



# Crack Detection and Segmentation Using Deep Learning with 3D Reality Mesh Model for Quantitative Assessment and Integrated Visualization

Rony Kalfarisi<sup>1</sup>; Zheng Yi Wu, M.ASCE<sup>2</sup>; and Ken Soh<sup>3</sup>

**Abstract:** Crack detection has been an active research topic for civil infrastructure inspection. Over the last few years, many research efforts have focused on applying deep learning-based techniques to automatically detect cracks in images. Good results have been reported with bounding boxes around the detected cracks in images. However, there is no accurate crack segmentation, quantitative assessment, or integrated visualization in the context of engineering structures. In addition, most previously developed deep learning-based crack detection models have been trained with homogenous images collected under controlled conditions, rather than applying the models to images collected during real-world infrastructure inspections. In this paper, two deep learning-based approaches are developed for crack detection and segmentation. The first approach is to integrate the faster region-based convolutional neural network (FRCNN) with structured random forest edge detection (SRFED). The FRCNN is used to detect cracks with bounding boxes while SRFED is applied to segment the cracks within the boxes. The second approach is to directly apply Mask RCNN for crack detection and segmentation. The models have been trained with diverse images collected during real-world infrastructure inspections, enhancing the broad applicability of the models. Both approaches have been applied in a unified framework using three-dimensional (3D) reality mesh-modeling technology that enables quantitative assessment with the integrated visualization of an inspected structure. The effectiveness and robustness of the developed techniques are evaluated and demonstrated using various real cases including bridges, road pavements, underground tunnels, water towers, and buildings. DOI: [10.1061/\(ASCE\)CP.1943-5487.0000890](https://doi.org/10.1061/(ASCE)CP.1943-5487.0000890). © 2020 American Society of Civil Engineers.

**Author keywords:** Deep learning; Crack detection; Crack segmentation; Three-dimensional (3D)-mesh modeling.

## Introduction

Civil infrastructures, including but not limited to bridges, buildings, dams, railways and highways, 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 of the infrastructures over time and can 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 detections as preventive measures 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 for improving 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 the 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.

Because of rapid advancement of computer vision techniques, several vision-based methods, primarily image processing techniques (IPTs) have been applied for 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 et al. 1992, 1993; Tsao et al. 1994; Kim and Haas 2002; Koch and Brilakis 2011; Koch et al. 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 et al. 2003; Guo et al. 2009), detecting concrete cracks in tunnels (Yu et al. 2007), and surface buildings or reinforced concrete structures (Chen et al. 2006; Yamaguchi and Hashimoto 2006; Christen et al. 2009; Kabir et al. 2009; Zhu and Brilakis 2010; Zhu et al. 2011; German et al. 2013). The most widely used technique to detect cracks was to use a morphological approach (Mohan and Poobal 2018), in which a collection of nonlinear 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 digital images is that the pixel intensity within the cracks is lower than the background. Therefore, cracks are also

<sup>1</sup>Associate Research Engineer, Bentley Systems, Incorporated, 27 Siemon Co Dr., Watertown, CT 06795.

<sup>2</sup>Bentley Fellow, Bentley Systems, Incorporated, 27 Siemon Co Dr., Watertown, CT 06795 (corresponding author). Email: zheng.wu@bentley.com

<sup>3</sup>Senior Service Manager, Bentley Systems Singapore Pte. Ltd., 1 HarbourFront Place, #18-01 to 03, HarbourFront Tower One, Singapore 098633.

Note. This manuscript was submitted on March 13, 2019; approved on October 31, 2019; published online on February 22, 2020. Discussion period open until July 22, 2020; separate discussions must be submitted for individual papers. This paper is part of the *Journal of Computing in Civil Engineering*, © ASCE, ISSN 0887-3801.

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 bridge structures. Their study found that the best solution for crack detection was to use fast Haar transform (FHT). Over the 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 et al. 1997; Cheng et al. 1999; Li and Liu 2008; Oliveira and Correia 2009; Tsai et al. 2010; Kamaliardakani et al. 2016; Adhikari et al. 2014). Wavelet-based methods using multiscale features of the target have also been studied (Zhou et al. 2006; Subiras et al. 2006; Wang et al. 2007; Nejad and Zakeri 2011). However, these approaches are not effective at detecting noncontinuous or high-curvature cracks. 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 noise-inducing factors. These noises can be removed by implementing denoising techniques, however they are not always effective since images taken in real-world situations can vary extensively.

Several studies in using two-dimensional (2D) digital images with 3D scene reconstruction have also been explored for crack detection (Jahanshahi and Masri 2012, 2013; Torok et al. 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 et al. (2014) presented 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 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 et al. (1994) first presented a comparative study of the traditional and neural network classifiers to detect cracks on pavement roads (Kaseko et al. 1994). Wang et al. (1998) used neural network-based microchip Ni 1000 to classify and quantify the surface distress on highway pavement (Wang et al. 1998). Saar and Talvik (2010) proposed a system based on neural network to automatically detect and classify pavement cracks (Saar and Talvik 2010). Lattanzi and Miller (2014) presented a crack-detection method using Candy operator 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 et al. 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 convolutional neural network (CNN)—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 et al. 2017; Tong et al. 2017; Kim and Cho 2018; Zhang et al. 2018a, b). 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 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  $99 \times 99$  resolution. The cracks were successfully detected using sliding window with 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  $224 \times 224$  pixels, and a GoogleNet was used to classify each patch into crack and noncrack objects. Chen and Jahanshahi (2018) also performed crack detection in a nuclear power plant where video data was used for detecting cracks (Chen and Jahanshahi 2018). 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 et al. (2017) designed and trained the CNN with the image patch of  $256 \times 256$  pixels resolution in which the trained model was combined with a sliding window technique to scan any image size larger than  $256 \times 256$  and the model was able to achieve precision as high as 98% (Cha et al. 2017). Zhang et al. (2018a, b) used transfer learning-based CNN to classify pavement images into cracks and sealed-cracks (Zhang et al. 2018a, b). Using block size image of  $400 \times 400$  pixels, they developed block-wise thresholding to segment crack/sealed-crack pixels effectively and efficiently. Overall, the experimental result accurately distinguished cracks from sealed-cracks and achieved a 90.2% detection rate. Zhang et al. (2018a) used 3D images of asphalt surfaces and trained the improved AlexNet II to automatically detect cracks at the pixel level (Zhang et al. 2018a, b). In their study, they were able to detect finer (or hairline) cracks while eliminating local noises and maintaining fast processing speed. Park et al. (2019) used a two-step deep learning approach in automatically detecting 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 encoder-decoder network, such as done by Huang et al. (2018), Dung and Anh (2019), and Bang et al. (2019).

Although the CNN shows strong potential, directly applying CNN to small patches of images with a sliding window proves to be inefficient to precisely locate the objects (Vaillant et al. 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 R-CNN) (Girshick et al. 2014), SPP-Net (He et al. 2015), and Fast-RCNN (Girshick 2015). In 2017, Ren et al. (2017) further improved Fast-RCNN technique and developed Faster-RCNN,

6-8

3 notes:

9-10

2 notes:

Eric Bianchi 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 for precise crack identification (i.e., detection and segmentation), intuitive visualization, and quantitative assessment of the identified cracks using a 3D reality mesh model. Two deep learning-based approaches have been proposed for crack identification. The first approach is to use the Faster-RCNN technique along with structured random forest edge detection (SRFED) (Dollar and Zitnick 2013). The idea is to localize the cracks using Faster-RCNN by estimating bounding boxes that surround the cracks. Once the area containing cracks has been located, SRFED is used to segment the cracks. The second approach is to apply Mask region-based CNN (Mask-RCNN) (He et al. 2017) to detect and segment the cracks in one go. The identified cracks can be intuitively visualized in a 3D model and systematically quantified within the context of engineering structures including but not limited to buildings, bridges, and tunnels.

## Unified Framework for Crack Identification and Assessment

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

### Image Acquisition

Image acquisition can be done mainly by the inspector using a handheld camera or with the help of unmanned aerial vehicles (UAVs) 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 small part of the structure. It turns out 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 generating a 3D reality mesh model using photogrammetry software.

### Deep Learning-Based Crack Identification

Deep learning was built on the foundation of CNN, which was originated by work of Fukushima (Fukushima 1980) and significantly advanced by LeCun et al. (1998) for formally establishing 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 that 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.

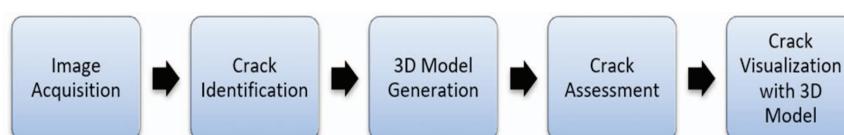
Deep-learning models are different from traditional machine-learning techniques in that they can learn the representations of the data without introducing any hand-crafted rules or knowledge. This means that a deep-learning technique is highly flexible and effective for solving a wide variety of challenging 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 et al. 2017). In this paper, two approaches based on deep-learning technique, Faster-RCNN with edge detection and Mask-RCNN, will be used to identify cracks.

### 3D Mesh Model Generation

Identified cracks can be intuitively visualized with a 3D reality mesh model for systematic assessment. 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 orthophoto (orthophoto), as well as the occlusion region. In contrast, a 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 versions 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 because of widely available and accessible digital cameras. Thus, a well-developed photogrammetry software (Bentley Systems 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



**Fig. 1.** (Color) Proposed framework for crack identification, assessment, and visualization.

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, the camera takes hundreds or thousands of images to reconstruct 3D model. To achieve a high-quality 3D model, a minimum of 50% is required, but an overlap of 70% is recommended (Bentley Systems 2017).

### Crack Assessment

To quantitatively assess cracks, it is necessary to figure out the total area of the detected cracks. Using mask images obtained from crack identification, the areas of every connected crack can be calculated by counting the number of white pixels, as shown in Fig. 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 Fig. 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 for assessing the structure condition. By combining the scale in the image and 3D reality mesh model, the exact dimension of the cracks can be determined.

### Crack Visualization

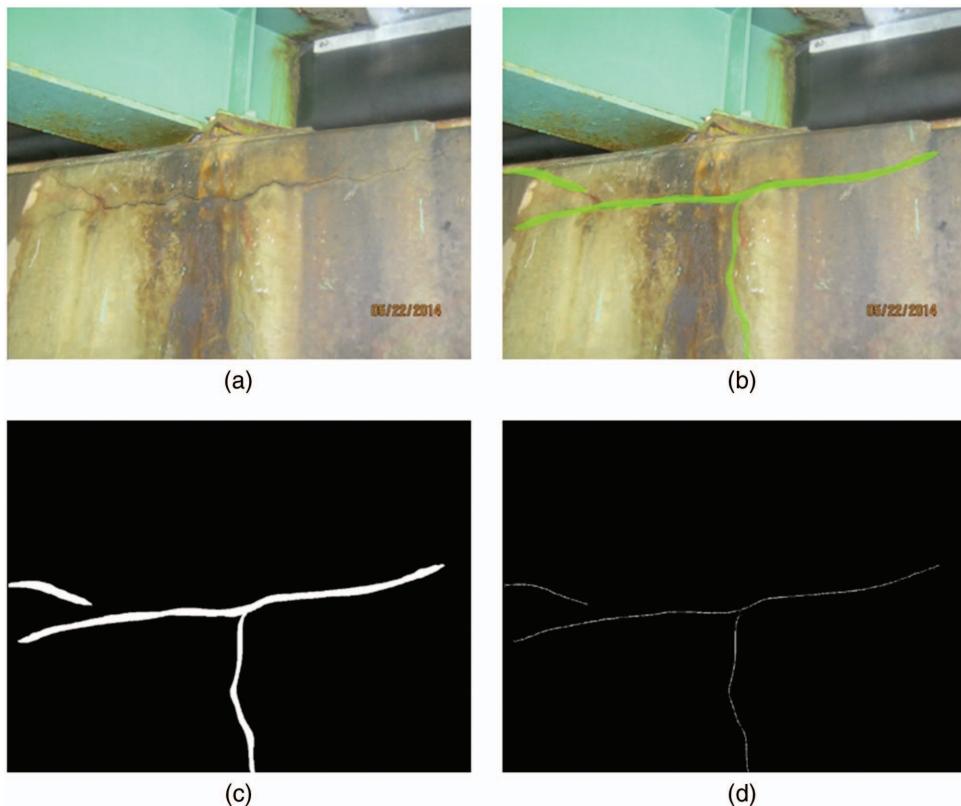
The last step is to visualize the identified cracks in the 3D model to be used for easy inspection and integrative assessment of the

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 bridge 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.

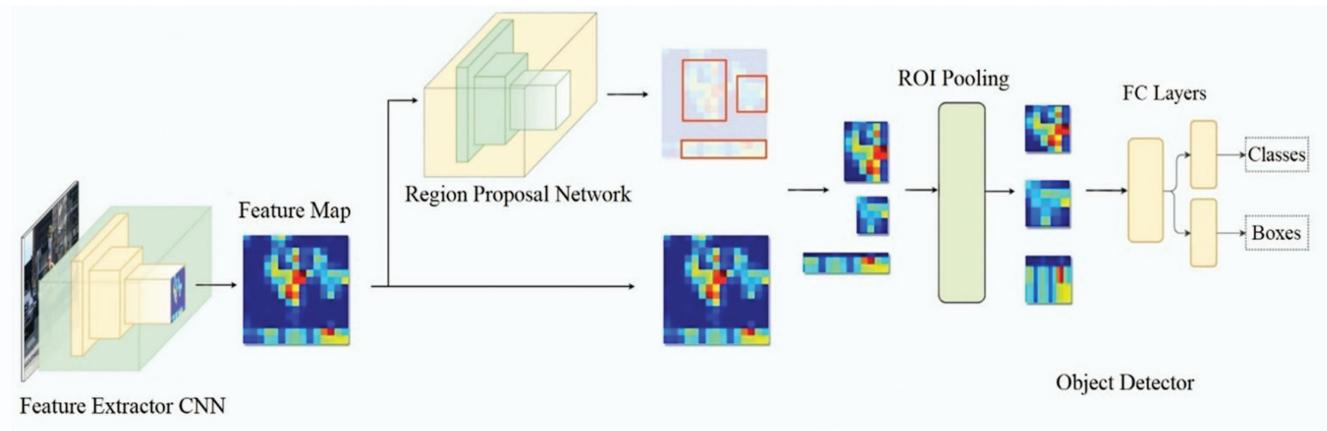
## Implementation

### Faster-RCNN with Edge Detection

Faster-RCNN 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. The algorithm will obtain bounding boxes, assign labels to the boxes, and provide probabilities for each label. In a simplified picture, as shown as in Fig. 3, Faster R-CNN consists of two main components, including a 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) that 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 a box-classification layers, called object



**Fig. 2.** (Color) Example of crack assessment with thinning algorithm: (a) original image; (b) crack identification; (c) mask image; and (d) mask image after thinning algorithm.



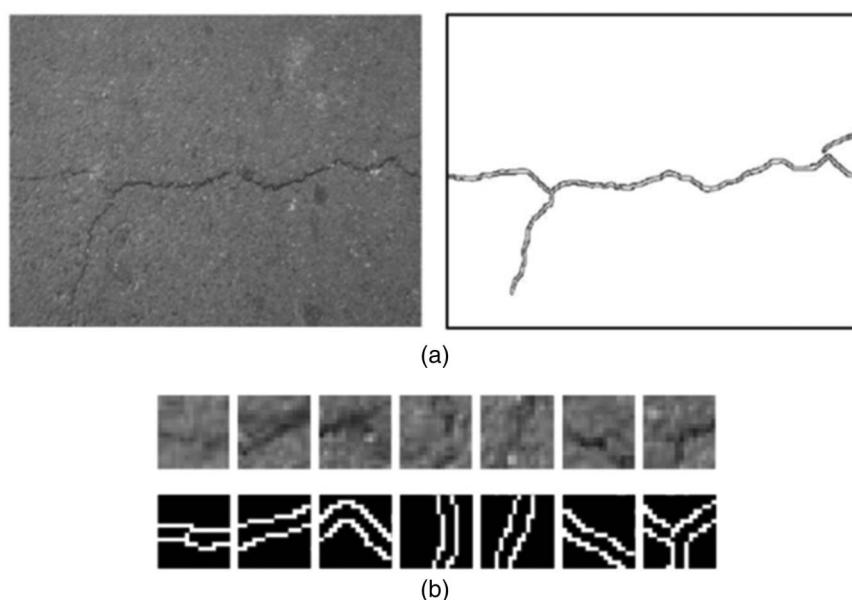
**Fig. 3.** (Color) Faster-RCNN architecture. (Adapted from Hui 2018b.)

detector. The algorithm runs through the network only once for the entire input image and then refines the object proposals. Because of 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.

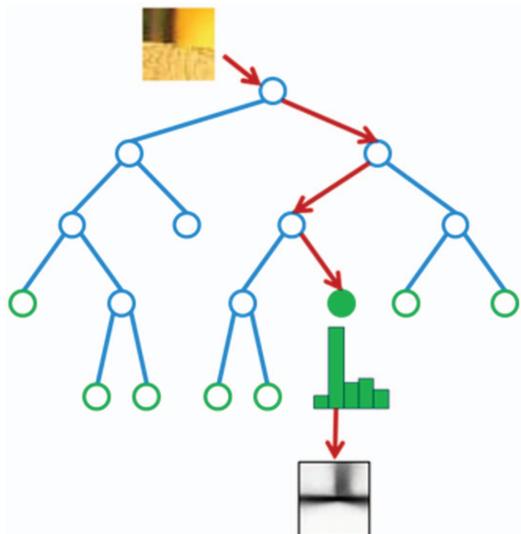
Faster-RCNN is an object detection algorithm, which is employed to localize the cracks with bounding boxes without segmentation. Segment the detected cracks inside the bounding box, structured random forest edge detection (SRFED) is used within these bounding boxes. Thus, coupling Faster-RCNN with SRFED results in an integrated approach, referred to as FRCNN-FED, for crack detection and segmentation. SRFED is an edge detection image processing technique that uses token (segmentation mask) illustrated in Fig. 4 to indicate the crack regions in an image. To train SRFED, a sliding window of  $16 \times 16$  is used to extract image patches from the original image. The image patch containing a labeled crack edge at its center pixel is regarded as a positive instance. Once extracted, to describe an image token, the features (such as mean and standard deviation) are calculated on these patches. These traditional features are computed on gray level

images and applied to describe the brightness and gradient information. To comprehensively characterize cracks, a set of channel features composed with color, gradient, and orientation gradient are also applied.

These image patches will be clustered using a structured random forest to generate an effective crack detector. Structured random forests can exploit the structured information and predict the segmentation mask (token) of a given image patch. In structured random forests, each decision tree, as shown in Fig. 5, classifies an image patch by recursively branching left or right down to the tree until a leaf is reached. And the class of the node is assigned to the image patch. The leaf stores the prediction of the input patch. By training such a tree tokens with the same structure will be gathered at one leaf. Finally, to ensure the connectivity of the detected cracks, erosion and the dilation operation were applied on the edge detection results. These operations can help reduce and eliminate some of the noises presented in the detected image. As a result, some small fragments of the detected region merge together, and the continuity of the crack is improved.



**Fig. 4.** Example of original image with its ground truth and set of image patches with their respective token: (a) original image and its ground truth; and (b) extracted image patches and their token. (Adapted from Shi et al. 2016.)



**Fig. 5.** (Color) Routing path of an image patch using structured random forest.

### Mask-RCNN

Built and extended on Faster-RCNN, Mask-RCNN also consists of two stages including Region proposal network (RPN) and the R-CNN object detector. However, in the second stage, Mask-RCNN adds a third branch, which is called a mask branch as shown in Fig. 6. The mask branch is a fully convolutional network that takes 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 bilinear interpolation in extracting the feature map.

### Transfer Learning for Crack Identification

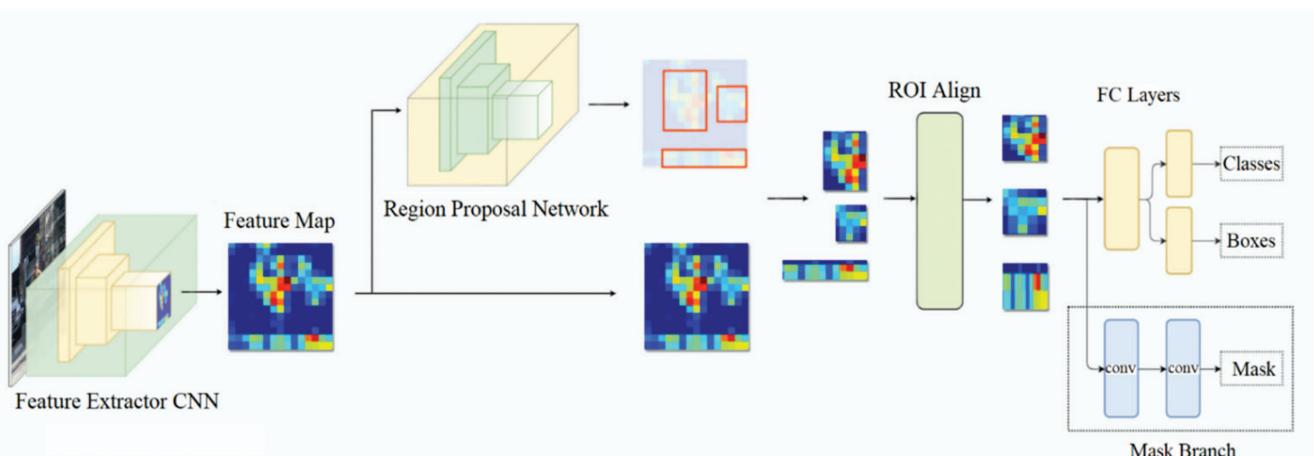
To train a deep-learning model, a relatively large dataset is needed. CNN with many layers usually contains millions of trainable parameters. For example, the network architecture of VGGNet with only eight layers (Krizhevsky et al. 2012), contains more than 60 million trainable parameters. Directly training 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. In transfer learning, one network is first trained on a base dataset and task, and then the learned features (base model) are repurposed or transferred to a second network (with the same network architecture from the first one) to be retrained or fine-tuned on a smaller target dataset and task (in this case, crack dataset). The improvement is significant in large-data-driven task like deep learning, therefore, the transfer-learning technique is adopted in many deep learning applications.

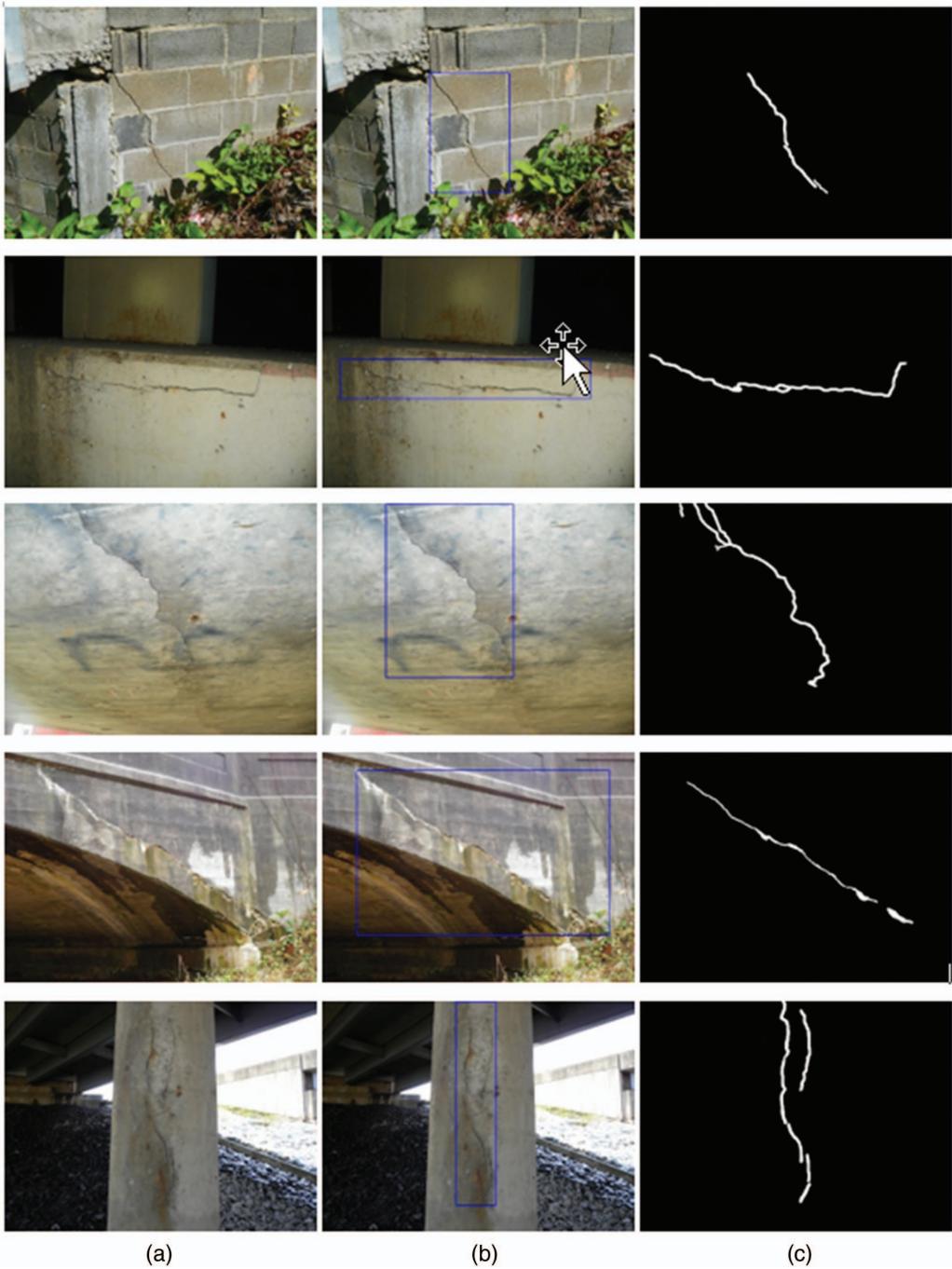
### Dataset Preparation

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 the United States ( Bentley Systems 2017). The images in the dataset were taken during field inspection and captured by professional inspectors using regular handheld cameras without any prior conditioning. The distance of the cracks to the cameras range from approximately 0.5 to 10 m. Some images have good lighting conditions with adequate sunlight (above the bridge), and some have dark backgrounds (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, tree branches, 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 working with the dataset. Among the large number of images, a total of 1,250 images where cracks can be seen visually are selected. The size of the images ranges from  $344 \times 296$  to  $1,024 \times 796$ . The dataset was then split into two categories, 1,000 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 labelled with tight fitting bounding boxes using an image labelling tool (Tzutalin 2015). As for Mask-RCNN, ground truth mask images in which the pixels show cracks were annotated manually. Fig. 7 illustrates several representative training images with the corresponding ground truth bounding boxes as well as mask images.



**Fig. 6.** (Color) Mask-RCNN architecture. (Reprinted with permission from Hui 2018a.)



**Fig. 7.** (Color) Some examples of the images in the datasets: (a) training image; (b) ground truth bounding box; and (c) ground truth mask image.

### Training and Fine-Tuning

For all the experiments described in this paper, Faster-RCNN and Mask-RCNN algorithm were implemented using the object detection API (Huang et al. 2017) by TensorFlow version 1.12 open-source software library. There is no real difference in how Mask-RCNN is trained from Faster-RCNN except a few changes regarding configuration parameters.

Faster-RCNN and Mask-RCNN are designed to work with variable image sizes and aspect ratios. However, previous studies [38-39] found 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 1,024 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 1,024 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 1,024 pixels, then the longer dimension is resized to 1,024 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 transfer learning technique, in which the base model is first trained for the image classification task. The selection of the feature extractor network architecture is very important because the number of parameters, the type of layers, and other properties directly affect the performance of the detector.

In this study, 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 selected and retrained via transfer learning for crack identification.

40-41

2 notes: With transfer learning, the models (Huang et al. 2017) are first initialized using weights obtained from pretraining on the MS-COCO dataset (Lin et al. 2014). When they are fine-tuned using the prepared training images, the weights are updated using the stochastic gradient descent (SGD) algorithm for 30,000 iterations

42

Eric Bianchi with a learning rate of 0.0002 and momentum of 0.9. Although

43-44

2 notes: training did not lead to noticeable improvement while shorter training led to underfit. The RPN, only 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 an exploding gradient problem. Additional experiments with other hyperparameter schedules were also performed, but they show no significant improvements and are not reported here.

45

Eric Bianchi 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 preprocessing step was taken before the training.

49

## Model Performance Evaluation

Eric Bianchi

In this section, the trained models of four different network architectures are evaluated for their performance in detection, segmentation, and computation efficiency for crack identification.

### Detection Performance

The detection performance of the proposed approaches was evaluated on the validation dataset as described earlier. The inferred boxes around the cracks on the validation images were produced by the trained models and were compared with the ground truth boxes. An inferred bounding box with an intersection-over-union (IoU) greater than a threshold of 0.5 is considered true positive. If multiple boxes are predicted for a single ground truth box, then the box with the highest IoU will be considered true positive and the rest are considered false positive. If a ground truth box does not possess any predicted box associated with it, it is designated false negative. The detection performance is measured in terms of average precision (AP) and is defined as the ratio of true positive to the sum of true positive and false positive. In other words, the AP value describes the percentage of the overall detections as correct detections. The AP values obtained from the proposed approaches with four different CNN architectures are reported in Table 1.

Table 1 indicates clearly that Mask-RCNN gives a much better result than Faster-RCNN in detecting cracks. The improvement in AP is thought to be largely because of the benefits arising from joint prediction of bounding boxes and segmentation masks. It is also observed that in both approaches, ResNet-50 and Inception-V2 perform very similarly; as is the case with the ResNet-101 and

**Table 1.** Crack detection average precision of the trained models with different CNN architectures

CNN architecture	Mask-RCNN	Faster-RCNN
ResNet-50	0.72	0.56
ResNet-101	0.77	0.64
Inception-V2	0.71	0.53
Inception-Resnet-V2	0.78	0.66

51

Eric Bianchi

Inception-ResNet-V2. When the crack detection model is trained without the segmentation module (i.e., Faster-RCNN), the highest AP that can be reached is 0.66. While training with segmentation module (i.e., Mask-RCNN), the AP value can reach as high as 0.78. This means the bounding-box prediction accuracy of the crack detection approach is higher when the system is trained simultaneously on crack detection and crack segmentation tasks. This is a common benefit of multitask learning, which is well-documented in the literature (Ren et al. 2017; He et al. 2017). The accuracy is improved when both tasks are learned in parallel, as the bounding box and segmentation modules use a shared representation of the input image (from the feature extractor). However, it should also be noted that training with the segmentation module is slower. The memory requirements for training and testing time are also higher, when object detection and instance segmentation are performed simultaneously compared to pure object detection.

### Segmentation Performance

The primary and common metric for the model evaluation in image segmentation is to use the Jaccard index (or referred to as intersection-over-union) or the Sørensen–Dice Similarity Coefficient (DSC) (Taha and Hanbury 2015). In this paper, DSC will be used to evaluate the segmentation performance of the trained models on different network architectures. DSC measures how close or similar the predicted mask is to the manually marked mask (the ground truth), ranging from 0 (no overlap) to 1 (complete congruence). Given two sets  $\mathbf{X}$  and  $\mathbf{Y}$ , DSC is defined as

$$DSC = \frac{2|\mathbf{X} \cap \mathbf{Y}|}{|\mathbf{X}| + |\mathbf{Y}|} \quad (1)$$

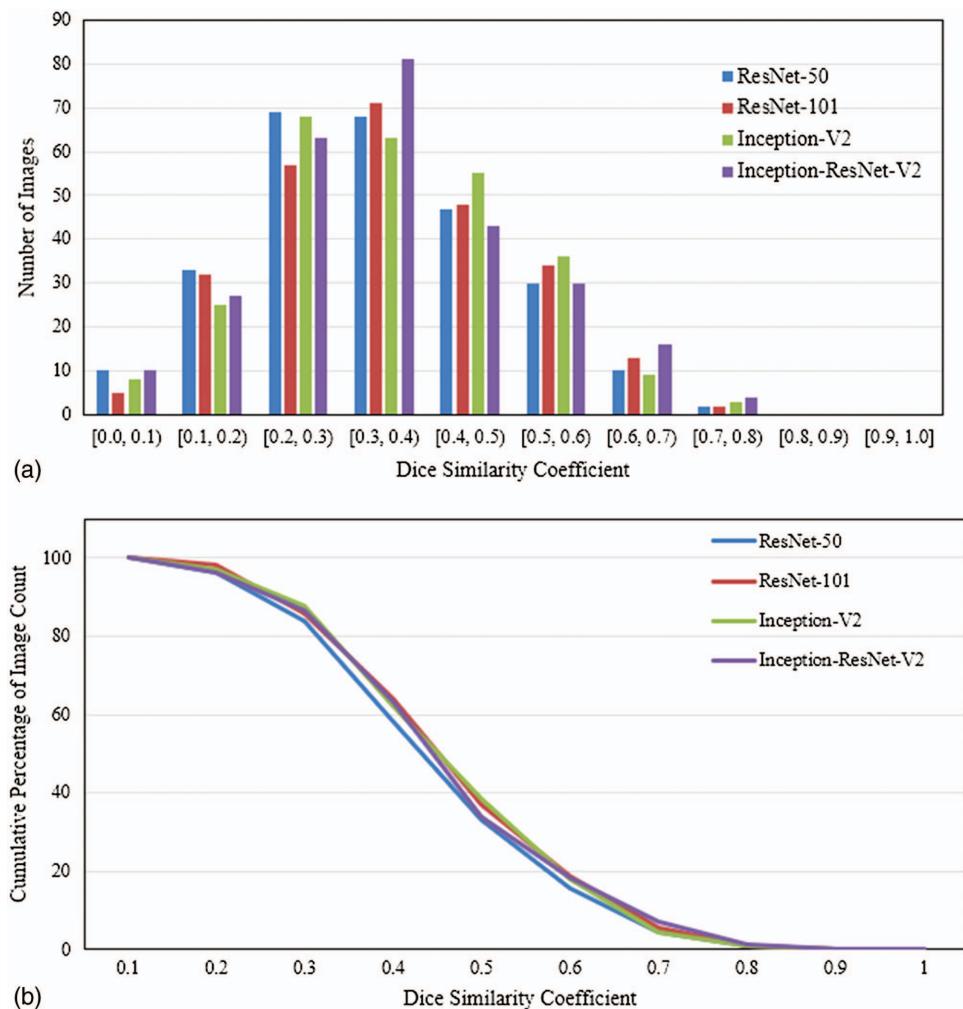
where  $|\mathbf{X}|$  and  $|\mathbf{Y}|$  are the cardinalities of the two sets.

The mean and standard deviation of DSC for two approaches studied in this paper are shown at Table 2. The greater the value of average DSC, the better the performance of the trained model. To examine the segmentation performance, 275 raw images that are not used for training and validation processes are used as the evaluation dataset. From Table 2, the mean values of DSC on the evaluation dataset using the Mask-RCNN approach are greater than those by the FRCNN-FED approach. This is true because the FRCNN-FED approach is a hybrid method that combines deep learning (Faster-RCNN) and image processing technique (SRFED). Using SRFED to segment cracks yields a noisy mask image because of a lot of false positive pixels. It happens because SRFED still suffers from false feature extraction. The features extracted using SRFED are still considered hand-crafted and do not necessarily represent the characteristics of cracks. Thus, the SRFED technique is the weak link and the limiting factor in the FRCNN-FED approach.

A careful analysis of the information presented in Table 2 on the FRCNN-FED approach indicates that there is no clear difference for the segmentation performance of the four CNN architectures.

**Table 2.** Crack segmentation evaluation of the trained models with different CNN architectures

CNN architecture	FRCNN-FED		Mask-RCNN	
	DSC mean	DSC STD	DSC mean	DSC STD
ResNet-50	0.34	0.15	0.51	0.17
ResNet-101	0.36	0.15	0.50	0.16
Inception-V2	0.36	0.15	0.39	0.20
Inception-Resnet-V2	0.36	0.15	0.54	0.13



**Fig. 8.** (Color) DSC distribution analysis of crack segmentation using FRCNN-FED based models: (a) DSC distribution of crack segmentation; and (b) complementary cumulative DSC distribution of crack segmentation.

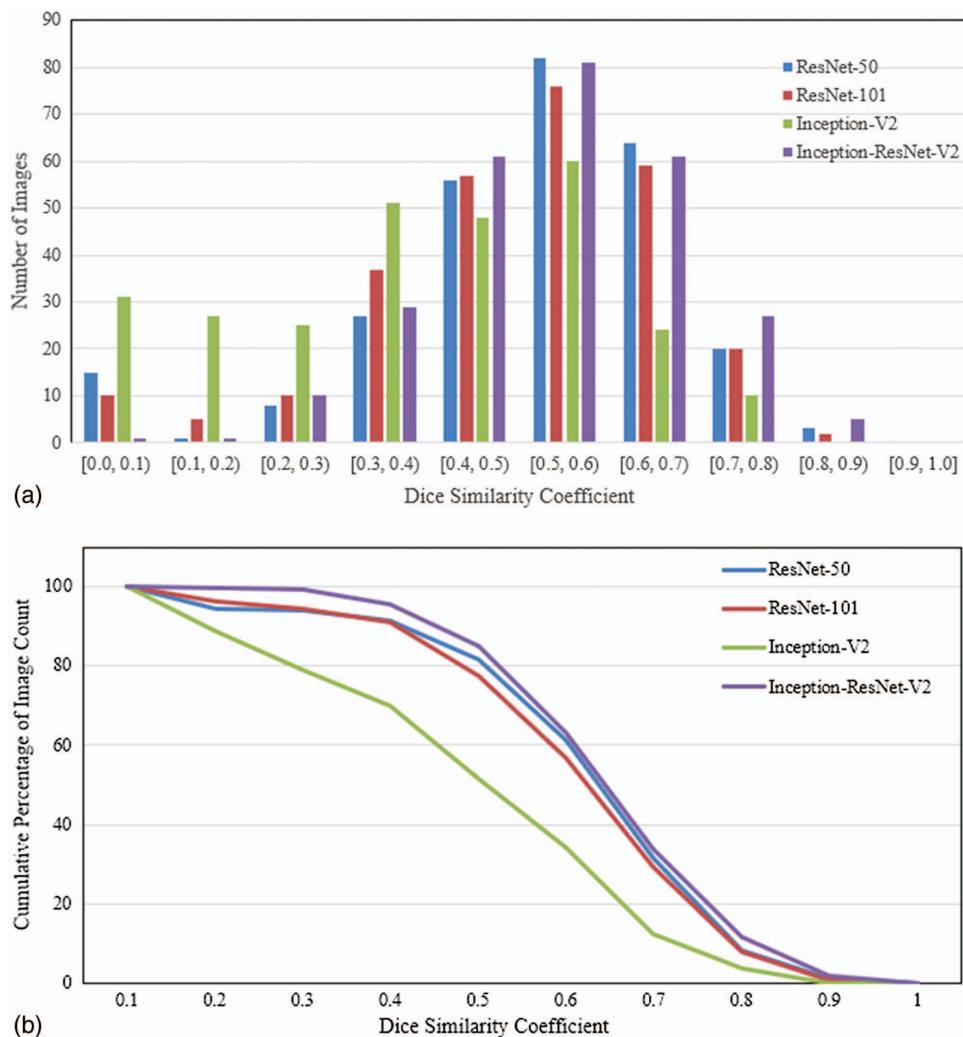
This claim is also supported by the DSC distribution in the evaluation dataset using the FRCNN-FED approach, as shown in Fig. 8(a). The figure shows that the shape of the distribution curve for all CNN architectures are almost identical. Also, with the help of complementary cumulative distribution, as shown in Fig. 8(b), it is clear to see that there's no significant difference in term of performance among four network architectures. The complementary cumulative distribution graph was obtained by counting the number of images with DSC values greater than a certain threshold. Therefore, it can be deduced that the option of choosing the network architecture doesn't influence the crack segmentation results. The segmentation by the FRCNN-FED approach is handled by SRFED and the only important thing to make this approach work is by supplying the accurate (or tight) bounding boxes around cracks. Although the number and the location of bounding boxes are different from each CNN architecture, they do not affect the segmentation result at all since the FRCNN-FED approach only needs one (or more) bounding box that surrounds the cracks entirely. It is also noteworthy to point out that all distributions are more populated around the lower values of the DSC scale with the center around 0.3–0.4.

As for Mask-RCNN, the information given in Table 2 shows that ResNet-50 and ResNet-101 perform very similarly in terms of crack segmentation results. Inception-ResNet-V2 outperformed the others while Inception-V2 shows the worst performance.

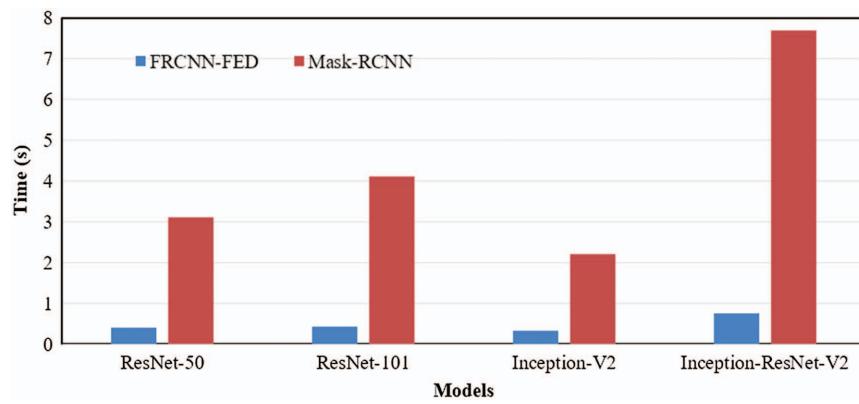
This analysis is further supported by the histogram of DSC distribution using the Mask-RCNN approach as shown in Fig. 9(a). As can be seen, Inception-V2 has more population in the lower DSC scale while Inception-ResNet-V2 only has a very small population in the range of 0.0–0.2. Fig. 9(b) shows the complementary cumulative distribution graph of DSC values using Mask-RCNN approach. As shown in Fig. 9(b), Inception-ResNet-V2 outperforms the others while Inception-V2 performs the worst. Therefore, it can be summarized that the performance of crack segmentation results using Mask-RCNN approach depends on the network architecture and it is safe to say that Inception-ResNet-V2 performs the best for crack identification task.

### Computation Performance

Apart from accuracy, another parameter that is often considered for comparing various detection and segmentation algorithms is computational efficiency, which is measured in terms of average processing time for a single image. Fig. 10 shows that the processing time of the evaluation dataset for crack detection and segmentation using both FRCNN-FED and Mask-RCNN. For all the models tested, it is illustrated that the Mask-RCNN approach is much slower than the FRCNN-FED approach. In fact, the processing time for the Mask-RCNN approach is 10 $\times$  less efficient than the FRCNN-FED approach. It was also observed that the



**Fig. 9.** (Color) DSC distribution analysis of crack segmentation using Mask-RCNN based models.



**Fig. 10.** (Color) Comparison of processing speed for both approaches on different models.

architectures that exhibited greater DSC values had slower processing speed. Comparing the information from Table 2 and Fig. 10 on the Mask-RCNN approach leads to the conclusion that the selection of a suitable detector is a trade-off between segmentation performance and processing speed. This finding also supports the previous study by Huang et al. (2017). As the absolute values presented in Fig. 10 are highly subjective and are dependent on the

image resolution and specific GPU hardware used, therefore, these numbers should not be taken at face value. However, the relationship of the processing time shown here should hold true and it can be deduced that Inception-V2 is the fastest of all network architectures considered in this study. ResNet-50 and ResNet-101 take about the same amount of processing time and Inception-ResNet-V2 was identified as the slowest to accomplish the same task.

## Zero-Shot Transfer Applications

2 notes:

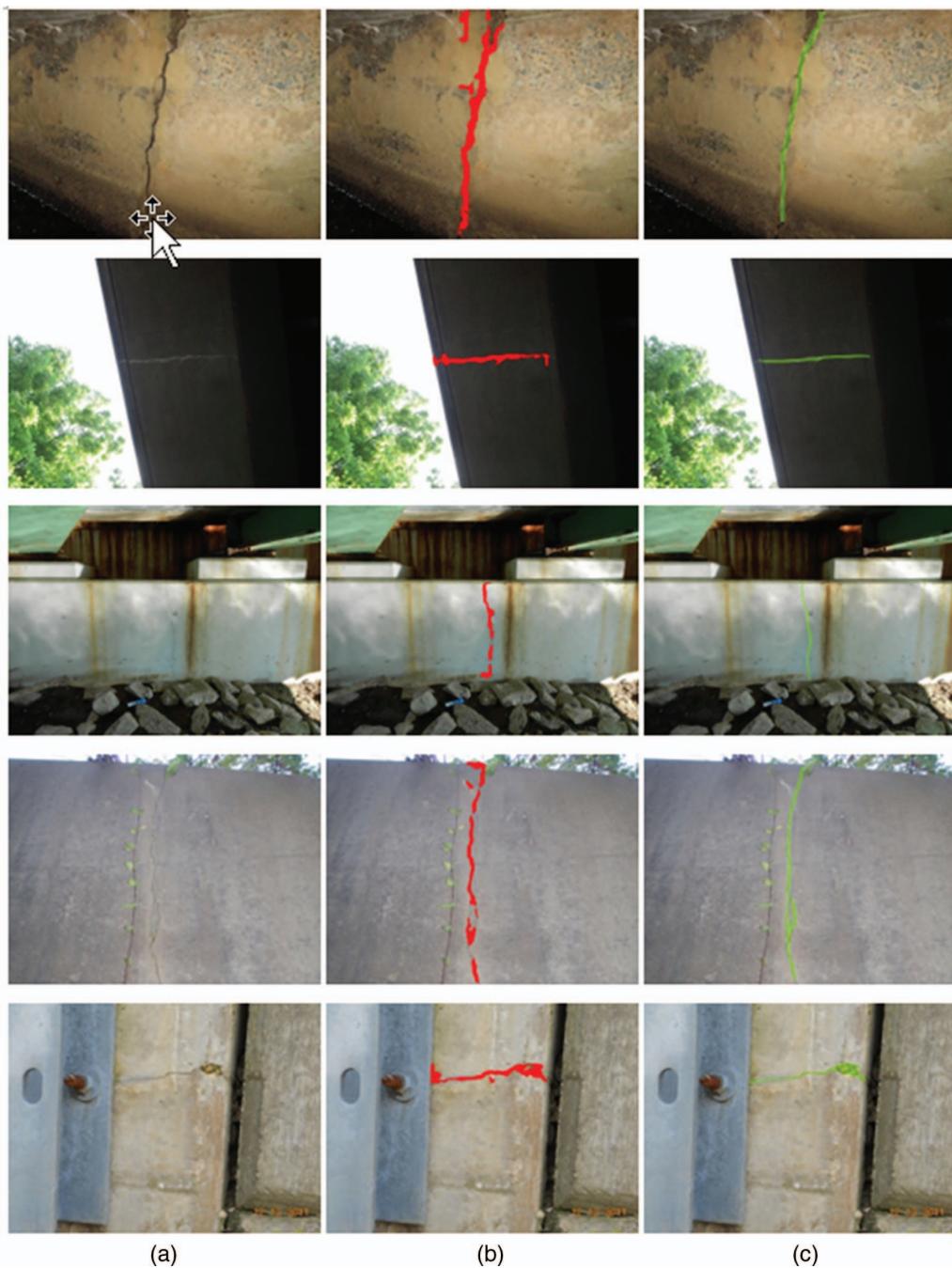
In the field of machine learning, the process of taking a trained model and using it without retraining to conduct inference on an



newly different dataset is called *zero-shot transfer* (Xian et al. 2018). To test the robustness of the proposed crack identification system, the trained models are tested on a range of images of different civil infrastructures including bridges, roads, and underground

**Table 3.** Summary of robustness test cases for crack identification

Test cases	Image count	Resolution	Mask-RCNN		FRCNN-FED	
			Number of cracks	Processing time (h)	Number of cracks	Processing time (h)
Bridge structure	485	1,080 × 768	1,323	0.6	5,029	0.06
Asphalt road	247	1,080 × 768	858	0.3	2,544	0.03
Concrete road	3,930	1,920 × 1,080	8,912	4.9	26,719	0.55
Tunnel	5,310	4,032 × 3,024	861	6.6	1,533	0.74



**Fig. 11.** (Color) Sample results of crack identification on bridge structures: (a) original image; (b) FRCNN-FED result; and (c) Mask-RCNN result.

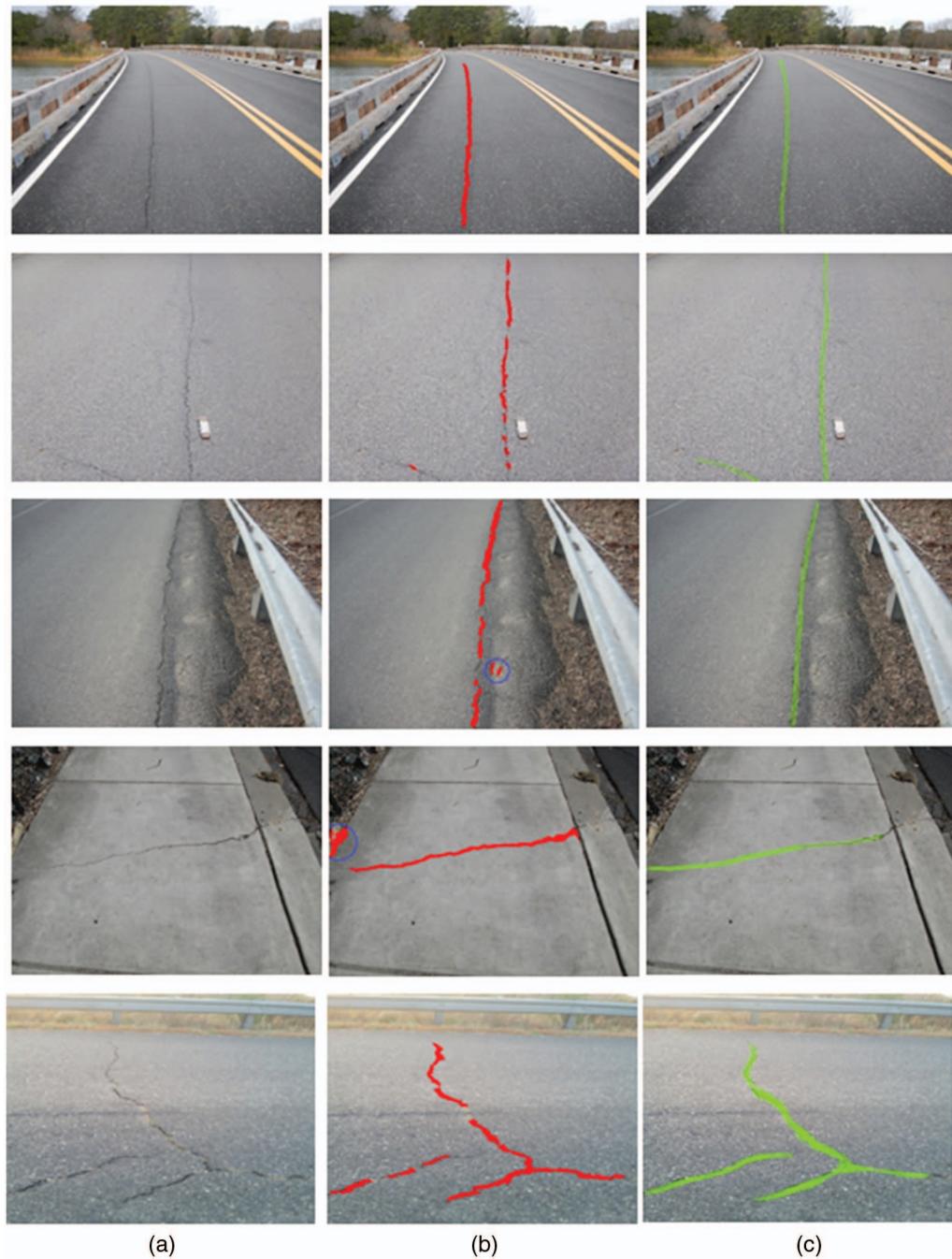
tunnels. In addition, the trained models have been applied to crack identification and assessment of real building inspection, together with 3D reality mesh model for integrated visualization.

### Robustness Test

To test the robustness of the proposed approach, the trained model has been tested on four different cases of bridge structures, road surfaces, and tunnels for crack identification. The images for each of the test cases were collected and provided by the professional infrastructure inspectors to the authors without prior knowledge or specific requirement for image collection and used as is, which strengthens the confidence of the test results. Table 3 shows the

summary of four test cases including bridge structures, asphalt road, concrete road, and underground tunnels. Each of the cases has been executed by using NVIDIA GV100 GPU card on Intel i9-7920X CPU@2.9 GHZ. The summarized table illustrates the scale (the number of images) of the test cases, resolutions of the images, the number of detected cracks, and processing time taken by both FRCNN-FED and Mask-RCNN for each of four test cases. It is believed that this summarized information is useful to provide good understanding of the robustness of the proposed approaches for crack identification.

The robust test was conducted on four real use cases that are never seen by the models during the training process. Because the images of four test cases were collected from the infrastructure



**Fig. 12.** (Color) Sample examples on crack identification on asphalt road pavement: (a) original image; (b) FRCNN-FED result; and (c) Mask-RCNN result.

inspection conducted by engineers, there is no ground truth of the whereabouts of the cracks available or provided for the performance evaluation. It is also prohibitive to manually generate the ground truth for a large number of images as shown in Table 3 for each case. Therefore, the number of detected crack segments is provided as the reference information, along with the image counts, image resolutions, and processing time, for the purpose of robustness test of the models. It is noticed that the number of crack segments identified using the FRCNN-FED approach is always greater than using the Mask-RCNN approach. This is because the FRCNN-FED approach uses SRFED to segment the cracks, which lead to more disconnected crack segments and some false positives or noisy structured edges.

To illustrate the results of four test cases, some example result images are presented for each case. The examples in Fig. 11 show crack identification for the bridge inspection and give a strong proof on how well both approaches work. Both FRCNN-FED

and Mask-RCNN models did a fairly good job in learning how to identify cracks in the images with consistent segmentation performance. The results also show how both approaches perform under various lighting conditions and varying distances.

Eric Bianchi

In the case of FRCNN-FED, the bounding boxes around the cracks on the test images are produced by the Faster-RCNN. SRFED algorithm was then applied to obtain the crack segments inside these bounding boxes. For Mask-RCNN, only one step was taken to get the segmented cracks, i.e., providing the test images into the trained models of Mask-RCNN will produce segmented crack image automatically.

The examples, as shown in Figs. 12 and 13, illustrate how both approaches correctly identify the cracks on asphalt and concrete pavement road images respectively. As can be seen, the trained models of both approaches correctly identify several cracks in images never seen before (there are no images of pavement road cracks in the training dataset). However, it is also observed that



**Fig. 13.** (Color) Sample examples on crack identification on concrete road pavement: (a) original image; (b) FRCNN-FED result; and (c) Mask-RCNN result.

the FRCNN-FED model produces a small number of false positive segmentations (in yellow circles).

The final zero-shot transfer robustness test for crack identification is performed on the tunnel inspection dataset. This dataset, more than 5,000 images have been taken inside an underground tunnel. Again, both approaches give very good segmentation performance. Fig. 14 shows the sample results for the tunnel test case. It was also observed that both techniques failed to identify some cracks. Nonetheless, the fact that both techniques are effective at identifying cracks in images for different datasets demonstrates the robustness of the trained models and gives strong confidence in the trained models to be useful for crack identification of various civil infrastructures.

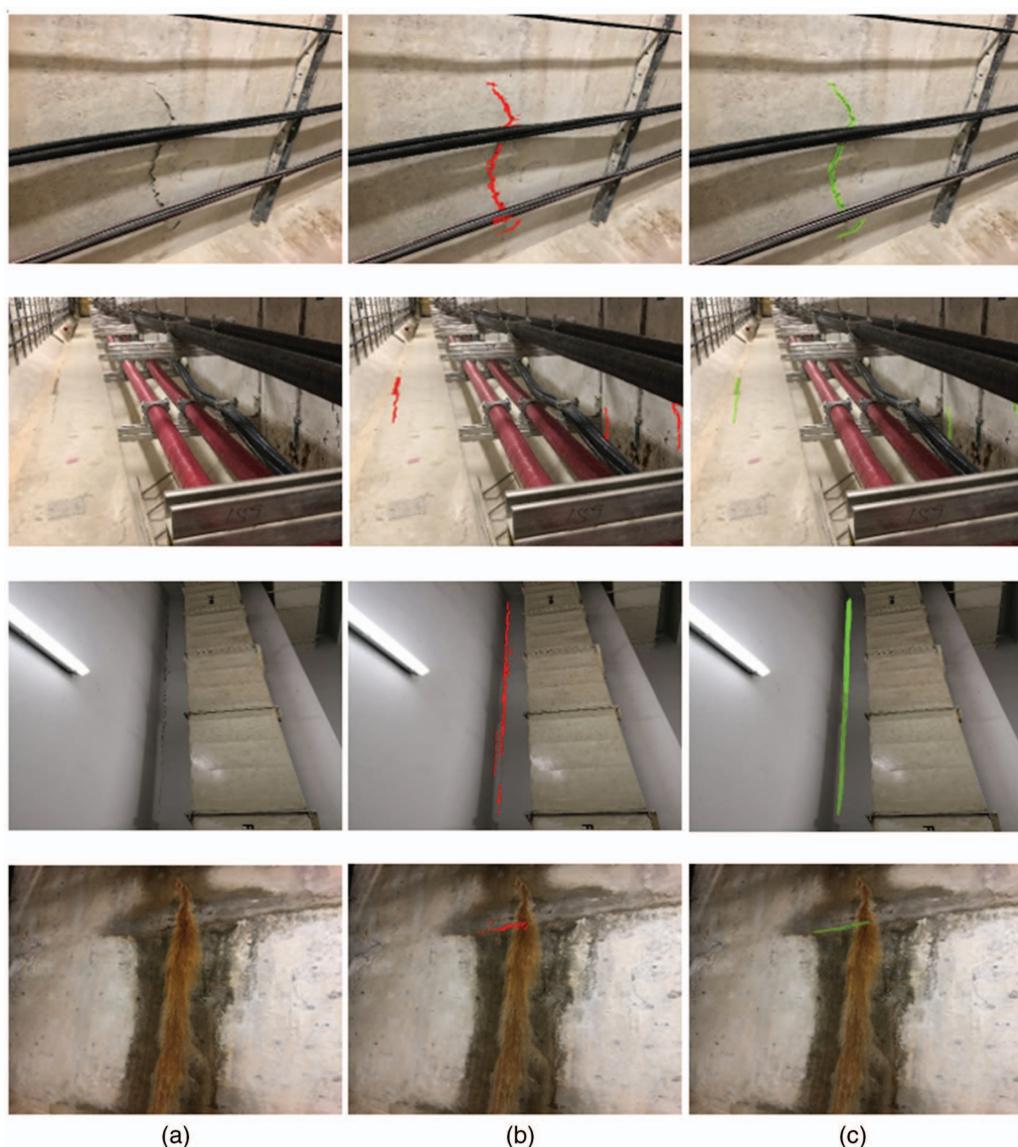
### Building Crack Identification and Assessment

Identifying cracks on buildings is one of the important tasks for building inspection, which is usually conducted for assessment of the general condition of an estate or property. It can be very time consuming to manually identify and quantify all the cracks on a building or difficult to do it for high rise

buildings. It is highly appealing and cost-effective to identify and assess the cracks via images that can be collected by hand-held cameras and/or the cameras mounted on UAVs. In this section, the previously trained model is applied to identify cracks on the internal and external walls of buildings with the help of 3D reality modeling technology. Following the workflow laid out in the unified framework (Fig. 1), the first step is to acquire images of the inspection target. In example, Fig. 15 shows some images of an office wall with approximately 50% overlap taken with the camera from a Google Pixel phone. A total of 10 images were taken within a distance about 1.5 m away from the wall.

After acquiring the images, previously trained Inception-ResNet-V2 model based on FRCNN-FED approach was chosen for aesthetic purposes and applied to identify the cracks. Fig. 16 illustrates the result of crack identification on images.

With these images, a 3D reality mesh model was constructed using photogrammetry software (Bentley Systems 2017), which facilitates crack visualization and crack assessment as can be seen in Figs. 17(a and b). Because the detected cracks in one image overlap with some portions of the cracks in another image, it is not possible to accurately quantify the cracks by using the images only.



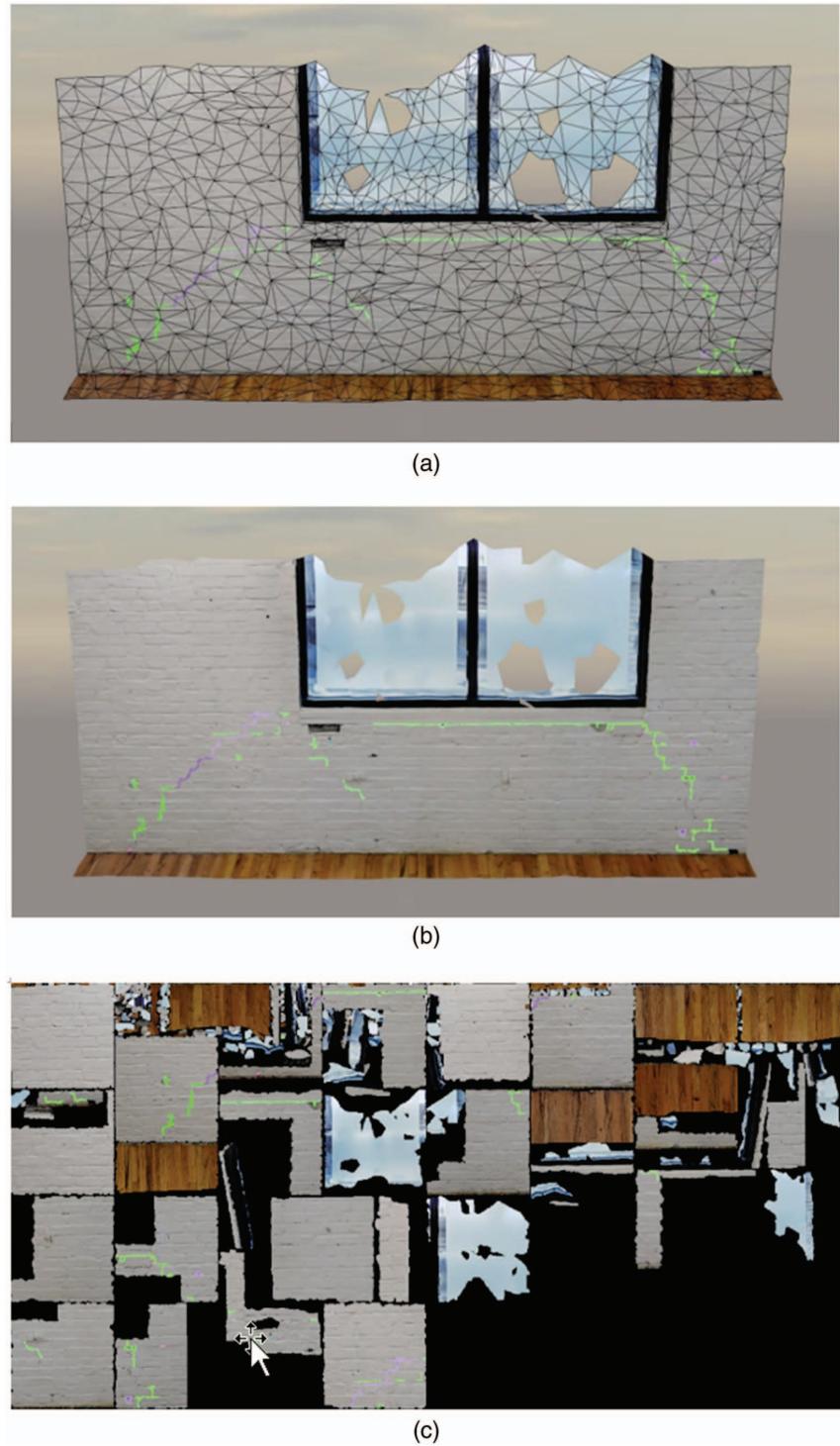
**Fig. 14.** (Color) Sample results of crack identification for tunnel inspection: (a) original image; (b) FRCNN-FED result; and (c) Mask-RCNN result.



**Fig. 15.** (Color) Image acquisition for crack identification on office wall.



**Fig. 16.** (Color) Crack identification on images using deep learning-based approach.

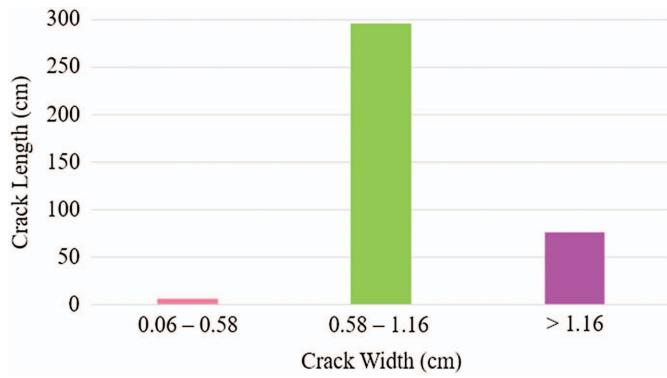


**Fig. 17.** (Color) 3D reality models for crack visualization and assessment: (a) crack visualization with 3D mesh model; (b) crack visualization with 3D texture model; and (c) fused texture cover of the constructed 3D model.

The fused texture (or skin cover), as shown in Fig. 17(c), was generated by the 3D model construction and provides a complete view of the inspected structure without any overlap. This skin cover enables us to adequately assess the crack segments. Using this skin cover tile, it is guaranteed that crack assessment can be accurately conducted for the inspected wall.

To quantify the cracks (width and length) in distance unit, a conversion factor must be worked out to convert the number of pixels to metric unit. The inner side of the glass window was measured to

be 108.60 cm wide on the building wall and covered by 1,880 pixels in the image. Therefore, a conversion factor of 0.06 cm/pixels is obtained. Using this scale factor, each crack segment can be quantified for its average width and length. Thus, the crack statistics are conducted and the result, as illustrated in Fig. 18, is color coded based on the width level of the crack. Purple is the crack area with average width greater than 1.16 cm, green is the crack area with an average width 0.58–1.16 cm, while pink is the crack area with an average width of 0.06–0.58 cm.



**Fig. 18.** (Color) Quantitative assessment of the identified cracks.

To further demonstrate the application of the proposed approach, the same techniques and steps are applied to identify the cracks on an external building  while images collected by using unmanned aerial vehicle (UAV). A total of 45 images have been collected with resolution of  $6,000 \times 4,000$ . Fig. 19 shows the 3D model which illustrate the identified cracks on both sides of the building. As can be seen, the cracks are well identified. With the help of 3D model visualization, the inspection can be done more easily and objectively.

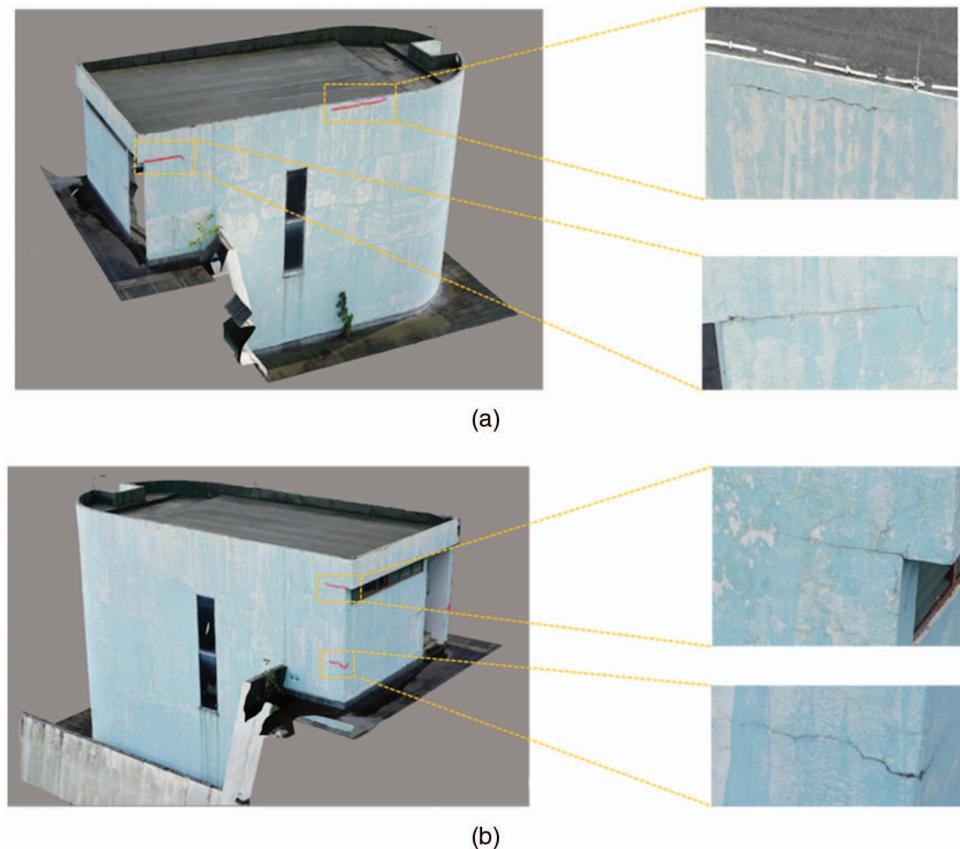
## Conclusions

In this study, two deep learning-based approaches, FRCNN-FED and Mask-RCNN, are used to automatically detect and segment

cracks on civil infrastructures. Four different network architectures, namely, Inception-V2, ResNet-50, ResNet-101, and Inception-ResNet-V2 are employed and the models pretrained on a COCO dataset have been retrained with total of 1,250 images collected from infrastructure inspection.

The performance of the trained models was evaluated on detection and segmentation tasks, which have been conducted with different datasets from training. Detection performance results showed that the training with segmentation module (i.e., Mask-RCNN) yields a much higher AP value. It also shows that the best performance is when the model was trained using Inception-ResNet-V2 network architecture. The segmentation results showed a consistent performance even though the test images were taken under various lighting conditions and distances. The results indicate that Inception-ResNet-V2 significantly outperforms the other networks for both crack detection and segmentation. In addition, the processing speed of the crack identification using the Mask-RCNN approach is found to be much slower than the FRCNN-FED approach. It was noted that the speed is inversely proportional with the performance. The robustness of both approaches was tested on images never seen before and the result shows a very good performance in identifying the cracks. Finally, a unified framework for crack identification, assessment, and visualization was demonstrated using 3D reality mesh model.

This research study shows that the identification and assessment of cracks on civil infrastructures can be used to facilitate and improve the inspection as the vision-based autonomous inspection of civil infrastructures is increasingly adopted by infrastructure owners and operators. For ongoing and future research, it is necessary and desirable to detect and segment multiple defects for civil



**Fig. 19.** (Color) Application of the unified crack identification framework on building using UAV: (a) identified cracks on the right side of the building; and (b) identified cracks on the left side of the building.

infrastructure inspections with focus on the practical implementation of defect detection, segmentation and assessment integrating it with UAVs and/or inspection robots with edge computing capacity.

## Data Availability Statement

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.

## Acknowledgments

Authors are grateful for Mr. Taylor Gilmore and his team from Bentley Asset Performance product advancement unit for providing tens of thousands of infrastructure inspection images for training, validating, and testing the models. Their support is essential for the research project and therefore greatly appreciated.

## References

- Abdel-Qader, I., O. Abudayyeh, and M. E. Kelly. 2003. "Analysis of edge-detection techniques for crack identification in bridges." *J. Comput. Civ. Eng.* 17 (3): 255–263. [https://doi.org/10.1061/\(ASCE\)0887-3801\(2003\)17:4\(255\)](https://doi.org/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." *Adv. Eng. Software* 37 (12): 771–778. <https://doi.org/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 *Proc., IAPR, Workshop on Machine Vision Applications*, 529–532. Tokyo: International Association for Pattern Recognition.
- Abe, S., T. Okano, and H. Sato. 1993. "System integration of road crack evaluation system." In *Proc., SPIE: Int. Society Optical Engineering*, 38–48. Albuquerque, New Mexico: Society of Photo Optical.
- Aboudi, J. 1987. "Stiffness reduction of cracked solids." *Eng. Fract. Mech.* 26 (5): 637–650. [https://doi.org/10.1016/0013-7944\(87\)90129-9](https://doi.org/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." *Autom. Constr.* 39 (Apr): 180–194. <https://doi.org/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 Int.* 42 (1): 2–8. <https://doi.org/10.1016/j.ndteint.2008.06.005>.
- Bang, S., S. Park, H. Kim, and H. Kim. 2019. "Encoder-decoder network for pixel-level road crack detection in black-box images." *Comput.-Aided Civ. Infrastruct. Eng.* 34 (8): 713–727. <https://doi.org/10.1111/mice.12440>.
- Bentley Systems. 2017. *ContextCapture: Quick Start Guide*. Watertown, CT: Bentley Systems.
- Bu, G. P., 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." *Electron. J. Struct. Eng.* 14 (1): 41–48p.
- Budiansky, B., and R. J. O'connel. 1976. "Elastic moduli of a cracked solid." *Int. J. Solids Struct.* 12 (2): 81–97. [https://doi.org/10.1016/0020-7683\(76\)90044-5](https://doi.org/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." *Comput.-Aided Civ. Infrastruct. Eng.* 32 (5): 361–378. <https://doi.org/10.1111/mice.12263>.
- Chen, F. C., and M. R. Jahanshahi. 2018. "NB-CNN: Deep learning-based crack detection using convolutional neural network and naive bayes data fusion." *IEEE Trans. Ind. Electron.* 65 (5): 4392–4400. <https://doi.org/10.1109/TIE.2017.2764844>.
- Chen, L. C., Y. C. Shao, H. H. Jan, C. W. Huang, and Y. M. Tien. 2006. "Measuring system for cracks in concrete using multitemporal images." *J. Surv. Eng.* 132 (2): 77–82. [https://doi.org/10.1061/\(ASCE\)0733-9453\(2006\)132:2\(77\)](https://doi.org/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." *J. Comput. Civ. Eng.* 13 (4): 270–280. [https://doi.org/10.1061/\(ASCE\)0887-3801\(1999\)13:4\(270\)](https://doi.org/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." *Meas. Sci. Tech.* 20 (7): 077001. <https://doi.org/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." *J. Comput. Civ. Eng.* 32 (1): 04018024. [https://doi.org/10.1061/\(ASCE\)CP.1943-5487.0000766](https://doi.org/10.1061/(ASCE)CP.1943-5487.0000766).
- Dollar, P., and C. L. Zitnick. 2013. "Structured forests for fast edge detection." In *Proc., IEEE ICCV*, 1841–1848. New York: IEEE.
- Dung, C. V., and L. D. Anh. 2019. "Autonomous concrete crack detection using deep fully convolutional neural network." *Autom. Constr.* 99 (Mar): 52–58. <https://doi.org/10.1016/j.autcon.2018.11.028>.
- Fukushima, K. 1980. "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position." *Biol. Cybern.* 36 (4): 193–202. <https://doi.org/10.1007/BF00344251>.
- German, S., J. S. Jeon, Z. Zhu, C. Bearman, I. Brilakis, R. DesRoches, and L. Lowes. 2013. "Machine vision enhanced post-earthquake inspection." *J. Comput. Civ. Eng.* 27 (6): 622–634. [https://doi.org/10.1061/\(ASCE\)CP.1943-5487.0000333](https://doi.org/10.1061/(ASCE)CP.1943-5487.0000333).
- Girshick, R. 2015. "Fast R-CNN." In *Proc., 15th IEEE Int. Conf. on Computer Vision (ICCV '15)*, 1440–1448. New York: IEEE.
- Girshick, R., J. Donahue, T. Darrel, and J. Malik. 2014. "Rich feature hierarchies for accurate object detection and semantic segmentation." In *Proc., 27th IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'14)*, 580–587. New York: IEEE.
- Guo, W., L. Soibelman, and J. Garret. 2009. "Visual pattern recognition supporting defect reporting and condition assessment of wastewater collection systems." *J. Comput. Civ. Eng.* 23 (3): 160–169. [https://doi.org/10.1061/\(ASCE\)0887-3801\(2009\)23:3\(160\)](https://doi.org/10.1061/(ASCE)0887-3801(2009)23:3(160)).
- He, K., G. Gkioxari, P. Dollár, and R. Girshick. 2017. "Mask R-CNN." Preprint, submitted March 20, 2017. <https://arxiv.org/abs/1703.06870>.
- He, K., X. Zhang, S. Ren, and J. Sun. 2015. "Spatial pyramid pooling in deep convolutional networks for visual recognition." *IEEE Trans. Pattern Anal. Mach. Intell.* 37 (9): 1904–1916. <https://doi.org/10.1109/TPAMI.2015.2389824>.
- He, K., X. Zhang, S. Ren, and J. Sun. 2016. "Deep residual learning for image recognition." In *Prod., IEEE Conf. on Computer Vision and Pattern Recognition*, 770–778. New York: IEEE.
- 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 Underground Space Technol.* 77 (Jul): 166–176. <https://doi.org/10.1016/j.tust.2018.04.002>.
- Huang, J., et al. 2017. "Speed/accuracy trade-offs for modern convolutional object detectors." In *Proc., IEEE Conf. on Computer Vision and Pattern Recognition*, 7310–7311. New York: IEEE.
- Hui, J. 2018a. "Image segmentation with Mask R-CNN." Accessed July 20, 2018. [https://medium.com/@jonathan\\_hui/image-segmentation-with-mask-r-cnn-ebe6d793272](https://medium.com/@jonathan_hui/image-segmentation-with-mask-r-cnn-ebe6d793272).
- Hui, J. 2018b. "What do we learn from region based object detectors (Faster R-CNN, R-FCN, FPN)?" Accessed July 20, 2018. [https://medium.com/@jonathan\\_hui/what-do-we-learn-from-region-based-object-detectors-faster-r-cnn-r-fcn-fpn-7e354377a7c9](https://medium.com/@jonathan_hui/what-do-we-learn-from-region-based-object-detectors-faster-r-cnn-r-fcn-fpn-7e354377a7c9).
- Ioffe, S., and C. Szegedy. 2015. "Batch normalization: Accelerating deep network training by reducing internal covariate shift." In *Proc., Int. Conf. on Machine Learning*, 448–456. Lille, France: JMLR.org.
- Jahanshahi, M. R., and S. F. Masri. 2012. "Adaptive vision-based crack detection using 3D scene reconstruction for condition assessment of structures." *J. Autom. Constr.* 22 (Mar): 567–576. <https://doi.org/10.1016/j.autcon.2011.11.018>.
- Jahanshahi, M. R., and S. F. Masri. 2013. "A new methodology for non-contact accurate crack width measurement through photogrammetry for automated structural safety evaluation." *Smart Mater. Struct.* 22 (3): 035019. <https://doi.org/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." *Mach. Vision Appl.* 24 (2): 227–241. <https://doi.org/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." *Const. Build. Mater.* 23 (10): 3166–3174. <https://doi.org/10.1016/j.conbuildmat.2009.06.013>.
- Kamaliardakani, M., L. Sun, and M. Ardakani. 2016. "Sealed-crack detection algorithm using heuristic thresholding approach." *J. Comput. Civ. Eng.* 30 (1): 04014110. [https://doi.org/10.1061/\(ASCE\)CP.1943-5487.0000447](https://doi.org/10.1061/(ASCE)CP.1943-5487.0000447).
- Kaseko, M. S., Z. P. Lo, and S. G. Ritchie. 1994. "Comparison of traditional and neural classifiers for pavement crack detection." *J. Transp. Eng.* 120 (4): 552–569. [https://doi.org/10.1061/\(ASCE\)0733-947X\(1994\)120:4\(552\)](https://doi.org/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* 18 (10): 3452. <https://doi.org/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." *Can. J. Civ. Eng.* 29 (3): 459–474. <https://doi.org/10.1139/I02-018>.
- Kirschke, K. R., and S. A. Velinsky. 1992. "Histogram-based approach for automated pavement-crack sensing." *J. Trans. Eng.* 118 (5): 700–710. [https://doi.org/10.1061/\(ASCE\)0733-947X\(1992\)118:5\(700\)](https://doi.org/10.1061/(ASCE)0733-947X(1992)118:5(700)).
- Koch, C., and I. Brilakis. 2011. "Pothole detection in asphalt pavement images." *J. Adv. Eng. Inf.* 25 (3): 507–515. <https://doi.org/10.1016/j.aei.2011.01.002>.
- Koch, C., G. M. Jog, and I. Brilakis. 2013. "Automated pothole distress assessment using asphalt pavement video data." *J. Comput. Civ. Eng.* 27 (4): 370–378. [https://doi.org/10.1061/\(ASCE\)CP.1943-5487.0000232](https://doi.org/10.1061/(ASCE)CP.1943-5487.0000232).
- Krizhevsky, A., I. Sutskever, and G. E. Hinton. 2012. "ImageNet classification with deep convolutional neural network." In *Proc., Advances in Neural Information Processing Systems*. La Jolla, CA: Neural Information Processing Systems Foundation.
- Lattanzi, D., and G. Miller. 2014. "Robust automated concrete damage detection algorithms for field applications." *J. Comput. Civ. Eng.* 28 (2): 253–262. [https://doi.org/10.1061/\(ASCE\)CP.1943-5487.0000257](https://doi.org/10.1061/(ASCE)CP.1943-5487.0000257).
- LeCun, Y., L. Bottou, Y. Bengio, and P. Haffner. 1998. "Gradient-based learning applied to document recognition." *Proc. IEEE* 86 (11): 2278–2324. <https://doi.org/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 *Proc., Int. Congress on Image Signal Processing*, 792–796. New York: IEEE.
- Lin, T. Y., M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. 2014. "Microsoft COCO: Common objects in context." In *Microsoft COCO: Common Objects in Context*, 740–755. Cham, Switzerland: Springer.
- Mohan, A., and S. Poobal. 2018. "Crack detection using image processing: A critical review and analysis." *Alexandria Eng. J.* 57 (2): 787–798. <https://doi.org/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 *Proc., 28th ISARC*, 1461–1467. Seoul, Korea: International Association for Automation and Robotics in Construction.
- Nejad, F. M., and H. Zakeri. 2011. "An expert system based on wavelet transform and radon neural network for pavement distress classification." *Expert Syst. Appl. Int. J.* 38 (6): 7088–7101. <https://doi.org/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." *Comput.-Aided Civ. Infrastruct. Eng.* 27 (1): 29–47. <https://doi.org/10.1111/j.1467-8667.2011.00716.x>.
- NYSDOT Office of Structure. 2016. *Bridge manual inspection*. Albany, NY: New York State Dept. of Transportation.
- O'Byrne, M., F. Schoefs, B. Ghosh, and V. Pakrashi. 2013. "Texture analysis based damage detection of ageing infrastructural elements." *Comput.-Aided Civ. Infrastruct. Eng.* 28 (3): 162–177. <https://doi.org/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 *Proc., Int. Conf. on Imaging Technologies: Techniques and Applications in Civil Engineering*, 138–147. Cambridge, MA: Society for Imaging Science and Technology.
- Oh, J. K., G. Jang, S. Oh, J. H. Lee, B. J. Yi, Y. S. Moon, J. S. Lee, and Y. Choi. 2009. "Bridge inspection robot system with machine vision." *Autom. Constr.* 18 (7): 929–941. <https://doi.org/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 *Proc., European Signal Processing Conf. (EUSPICO'09)*, 622–626. Glasgow, Scotland: EURASIP.
- Pan, S. J., and Q. Yang. 2010. "A survey on transfer learning." *IEEE Trans. Knowl. Data Eng.* 22 (10): 1345–1359. <https://doi.org/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." *J. Comput. Civ. Eng.* 33 (3): 04019017. [https://doi.org/10.1061/\(ASCE\)CP.1943-5487.0000831](https://doi.org/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 Trans. Pattern Anal. Mach. Intell.* 39 (6): 1137–1149. <https://doi.org/10.1109/TPAMI.2016.2577031>.
- Saar, T., and O. Talvik. 2010. "Automatic asphalt pavement crack detection and classification using neural network." In *Proc., 2010 12th Biennial Baltic Electronics Conf.*, 345–348. New York: IEEE.
- Schmugge, S. J., 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 *Proc., 2016 IEEE Winter Conf. on Applications of Computer Vision (WACV)*, 1–7. New York: IEEE.
- Sermanet, P., K. Kavukcuoglu, S. Chintala, and Y. LeCun. 2013. "Pedestrian detection with unsupervised multi-stage feature learning." In *Proc., 26th IEEE Conf. on Computer Vision and Pattern Recognition (CVPR '13)*, 3626–3633. New York: IEEE.
- Shi, Y., L. Cui, Z. Qi, F. Meng, and Z. Chen. 2016. "Automatic road crack detection using random structured forests." *IEEE Trans. Intell. Transp. Syst.* 17 (12): 3434–3445. <https://doi.org/10.1109/TITS.2016.2552248>.
- Sinha, S. K., and P. W. Fieguth. 2006. "Automated detection of cracks in buried concrete pipe images." *Autom. Constr.* 15 (1): 58–72. <https://doi.org/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." *Comput. Aided Civ. Infrastruct. Eng.* 18 (2): 95–112. <https://doi.org/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." *J. Comput. Civ. Eng.* 30 (6): 04016032. [https://doi.org/10.1061/\(ASCE\)CP.1943-5487.0000597](https://doi.org/10.1061/(ASCE)CP.1943-5487.0000597).
- Song, M., and D. Civco. 2004. "Road extraction using SVM and image segmentation." *Photogramm. Eng. Remote Sens.* 70 (12): 1365–1371. <https://doi.org/10.14358/PERS.70.12.1365>.
- Subiras, P., J. Dumoulin, V. Legay, and D. Barba. 2006. "Automation of pavement surface crack detection using the continuous wavelet transform." In *Proc., Int. Conf. on Image Processing*, 3037–3040. New York: IEEE.
- Szegedy, C., S. Ioffe, V. Vanhoucke, and A. A. Alemi. 2017. "Inception-v4, inception-resnet and the impact of residual connections on learning." In *Proc., 31st AAAI Conf. on Artificial Intelligence*, 4278–4284. San Francisco: Association for the Advancement of Artificial Intelligence.
- Taha, A. A., and A. Hanbury. 2015. "Metrics for evaluating 3D medical image segmentation: Analysis, selection, and tool." *BMC Med. Imaging* 15 (1): 29. <https://doi.org/10.1186/s12880-015-0068-x>.
- Tong, Z., J. Gao, Z. Han, and Z. Wang. 2017. "Recognition of asphalt pavement crack length using deep convolutional neural networks."

- Road Mater. Pavement Des.* 19 (6): 1334–1349. <https://doi.org/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.” *J. Comput. Civ. Eng.* 28 (5): A4014004. [https://doi.org/10.1061/\(ASCE\)CP.1943-5487.0000334](https://doi.org/10.1061/(ASCE)CP.1943-5487.0000334).
- Tsai, Y., V. Kaul, and R. M. Mersereau. 2010. “Critical assessment of pavement distress segmentation methods.” *J. Transp. Eng.* 136 (1): 11–19. [https://doi.org/10.1061/\(ASCE\)TE.1943-5436.0000051](https://doi.org/10.1061/(ASCE)TE.1943-5436.0000051).
- Tsao, S., N. Kehtarnavaz, P. Chan, and R. Lytton. 1994. “Image-based expert system approach to distress detection on CRC pavement.” *J. Transp. Eng.* 120 (1): 52–64. [https://doi.org/10.1061/\(ASCE\)0733-947X\(1994\)120:1\(52\)](https://doi.org/10.1061/(ASCE)0733-947X(1994)120:1(52)).
- Tzutalin. 2015. “LabelImg Git Code.” Accessed May 1, 2018. <https://github.com/tzutalin/labelImg>.
- Vaillant, R., C. Monrocq, and Y. LeCun. 1994. “Original approach for the localization of objects in images.” In Vol. 141 of *IEEE Proc.: Vision, Image and Signal Processing*, 245–250. New York: IEEE.
- Vargas, R., A. Mosavi, and L. Ruiz. 2017. “Deep learning: A review.” In *Advances in intelligent systems and computing*. Berlin: Springer.
- Wang, K. C., S. Nallamothu, and R. P. Elliot. 1998. “Classification of pavement surface distress with an embedded neural net chip.” In *Artificial neural networks for civil engineers: Advanced features and applications*, 131–161. Reston, VA: ASCE.
- Wang, K. C. P., Q. Li, and W. Gong. 2007. “Wavelet-based pavement distress image edge detection with à Trou algorithm.” *Transp. Res. Rec.* 2024 (1): 73–81. <https://doi.org/10.3141/2024-09>.
- Wang, R. 2013. “3D building modeling using images and LiDAR: A review.” *Int. J. Image Date Fusion* 4 (4): 273–292. <https://doi.org/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.” *J. Comput. Eng.* 30 (1): 04014118. [https://doi.org/10.1061/\(ASCE\)CP.1943-5487.0000451](https://doi.org/10.1061/(ASCE)CP.1943-5487.0000451).
- Xian, Y., C. H. Lampert, B. Schiele, and Z. Akata. 2018. “Zero-shot learning: A comprehensive evaluation of the good, the bad and the ugly.” In *Proc., IEEE Transactions on Pattern Analysis and Machine Intelligence*. New York: IEEE.
- Yamaguchi, T., and S. Hashimoto. 2006. “Automated crack detection for concrete surface image using percolation model and edge information.” In *Proc., 32nd Annual Conf. on IEEE Industrial Electronics*, 3355–3360. New York: IEEE.
- Yamaguchi, T., S. Nakamura, R. Saegusa, and S. Hashimoto. 2008. “Image-based crack detection for real concrete surfaces.” *IEEJ Trans. Electr. Electron. Eng.* 3 (1): 759–770.
- 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.” *J. Autom. Constr.* 16 (3): 255–261. <https://doi.org/10.1016/j.autcon.2006.05.003>.
- Zhang, A., K. C. Wang, Y. Fei, Y. Liu, S. Tao, C. Chen, J. Q. Li, and B. Li. 2018a. “Deep learning-based fully automated pavement crack detection on 3D asphalt surfaces with an improved cracknet.” *J. Comput. Civ. Eng.* 32 (5): 04018041. [https://doi.org/10.1061/\(ASCE\)CP.1943-5487.0000775](https://doi.org/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.” *J. Comput. Civ. Eng.* 32 (2): 04018001. [https://doi.org/10.1061/\(ASCE\)CP.1943-5487.0000736](https://doi.org/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 *Proc., 2016 IEEE Int. Conf. on Image Processing (ICIP)*, 3708–3712. New York: IEEE.
- Zhou, J., P. S. Huang, and F. P. Chiang. 2006. “Wavelet-based pavement distress detection and evaluation.” *Opt. Eng.* 45 (2): 027007. <https://doi.org/10.1117/1.2172917>.
- Zhu, Z., and I. Brilakis. 2010. “Machine vision-based concrete surface quality assessment.” *J. Constr. Eng. Manage.* 136 (2): 210–218. [https://doi.org/10.1061/\(ASCE\)CO.1943-7862.0000126](https://doi.org/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.” *J. Autom. Const.* 20 (7): 874–883. <https://doi.org/10.1016/j.autcon.2011.03.004>.

# Crack Detection and Segmentation Using Deep Learning with 3D Reality Mesh Model for Quantitative Assessment and Integrated Visualization

Kalfarisi, Rony; Wu, Zheng Yi; Soh, Ken

- 
- 01 Eric Bianchi Page 1  
1/6/2020 16:53
- 02 Eric Bianchi Page 1  
1/6/2020 16:53
- 03 Eric Bianchi Page 1  
1/6/2020 16:53
- 04 Eric Bianchi Page 2  
1/6/2020 18:43
- 05 Eric Bianchi Page 2  
1/6/2020 18:43
- 06 Eric Bianchi Page 2  
1/6/2020 18:44
- 07 Eric Bianchi Page 2  
1/6/2020 18:44
- 08 Eric Bianchi Page 2  
1/6/2020 18:44
- 09 Eric Bianchi Page 2  
1/6/2020 18:44

- 10** Eric Bianchi Page 2  
1/6/2020 18:45
- 11** Eric Bianchi Page 3  
1/6/2020 18:45
- 12** Eric Bianchi Page 3  
1/6/2020 18:45
- 13** Eric Bianchi Page 3  
1/6/2020 18:45
- 14** Eric Bianchi Page 3  
1/6/2020 18:46
- 15** Eric Bianchi Page 3  
1/6/2020 18:46
- 16** Eric Bianchi Page 4  
1/6/2020 19:21
- 17** Eric Bianchi Page 5  
1/6/2020 18:48
- 18** Eric Bianchi Page 5  
1/6/2020 18:48
- 19** Eric Bianchi Page 5  
1/6/2020 18:47
- 20** Eric Bianchi Page 5  
1/6/2020 18:48
- 21** Eric Bianchi Page 5  
1/6/2020 18:47

- 22** Eric Bianchi Page 5  
1/6/2020 18:49
- 23** Eric Bianchi Page 6  
1/6/2020 18:49
- 24** Eric Bianchi Page 6  
1/6/2020 18:50
- 25** Eric Bianchi Page 6  
1/6/2020 18:50
- 26** Eric Bianchi Page 6  
1/6/2020 18:50
- 27** Eric Bianchi Page 6  
1/6/2020 18:50
- 28** Eric Bianchi Page 6  
1/6/2020 18:50
- 29** Eric Bianchi Page 6  
1/6/2020 18:49
- 30** Eric Bianchi Page 6  
1/6/2020 18:49
- 31** Eric Bianchi Page 6  
1/6/2020 19:35
- 32** Eric Bianchi Page 6  
1/6/2020 18:50
- 33** Eric Bianchi Page 6  
1/6/2020 18:49

- 34 Eric Bianchi Page 6  
1/6/2020 18:51
- 35 Eric Bianchi Page 6  
1/6/2020 18:51
- 36 Eric Bianchi Page 6  
1/6/2020 18:49
- 37 Eric Bianchi Page 7  
1/6/2020 17:02
- 38 Eric Bianchi Page 7  
1/6/2020 20:02
- 39 Eric Bianchi Page 7  
1/6/2020 17:02
- 40 Eric Bianchi Page 8  
1/6/2020 17:04
- 41 Eric Bianchi Page 8  
1/6/2020 17:04
- 42 Eric Bianchi Page 8  
1/6/2020 17:03
- 43 Eric Bianchi Page 8  
1/6/2020 17:05
- 44 Eric Bianchi Page 8  
1/6/2020 20:03
- 45 Eric Bianchi Page 8  
1/6/2020 17:03

- 46 Eric Bianchi Page 8  
1/6/2020 17:03
- 47 Eric Bianchi Page 8  
1/6/2020 18:51
- 48 Eric Bianchi Page 8  
1/6/2020 18:51
- 49 Eric Bianchi Page 8  
1/6/2020 18:52
- 50 Eric Bianchi Page 8  
1/6/2020 17:04
- 51 Eric Bianchi Page 8  
1/6/2020 18:53
- 52 Eric Bianchi Page 9  
1/6/2020 18:54
- 53 Eric Bianchi Page 9  
1/6/2020 18:54
- 54 Eric Bianchi Page 9  
1/6/2020 18:55
- 55 Eric Bianchi Page 9  
1/6/2020 18:54
- 56 Eric Bianchi Page 9  
1/6/2020 18:55
- 57 Eric Bianchi Page 11  
1/6/2020 18:56

58 Eric Bianchi

Page 11

1/6/2020 18:56

59 Eric Bianchi

Page 12

1/6/2020 18:57

60 Eric Bianchi

Page 13

1/6/2020 18:57

61 Eric Bianchi

Page 13

1/6/2020 16:59

62 Eric Bianchi

Page 14

1/6/2020 17:00

63 Eric Bianchi

Page 14

1/6/2020 18:58

64 Eric Bianchi

Page 14

1/6/2020 18:58

65 Eric Bianchi

Page 17

1/6/2020 17:01

66 Eric Bianchi

Page 17

1/6/2020 18:58