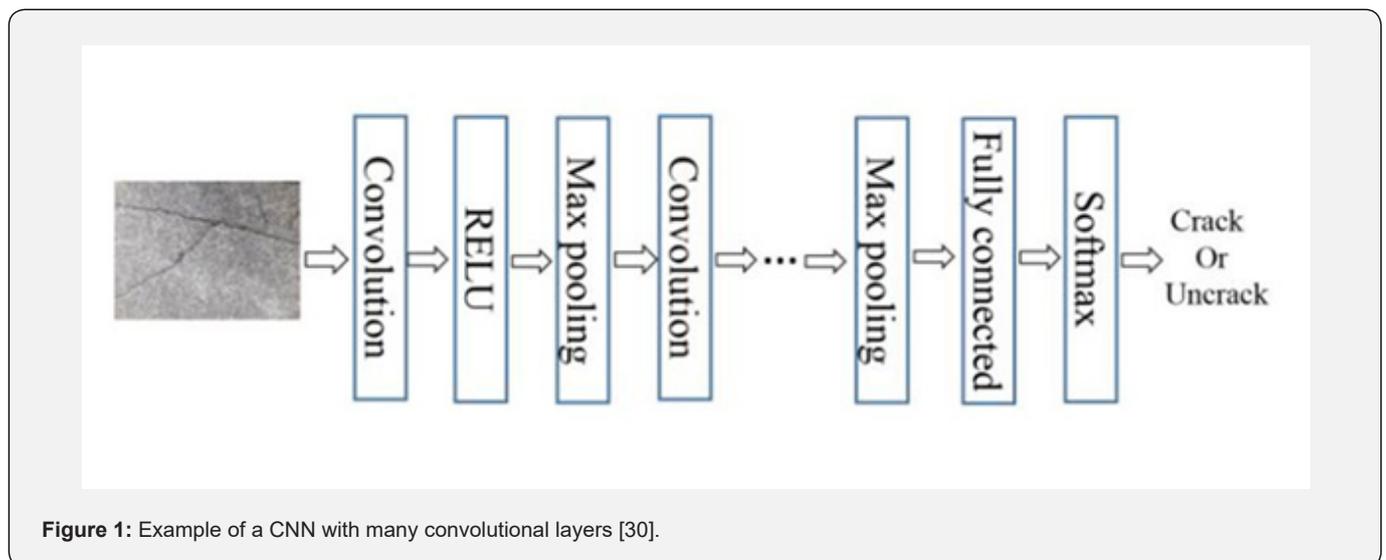


Figure 1: Example of a CNN with many convolutional layers [30].

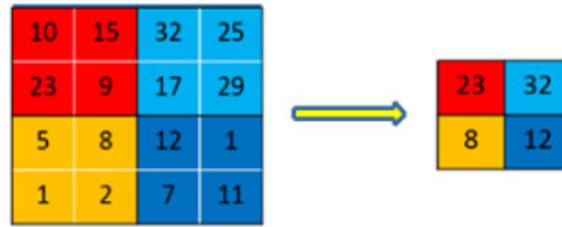


Figure 2: Pooling layer.

Regions with CNN (R-CNN)

R-CNN propose a bunch of boxes in the image and see if any of them actually correspond to an object or not. R-CNN creates these bounding boxes, or region of interest (ROI), using a process called Selective Search. The second step is the same as the CNN classification task, that is, the candidate region is input into the trained CNN model to calculate the category score so as to classify. In the third step, the target frame is further modified (translated, scaled) using the regression algorithm so as to accurately complete the target class positioning task. Although R-CNN can localize the interest object in images, it applies a region proposal method to extract ROI, which has notable drawbacks [22]: (1) training is a multi-stage pipeline; (2) training is expensive in space and time; (3) object detection is slow. Girshick [22], Ren et al. [17] and Lin et al. (2017) promote R-CNN to fast/faster R-CNN by using a lightweight CNN to propose ROI with a Region Proposal Network (RPN) instead of the selective search algorithm.

Fast R-CNN

To address the drawbacks of R-CNN, Girshick [22] introduced

Fast R-CNN. Fast R-CNN is trained end-to-end in one stage and shows higher speed and accuracy than the R-CNN. The Fast R-CNN takes precomputed object proposals from an external method (selective search) and uses CNNs to extract a features map from the input image. Features bound by object proposals are called a region of interest (RoI). In the RoI pooling layer (Figure 3), precomputed object proposals are overlaid on the feature map. RoI pooling takes RoIs and applies max pooling operation to extract a fixed-size feature vector from each RoI. These vectors are fed into FC layers, followed by two regression and softmax layers, to calculate the location of bounding boxes and classify objects in the boxes. Despite the better performance of Fast RCNN, it is also slow and has limited accuracy because generating object proposals through an external method (selective search) is time consuming and is not optimized during the training process. Fast R-CNN replaced the SVM classifier with a softmax layer on top of the CNN to output a classification. It also added a linear regression layer parallel to the softmax layer to output bounding box coordinates.

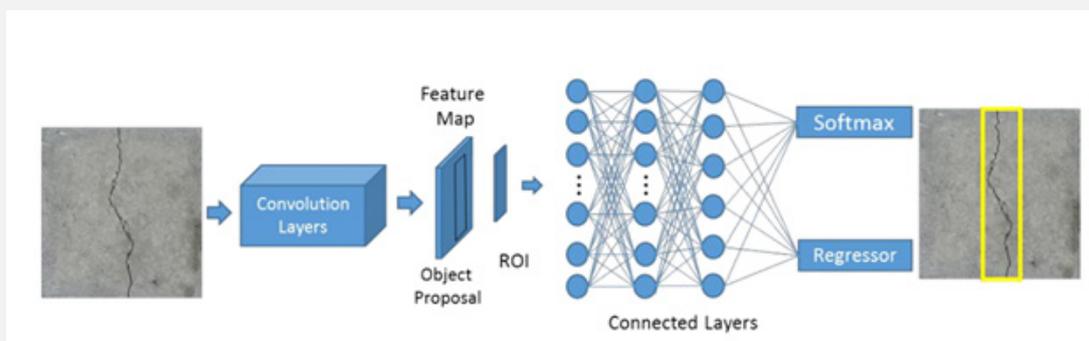


Figure 3: Fast R-CNN.

Faster R-CNN

The first step in object detection using Fast R-CNN is generating a bunch of potential bounding boxes or regions of interest (ROI). In Fast R-CNN, these proposals were created using selective

search, a fairly slow process that was found to be the bottleneck of the overall process. To overcome this problem, in Faster R-CNN, a lightweight CNN to propose ROI with a Region Proposal Network (RPN) instead of the selective search algorithm is used. RPN is a

neural network that scans the image in a sliding-window fashion and finds areas that contain objects, which simultaneously predicts object bounds and object scores at each position. The regions that the RPN scans over images are called anchors, which are boxes distributed over the image area. Practically, there are around 200K anchors of different sizes and aspect ratios that overlap to cover as much of the input image as possible. Although there are large numbers of anchors, the RPN can scan very fast. Since the sliding window is handled by the convolutional nature of

the RPN, which allows it to scan all regions in parallel (on a GPU). Further, the RPN does not scan over the image directly. Instead, the RPN scans over the backbone feature map. This allows the RPN to reuse the extracted features efficiently and avoid duplicate calculations. In other words, feature extraction is shared between Region Proposal Network (RPN) and Fast R-CNN which reduce the computational significantly. (Figure 4) Show shows the faster R-CNN algorithm architecture.

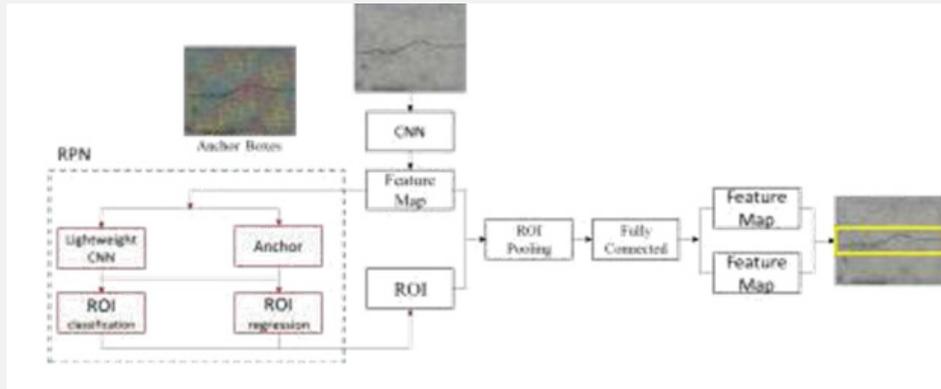


Figure 4: Faster R-CNN.

Point Feature Matching

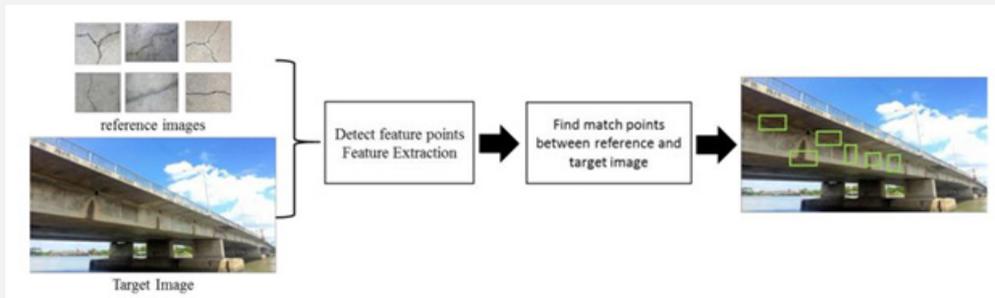


Figure 5: Point Feature Matching technique.

The feature is defined as an “interesting” part of an image such as a corner, blob, edge, or line, and are used as a starting point for many computer vision algorithms. The objective of the point features matching algorithm is to locate a specific object (reference image) inside the target image based on finding point correspondences between the reference and the target image. It can locate the reference images inside the target image despite a scale change or in-plane rotation for reference images. It is also robust to a small amount of out-of-plane rotation and occlusion. This method of object detection works best for objects that exhibit non-repeating texture patterns (like crack), which give rise to unique feature matches. (Figure 5) shows the basic steps for the point features matching algorithm. Speeded-Up

Robust Features (SURF) algorithm is used to achieve these goals. The SURF algorithm is developed by Bay [26,27] as a way for detection, description, and matching images features. For detection, SURF uses a blob detector based on the Hessian matrix to find points of interest. The determinant of the Hessian matrix is used as a measure of local change around the point and points are chosen where this determinant is maximal. To describe each feature, SURF summarizes the pixel information within a local neighborhood. The first step is determining an orientation for each feature, by convolving pixels in its neighborhood with the horizontal and the vertical Haar wavelet filters (Figure 6). These filters can be thought of as block-based methods to compute directional derivatives of the image’s intensity. By using intensity

changes to characterize orientation, this descriptor can describe features in the same manner regardless of the specific orientation of objects or of the camera. This rotational invariance property allows SURF features to accurately identify objects within images taken from different perspectives. For matching, by comparing the

descriptors obtained from different images, matching pairs can be found (Figure 7), and the reference image can be transformed into the coordinate system of the target image. The transformed image indicates the location of the object in the scene.

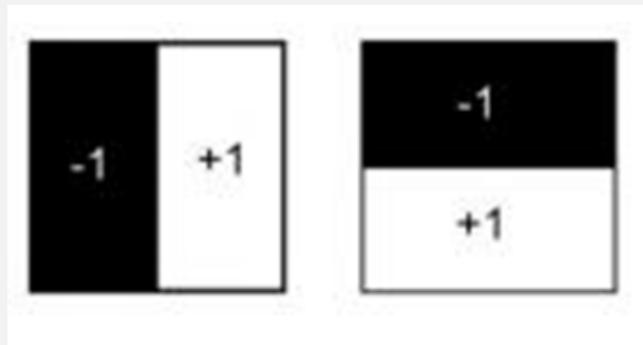


Figure 6: Horizontal and vertical wavelet filters [27].

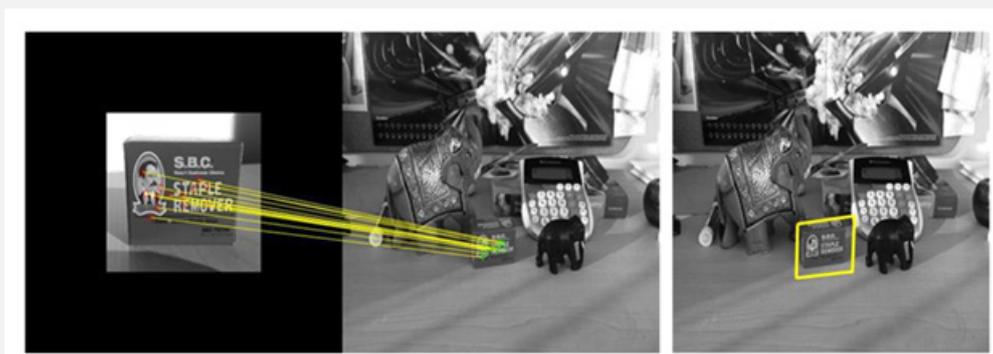


Figure 7: Features matching and detecting box.

Case Study

In this paper the faster R-CNN technique is adopted to find surface cracks from UAV input images. The training datasets have two parts; one is from an open resource [14], which has a total of 2000 images with 227 x 227 pixels with RGB channels; another is from an open resource [23], which consists 960 cracks images. Of that both datasets 90% are randomly chosen to training and remaining images are used to validation images. The training process was implemented on tensorflow [28]. We use a batch size of four per NVIDIA Tesla P100 16GB GPUs with four total GPUs. The popular deep CNN named ResNet-101(He et al. 2016) was applied as the backbone of faster R-CNN implementing in tensorflow. The final accuracy of this model on validation datasets is 89.28%. Then, this training crack detector was applied for recognizing cracks from images captured by the camera of UAV in real time. After finishing inspection, the drone will take a photo that shows the whole element. Then the Point Feature Matching algorithm is applied to locate the crack on the inspected part photo. The drone

used in this paper is called Crazyflie 2.0 manufactured by Bitcraze. Crazyflie 2.0 is a versatile open source flying development platform that only weighs 27g and fits in the palm of your hand. Crazyflie 2.0 is equipped with low latency/long-range radio as well as Bluetooth LE. The drone can be controlled by user mobile device or using a computer. The drone is equipped with high resolution FPV Camera which allowed capture pictures and record videos. The experiment was done in University of Alabama at Birmingham (UAB) structural lab. (Figure 8) shows the crack detection process for a concrete block that has a crack in the middle. Figure 8 (a, and b) show the drone and the done during inspection process, respectively. Figure 8 (c, d) show the crack (reference image) and the bridge photos (target image) respectively. Figure 8-e shows features matching, and Figure 8-f shows the box around the crack. (Figure 9) shows the same process for another block [29-32]. It should be noted that the crack picture and the block picture were taken from different view angel. The features extraction process that displayed in Figure 9 (c, d) helped the algorithm to match the two pictures.

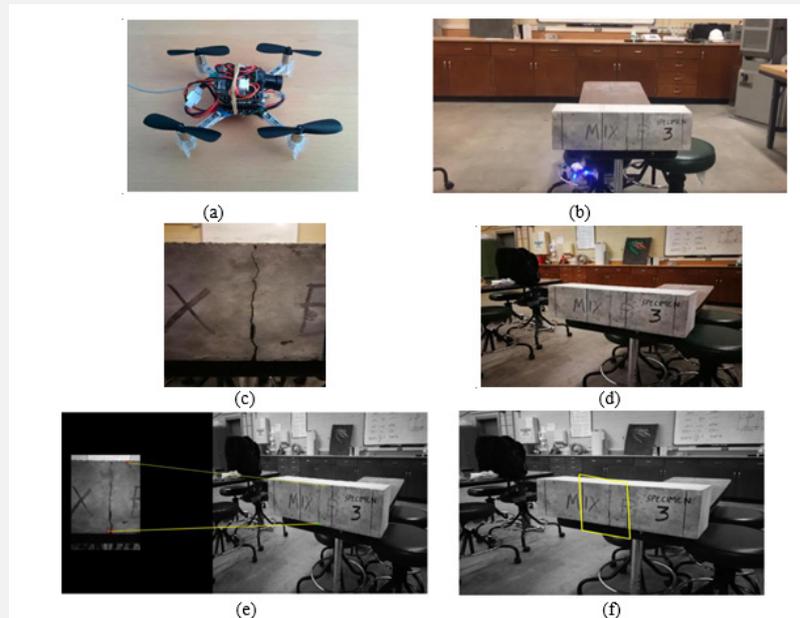


Figure 8: (a) Drone, (b) Drone captures pictures, (c) crack photo (d) beam photo, (e) matched points between the two photos, (f) detection box.



Figure 9: (a) crack photo, (b) photo for the concrete block, (c) 200 strongest feature points for a crack photo (d) 200 strongest feature points for block photo, (e) matched points between the two photos, (f) detection box.

Conclusion

In conclusion, this paper proposes a new method to detect cracks and display them in an easy way. The new method uses the drone as an inspection tool and Faster R-CNN (image processing

algorithm) for real time crack detection. In the beginning drone should take a photo for the inspection region, then Fast R-CNN will be applied to capture photos for cracks from the inspection live video if found. Another technique called Point Feature Matching

has been applied to match the crack photos with the inspection element photo. Finally, the inspector will receive one photo for each region or element that shows the cracks locations.

Acknowledgments

The authors would like to express their gratitude for the financial support received from the National Science Foundation (NSF-CNS- 1645863) and for this investigation.

References

1. Gattulli V, Chiamonte L (2005) Condition assessment by visual inspection for a bridge management system. *Computer-Aided Civil and Infrastructure Engineering* 20(2): 95-107.
2. Graybeal BA, Phares BM, Rolander DD, Moore M, Washer G (2002) Visual inspection of highway bridges. *Journal of nondestructive evaluation* 21(3): 67-83.
3. Abdel-Qader I, Abudayyeh O, Kelly ME (2003) Analysis of edge-detection techniques for crack identification in bridges. *Journal of Computing in Civil Engineering* 17(4): 255-263.
4. Nishikawa T, Yoshida J, Sugiyama T, Fujino Y (2012) Concrete crack detection by multiple sequential image filtering. *Computer-Aided Civil and Infrastructure Engineering* 27(1): 29-47.
5. Yamaguchi T, Hashimoto S (2010) Fast crack detection method for large-size concrete surface images using percolation-based image processing. *Machine Vision and Applications* 21(5): 797-809.
6. Zhao H, Qin G, Wang X (2010) Improvement of canny algorithm based on pavement edge detection. *Proc., Image and Signal Processing (CISP), 2010 3rd International Congress on, IEEE* pp: 964-967.
7. Lei B, Wang N, Xu P, Song G (2018) New Crack Detection Method for Bridge Inspection Using UAV Incorporating Image Processing. *Journal of Aerospace Engineering* 31(5): 04018058.
8. Marques O (2011) Practical image and video processing using MATLAB. John Wiley & Sons.
9. Nishikawa T, Yoshida J, Sugiyama T, Fujino Y (2012) Concrete crack detection by multiple sequential image filtering. *Computer-Aided Civil and Infrastructure Engineering* 27(1): 29-47.
10. Cord A, Chambon S (2012) Automatic road defect detection by textural pattern recognition based on AdaBoost. *Computer-Aided Civil and Infrastructure Engineering* 27(4): 244-259.
11. Ying L, Salari E (2010) Beamlet transform-based technique for pavement crack detection and classification. *Computer-Aided Civil and Infrastructure Engineering* 25(8): 572-580.
12. Yeum CM, Dyke SJ (2015) Vision-based automated crack detection for bridge inspection. *Computer-Aided Civil and Infrastructure Engineering* 30(10): 759-770.
13. Cha YJ, You K, Choi W (2016). Vision-based detection of loosened bolts using the Hough transform and support vector machines. *Automation in Construction*, 71: 181-188.
14. Özgenel ÇF (2018) Concrete Crack Images for Classification. Park J, Kim T, Kim J Image-based bolt-loosening detection technique of bolt joint in steel bridges. *Proc., 6th international conference on advances in experimental structural engineering, University of Illinois, Urbana-Champaign.*
15. Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. *Proc., Advances in neural information processing systems* pp: 1097-1105.
16. Simonyan, K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556.*
17. He K, Zhang X, Ren S, Sun J (2015) Deep residual learning for image recognition. *Proc., Proceedings of the IEEE conference on computer vision and pattern recognition* pp: 770-778.
18. Long, J, Shelhamer E, Darrell (2015) Fully convolutional networks for semantic segmentation. *Proc., Proceedings of the IEEE conference on computer vision and pattern recognition* pp: 3431-3440.
19. Uijlings JR, Van De Sande KE, Gevers T, Smeulders AW (2013) Selective search for object recognition. *International journal of computer vision* 104(2): 154-171.
20. Girshick R Fast R-CNN. *Proc Proceedings of the IEEE international conference on computer vision* pp: 1440-1448.
21. He K, Zhang X, Ren S, Sun J (2014) Spatial pyramid pooling in deep convolutional networks for visual recognition. *Proc., european conference on computer vision, Springer* pp: 346-361.
22. Girshick R (2015) Fast r-cnn. *arXiv preprint arXiv:1504.08083.*
23. Cha YJ, Choi W, Büyüköztürk O (2017) Deep Learning-Based Crack Damage Detection Using Convolutional Neural Networks. *Computer-Aided Civil and Infrastructure Engineering*, 32(5): 361-378.
24. He K, Gkioxari G, Dollár P, Girshick R (2017) Mask r-cnn *Proc, Computer Vision (ICCV) IEEE International Conference on, IEEE*, 2980-2988.
25. Johnson JW (2018) Adapting Mask-RCNN for Automatic Nucleus Segmentation. *arXiv preprint arXiv:1805.00500.*
26. Bay H, Ess A, Tuytelaars T, Van Gool L (2008) Speeded-up robust features (SURF). *Computer vision and image understanding* 110(3): 346-359.
27. Bay H, Tuytelaars T, Van Gool L (2006) Surf: Speeded up robust features. *Proc, European conference on computer vision* pp: 404-417.
28. (2018) tensorflow <<https://github.com/tensorflow>>.
29. Girshick R, Donahue J, Darrell T, Malik J Rich feature hierarchies for accurate object detection and semantic segmentation. *Proc., Proceedings of the IEEE conference on computer vision and pattern recognition* pp: 580-587.
30. (2018) MathWork Convolutional Neural Network <<https://www.mathworks.com/solutions/deep-learning/convolutional-neural-network.html>>.
31. Lin TY, Dollár P, Girshick R, He K, Hariharan B, Belongie S Feature pyramid networks for object detection. *Proc., CVPR*, 4.
32. Ren S, He K, Girshick R, Sun J Faster r-cnn: Towards real-time object detection with region proposal networks. *Proc., Advances in neural information processing systems* pp: 91-99.



This work is licensed under Creative Commons Attribution 4.0 License
DOI: [10.19080/CERJ.2020.10.555790](https://doi.org/10.19080/CERJ.2020.10.555790)

**Your next submission with Juniper Publishers
will reach you the below assets**

- Quality Editorial service
- Swift Peer Review
- Reprints availability
- E-prints Service
- Manuscript Podcast for convenient understanding
- Global attainment for your research
- Manuscript accessibility in different formats

(Pdf, E-pub, Full Text, Audio)

- Unceasing customer service

Track the below URL for one-step submission

<https://juniperpublishers.com/online-submission.php>