Urban Water Journal

ISSN: 1573-062X (Print) 1744-9006 (Online) Journal homepage: www.tandfonline.com/journals/nurw20

Applying deep convolutional neural network with 3D reality mesh model for water tank crack detection and evaluation

Z. Y. Wu, R. Kalfarisi, F. Kouyoumdjian & C. Taelman

To cite this article: Z. Y. Wu, R. Kalfarisi, F. Kouyoumdjian & C. Taelman (2020) Applying deep convolutional neural network with 3D reality mesh model for water tank crack detection and evaluation, Urban Water Journal, 17:8, 682-695, DOI: 10.1080/1573062X.2020.1758166

To link to this article: https://doi.org/10.1080/1573062X.2020.1758166

URBAN WATER JOURNAL
2020, VOL. 17, NO. 8, 682–695
https://doi.org/10.1080/1573062X.2020.1758166

Taylor & Francis
Taylor & Francis Group

RESEARCH ARTICLE

Applying deep convolutional neural network with 3D reality mesh model for water tank crack detection and evaluation

Z. Y. Wu[a], R. Kalfarisi[a], F. Kouyoumdjian[b] and C. Taelman[b]

[a]Applied Research, Bentley Systems, Incorporated, Watertown, CT, USA; [b]Department of Engineering, Société Wallonne Des Eaux, Verviers, Belgium

ABSTRACT

Water tanks have been built for decades and likely deteriorated over time. To ensure its integrity, Unmanned Aerial Vehicle (UAV) e.g. drones with cameras are used for inspecting the elevated tanks and the images are collected for detecting defects such as cracks. To automatically detect and segment the defect in images, Mask Regional Convolutional Neural Network (Mask-RCNN) the latest convolution neural network model is trained with the real-world infrastructure inspection images. The trained model has been applied to crack detection and segmentation for a water tower of 50 meters height and storage of 500 m^3. The images are used to construct a 3D mesh model by photogrammetry technology. The 3D model with annotated cracks enables intuitive visualization and quantitative assessment of 1704 detected cracks, which are evaluated in range of 2 mm to 10 mm wide with total crack length of 77.6 m and total crack area of 0.28 m^2.

ARTICLE HISTORY
Received 11 December 2019
Accepted 13 April 2020

KEYWORDS
Tank inspection; deep learning; crack detection; crack segmentation; convolutional neural network; 3D reality mesh model

Introduction

Water storage tanks or towers are highly elevated critical infrastructure for a water distribution system. They are fabricated and installed for multiple purposes of storing adequate water for emergent usages, e.g. as fire fighting, peak consumption, pump outage, etc., maintaining good pressure throughout a system and preventing pressure surges. However, like many other civil infrastructures, including but not limited to bridges, buildings, dams, railways, and highways, water tanks are prone to lose their designed functions as they deteriorate over time. These structures are often subjected to fatigue stress, thermal expansion and contraction, and external load. These effects can have negative impacts on the service performances overtime and may lead to cracks on the structures' surface. The cracks on the structure reduce the local stiffness and cause material discontinuities (Budiansky and O'connel 1976; Aboudi 1987). This inevitable process has prompted for early detection as a preventive measure to avoid further damage and possible failures. Although this concern has motivated people to inspect infrastructures on a regular basis, onsite inspections still require a lot of effort and resources to be performed accurately and efficiently. Therefore, much research has been done to improve the process of civil infrastructure inspection.

Cracks on the concrete surface are one of the earliest indications of degradation of the infrastructures and manual inspection is still the standard practice for detecting cracks and other defects (NYSDOT Office of Structure 2016). In manual inspection, the process is performed by a human inspector who prepares a sketch of the cracks manually by recording the irregular conditions on structures. Since this approach completely depends on an inspector's judgment and expertise, it lacks objectivity in the quantitative analysis. Therefore, much research has been conducted to improve the inspection with computer vision-based methods.

Due to the rapid advancement of computer vision techniques, several vision-based methods, primarily image processing techniques (IPTs), have been applied to crack detection. Mohan and Poobal (2018) conducted a detailed survey of around 50 papers to identify research challenges and the achievements in the field (Mohan and Poobal 2018). Some research works have been done in detecting cracks and potholes on asphalt roads (Abe, Okano, and Sato 1992, 1993; Tsao et al. 1994; Kim and Haas 2002; Koch and Brilakis 2011; Koch, Jog, and Brilakis 2013), detecting cracks using 3D pavement data (Sollazzo et al. 2016), detecting cracks on concrete pavement and bridge surfaces (Abdel-Qader et al. 2006; Oh et al. 2009), structural assessment on underground pipes (Sinha, Fieguth, and Polak 2003; Guo, Soibelman, and Garret 2009; Myrans, Everson, and Kapelan 2019), detecting concrete cracks in tunnels (Yu, Jang, and Han 2007), and surface buildings or reinforced concrete structures (Chen et al. 2006; Yamaguchi and Hashimoto 2006; Christen, Bergamini, and Motavalli 2009; Kabir et al. 2009; Zhu and Brilakis 2010; Zhu, German, and Brilakis 2011; German et al. 2013). The most widely used technique to detect a crack was to use a morphological approach (Mohan and Poobal 2018), in which a collection of non-linear operations (such as erosion, dilation, opening, closing, top-hat, and watershed transform) was applied to the image. One of the fundamental principles of crack recognition in a digital image is that the pixel intensity within the cracks is lower than the background. Therefore, cracks are also detected using thresholding or edge-detection-based methods. Abdel-Qader et al. (2003) did an early comparative study of four edge detection methods for detecting cracks on engineering structures. Their study found that the best solution for crack detection was to use fast Haar transform (FHT). Over the

CONTACT Z. Y. Wu zheng.wu@bentley.com

next several years, the study was expanded and improved by others to include modified edge detection methods (Song and Civco 2004; Sinha and Fieguth 2006; Yamaguchi et al. 2008; Alaknanda and Kumar 2009; Nishikawa et al. 2012). Because of its simplicity, thresholding methods have also been used to detect cracks over the past several decades (Kirschke and Velinsky 1992; Oh, Garrick, and Achenie 1997; Cheng et al. 1999; Li and Liu 2008; Oliveira and Correia 2009; Tsai, Kaul, and Mersereau 2010; Kamaliardakani, Sun, and Ardakani 2014; Adhikari, Moselhi, and Bagchi 2014). Wavelet-based methods using multiscale features of the target have also been studied (Zhou, Huang, and Chiang 2006; Subitras et al. 2006; Wang, Li, and Gong 2007; Nejad and Zakeri 2011). However, these approaches are not effective at detecting non-continuous or high-curvature cracks. Dorafshan, Thomas, and Maguire (2019) reported that Laplacian of Gaussian (LoG) produced the best results among six edge detection algorithms for the testing images with a consistently simple and clean background. It is unknown if LoG will be effective for a complex image background, which is always the case in real world infrastructure inspection. They indicated that edge detectors are used in combination with more contemporary techniques such as deep learning convolutional neural networks for UAS applications, reducing false positive cases by 20 times compared to the sole use of edge detectors (Dorafshan et al. 2018a). Although there are many studies, most IPT-based approaches easily fail to separate cracks when the images show complicated backgrounds, such as dirt, shadows, vegetation, and other noise-inducing factors. These noises can be removed by implementing denoising techniques, however they are not always effective since images taken in real-world situation can vary extensively.

Several studies in using 2D digital images with 3D scene reconstruction have also been explored for crack detection (Jahanshahi and Masri 2012, 2013; Torok, Golparvar-Fard, and Kochersberger 2014). Jahanshahi and Masri (2012) proposed an adaptive crack detection procedure where the crack segmentation parameters are adjusted automatically using depth parameters obtained using 3D scene reconstruction. They used a regular edge-based approach with morphological operation to extract the crack from its background. Torok, Golparvar-Fard, and Kochersberger (2014) presented a crack detection algorithm to operate on 3D mesh models that was constructed with the post-disaster images. The crack algorithm was derived by assuming if an element of a building (e.g. column) is undamaged, then its surface mesh normal should also be perpendicular to the elements' axial direction. Since, this assumption is based on the building elements under disaster events, the application of such a technique cannot be extended to civil infrastructures that are still in service and without disaster-scale damage.

One promising solution for real-world applications is to incorporate learning-based approaches in which the techniques can learn patterns (or features) from images to predict cracks. This in turn can alleviate negative effects of background noises. Several research studies have implemented a combination of machine learning algorithms-based (MLAs) classifications with IPT-based feature extractions for crack detection (Moon and Kim 2011; O'Byrne et al. 2013; Jahanshahi et al. 2013; Wu et al. 2014). Kaseko, Lo, and Ritchie (1994) first presented a comparative study of the traditional and neural-network classifiers to detect cracks on pavement road (Kaseko, Lo, and Ritchie 1994). Wang, Nallamothu, and Elliot (1998) used neural network-based microchip, Ni 1000, to classify and quantify the surface distress on highway pavement (Wang, Nallamothu, and Elliot 1998). Saar and Talvik (2010) proposed a system based on neural network to automatically detect and classify pavement crack (Saar and Talvik 2010). Lattanzi and Miller (2014) presented a crack detection method using Canny edge detector and K-means clustering algorithm (Lattanzi and Miller 2014). Bu et al. (2015) proposed an automatic crack detection scheme where wavelet features were first extracted using a sliding window texture analysis technique, then the features were classified by SVM (Bu et al. 2015). Recently, various IPTs and MLA techniques were applied to analyzing surface cracks for quantitative estimation of structural load on reinforced concrete beams and slabs (Davoudi, Miller, and Kutz 2019). In their study, the IPTs were primarily used to analyze the crack patterns while MLA techniques were focused on identifying image features in estimating the load levels in structural components. Even after incorporating MLAs, the results of these approaches still suffer from the false feature extraction. This is because the features extracted using IPTs are still considered hand-crafted and do not necessarily represent the true characteristics of cracks. To overcome this challenge, many researchers have adopted a convolutional neural network (CNN) as a deep learning model for crack detection.

In recent years, there have been several efforts to improve crack detection using CNN techniques (Zhang et al. 2016; Cha, Choi, and Buyukozturk 2017; Tong et al. 2018; Kim and Cho 2018; Zhang, Cheng, and Zhang 2018b). A typical method of applying CNN is to utilize a scanning window where the input images are divided into several image patches with a fixed size. These image patches are manually categorized as either a cracked surface or as an intact surface to build a classification model, which is used to determine the locations of the cracks. This method is known as a block-wise CNN method. Zhang et al. (2016) first proposed a method to detect pavement cracks in images using deep CNN by training a square image patch of 99 × 99 resolution. The cracks were successfully detected using sliding window with a step of 1 pixel. Schmugge et al. (2016) proposed a deep neural network-based method for nuclear power plant crack detection (Schmugge et al. 2016). In their study, training and testing images were divided into patches of 224 × 224 pixels, and a GoogleNet was used to classify each patch into crack and non-crack objects. Chen and Jahanshahi (2017) also performed crack detection in a nuclear power plant where video data was used for detecting cracks (Chen and Jahanshahi 2017). They used CNN to detect cracks in image patches for each frame and using data fusion to aggregate the information obtained from multiple frames. Cha, Choi, and Buyukozturk (2017) designed and trained the CNN with an image patch of 256 × 256 pixels resolution in which the trained model was combined with a sliding window technique to scan any image size larger than 256 × 256 and the model was able to achieve precision as high as 98% (Cha, Choi, and Buyukozturk 2017). Dorafshan, Thomas, and Maguire (2018b) applied AlexNet with the image patches of 227 × 227 pixels for crack detection and reported good results by fully training the model and transferring learning mode, which improved the computation efficiency and detection accuracy.

Zhang et al. (2018a) used transfer learning-based CNN to classify a pavement image into cracks and sealed-cracks (Zhang, Cheng, and Zhang 2018b). Using a block size image of 400 × 400 pixels, they developed block-wise thresholding to segment crack/sealed-crack pixels effectively and efficiently. Overall, the experimental result accurately distinguished crack from sealed-cracks and achieved 90.2% detection rate. Zhang et al. (2018a) used 3D images of asphalt surface and trained the improved CrackNet II to automatically detect cracks at pixel level (Zhang, Cheng, and Zhang 2018b). In their study, they were able to detect finer (or hairline) cracks while eliminating local noise and maintaining a fast processing speed. Park et al. (2019) used a two-step deep learning approach to automatically detect cracks on road pavement (Park et al. 2019). With their approach a segmentation process is first performed to extract the road surface. Subsequently, cracks are detected through analysis of a unit patch within the extracted road surface. Several semantic crack segmentation researches have been also been conducted using fully convolutional networks (FCN) and an encoder-decoder network, such as done by (Huang, Li, and Zhang 2018; Dung and Anh 2019; Bang et al. 2019).

Although the CNN shows strong potential, directly applying CNN on small patches of an image with a sliding window proves to be inefficient to precisely locate the objects (Vaillant, Monrocq, and LeCun 1994; Sermanet et al. 2013). The intact surface takes up most of the image and, therefore, it has the highest influence in the training. To address this issue, several techniques have been proposed to improve object detection performance, such as region-based CNN (or RCNN) (Girshick et al. 2014), SPP-Net (He et al. 2015), and Fast-RCNN (Girshick 2015). In 2017, Ren et al. (2017) further improved the Fast-RCNN technique and developed Faster-RCNN, which has become the state-of-the-art object detector in terms of both accuracy and efficiency (Ren et al. 2017). In addition, all previous CNN-based crack detection studies using 2D images have only tried to locate the cracks by using a small image patch, without crack segmentation and quantitative assessment. Moreover, the models were trained with homogenous (or near-ideal condition) images, which prevent the approaches from being practically useful. In this paper, a unified framework is developed and applied to precise crack identification (i.e. detection and segmentation), intuitive visualization, and quantitative assessment of the large water tower that is more than 50 meters in height and around 500 m^3 in storage. Mask region-based CNN (Mask-RCNN) (He et al. 2017) is employed to detect and segment the cracks. The identified cracks can be intuitively visualized in a 3D model and systematically quantified for the elevated water tower, which is difficult to inspect by a conventional hands-on approach. Thus, the proposed approach facilitates the effective and comprehensive assessment of the entire water tower.

Unified framework for crack identification and assessment

A unified framework, as shown in Figure 1, is proposed for crack identification, assessment and visualization. It consists of five main stages, including (1) image acquisition using handheld (or mounted) camera; (2) crack identification using deep learning-based techniques; (3) the generation of a 3D model of the civil infrastructures; (4) crack quantification using image processing technique; and (5) visualization of the identified cracks on a 3D model. Each of the steps is elaborated in the following sections.

Image acquisition

Image acquisition may be done manually by the inspector using a handheld camera or with the help of unmanned aerial vehicles (UAV) for areas that are hard to reach. The image resolution is critical in detecting cracks. The higher the resolution of the images, the more detailed texture the images can capture, the better the model will perform. Therefore, it is suggested to maintain the same distance between the camera and the cracks in collecting the images. This also helps in ensuring a consistent distribution throughout the image dataset. To conduct a complete inspection of a civil structure, it is necessary to collect hundreds or even thousands of images, each of which contains only a small part of the structure. It turns out that these images can be used to generate a 3D model with the help of photogrammetry technique. A 3D reality mesh model is the essential tool for engineers to assess and visualize the cracks within the context of civil infrastructures. It is worth noting that the images with 50% overlapping are usually required for the generation of a 3D reality mesh model using photogrammetry software.

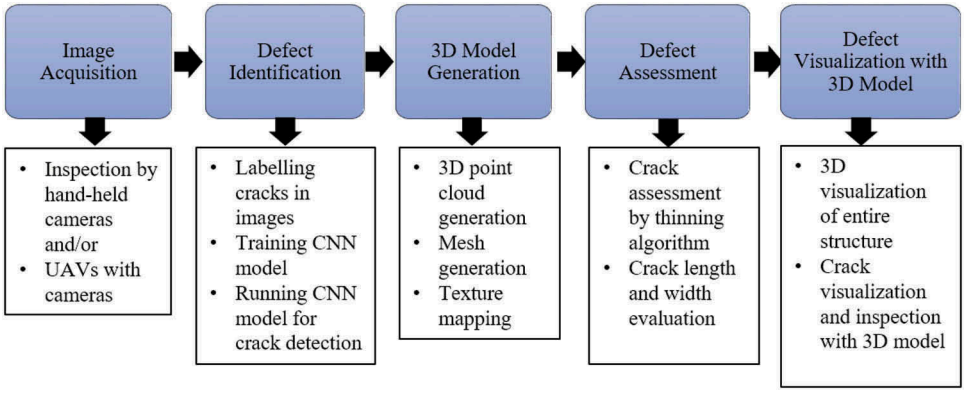


Figure 1. Overview of unified framework for defect identification and evaluation.

Deep learning-based crack identification

Deep learning was built on the foundation of CNN, which was originated by the work of Fukushima (Fukushima 1980) and significantly advanced by LeCun et al. (1998) who formally established how CNN is widely applied today. With CNN, the pixels from each image are converted to a feature representation through a series of mathematical operations. The input image sequentially goes through several processing steps, commonly referred to as layers. The outputs of a layer are often referred to as a feature map. By combining multiple layers, it is possible to develop a complex nonlinear function which can map high-dimensional data (such as images) to useful outputs (such as classification labels). There are several layer types such as convolution layers, pooling layers, and batch normalization layers in most CNN architectures. The first few convolutional layers extract features like edges and textures. Convolutional layers deep in the network can extract features that span a great spatial area of the image, such as object shapes.

Deep learning models are different from traditional machine learning techniques in that they can learn the representations of the data without introducing any hand-crafted rules or knowledge. This means that the deep learning technique is highly flexible and effective for solving a wide variety of challenging problems, such as natural language processing, image recognition and segmentation, speech recognition, etc. Deep learning has been applied in many fields including but not limited to computer vision, medicine and health care, biometrics, and engineering (Vargas, Mosavi, and Ruiz 2017). In this paper, Mask-RCNN will be used to identify cracks.

3D mesh model generation

Identified cracks can be intuitively visualized with 3D reality mesh model for systematic assessment (Bentley 2017). In principle, the 2D model based on the image stitching method could be utilized, but it is difficult to apply the 2D image modeling method to create a digital orthoimage (orthophoto) (Wang 2013), as well as the occlusion region. In contrast, the photogrammetry-based 3D model employs the point cloud technique to build a 3D mesh model and 3D spatial information by referring to the point information extracted from the 2D images (Wang 2013). With the constructed 3D reality mesh model, the inspector can intuitively visualize and systematically assess the defect within the context of engineering structure, instead of individual images.

Photogrammetry is a very useful technique for various applications, especially 3D reality scanning. It is a way to create digital version of objects or landscapes. It can also capture very large objects like buildings, or even an entire city, that would be otherwise impossible to scan using other methods. Moreover, photogrammetry is extremely affordable due to the widely available and accessible digital camera. Thus, a well-developed photogrammetry software (Bakker 2017) is essential for creating 3D models of civil infrastructure.

Photogrammetry works by extracting the geometric information of a 2D image. By combining a lot of pictures, a 3D model can be generated. The way photogrammetry software works is by allowing the program to automatically register shared points among the images and then calculates the distances in 3D space. The result is a point cloud that can be transformed into a 3D mesh. For photogrammetry software to work properly, enough overlapped area is necessary within the images. In close-range photogrammetry, a camera can be used to take hundreds or thousands of images to reconstruct a 3D model. To achieve a high-quality 3D model, a minimum of 50% is required, but an overlap of 70% is recommended (Bakker 2017).

Crack assessment

To quantitatively assess cracks, it is necessary to figure out the total area of the detected cracks. Using a mask image obtained from crack identification, the areas of every connected crack can be calculated by counting the number of white pixels, as shown in Figure 2(c). The length of a crack is obtained by applying a thinning algorithm iteratively until the shape of the cracks only shows one-pixel width, as illustrated in Figure 2(d). Once this is achieved, the average crack width is estimated by dividing the area with the crack length. Finally, all the quantified cracks are classified into different categories or levels according to the estimated average crack width. The crack statistics can then be summarized to assess the structure's condition. By combining the scale in the image and a 3D reality mesh model, the exact dimension of the cracks can be determined.

Crack visualization

The last step is to visualize the identified cracks in the 3D model to be used for easy inspection and integrative assessment of infrastructure condition. This inspection model can be helpful since most aging infrastructures, such as bridges, have no digital information about their geometrical shape and texture conditions. The generated 3D texture and mesh model can be used to inspect the engineering structures for crack assessment. Identification and assessment of the crack information on the 3D model is essential for recording and presenting the inspection results. The generated 3D inspection model can be stored in the database and used for future reference.

Implementation

Computation frameworks

Training a deep CNN is a computationally intensive task since it consists of many nodes, and every node has many connections that must be updated constantly during the learning process. Fortunately, most of the calculations for the neurons performed at each layer are identical. Graphic processing units (GPUs), which are designed to perform the same computational instructions in parallel, fit very well for training deep CNN models. Therefore, GPUs have been extensively used in performing deep learning modeling.

As deep learning techniques have improved extremely quickly over the last few years, several deep learning frameworks are being introduced by tech companies such as Google, Microsoft, Amazon, and Facebook. Deep learning frameworks offer building blocks for designing, training and validating

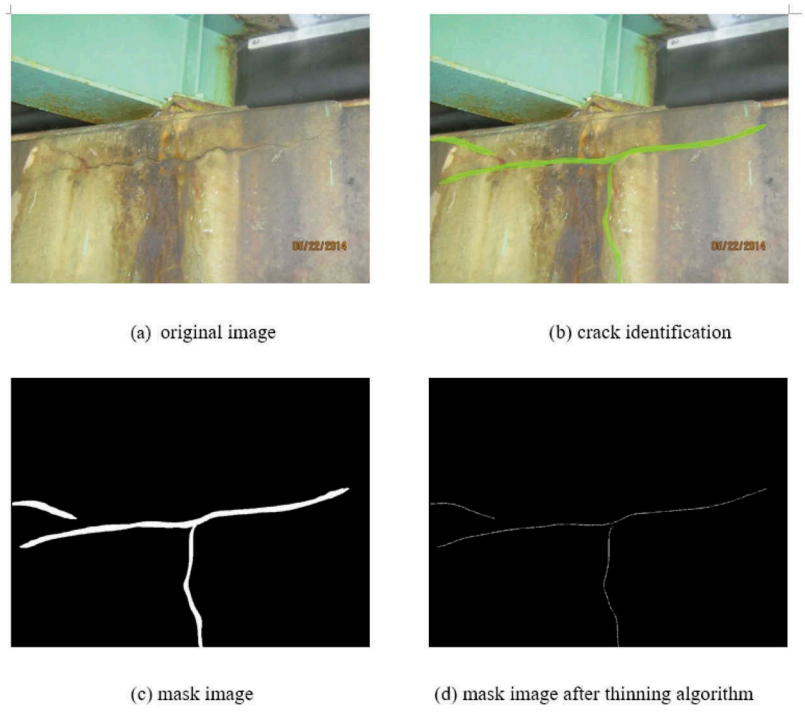


Figure 2. Crack assessment with thinning algorithm.

deep neural networks, through a high-level programming interface. They are created with the goal to run deep learning systems efficiently on GPUs. All the frameworks rely on the concept of *computational graphs* which define the order of computations that need to be performed. These frameworks provide a language that sets up the computational graph and an execution mechanism that can be different from the host language. The graph can be optimized and run in parallel on the target GPU. Figure 3 shows the widely used deep learning frameworks available for professional implementation of deep CNN models to detect and segment the desired objects in images and videos.

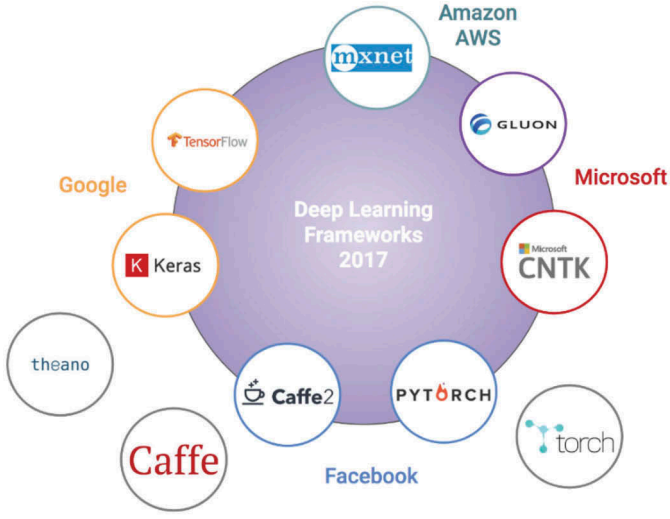


Figure 3. Open source deep learning frameworks (Bakker 2017).

Mask-RCNN

Mask-RCNN is the extension of Faster-RCNN, which essentially utilizes the CNN computed features (or convolutional feature maps) as an input to detect bounding boxes that have a probability of containing the object(s) of interest. In a simplified picture, as shown as in Figure 4, Faster R-CNN consists of two main components, including regional proposal network (RPN) and the object detector. RPN is a fully convolutional network that efficiently generates proposal regions (or candidates) on a wide range of scales (anchors). The region proposals are rectangular regions of interest (ROI) which may or may not contain the objects. The network utilizes a sliding window on the feature map (obtained from the feature extractor CNN) and then maps the window to a lower-dimensional feature using ROI Pooling. The newly created feature is then fed into the second component, a fully connected layer of a box-regression and box-classification layers, called object detector. The algorithm runs through the network only once for the entire input image and then refines the object proposals. Due to the sharing of convolutional layers, it is possible to use a very deep network as the network backbone to generate high quality object proposals and detections.

Built and extended upon Faster R-CNN, Mask-RCNN also consists of two main stages including regional proposal network (RPN) and the Faster R-CNN object detector. However, in the second stage, Mask-RCNN adds a third branch, which is called the mask branch as shown in Figure 4. The mask branch is a fully convolutional network that takes a positive region (region with object inside) and generates a mask image for them. In addition, to get a more precise pixel-wise mask image, the original ROI pooling is replaced with ROI align that uses a bilinear interpolation in extracting the feature map.

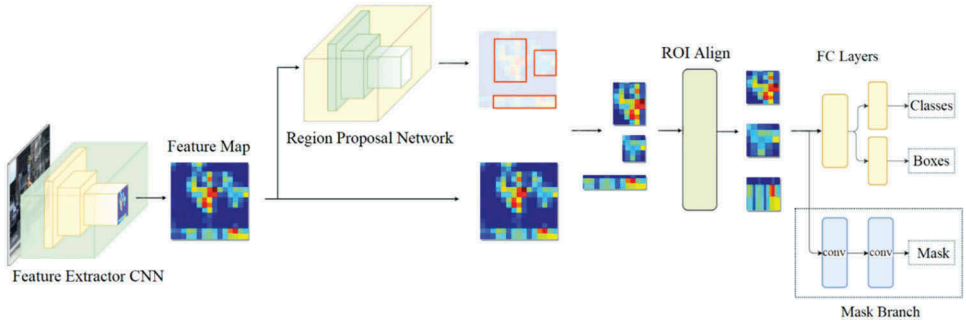


Figure 4. Mask-RCNN architecture (Hui 2018).

In general, the object of a crack segment is detected by minimizing the loss function, given as:

$$L = L_{cls} + L_{box} + L_{mask} \tag{1}$$

$$L_{cls} = \frac{1}{N_{cls}} \sum_i -p_i^* \log p_i - \left(1 - p_i^*\right) \log(1 - p_i) \tag{2}$$

$$L_{box} = \frac{\lambda}{N_{box}} \sum_i p_i^* \cdot L_1^{smooth}\left(t_i - t_i^*\right) \tag{3}$$

$$L_{mask} = -\frac{1}{m^2} \sum_{1 \le i,j \le m} \left[y_{ij} \log \hat{y}_{ij}^k + \left(1 - y_{ij}\right) \log\left(1 - \hat{y}_{ij}^k\right) \right] \tag{4}$$

Where p_i is the predicted probability of anchor i being an object; p_i^* is the ground truth label (binary) of whether anchor i is an object; t_i is the predicted four parameterized coordinates; t_i^* is the ground truth coordinates; N_{cls} is the normalization term, set to be mini-batch size (~256) in the paper; N_{box} is the normalization term, set to the number of anchor locations (~2400) in the paper; λ is a balancing parameter, set to be ~10 in the paper (so that both L_{cls} and L_{box} terms are roughly equally weighted);y_{ij} is the label of a cell (i, j) in the true mask for the region of size m × m and $\hat{y}_{ij}^k$ is the predicted value of the same cell in the mask learned for the ground-truth class k.

Transfer learning for crack identification

To train a deep learning model, a relatively large dataset is needed. CNN with many layers usually contains millions of trainable parameters. For example, the network architecture of AlexNet with only eight layers (Krizhevsky, Sutskever, and Hinton 2012), contains more than 60 million trainable parameters. Directly training such a huge number of parameters is problematic, especially when the training dataset is small. Transfer learning offers one way to transfer the knowledge learned from one task to be used to deal with different tasks (Pan and Yang 2010). In other words, the learning of one task can be improved through the transfer of knowledge from another task that has been trained previously. For example, if a CNN model is to be trained for detecting a car in images, the knowledge and or the features that a previously trained model for recognizing other objects such as buses, or bicycles, etc. can be reused without retraining the model. With transfer learning, we try to exploit what the model has learned in one task to improve generalization in another task. In technical terms, we re-use (transfer) the model weights (or parameters) that a network has learned at 'task A' to a new 'task B'. Instead of starting the learning process from scratch, transfer learning enables us to use a previously trained model as the baseline to further train for a given task. For our study on crack detection, the models (Huang et al. 2017) are first initialized using weights obtained from pre-training on the MS-COCO dataset (Lin et al. 2014) then they are fine-tuned using the prepared training images.

Dataset preparation

The crack image dataset used for this study was acquired from a data repository for hosting field bridge inspection applications and asset management services in USA (Bakker 2017). The images in the dataset were taken during field inspection and captured by professional inspectors using a regular handheld camera without any prior conditioning. The distance of the cracks to the camera ranges from approximately 0.5 m to 10 m. Some images have a good lighting condition with adequate sunlight (above the bridge), and some have dark background (under the bridge). Therefore, the images in the dataset have a broad range of light intensity and shading. In addition, many of the cracks are occluded by other objects such as steel beams, shadows, water stains, branches of trees, etc. Also, the dataset is considered imbalanced with regard to the number of pixels representing cracks and the background. This complexity adds another layer of difficulty to work with the dataset. Among the large number of images, a total of 1250 images where cracks can be seen visually are selected. The size of the image ranges from 344 × 296 to 1024 × 796. The dataset was then split into two categories, 1000 images are used for training and 250 images are used for validation. Images are randomly chosen from the dataset for generating training and validation sets.

To train Faster-RCNN, the cracks in each image are labeled with tight fitting bounding boxes using an image labeling tool (Tzutalin 2015). As for Mask-RCNN, ground truth mask images in which the pixels show cracks were annotated manually. Figure 5 illustrates several representative training images with the corresponding ground truth bounding boxes as well as mask images.

Training and fine-tuning

For all the experiments described in this paper, Faster-RCNN and Mask-RCNN algorithms were implemented using the object

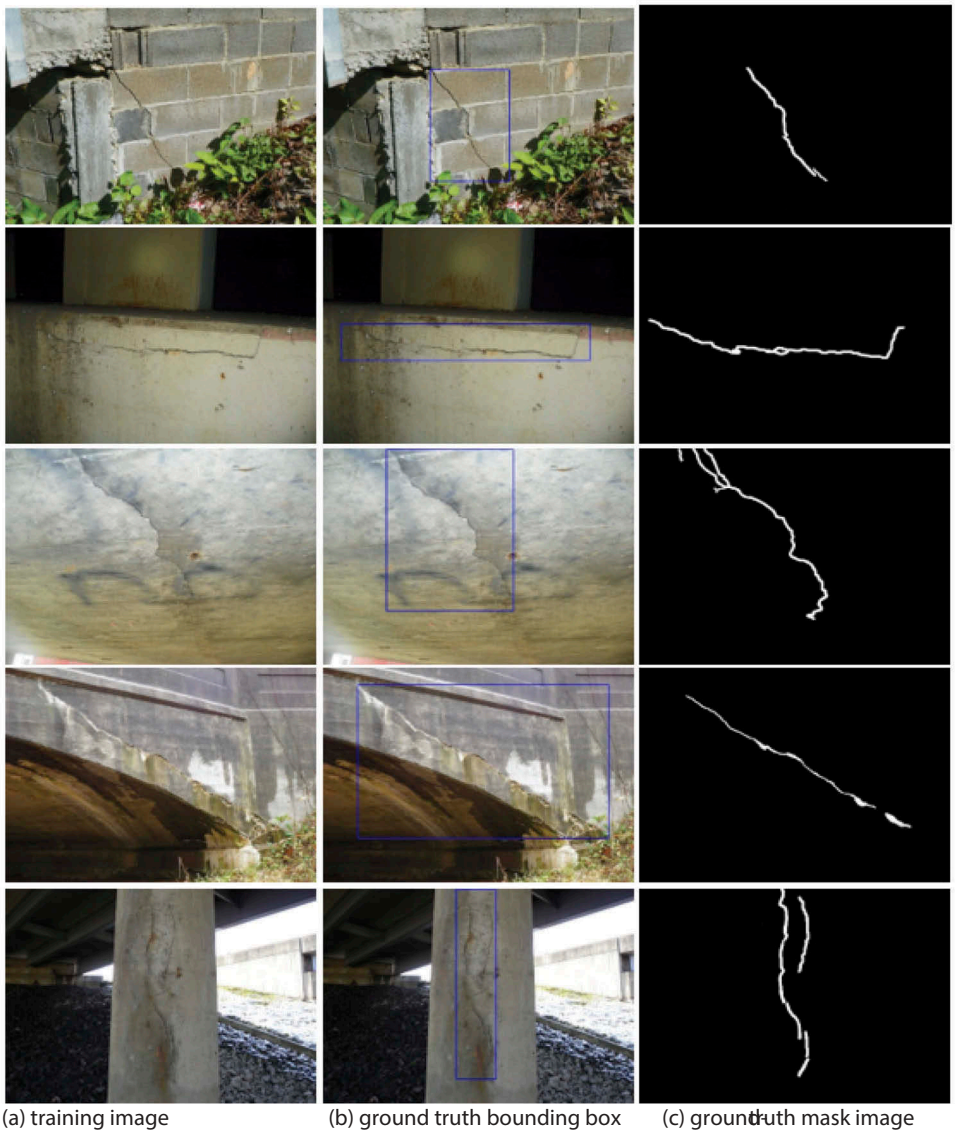


Figure 5. Sample images for training Mask-RCNN.

detection API (Huang et al. 2017) by TensorFlow open-source software library. There is no real difference on how Mask-RCNN is trained from Faster-RCNN except a few changes regarding configuration parameters which are elaborated as follows.

Faster-RCNN and Mask-RCNN are designed to work with variable image size and aspect ratio. However, previous studies suggested that resizing the images can enhance the performance. Therefore, the input images were resized to a minimum dimension of 600 pixels and maximum dimension of 1024 pixels (either width or height) while retaining the original aspect ratio. In other words, if the longer dimension of the input image is less than 1024 pixels, then the shorter dimension will be resized to 600 pixels, and the longer dimension is modified proportionally to keep the aspect ratio the same. On the other hand, if the longer dimension is greater than 1024 pixels, then the longer dimension is resized to 1024 pixels while the shorter dimension is resized appropriately to keep the aspect ratio unchanged.

Like most object detectors, the implementation of object detection API by TensorFlow also uses the transfer learning technique, in which the base model is first trained for an image classification task. The selection of the feature extractor network architecture is of considerable importance because the number of parameters, the type of layers, and other properties directly affect the performance of the detector. Four representative network architectures, including ResNet-50, ResNet-101 (He et al. 2016), Inception-V2 (Ioffe and Szegedy 2015), and Inception-Resnet-V2 (Szegedy et al. 2017) were retrained via transfer learning and evaluated for crack identification. Inception-Resnet-V2 consistently outperformed the other networks with the average detection accuracy of 0.78 and processing time of less than 5 seconds per image the size of 4000 by 4000 pixels on various datasets (Kalfarisi, Wu, and Soh 2020) and was employed in this study.

With transfer learning, the models (Huang et al. 2017) are first initialized using weights obtained from pre-training on the MS-COCO dataset (Lin et al. 2014) then they are fine-tuned using the prepared training images. The weights are updated

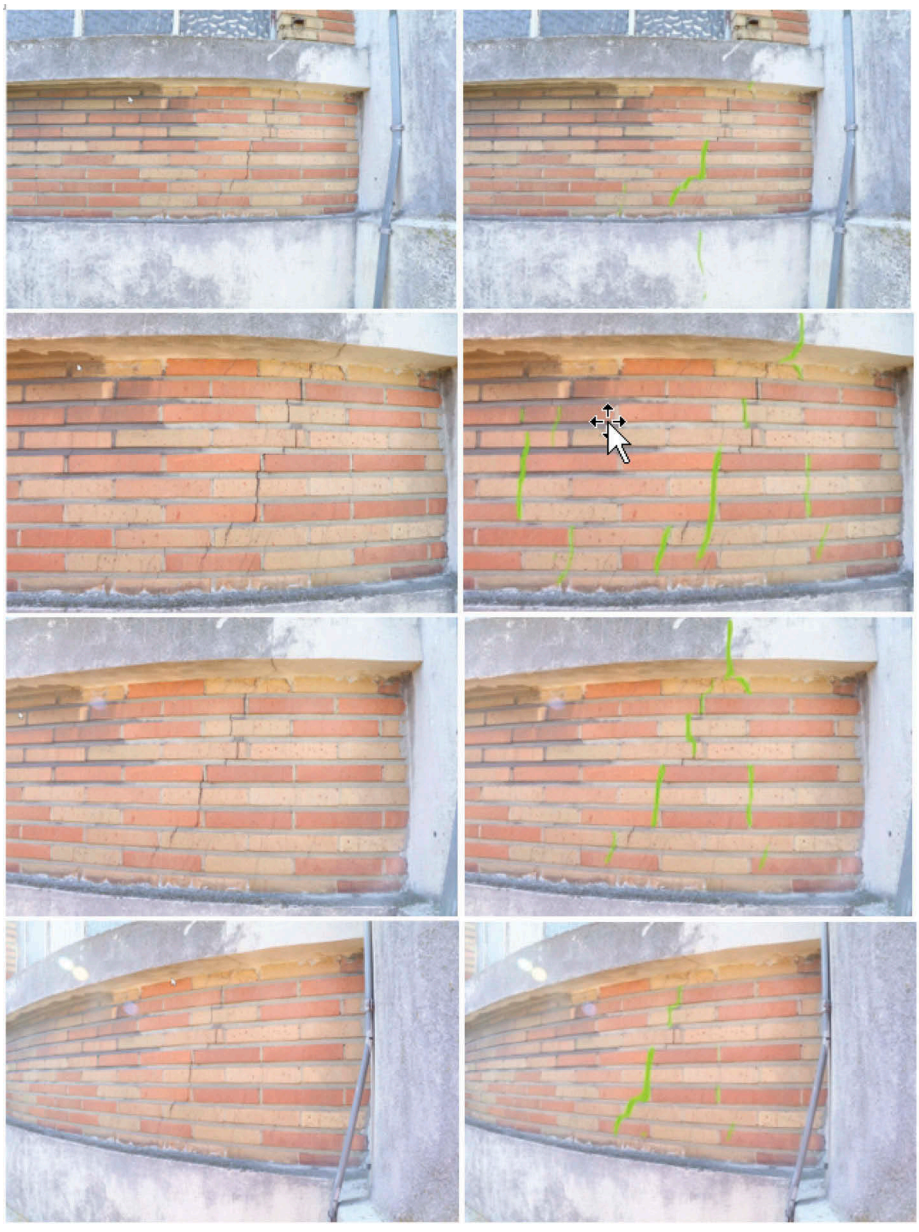
using the stochastic gradient descent (SGD) algorithm for 30,000 iterations with a learning rate of 0.0002 and momentum of 0.9. Although a greater number of training iterations were performed, additional training did not lead to noticeable improvement while shorter training led to underfit. For the RPN, only the top 300 crack proposals were chosen. The experiments were performed on a single NVIDIA K80 GPU with a mini-batch size of 1. The gradients are clipped to a threshold of 10.0 during the training to avoid the exploding gradient problem. Additional experiments with other hyperparameter schedules were also performed, but they gave no significant improvements and are not reported here. To help avoid overfitting, the dataset was augmented using 90° random rotations, random vertical, and random horizontal flips with a probability of 0.5. No other image pre-processing step was taken before the training.

Application examples

The model trained with the engineering inspection images has been applied to two examples of water tank crack detection and elaborated as follows.

Example I

To test and demonstrate the process of image-based crack detection using the trained Mask-RCNN model and photogrammetry approach, images in the resolution of 6000 by 4000 were collected by using a hand-held camera (Canon Eos 80D 18–135 mm with 24.2 million pixel CMOS sensor) of a small area of a water tower. Two columns of images, as shown in Figure 6, illustrate the original images as the input to the trained model and the output images with the detected cracks respectively. The cracks are detected and



(a) Original images (b) Result images with detected cracks

Figure 6. Crack detection results for Example I.

segmented as highlighted in green as shown in Figure 6(b). Although the deep learning model was not trained with the images with the brick texture, the cracks are still successfully detected within the images of brick texture.

It is easily noticeable that not every segment of cracks in each image is detected, but multiple images taken from a slightly different angle help to detect more segments of cracks. In the meantime, some crack segments are detected in multiple images. Therefore, it is difficult to quantitatively evaluate the detected cracks using individual images for the inspection area because the one actual crack segment will be counted multiple times in images. This challenge is addressed by constructing a 3D mesh model via a photogrammetry approach, which essentially requires the images with a large overlap. As shown in Figure 7, a 3D mesh model was constructed by the photogrammetry modeling technique (Bakker 2017) using the images with the detected cracks as the input. The 3D mesh model provides a holistic visualization of the inspected area of the water tower and facilitates the quantitative evaluation of the detected crack as shown in Figure 8, which indicates that the detected cracks range from less than 3 mm to greater than 8 mm wide, with a conversion factor of 0.6 mm/pixel.

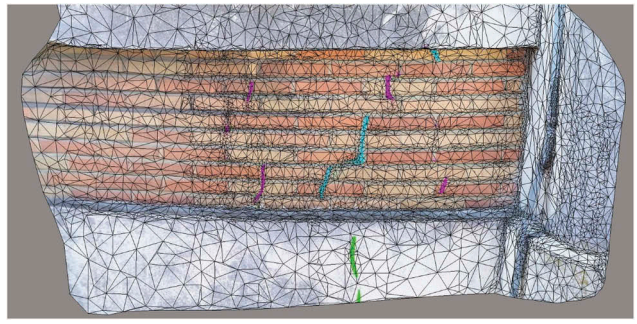


Figure 7. Visualization of detected cracks with 3D mesh model.

Example II

A real water tower, owned and maintained by Société Wallonne Des Eaux (SWDE) a regional water corporation, is a major structure built in 1981 with a storage capacity of 500 m^3 in Juprelle, Belgium. The bearing structure consists of eight columns and intermediate beams on which the water tank rests. The sandblasted brick siding is connected to the supporting wall only by galvanized metal anchors. During the surveys, the presence of rust is noted reducing the section of hooks. The cylindrical cladding, as shown in Figure 9, with a height of 50 m, rests on a blue stone base.

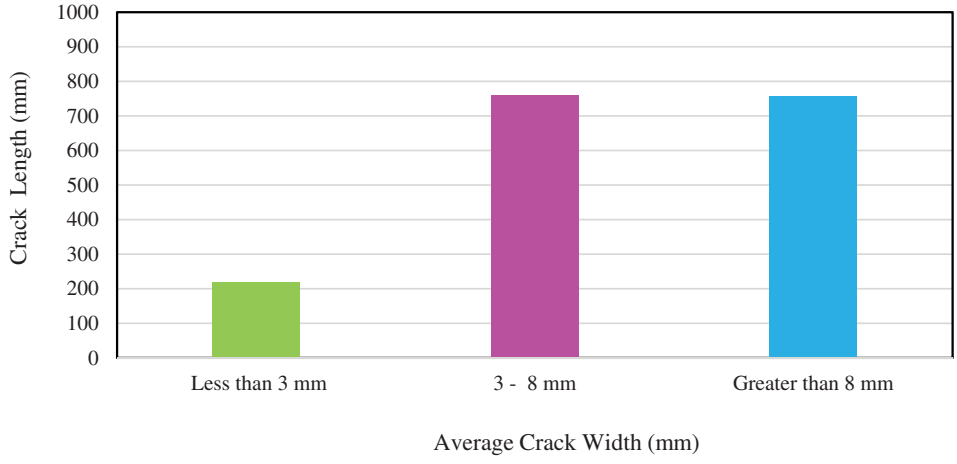


Figure 8. Statistical evaluation of detected cracks.

(a) Street view of water tower (b) Drone image showing upper floor of the structure

Figure 9. Water tower overview.

The exterior is made up of a siding brick, a concrete-filled slide and a terracotta supporting wall inside. The idea of filling the slide with concrete was probably on the one hand to reduce the flaming stresses in the bricks given the height of the facades of 52 m and on the other hand, to provide the cladding a strong link with the inner bearing structure. Unfortunately, this filling of the slide accentuated the damage to the masonry, the absence of a ventilated slide and ventilation within the building favored the formation of condensation and its transfer to the walls. The negative impact of this condensation is evident on the walls and elements attached to them, both on the inside and outside of the building.

The southwest side of the tower, exposed to heavy rain, is the most degraded. The impact of freeze/thaw cycles has

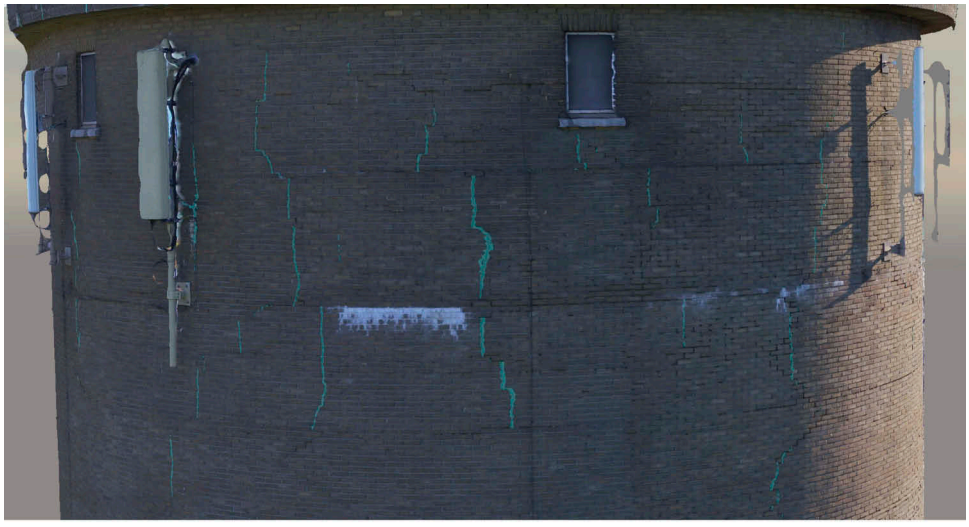
Figure 10. Sample images of detected cracks for the water tower.

(a) 3D model with meshes (b) 3D model with reality texture

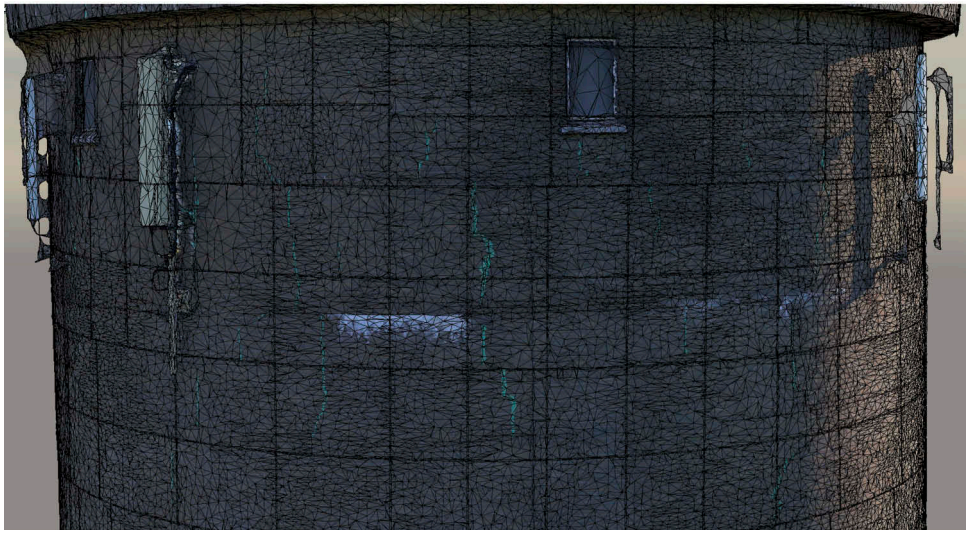
Figure 11. 3D models of the water tower.

particularly affected and accentuated the degradations which have resulted in burst joints, break-ups of hard-work edifice bricks, and the appearance of multiple cracks. The porous sandy bricks of the siding have favored degradation due to climatic conditions. The image as shown in Figure 10(b), taken using the drone, shows significant damage to the upper floor of the structure. The SWDE's operating inspectors have seen superficial cracking and alterations on the brick siding. To get a global view of the building's pathologies and given its height, it is decided to apply the photogrammetry approach by using a Blade Chroma drone to record videos (in a resolution of 3840 × 2160), and subsequently using DJI Inspire 1 with Zenmuse x5 to collect the images at a resolution of 4000 × 2250 to compliment the videos. The drones were piloted by the authorized staff from the design office. The videos were first employed to detect and segment the cracks on the brick siding by applying the trained deep learning model. Over 3000 images are extracted from the video, each image was used as the input of the trained model and the cracks were detected in the resulting images. Some sample results have been illustrated in Figure 10. Subsequently the video images were used to build a 3D model, as shown in Figures 11 and 12 for crack visualization and assessment.

Crack assessment was conducted by using the 3D model texture skin images, which are produced by aerotriangulation of the individual images (with detected and segmented cracks) and the 3D mesh model construction. The texture skin images provide a holistic view of the whole water tower without any overlapping, Thus, each of the detected crack segments can be quantitatively assessed in terms of area, length, and average width by applying the assessment algorithm as elaborated in previous sections. The evaluation was first completed in units of pixels, then converted to metric units by using the conversion factor of 1.4667 mm/pixel, which was worked out by dividing the brick length over the number of pixels of brick length in the image. Table 1 shows the evaluation summary of



(a) Top portion of water tower 3D texture model with detected crack



(b) Top portion of water tower 3D mesh model with detected cracks

Figure 12. Illustration of detected cracks on the 3D model of the water tower.

Table 1. Statistics of detected cracks for the water tower, Example II.

Width (mm)	Crack segments (No.)	Crack length (mm)	Crack area (mm^2)
<2	520	14,583	22,964
2~4	866	36,645	108,371
4~6	230	17,279	82,248
6~8	61	6794	45,214
8~10	25	2209	19,010
>10	2	89	995

the detected cracks for Example II. With 1704 crack segments detected, many more small cracks with an average width less than 6 mm were detected than cracks with a width greater than 6 mm wide. Among them, 520 of them have a width of less than 2 mm, length of more than 14 meters, 866 crack segments have a width between 2 and 4 mm, and length of more than 36 meters, 230 crack segments have a width between 4 and 6 mm, and length greater than 17 meters, only two crack segments are detected in the width of greater than 10 mm, and length of 89 mm. The statists of the detected cracks offer good information for an insightful assessment of the water tower condition.

Conclusions

An integrated approach for water tank inspection has been developed by applying the deep convolutional neural network for crack detection, segmentation, and constructing a 3D mesh model for holistic visualization and quantitative assessment of the detected cracks. The Mask-RCNN model was trained via transfer learning by using the images of engineering structure inspection. The image set was split into training and testing datasets. The trained model is applied to detect the cracks on a water tower, which is 50 meters in height and about 5 meters in diameter. With more than 3000 images collected by using a drone, more than 1700 cracks have been detected and segmented in the range of 2 mm to 10 mm width with a total crack length of 77.6 m and a total crack area of 0.28 m^2. Due to the size of the water tower, it is difficult to perform a hands-on inspection of the entire structure. The proposed approach enables engineers to effectively inspect and assess the defect of the overall tower with the 3D reality model constructed by a photogrammetry technique. The 3D model with annotated cracks does not only facilitate the defect inspection but also serves as a good baseline for comparing and evaluating the structural condition in future.

In general, this research study shows that the identification and assessment of cracks on civil infrastructures can be used to facilitate and improve on the vision-based autonomous inspection of civil infrastructures including but not limited to elevated water towers, pipelines, tunnels, and dams and is being increasingly adopted by infrastructure owners and operators. For on-going and future research, it is necessary and desirable to detect and segment multiple defects for civil infrastructure inspections with a focus on the practical implementation of defect detection, segmentation and assessment integrating it with UAVs and/or inspection robots with edge computing capacity. In the meantime, it is essential to ensure the model's robustness by training the model with a large number of images to detect other defects including but not limited to corrosions, spalling, and water leakages.

Data availability

Some or all data, models, or code generated or used during the study are proprietary or confidential in nature and may only be provided with restrictions.

Disclosure statement

No potential conflict of interest was reported by the author(s).

References

Abdel-Qader, I., O. Abudayyeh, and M. E. Kelly. 2003. "Analysis of Edge-detection Techniques for Crack Identification in Bridges." *Journal of Computing in Civil Engineering* 17 (4): 255-263. doi:10.1061/(ASCE)0887-3801(2003)17:4(255).

Abdel-Qader, I., S. Pashaie-Rad, O. Abudayyeh, and S. Yehia. 2006. "PCA-based Algorithm for Unsupervised Bridge Crack Detection." *Advances in Engineering Software* 37 (12): 771–778. doi:10.1016/j.advengsoft.2006.06.002.

Abe, S., T. Okano, and H. Sato. 1992. "A High Speed Image Processor for Detection of Pavement Cracks." In *Proceedings of the IAPR Workshop on Machine Vision Applications*, Tokyo, Japan, 529–532.

Abe, S., T. Okano, and H. Sato. 1993. "System Integration of Road Crack Evaluation System." In *Proceedings of SPIE - The International Society for Optical Engineering*, San Diego, CA, USA, 38–48.

Aboudi, J. 1987. "Stiffness Reduction of Cracked Solids." *Engineering Fracture Mechanics* 26 (5): 637–650. doi:10.1016/0013-7944(87)90129-9.

Adhikari, R., O. Moselhi, and A. Bagchi. 2014. "Image-based Retrieval of Concrete Crack Properties for Bridge Inspection." *Automation in Construction* 39: 180–194. doi:10.1016/j.autcon.2013.06.011.

Alaknanda, A. R. S., and P. Kumar. 2009. "Flaw Detection in Radiographic Weldment Images Using Morphological Watershed Segmentation Technique." *NDT & E International* 42 (1): 2–8. doi:10.1016/j.ndteint.2008.06.005.

Bakker, I. D. 2017. https://towardsdatascience.com/battle-of-the-deep-learning-frameworks-part-i-cff0e3841750

Bang, S., S. Park, H. Kim, and H. Kim. 2019. "Encoder-decoder Network for Pixel-level Road Crack Detection in Black-box Images." *Computer-Aided Civil and Infrastructure Engineering* 34 (8): 713–727. doi:10.1111/mice.12440.

Bentley Systems, I. 2017. *ContextCapture: Quick Start Guide*. Bentley Institute Press.

Bu, G., S. Chanda, H. Guan, J. Jo, M. Blumenstein, and Y. C. Loo. 2015. "Crack Detection Using a Texture Analysis-based Technique for Visual Bridge Inspection." *Electronic Journal of Structural Engineering* 14 (1): 41–48.

Budiansky, B., and R. J. O'connel. 1976. "Elastic Moduli of a Cracked Solid." *International Journal of Solids and Structures* 12 (2): 81–87. doi:10.1016/0020-7683(76)90044-5.

Cha, Y.-J., W. Choi, and O. Buyukozturk. 2017. "Deep Learning-based Crack Damage Detection Using Convolutional Neural Networks." *Computer-Aided Civil and Infrastructure Engineering* 32 (5): 361–378. doi:10.1111/mice.12263.

Chen, F. C., and M. R. Jahanshahi. 2017. "NB-CNN: Deep Learning-based Crack Detection Using Convolutional Neural Network and Naıve Bayes Data Fusion." *IEEE Transaction on Industrial Electronics* 65 (5): 4392–4400. doi:10.1109/TIE.2017.2764844.

Chen, L. C., Y.-C. Shao, -H.-H. Jan, C.-W. Huang, Y.-M. Tien. 2006. "Measuring System for Cracks in Concrete Using Multitemporal Images." *Journal of Surveying Engineering* 132 (2): 77–82. doi:10.1061/(ASCE)0733-9453-(2006)132:2(77).

Cheng, H. D., J. Chen, C. Glazier, and Y. Hu. 1999. "Novel Approach to Pavement Cracking Detection Based on Fuzzy Set Theory." *Journal of Computing in Civil Engineering* 13 (4): 270–280. doi:10.1061/(ASCE)0887-3801(1999)13:4(270).

Christen, R., A. Bergamini, and M. Motavalli. 2009. "High Precision Measurement of Surface Cracks Using an Optical System."

Measurement Science and Technology 20 (7): 077001. doi:10.1088/0957-0233/20/7/077001.

Davoudi, R., G. R. Miller, and J. N. Kutz. 2019. "Structural Load Estimation Using Machine Vision and Surface Crack Patterns for Shear-critical RC Beams and Slabs." *Journal of Computing in Civil Engineering* 32 (4): 04018024.

Dorafshan, S., R. J. Thomas, C. Coopmans, and M. Maguire. 2018a. "Deep Learning Neural Networks for sUAS-assisted Structural Inspections: Feasibility and Application." In *Proceedings of the 2018 International Conference on Unmanned Aircraft Systems (ICUAS)*, 874–882. 28. Dallas, TX, June 12–15.

Dorafshan, S., R. J. Thomas, and M. Maguire. 2018b. "Comparison of Deep Convolutional Neural Networks and Edge Detectors for Image-based Crack Detection in Concrete." *Construction and Building Materials* 186: 1031–1045. doi:10.1016/j.conbuildmat.2018.08.011.

Dorafshan, S., R. J. Thomas, and M. Maguire. 2019. "Benchmarking Image Processing Algorithms for Unmanned Aerial System-Assisted Crack Detection in Concrete Structures." *Infrastructures* 4: 19. doi:10.3390/infrastructures4020019.

Dung, C. V., and L. D. Anh. 2019. "Autonomous Concrete Crack Detection Using Deep Fully Convolutional Neural Network." *Automation in Construction* 99: 52–58. doi:10.1016/j.autcon.2018.11.028.

Fukushima, K. 1980. "Neocognitron: A Self-organizing Neural Netowk Model for a Mechanism of Patter Recognition Unaffected by Shift in Position." *Biological Cybernetics* 36: 193–202. doi:10.1007/BF00344251.

German, S., J.-S. Jeon, Z. Zhu, C. Bearman, I. Brilakis, R. DesRoches, L. Lowes et al. 2013. "Machine Vision Enhanced Post-earthquake Inspection." *Journal of Computing in Civil Engineering* 27 (6): 622–634. doi:10.1061/(ASCE)CP.1943-5487.0000333.

Girshick, R. 2015. "Fast R-CNN." In *Proceedings of the 15th IEEE international Conference on Computer Vision (ICCV '15)*, Santiago, Chile, 1440–1448.

Girshick, R., J. Donahue, T. Darrel, and J. Malik. 2014. "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation." In *Proceedings of the 27th IEEE Conference on Computer Vision and Patern Recognition (CVPR' 14)*, Columbus, Ohio, 580–587.

Guo, W., L. Soibelman, and J. Garret. 2009. "Visual Pattern Recognition Supporting Defect Reporting and Condition Assessment of Wastewater Collection Systems." *Journal of Computing in Civil Engineering* 23 (3): 160–169. doi:10.1061/(ASCE)0887-3801(2009)23:3(160).

He, K., G. Gkioxari, P. Dollár, and R. Girshick. 2017. *Mask R-CNN*. arXiv:1703.06870.

He, K., X. Zhang, S. Ren, and J. Sun. 2015. "Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37 (9): 1904–1916. doi:10.1109/TPAMI.2015.2389824.

He, K., X. Zhang, S. Ren, and J. Sun. 2016. "Deep Residual Learning for Image Recognition." In *Prodeedings of the IEEE conference on computer vision and pattern recognition*, Las Vegas, NV, USA, 770–778.

Huang, H.-W., Q.-T. Li, and D.-M. Zhang. 2018. "Deep Learning Based Image Recognition for Crack and Leakage Defects of Metro Shield Tunnel." *Tunnelling and Underground Space Technology* 77: 166–176. doi:10.1016/j.tust.2018.04.002.

Huang, J., V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, T. Song, S. Guadarrama, and K. Murphy. 2017. *Speed/accuracy Trade-offs for Modern Convolutional Object Detectors*. Honolulu, USA: CVPR.

Hui, J., 2018. "Medium." [Online]. https://medium.com/@jonathan_hui/image-segmentation-with-mask-r-cnn-ebe6d793272

Ioffe, S., and C. Szegedy. 2015. "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift." In *International Conference on Machine Learning*, Lille, France, 448–456.

Jahanshahi, M. R., and S. F. Masri. 2012. "Adaptive Vision-based Crack Detection Using 3D Scene Reconstruction for Condition Assessment of Structures." *Journal of Automation in Construction* 22: 567576.

Jahanshahi, M. R., and S. F. Masri. 2013. "A New Methodology for Non Contack Accurate Crack Width Measurement through Photogrammetry for Automated Structural Safety Evaluation." *Smart Materials and Structures* 22 (3): 035019. doi:10.1088/0964-1726/22/3/035019.

Jahanshahi, M. R., S. F. Masri, C. W. Padgett, and G. S. Sukhatme. 2013. "An Innovative Methodology for Detection and Quantification of Cracks through Incorporation of Depth Perception." *Machine Vision and Applications* 24 (2): 227–241. doi:10.1007/s00138-011-0394-0.

Kabir, S., P. Rivard, D. C. He, and P. Thivierge. 2009. "Damage Assessment for Concrete Structure Using Image Processing Techniques on Acoustic Borehole Imagery." *Construction and Building Materials* 23 (10): 3166–3174. doi:10.1016/j.conbuildmat.2009.06.013.

Kalfarisi, R., Z. Y. Wu, and K. Soh. 2020. "Crack Detection and Segmentation Using Deep Learning with 3D Reality Mesh Model for Quantitative Assessment and Integrated Visualization." *Journal of Computing in Civil Engineering* 34 (3): 04020010. doi:10.1061/19(ASCE)CP.1943-5487.0000890.

Kamaliardakani, M., L. Sun, and M. Ardakani. 2014. "Sealed-crack Detection Algorithm Using Heuristic Thresholding Approach." *Journal of Computing in Civil Engineering* 30 (1): 04014110.

Kaseko, M. S., Z. P. Lo, and S. G. Ritchie. 1994. "Comparison of Traditional and Neural Classifiers for Pavement Crack Detection." *Journal of Transportation Engineering* 120 (4): 552–569. doi:10.1061/(ASCE)0733-947X(1994)120:4(552).

Kim, B., and S. Cho. 2018. "Automated Vision-Based Detection of Cracks on Concrete Surfaces Using a Deep Learning Technique." *Sensors (Basel)* 18 (10): 3452. doi:10.3390/s18103452.

Kim, Y. S., and C. Haas. 2002. "A Man-machine Balanced Rapid Object Model for Automation of Pavement Crack Sealing and Maintenance." *Canadian Journal of Civil Engineering* 29 (3): 1–16. doi:10.1139/l02-018.

Kirschke, K. R., and S. A. Velinsky, 1992. "Histogram-based Approach for Automated Pavement-crack Sensing." *Journal of Transportation Engineering* 118: 700–710.

Koch, C., G. M. Jog, and I. Brilakis, 2013. "Automated Pothole Distress Assessment Using Asphalt Pavemen Video Data." *Journal of Computing in Civil Engineering* 27: 370–378.

Koch, C., and I. Brilakis. 2011. "Pothole Detection in Asphalt Pavement Images." *Advanced Engineering Informatics* 25 (3): 507–515. doi:10.1016/j.aei.2011.01.002.

Krizhevsky, A., I. Sutskever, and G. E. Hinton. 2012. "ImageNet Classification with Deep Convolutional Neural Network." In *Proceedings of the 2002 Neural Information Processing Systems*, Las Vegas, NV, USA.

Lattanzi, D., and G. Miller. 2014. "Robust Automated Concrete Damage Detection Algorithms for Field Applications." *Journal of Computing in Civil Engineering* 28 (2): 253–262. doi:10.1061/(ASCE)CP.1943-5487.0000257.

LeCun, Y., L. Bottou, Y. Bengio, and P. Haffner. 1998. "Gradient-based Learning Applied to Document Recognition." *Proceedings of the IEEE* 86 (11): 2278–2323. doi:10.1109/5.726791.

Li, Q. Q., and X. L. Liu. 2008. "Novel Approach to Pavement Image Segmentation Based on Neighboring Difference Histogram Method." In *Proceedings of the International Congress on Image Signal Processing*, Zurich, 792–796.

Lin T. -Y., M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollar, and L. Zitnicksung. 2014. *Microsoft COCO: Common Objects in Context*. European Conference on Computer Vision (ECCV).

Mohan, A., and S. Poobal. 2018. "Crack Detection Using Image Processing: A Critical Review and Analysis." *Alexandria Engineering Journal* 57 (2): 787–798. doi:10.1016/j.aej.2017.01.020.

Moon, H., and J. Kim. 2011. "Intelligent Crack Detecting Algorithm on the Concrete Crack Image Using Neural Network." In *Preceedings of the 28th ISARC*, Seoul, Korea.

Myrans, J., R. Everson, and Z. Kapelan. 2019. "Automated Detection of Fault Types in CCTV Surveys." *Journal of Hydroinformatics* 21 (1): 153–163. doi:10.2166/hydro.2018.073.

Nejad, F. M., and H. Zakeri. 2011. "An Expert System Based on Wavelet Transform and Radon Neural Network for Pavement Distress Classification." *Expert Systems with Applications* 38 (6): 7088–7101. doi:10.1016/j.eswa.2010.12.060.

Nishikawa, T., J. Yoshida, T. Sugiyama, and Y. Fujino. 2012. "Concrete Crack Detection by Multiple Sequential Image Filtering." *Computer-aided Civil and Infrastructure Engineering* 27 (1): 29–47. doi:10.1111/j.1467-8667.2011.00716.x.

NYSDOT Office of Structure. 2016. *Bridge Manual Inspection*. New York, USA: Department of Transportation.

O'Byrne, M., F. Schoefs, B. Ghosh, and V. Pakrashi. 2013. "Texture Analysis Based Damage Detection of Ageing Infrastructural Elements." *Computer-aided Civil and Infrastructure Engineering* 28 (3): 162–177. doi:10.1111/j.1467-8667.2012.00790.x.

Oh, H., N. W. Garrick, and L. E. K. Achenie. 1997. "Segmentation Algorithm Using Iterated Clipping for Processing Noisy Pavement Images." In *Proceedings of the International Conference on Imaging Technologies: Techniques and Applications in Civil Engineering*, Davos, Switzerland, 138–147.

Oh, J. K., G. Jang, S. Oh, J. H. Lee, B.-J. Yi, Y. S. Moon, J. S. Lee et al. 2009. "Bridge Inspection Robot System with Machine Vision." *Automation in Construction* 18 (7): 929–941. doi:10.1016/j.autcon.2009.04.003.

Oliveira, H., and P. L. Correia. 2009. "Automatic Road Crack Segmentation Using Entropy and Image Dynamic Thresholding." In *Proceedings of the European Signal Processing Conf. (EUSPICO'09)*, Glasgow, Scotland, 622–626.

Pan, S. J., and Q. Yang. 2010. "A Survey on Transfer Learning." *IEEE Transactions on Knowledge and Data Engineering* 22 (10): 1345–1359. doi:10.1109/TKDE.2009.191.

Park, S., S. Bang, H. Kim, and H. Kim. 2019. "Patch-based Crack Detection in Black Box Images Using Convolutional Neural Networks." *Journal of Computing in Civil Engineering* 33 (3): 04019017. doi:10.1061/(ASCE)CP.1943-5487.0000831.

Ren, S., K. He, R. Girshick, and J. Sun. 2017. "Faster R-CNN: Towards Real-time Object Detection with Region Proposal Networks." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39 (6): 1137–1149. doi:10.1109/TPAMI.2016.2577031.

Saar, T., and O. Talvik. 2010. "Automatic Asphalt Pavement Crack Detection and Classification Using Neural Network." In *Proceedings of the Biennial Baltic Electronic*, Tallinn, Estonia, 345–348.

Schmugge, S., L. Rice, N. R. Nguyen, J. Lindberg, R. Grizzi, C. Joffe, and M. C. Shin. 2016. "Detection of Cracks in Nuclear Power Plant Using Spatial-temporal Grouping of Local Patches." In *Proceedings of the 2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, Lake Placid, NY, USA, 1–7.

Sermanet, P., K. Kavukcuoglu, S. Chintala, and Y. LeCun. 2013. "Pedestrian Detection with Unsupervised Multi-stage Feature Learning." In *Proceedings of the 26th IEEE Conference on Computer Vision and Patern Recognition (CVPR '13)*, Portland, Oregon, 3626–3633.

Sinha, S. K., and P. W. Fieguth. 2006. "Automated Detection of Cracks in Buried Concrete Pipe Images." *Automation in Construction* 15 (1): 58–72. doi:10.1016/j.autcon.2005.02.006.

Sinha, S. K., P. W. Fieguth, and M. A. Polak. 2003. "Computer Vision Techniques for Automatic Structural Assessment of Underground Pipes." *Computer-Aided Civil and Infrastructure Engineering* 18 (2): 95–112. doi:10.1111/1467-8667.00302.

Sollazzo, G., K. C. P. Wang, G. Bosurgi, and A. J. Q. Li. 2016. "Hybrid Procedure for Automated Detection of Cracking with 3D Pavement Data." *Journal of Computing in Civil Engineering* 30 (6): 04016032. doi:10.1061/(ASCE)CP.1943-5487.0000597.

Song, M., and D. Civco. 2004. "Road Extraction Using SVM and Image Segmentation." *Photogrammetric Engineering and Remote Sensing* 70 (12): 1365–1371. doi:10.14358/PERS.70.12.1365.

Subitras, P., J. Dumoulin, V. Legay, and D. Barba. 2006. "Automation of Pavement Surface Crack Detection Using the Continuous Wavelet Transform." In *Proceedings of the International Conference on Image Processing*, Atlanta, GA, USA, 3037–3040.

Szegedy, C., S. Ioffe, V. Vanhoucke, and A. A. Alemi. 2017. "Inception-v4, Inception-resnet and the Impact of Residual Connections on Learning." In *AAAI*, San Francisco, California, USA, 4278–4284.

Tong, Z., J. Gao, Z. Han, and Z. Wang. 2018. "Recognition of Asphalt Pavement Crack Length Using Deep Convolutional Neural Networks." *Road Materials and Pavement Design* 19 (6): 1334–1349. doi:10.1080/14680629.2017.1308265.

Torok, M. M., M. Golparvar-Fard, and K. B. Kochersberger. 2014. "Image-based Automated 3D Crack Detection for Post-disaster Building Assessment." *Journal of Computing in Civil Engineering* 28 (5): A4014004.

Tsai, Y., V. Kaul, and R. M. Mersereau. 2010. "Critical Assessment of Pavement Distress Segmentation Methods." *Journal of Transportation Engineering* 136 (1): 11–19. doi:10.1061/(ASCE)TE.1943-5436.0000051.

Tsao, S., N. Kehtarnavaz, P. Chan, and R. Lytton. 1994. "Image-based Exper System Approach to Distress Detection on CRC Pavement." *The Journal of Transportation Engineering* 120 (1): 62–64.

Tzutalin. 2015. "LabelImg Git Code." [Online]. https://github.com/tzutalin/labelImg

Vaillant, R., C. Monrocq, and Y. LeCun. 1994. "Original Approach for the Localization of Objects in Images." *IEEE Proceedings: Vision, Image and Signal Processing* 141 (4): 245–250.

Vargas, R., A. Mosavi, and L. Ruiz. 2017. "Deep Learning: A Review." *Advances in Intelligent Systems and Computing*. doi:10.20944

Wang, K. C., S. Nallamothu, and R. P. Elliot. 1998. *Classification of Pavement Surface Distress with an Embedded Neural Net Chip*, 131–161. Reston, VA: ASCE.

Wang, K. C. P., Q. Li, and W. Gong, 2007. "Wavelet-based Pavement Distress Image Edge Detection with À Trous Algorithm." *Transportation Research Record* 2024: 73–81.

Wang, R. 2013. "3D Building Modeling Using Images and LiDAR: A Review." *International Journal of Image and Data Fusion* 4 (4): 273–292. doi:10.1080/19479832.2013.811124.

Wu, L., S. Mokhtari, A. Nazef, B. Nam, and H. B. Yun. 2014. "Improvement of Crack Detection Accuracy Using a Novel Crack Defragmentation Technique in Image-based Road Assessment." *Journal of Computing in Engineering* 30 (1): 040141181–040141189.

Yamaguchi, T., and S. Hashimoto. 2006. "Automated Crack Detection for Concrete Surface Image Using Percolation Model and Edge Information." In *Proceedings of the 32nd Annual Conf. on IEEE Industrial Electronics*, Paris, France, 3355–3360.

Yamaguchi, T., S. Nakamura, R. Saegusa, and S. Hashimoto. 2008. "Image-based Crack Detection for Real Concrete Surfaces." *IEEJ Transactions on Electrical and Electronic Engineering* 3 (1): 759–770. doi:10.1002/tee.20244.

Yu, S. N., J. H. Jang, and C. S. Han. 2007. "Auto Inspection System Using a Mobile Robot for Detecting Concrete Cracks in a Tunnel." *Automation in Construction* 16 (3): 255–261. doi:10.1016/j.autcon.2006.05.003.

Zhang, A., K. C. P. Wang, Y. Fei, Y. Liu, S. Tao, C. Chen, J. Q. Li et al. 2018a. "Deep Learning-based Fully Automated Pavement Crack Detection on 3D Asphalt Surfaces with an Improved Cracknet." *Journal of Computing in Civil Engineering* 32 (5): 04018041. doi:10.1061/(ASCE)CP.1943-5487.0000775.

Zhang, K., H. D. Cheng, and B. Zhang. 2018b. "Unified Approach to Pavement Crack and Sealed Crack Detection Using Preclassification Based on Transfer Learning." *Journal of Computing in Civil Engineering* 32 (2): 04018001. doi:10.1061/(ASCE)CP.1943-5487.0000736.

Zhang, L., F. Yang, Y. Zhang, and Y. Zhu. 2016. "Road Crack Detection Using Deep Convolutional Neural Network." In *Proceedings of the 2016 IEEE International Conference on Image Processing (ICIP)*, Phoenix, Arizona, USA, 3708–3712.

Zhou, J., P. S. Huang, and F. P. Chiang. 2006. "Wavelet-based Pavement Distress Detection and Evaluation." *Optical Engineering* 45 (2): 027007. doi:10.1117/1.2172917.

Zhu, Z., and I. Brilakis. 2010. "Machine Vision-based Concrete Surface Quality Assessment." *Journal of Construction Engineering and Management* 136 (2): 210–218. doi:10.1061/(ASCE)CO.1943-7862.0000126.

Zhu, Z., S. German, and I. Brilakis. 2011. "Visual Retrieval of Concrete Crack Properties for Automated Post-earthquake Structural Safety Evaluation." *Automation in Construction* 20 (7): 874–883. doi:10.1016/j.autcon.2011.03.004.