LIDAR, DRONES AND BRIM FOR RAPID BRIDGE INSPECTION AND MANAGEMENT

FINAL PROJECT REPORT

by

Yelda Turkan, Ph.D.
Paolo Calvi, Ph.D.
Donghoon Ji

Oregon State University
University of Washington

Sponsorship
PacTrans, Oregon State University, and the University of Washington

for

Pacific Northwest Transportation Consortium (PacTrans)
USDOT University Transportation Center for Federal Region 10
University of Washington
More Hall 112, Box 352700
Seattle, WA 98195-2700

In cooperation with U.S. Department of Transportation,
Office of the Assistant Secretary for Research and Technology (OST-R)

DISCLAIMER

The contents of this report reflect the views of the authors, who are responsible for the facts and the accuracy of the information presented herein. This document is disseminated under the sponsorship of the U.S. Department of Transportation's University Transportation Centers Program, in the interest of information exchange. The Pacific Northwest Transportation Consortium, the U.S. Government and matching sponsor assume no liability for the contents or use thereof.

TECHNICAL REPORT DOCUMENTATION PAGE

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
	01784893	

4. Title and Subtitle	5. Report Date
LIDAR, DRONES AND BRIM FOR RAPID BRIDGE INSPECTION AND MANAGEMENT	02/20/2023
	6. Performing Organization Code

7. Author(s) and Affiliations	8. Performing Organization Report No.
Yelda Turkan, Ph.D., 0000-0002-3224-5462; Oregon State University Paolo Calvi, Ph.D., 0000-0002-0051-2677; University of Washington Donghoon Ji, Oregon State University	2021-M-OSU-3

9. Performing Organization Name and Address	10. Work Unit No. (TRAIS)
PacTrans Pacific Northwest Transportation Consortium University Transportation Center for Federal Region 10 University of Washington More Hall 112 Seattle, WA 98195-2700	**11. Contract or Grant No.** 69A355174110

12. Sponsoring Organization Name and Address	13. Type of Report and Period Covered
United States Department of Transportation Research and Innovative Technology Administration 1200 New Jersey Avenue, SE Washington, DC 20590	Final Report, 03/16/2021 - 02/23/2023 **14. Sponsoring Agency Code**

15. Supplementary Notes

Report uploaded to: www.pactrans.org

16. Abstract

Cracks are one of the main defects on concrete surfaces, and they are indicators of concrete structures' state of health. Because traditional methods to identify and assess cracks rely on manual measurements, a significant number of studies have focused on identifying ways to automate this process. Accordingly, this study proposed to combine convolutional neural network (CNN)-based algorithms and traditional image morphological operations for crack detection and measurement. The proposed approach was tested on data from six images containing ten cracks of various sizes and shapes that were obtained from laboratory experiments in a controlled environment. The proposed methodology achieved an average F1 score of 0.93, with 88.17 percent accuracy in crack length measurements and 94.40 percent accuracy in width measurements. Future research will focus on fine-tuning the proposed crack detection and measurement methodology and will evaluate it with a set of images acquired from a full-scale structure such as a bridge or a building.

17. Key Words	18. Distribution Statement
Cracking, Automation, Structural Image Processing, Neural Networks, Edge Detection, Measurement	

19. Security Classification (of this report)	20. Security Classification (of this page)	21. No. of Pages	22. Price
Unclassified.	Unclassified.	35	N/A

Form DOT F 1700.7 (8-72) **Reproduction of completed page authorized.**

SI* (MODERN METRIC) CONVERSION FACTORS

APPROXIMATE CONVERSIONS TO SI UNITS

Symbol	When You Know	Multiply By	To Find	Symbol
		LENGTH		
in	inches	25.4	millimeters	mm
ft	feet	0.305	meters	m
yd	yards	0.914	meters	m
mi	miles	1.61	kilometers	km
		AREA		
in^2	square inches	645.2	square millimeters	mm^2
ft^2	square feet	0.093	square meters	m^2
yd^2	square yard	0.836	square meters	m^2
ac	acres	0.405	hectares	ha
mi^2	square miles	2.59	square kilometers	km^2
		VOLUME		
fl oz	fluid ounces	29.57	milliliters	mL
gal	gallons	3.785	liters	L
ft^3	cubic feet	0.028	cubic meters	m^3
yd^3	cubic yards	0.765	cubic meters	m^3
	NOTE: volumes greater than 1000 L shall be shown in m^3			
		MASS		
oz	ounces	28.35	grams	g
lb	pounds	0.454	kilograms	kg
T	short tons (2000 lb)	0.907	megagrams (or "metric ton")	Mg (or "t")
		TEMPERATURE (exact degrees)		
°F	Fahrenheit	5 (F-32)/9 or (F-32)/1.8	Celsius	°C
		ILLUMINATION		
fc	foot-candles	10.76	lux	lx
fl	foot-Lamberts	3.426	$candela/m^2$	cd/m^2
		FORCE and PRESSURE or STRESS		
lbf	poundforce	4.45	newtons	N
lbf/in^2	poundforce per square inch	6.89	kilopascals	kPa

APPROXIMATE CONVERSIONS FROM SI UNITS

Symbol	When You Know	Multiply By	To Find	Symbol
		LENGTH		
mm	millimeters	0.039	inches	in
m	meters	3.28	feet	ft
m	meters	1.09	yards	yd
km	kilometers	0.621	miles	mi
		AREA		
mm^2	square millimeters	0.0016	square inches	in^2
m^2	square meters	10.764	square feet	ft^2
m^2	square meters	1.195	square yards	yd^2
ha	hectares	2.47	acres	ac
km^2	square kilometers	0.386	square miles	mi^2
		VOLUME		
mL	milliliters	0.034	fluid ounces	fl oz
L	liters	0.264	gallons	gal
m^3	cubic meters	35.314	cubic feet	ft^3
m^3	cubic meters	1.307	cubic yards	yd^3
		MASS		
g	grams	0.035	ounces	oz
kg	kilograms	2.202	pounds	lb
Mg (or "t")	megagrams (or "metric ton")	1.103	short tons (2000 lb)	T
		TEMPERATURE (exact degrees)		
°C	Celsius	1.8C+32	Fahrenheit	°F
		ILLUMINATION		
lx	lux	0.0929	foot-candles	fc
cd/m^2	$candela/m^2$	0.2919	foot-Lamberts	fl
		FORCE and PRESSURE or STRESS		
N	newtons	0.225	poundforce	lbf
kPa	kilopascals	0.145	poundforce per square inch	lbf/in^2

*SI is the symbol for the International System of Units. Appropriate rounding should be made to comply with Section 4 of ASTM E380.
(Revised March 2003)

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

3D:	Three-dimensional
ACC:	Accuracy
ASCE:	American Society of Civil Engineers
CNN:	Convolutional neural network
F1:	F1 score
FN:	False negative
FP:	False positive
NPV:	Negative prediction value
PPV:	Positive prediction value
TN:	True negative
TNR:	True negative rate
TP:	True positive
TPR:	True positive rate
UAS:	Unmanned aerial system
UHPC:	Ultra-high-performance concrete
UW:	University of Washington

ACKNOWLEDGMENTS

The authors would like to thank the Pacific Northwest Transportation Consortium (PacTrans) for funding this research. The authors would also like to thank John Paul Gaston for his help with data collection.

EXECUTIVE SUMMARY

Cracks are one of the main defects on concrete surfaces, and they are indicators of concrete structures' state of health. Because traditional methods to identify and assess cracks rely on manual measurements, a significant number of studies have focused on identifying ways to automate this process. Accordingly, this study proposed to combine convolutional neural network (CNN)-based algorithms and traditional image morphological operations for crack detection and measurement. The proposed approach was tested on data from six images containing ten cracks of various sizes and shapes that were obtained from laboratory experiments in a controlled environment. The proposed methodology achieved an average F1 score of 0.93, with 88.17 percent accuracy in crack length measurement and 94.40 percent accuracy in width measurement. Future research will focus on fine-tuning the proposed crack detection and measurement methodology and will evaluate it with a set of images acquired from a full-scale structure such as a bridge or a building

.

CHAPTER 1. INTRODUCTION

The American Society of Civil Engineers (ASCE) reported that approximately 231,000 bridges across the United States, corresponding to 37.4 percent of the nation's bridges, are in critical condition and need to be repaired immediately. Moreover, 46,154, or 7.5 percent of all bridges in the nation, are considered structurally deficient (ASCE 2021). The condition of these bridges is established through periodic inspections that are conducted manually, a process that is time consuming, labor intensive, and prone to error. Moreover, when rapid reconnaissance to determine the condition of a structure is needed, right after catastrophic disasters such as earthquakes or hurricanes, traditional inspections cannot guarantee optimal results. Therefore, various efforts to establish automated approaches for structural inspections have been proposed.

For structural assessments, visual inspections, which mainly focus on the identification and assessment of surface defects, are very important. Cracks on concrete surfaces are considered to be main defect features, and they are the major indicators of the stability and durability of concrete structures (Chen et al. 2021). Therefore, the identification and assessment of cracks play a big role in concrete structure inspections, and the relevant studies have focused on automating the process of crack detection and measurement by leveraging computer vision and machine learning techniques.

To date, various image-based crack detection techniques have been proposed to automate crack identification and assessment. Several of the studies proposed crack detection methods have relied on image enhancement and have implemented various computer vision techniques such as edge detection algorithms (Kanopoulos et al. 2002, Wang and Zhou 2008, Li et al. 2009), automatic threshold image segmentation (Otsu 1979), and image morphological operations (Nguyen et al. 2011, Xu and Turkan 2020). Not only have these studies achieved reliable results in terms of accuracy, but they also have provided a well-defined base for automated crack assessment. However, most of these studies have been performed on ideal test images containing cracks with clear edge-gradient changes and minimal numbers of non-crack features such as stains and shadows.

Simultaneously, deep learning algorithms, including several convolutional neural networks (CNN) architectures, have been proposed and utilized to improve the robustness and the accuracy of crack detection results, especially when they are applied to images containing stains and shadows (Krizhevsky et al. 2012, Simonyan and Zisserman, 2015, Szegedy et al.

2015, He et al. 2016). These CNN-based crack detection methods have a significant advantage over the computer vision-based techniques described above because they discard non-crack features from the original images. As a result, they provide better results than the results that can be obtained with conventional image morphological operations (Liu et al. 2019, Yang et al. 2018, Gopalakrishnan et al. 2017, Cha et al. 2017, Lee et al. 2019, Islam and Kim, 2019. Bhowmick et al. 2020), and they offer new directions for automated crack assessment, which can potentially replace manual measurements. Unfortunately, like other deep learning-based algorithms, larges number of training images and significant computing resources are required to ensure their performance.

Recent studies have focused on overcoming these limitations to achieve a high accuracy when images obtained from real-life structures are used. Accordingly, they have proposed to combine different techniques to compensate for each other's limitations, such as combining the results obtained with a deep learning-based technique applied to digital images with the results obtained using a 3D point cloud. Mohan and Poobal (2018) concluded that the fusion of various existing approaches can provide better crack detection results.

Accordingly, this study proposed a new methodology for automated crack detection and measurement. First, a crack detection approach combining conventional image morphological operations and a CNN-based algorithm was proposed. Next, skeletonization and orthogonal projection algorithms were utilized to measure the length and width of cracks. Finally, the overall performance of the proposed crack detection and measurement approaches were evaluated with data obtained from laboratory experiments in a controlled environment. Because the test images used in this study contained cracks with various branches that had different widths, the overall performance of the methodology used in this study could be evaluated comprehensively, i.e., the accuracy and the sensitivity of the methodology could be evaluated for cracks with different types and shapes.

CHAPTER 2. LITERATURE REVIEW

This section provides a comprehensive review of previous studies that focused on crack detection and measurements. The crack detection methods are based on conventional image morphological operations and deep learning algorithms, and the approaches that combine multiple methods are reviewed to identify their strengths and weaknesses. Relevant studies on crack measurements are also reviewed to identify an optimal method for measuring crack length and width.

2.1. Crack Detection

Previous studies that focused on automating crack detection mainly utilized digital images. As such, conventional object detection and segmentation algorithms have played a major role in image-based crack detection. Researchers have focused on detecting the edges of cracks by using various edge detection methods, such as Sobel, Prewitt, and Canny (Kanopoulos et al. 2002, Wang and Zhou 2008, Li et al. 2009). Using these methods, crack edges, where prominent gradient changes are especially intense in certain directions along the feature of interest, can be detected. Unfortunately, if the gradient changes are not significant, meaning that the crack is not clearly distinguishable, the overall crack detection performance decreases drastically.

To effectively segment any crack, whether gradient changes were distinctive or not, Otsu (1979) presented a segmentation method based on image threshold that computed the maximum grayscale intensity value of an image and utilized it as a threshold. Nguyen et al. (2011) proposed the Free-Form Anisotropy method, which simultaneously considered not only intensity values but also texture and other crack features to detect cracks accurately. Several studies applied these two approaches to data collected from actual structures. For instance, on the basis of image morphological operations proposed by Otsu (1979), Xu and Turkan (2020) introduced an automated crack detection approach tested with images of a bridge obtained with an unmanned aircraft system (UAS). They also utilized image pre-processing steps, such as contrast adjustment and noise reduction, to improve their crack detection results. Although most crack detection methods based on traditional image morphological operations have yielded meaningful results, they do not perform as well when they are applied to images that contain shadows or stains, i.e., features that are similar to cracks.

In the meantime, deep learning-based feature detection methods, which became the mainstream application for object/feature segmentation, were also utilized for crack detection.

The major CNN-based vision architectures, such as AlexNet (Krizhevsky et al. 2012), VGG-Net (Simonyan and Zisserman 2015), Inception Network (Szegedy et al. 2015), and ResNet (He et al. 2016), offered improved results in detecting and classifying target objects. Consequently, these CNN-based vision architectures were adopted for image-based crack detection in several recent studies. For example, Dorafshan et al. (2018) focused on comparing the performance of conventional edge detectors (Roberts, Prewitt, Sobel, Laplacian of Gaussian, Butterworth, and Gaussian) and a CNN-based crack detector. In terms of the CNN-based crack detector, an AlexNet crack detector with fully trained, partially re-trained, and pre-trained datasets were tested to determine their performance under different circumstances, and images from and SDNET dataset that contained various surface defects were used as training, validation, and testing datasets (Maguire et al. 2018). They concluded that the CNN-based crack detector performed best, resulting in 99 percent precision and 66 percent recall values when the fully trained dataset was used. The precision values for partially trained and pre-trained datasets were 92 percent and 80 percent, respectively, while the recall values were 86 percent and 84 percent, respectively. Meanwhile, the precision and recall values obtained with the conventional edge detector based on the Laplacian of Gaussian algorithms were 60 percent and 79 percent, respectively. Wang et al. (2021) evaluated the crack detection accuracy of six existing CNNs (VGG 16, Inception V2, V3, V4, Inception-ResNet-v2, and ResNet V1 50) using an image dataset obtained from the inspection of a slab element, and the highest accuracy, 80.1 percent, was achieved when the Inception-ResNet-v2 network was used.

As discussed above, the most recent studies on automated crack detection combined different methods to improve the accuracy and reliability of the results. Several studies focused on augmenting digital images with three-dimensional (3D) data such as point clouds acquired with a laser scanner. This concept was intended to help solve issues related to the interference of non-crack features such as shadows because the laser scan data were not affected by light or shadow. Chen et al. (2021) used point cloud data to obtain a depth image, which classified different features in the image on the basis of their measured depth. In the next step, the depth image was used together with the pre-processed image, and by adopting Otsu's segmentation algorithm, the proposed approach achieved, on average, 89.0 percent precision, 84.8 percent recall, and 86.7 percent F1 scores. (The F1 score is a machine learning evaluation metric that measures a model's accuracy. It computes how many times a model made a correct prediction

across the entire dataset.) Yan et al. (2021) also utilized point cloud data for crack detection, which enabled them to segment and exclude background and other non-crack features, thus minimizing the impacts of shadows or stains during data processing, a major challenge for images obtained in real life (e.g., images of a bridge). They applied this method to images obtained from a bridge with a VGG16-based crack detector. By combining the results obtained with the point cloud and the images, they achieved 93 percent crack detection accuracy on average, and obtained a precision value of 93.9 percent and a recall value of 89.4 percent.

2.2. Crack Measurement

The majority of studies on image-based crack measurement have implemented a skeletonization algorithm to guarantee the pixel-wise accuracy of the measurement (Yan et al. 2021, Wang et al. 2021, Qiu et al. 2017). Using binary images to illustrate the crack and non-crack pixels as 1 and 0, the skeletonization algorithm shrank the crack area and created the centerline of the crack (the crack thickness equal to one pixel). From this centerline, referred to as the crack skeleton, the length of the crack could be measured by counting the number of pixels along the centerline. To measure crack width, which is more challenging, the most accurate method proposed to date involved calculating the continuous width of each pixel on the crack skeleton by using the orthogonal projection algorithm (Yan et al. 2021, Wang et al. 2021, Qiu et al. 2017). First, the orientation, i.e., the slope, of each crack pixel was computed by fitting a line over two neighboring pixels of the target pixel on the crack skeleton. Next, an orthogonal line perpendicular to the orientation of the crack at the target pixel was projected. This orthogonal line defined the direction of the crack width at the target pixel, and two intersecting points between the orthogonal line and crack boundaries defined each end of the crack width. As such, the width of the crack was calculated as the distance between these two intersecting points.

Qiu et al. (2017) evaluated the performance of crack width measurements based on the skeletonization and the orthogonal projection algorithms. Using images that contained ten different cracks, with widths ranging between 18.1 mm and 66.3 mm (the average width was 36.4 mm), they calculated a 1.4-mm difference, on average, between the ground truth values and their results. Yan et al. (2021) implemented a CNN-based algorithm, which utilized both digital images and point cloud data, and tested the performance of their crack measurement approach, which was based on skeletonization and orthogonal projection algorithms. The test images they used included cracks with lengths ranging between 44.5 mm and 559.0 mm and widths ranging

between 1.0 mm and 5.0 mm. The results showed error rates of ±3 percent for length measurements and ±8 percent for width measurements, on average. Wang et al. (2021), who also utilized skeletonization and orthogonal projection methods for crack measurements, focused on classifying the cracks into three categories based on average crack widths (i.e., severity levels). They used an image dataset containing cracks narrower than 1.0 mm on average, and the results showed that the cracks were successfully classified into three categories with an average accuracy of 97.41 percent.

One of the most essential steps in image-based crack measurements has been to convert the measured value from pixel dimensions into real scale, such as millimeters or inches (Ni et al. 2020). For instance, Wang et al. (2021) used the width of a railhead in the image as a reference. Because the actual width of the railhead can be identified from its specifications document, they were able to calculate the conversion factor easily. Although this approach was fairly straightforward, it could be used only if the image contained a certain target feature with known dimensions. Furthermore, if the angle of the camera was not parallel to the surface of the area of interest, the accuracy of the scale conversion was not guaranteed. Another approach for scale conversion was to utilize information from other sources. For example, Yan et al. (2021) used values of depth and focal length obtained from lidar data to compute the scale factor. Kalfarisi et al. (2020) utilized dimensions obtained from a 3D mesh model, which was reconstructed from 2D images. The 3D mesh model provided the dimensions in both pixels and metric units.

CHAPTER 3. METHODOLOGY

3.1. Crack Detection

This study used MATLAB™ image processing tools to implement an image-based crack detection methodology that combines a CNN-based crack segmentation algorithm called DeepCrack, proposed by Liu et al. (2019), and a traditional image morphological operation based on Otsu's image segmentation algorithm (1979). The first step in this methodology involves implementing DeepCrack, a CNN-based crack segmentation algorithm that is based on the VGG-16 network. To decrease the time required for training, the study used a pre-trained DeepCrack algorithm model that was trained with 4,800 images and tested (for validation) with 3,792 images. Note that the pre-trained DeepCrack model's global prediction accuracy was 98.73 percent, and its precision, recall, and F1 scores were 85.82 percent, 84.56 percent, and 85.18 percent, respectively (Liu et al. 2019). The next step of the methodology involves utilizing an image morphological operation that is based on Otsu's image segmentation algorithm, together with additional pre-processing (e.g., contrast adjustment, smoothing) and post-processing (e.g., area filtering) steps (Xu and Turkan 2019). The steps of the proposed crack detection methodology are detailed in Figure 3.1.

First, the original image is pre-processed with grayscale conversion, brightness adjustment, and contrast enhancement techniques, respectively. In particular, the brightness of the converted grayscale images is increased by 10 percent of the average gray level of each image. This reduces interference with light-shaded non-crack features such as stains or shadows to differentiate and accurately extract the target crack. Next, during the contrast enhancement step, the adjusted image is saturated with the bottom and top 25 percent of all pixel intensity values to identify the target crack.

After these three pre-processing steps, two different methods are applied to the pre-processed images in parallel: The first method involves applying the pre-trained DeepCrack model to the pre-processed image. The pre-trained DeepCrack algorithm produces a grayscale image with predicted cracks, and this image is post-processed by using image contrast saturation with the bottom and top 10 percent of all pixel intensity values, and then binarized.

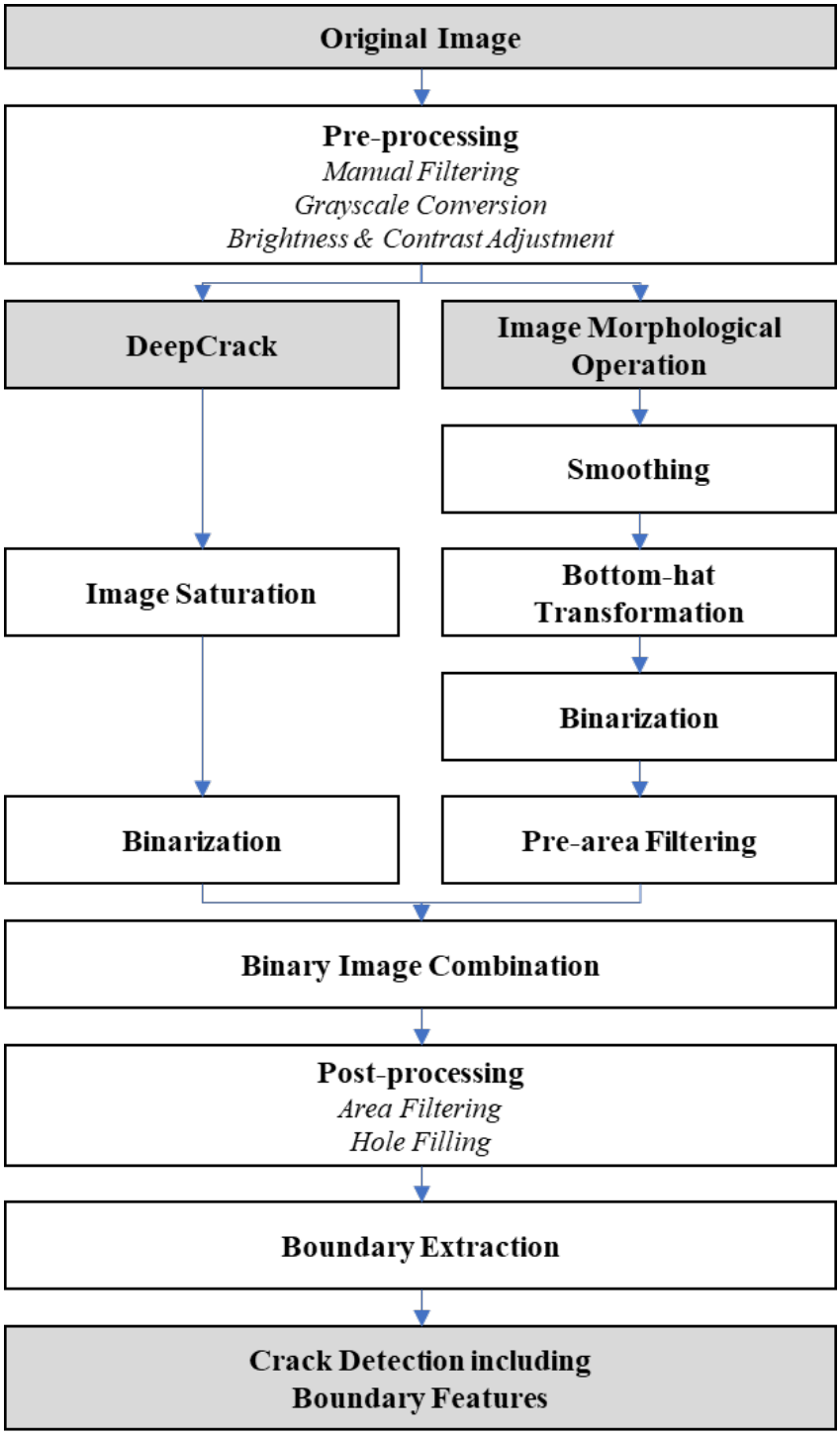

Figure 3.1 Crack Detection Procedure

The second method applied to the pre-processed image is a conventional image morphological operation process. This consists of procedures such as smoothing via median filtering to reduce the interference of non-crack features, bottom-hat transformation to extract

darker regions (that may be mistaken for target cracks) in the preprocessed image, and pre-area filtering that uses a threshold value to exclude smaller features in the image. More specifically, the median filter targets "salt and pepper" noise and aims to replace a pixel value with the median value of the surrounding three-by-three neighborhood. Bottom-hat transformation is used to extract the darker regions in an image by dilating the morphological closing of the image and subtracting it from the original image (Bai et al., 2012). Pre-area filtering is used to remove smaller features (smaller numbers of pixels) than the pre-determined threshold value. In this study, the threshold value was set at ten pixels to distinguish major cracks and smaller features that might be mistaken for cracks.

After this two-step process that implements both DeepCrack and image morphological operations, the preliminary results obtained from each step are integrated such that two binary images are simply combined to preserve the characteristics of each process. In the next step, the image is post-processed using second area filtering and a hole-filling operations. The second area filtering discards features that contain fewer than 30 pixels to preserve minor cracks that are detected by the DeepCrack algorithm. The hole-filling operation is applied to link features that make up the same crack. Finally, crack boundaries are extracted by using an image gradient threshold technique.

3.2. Crack Measurement

In this process, two algorithms, namely skeletonization and orthogonal projection algorithms, are used (Figure 3.2) for crack measurements. The binary image obtained at the end of the crack detection process (detailed in Section 3.1) is first skeletonized. Because the original crack skeleton contains several branches that are not associated with the crack length or the main orientation of the crack, a post-processing method is used to filter those branches (i.e., pruning). Next, an orthogonal projection algorithm is applied to the crack skeleton. The orientation of the crack skeleton is computed for each pixel, and the target pixel's orthogonal line is projected. By merging these projected orthogonal lines and the binary image of crack boundaries (obtained from the crack detection step), two intersecting points are obtained between the orthogonal line and crack boundaries. As detailed in Chapter 2, crack length is measured by counting the number of pixels along the crack skeleton, and the width of the crack is measured by calculating the distance between the orthogonal line and crack boundaries. Finally, the crack length and width

dimensions, which have been calculated in number of pixels, are converted into metric dimensions by using the actual dimensions of the specimen used in this study.

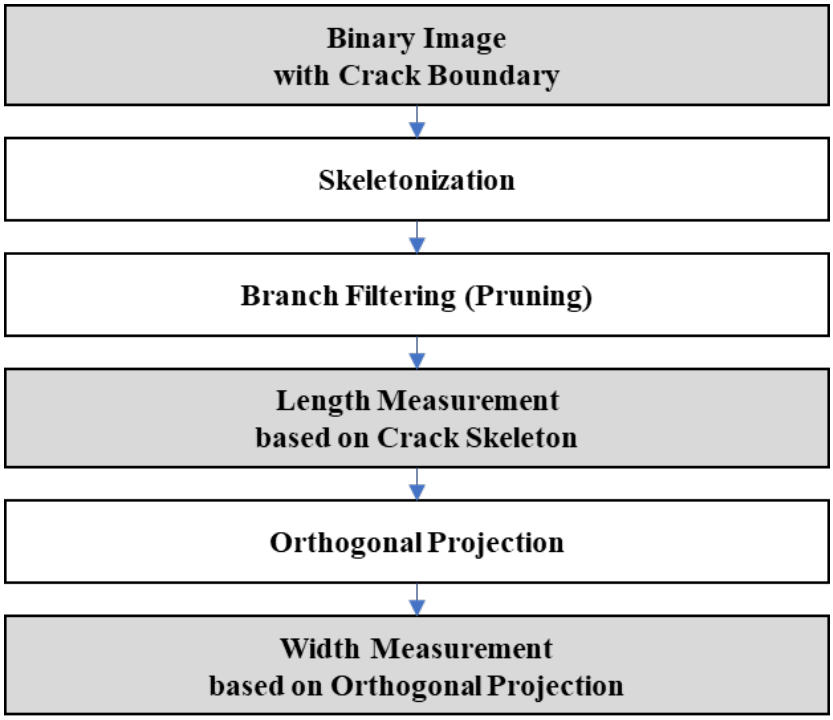

Figure 3.2 Crack Measurement Procedure

CHAPTER 4. EXPERIMENTS

The proposed methodology described in Chapter 3 was applied to six images containing ten cracks obtained from shear strength experiments conducted at the University of Washington (UW) Structures Laboratory. Detailed information about the experimental data and procedure is presented in this section.

4.1. Experimental Data

Six images obtained from shear strength experiments conducted on ultra-high-performance concrete (UHPC) were used in this test (Voytko et al. 2022). For each experiment, an 890 × 890 × 70 mm UHPC panel was tested under shear loads using the UW Panel Element Tester, and cracks in various shapes were expected to occur. With this series of experiments, large-sized cracks occurred along with multiple smaller and thin cracks. These large-sized cracks were great for testing the performance of the proposed crack detection and measurement methods, as they included multiple thin crack branches.

Because the original images contained excessive amounts of non-target features, such as the panel test frame and markers indicating the cracks, the test images were manually cropped and trimmed to extract the regions of interest, i.e., large cracks and their branches. When the region of interest in an image was extracted, the shape of the target crack was preserved, and the properties of the original images were kept as-is (Figure 4.1). In the meantime, the background containing the panel test frame and the markers were manually removed from the image. The size of the manually cropped test images were 1808 × 1748, 1350 × 1394, 2044 × 2011, 2358 × 2336, 2299 × 2263, and 2146 × 2055 pixels, respectively.

Test images 1 and 2 both contained one large crack with multiple branches. Test image 3 contained one circular defect and a large crack, but the purpose of this test was to extract the crack (Crack 3-2), not the circular defect. Test images 4, 5, and 6 contained multiple cracks of various shapes and sizes. When the crack detection algorithm was run on test images 4, 5 and 6, the cracks were defined on the basis of the segmentation results (i.e., if there was a clear, connected white panel area between each crack, these cracks were regarded as separate cracks). On the basis of the segmentation results, the test image 4 was found to contain three cracks, and test images 5 and 6 both contained two cracks. To summarize, a total of ten cracks from six images were extracted by using the proposed crack detection and measurement methods in this study.

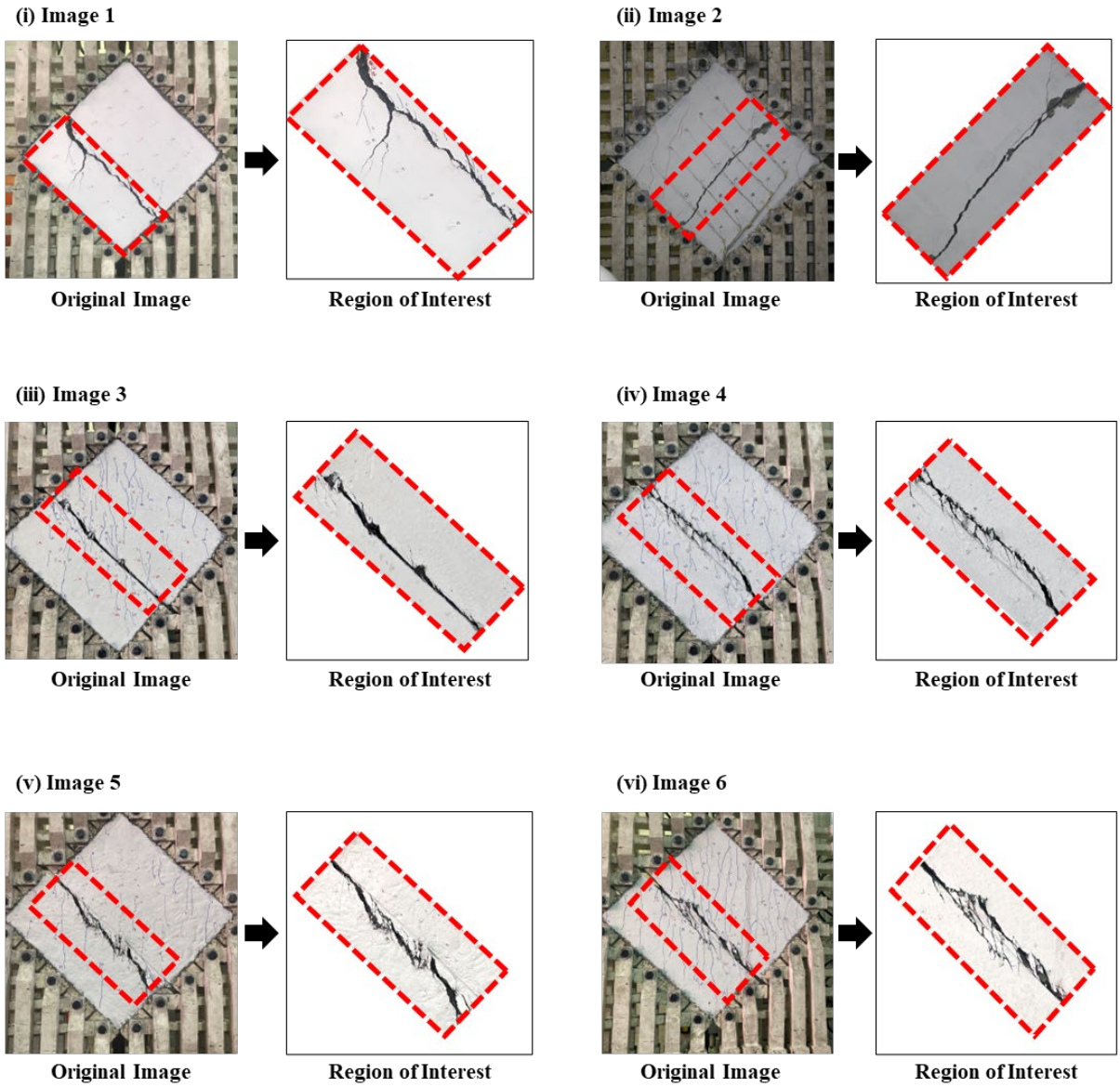

Figure 4.1 Original Images (Left) and Region of Interest in the Test Images (Right)

4.2. Experimental Procedure

The six test images containing ten cracks were processed with the crack detection and measurement methods proposed in this study. First, the performance of the crack detection method was evaluated. By utilizing manually annotated ground truth, the precision, recall, and F1 score values of the crack detection method were calculated. The ground truths (crack borders) used in this evaluation were selected on the basis of human judgment. In the next step, using the

manually annotated borders, the test images were converted into binary images, i.e., images made of crack and non-crack pixels.

In the next step, the automatically measured crack length values were compared to the manually annotated length values. To calculate the ground truth for crack lengths, each crack was divided into ten segments and measured manually. The orientation of each segment was preserved to enhance the overall accuracy. The ground truth for the crack length was calculated by adding up the length of each segment.

To determine the performance of width measurements, five checkpoints along a given crack were selected in each test image, and the automatically measured width values at those checkpoints were compared to the manually annotated width values. Similarly, the ground truths for crack widths were measured manually. The crack width at each checkpoint was measured manually, and the orientation of crack widths were determined on the basis of the proposed approach in this study to compare the performance of width measurements.

CHAPTER 5. RESULTS

As detailed in Chapter 4, the overall performance of the proposed crack detection and measurement methodology was evaluated by using data obtained from shear strength experiments conducted at the UW Structures Laboratory. First, crack detection accuracy was evaluated against manually annotated ground truths. Furthermore, to determine the limitations of the proposed crack detection approach, crack detail losses were evaluated by using the maximum widths of those crack sections. The accuracy of the crack measurement results was evaluated by comparing them to manually measured ground truths. Similarly, to determine the limitations of the proposed crack measurement approach, detail losses that occurred during the branch filtering step (i.e., pruning) were calculated.

5.1. Crack Detection

The ground truth images used to evaluate the crack detection performance were generated manually. Starting from the original test images, the boundaries of the target cracks were determined on the basis of human judgment. Next, the target crack boundaries were annotated. Following the annotation step, these images were converted into binary images to differentiate between crack and non-crack pixels.

By comparing the automated crack detection results, which were also binary images, with the ground truths, each pixel in the test images was classified as true positive (TP), false negative (FN), false positive (FP), and true negative (TN), as described in Table 5.1. Finally, on the basis of the classification results, the performance of the proposed crack detection approach was evaluated by using a positive prediction value (PPV; also referred to as precision), negative prediction value (NPV), true positive rate (TPR; also referred to as recall or sensitivity), true negative rate (TNR), accuracy (ACC), and F1 score (F1), which are defined below.

$$PPV = \frac{TP}{TP + FP}$$

$$NPV = \frac{TN}{TN + FN}$$

$$TPR = \frac{TP}{TP + FN}$$

$$TNR = \frac{TN}{FP + TN}$$

$$ACC = \frac{TP + TN}{TP + FP + FN + TN}$$

$$F1 = \frac{(2 \times PPV \times TPR)}{PPV + TPR}$$

Table 5.1 Crack Detection Results Confusion Matrix

Image	Predicted Crack	Predicted Non-crack	-
Actual Crack	True Positive (TP)	False Negative (FN)	True Positive Rate (TPR)
Actual Non-crack	False Positive (FP)	True Negative (TN)	True Negative Rate (TNR)
-	Positive Predictive Value (PPV)	Negative Predictive Value (NPV)	Accuracy (ACC)

As stated above, the images used in this study were collected from six different specimens, and the average number of total pixels per image was 4,045,621. Because every ground truth image was generated from the original image, the number of total pixels per ground truth image was same as those of the original test image.

The average crack detection accuracy, positive predictive value (precision), true positive rate (recall), and F1 scores obtained for each image are presented in tables 5.2 through 5.7, and Table 5.8 presents the average accuracy, precision, recall, and F1 values, which were 0.9968, 0.9123, 0.9481, and 0.9290, respectively. Even though a high level of accuracy was achieved for all the test images, the precision values were relatively low for images 2 and 5, and the recall rate obtained for images 1 and 3 were relatively low as well. The low precision values in images 2 and 5 stemmed from the FP values (i.e., non-crack pixels that were predicted/classified as crack pixels), which resulted in several thin cracks being detected as larger/wider cracks. The low recall rate obtained for images 1 and 3 stemmed from the FN values (i.e., crack pixels that were predicted/classified as non-crack pixels), which means that several thin cracks were lost during this process. The cause of both the low precision and low recall values stemmed from the post-processing procedure adopted in this study, which used definitive image adjustment algorithms to detect clearly identifiable cracks, resulting in FPs and FNs in dealing with thin cracks.

Table 5.2 Crack Detection Results for Test Image 1

Image	Predicted Crack	Predicted Non-crack	-
Actual Crack	TP = 75,235	FN = 8,069	TPR = 0.9031
Actual Non-crack	FP = 3,364	TN = 3,073,716	TNR = 0.9989
-	PPV = 0.9572	NPV = 0.9974	ACC = 0.9964

Table 5.3 Crack Detection Results for Test Image 2

Image	Predicted Crack	Predicted Non-crack	-
Actual Crack	TP = 38,037	FN = 917	TPR = 0.9765
Actual Non-crack	FP = 6,572	TN = 1,836,374	TNR = 0.9964
-	PPV = 0.8527	NPV = 0.9995	ACC = 0.9960

Table 5.4 Crack Detection Results for Test Image 3

Image	Predicted Crack	Predicted Non-crack	-
Actual Crack	TP = 78,560	FN = 7,785	TPR = 0.9098
Actual Non-crack	FP = 3,629	TN = 4,020,510	TNR = 0.9991
-	PPV = 0.9558	NPV = 0.9981	ACC = 0.9972

Table 5.5 Crack Detection Results for Test Image 4

Image	Predicted Crack	Predicted Non-crack	-
Actual Crack	TP = 97,702	FN = 6,962	TPR = 0.9335
Actual Non-crack	FP = 9,461	TN = 5,394,163	TNR = 0.9982
-	PPV = 0.9117	NPV = 0.9987	ACC = 0.9970

Table 5.6 Crack Detection Results for Test Image 5

Image	Predicted Crack	Predicted Non-crack	-
Actual Crack	TP = 107,226	FN = 1,158	TPR = 0.9893
Actual Non-crack	FP = 16,409	TN = 5,077,844	TNR = 0.9968
-	PPV = 0.8673	NPV = 0.9998	ACC = 0.9966

Table 5.7 Crack Detection Results for Test Image 6

Image	Predicted Crack	Predicted Non-crack	-
Actual Crack	TP = 125,779	FN = 3,046	TPR = 0.9764
Actual Non-crack	FP = 8,849	TN = 4,272,356	TNR = 0.9979
-	PPV = 0.9343	NPV = 0.9993	ACC = 0.9973

Table 5.8 Crack Detection Performance Results

Image	Accuracy	PPV (= Precision)	TPR (= Recall)	F1 Score
1	0.9964	0.9572	0.9031	0.9294
2	0.9960	0.8527	0.9765	0.9104
3	0.9972	0.9558	0.9098	0.9323
4	0.9970	0.9117	0.9335	0.9225
5	0.9966	0.8673	0.9893	0.9243
6	0.9973	0.9343	0.9764	0.9548
Average	**0.9968**	**0.9132**	**0.9481**	**0.9290**

To supplement the quantitative metrics used to measure the crack detection performance of the proposed approach, a qualitative approach was also used to further evaluate the preservation of details. As depicted in figures 5.1 and 5.2, shapes of target cracks in all six

images were captured successfully. However, the branches that were very thin were lost, i.e., not captured (Figure 5.3). Note that the average width where the detail losses occurred was 1.05 mm (or 2.18 pixels). In general, thin cracks on reinforced concrete structure surfaces are negligible according to the standards (Calvi et al. 2018) because their impact on the integrity of a structure is negligible.

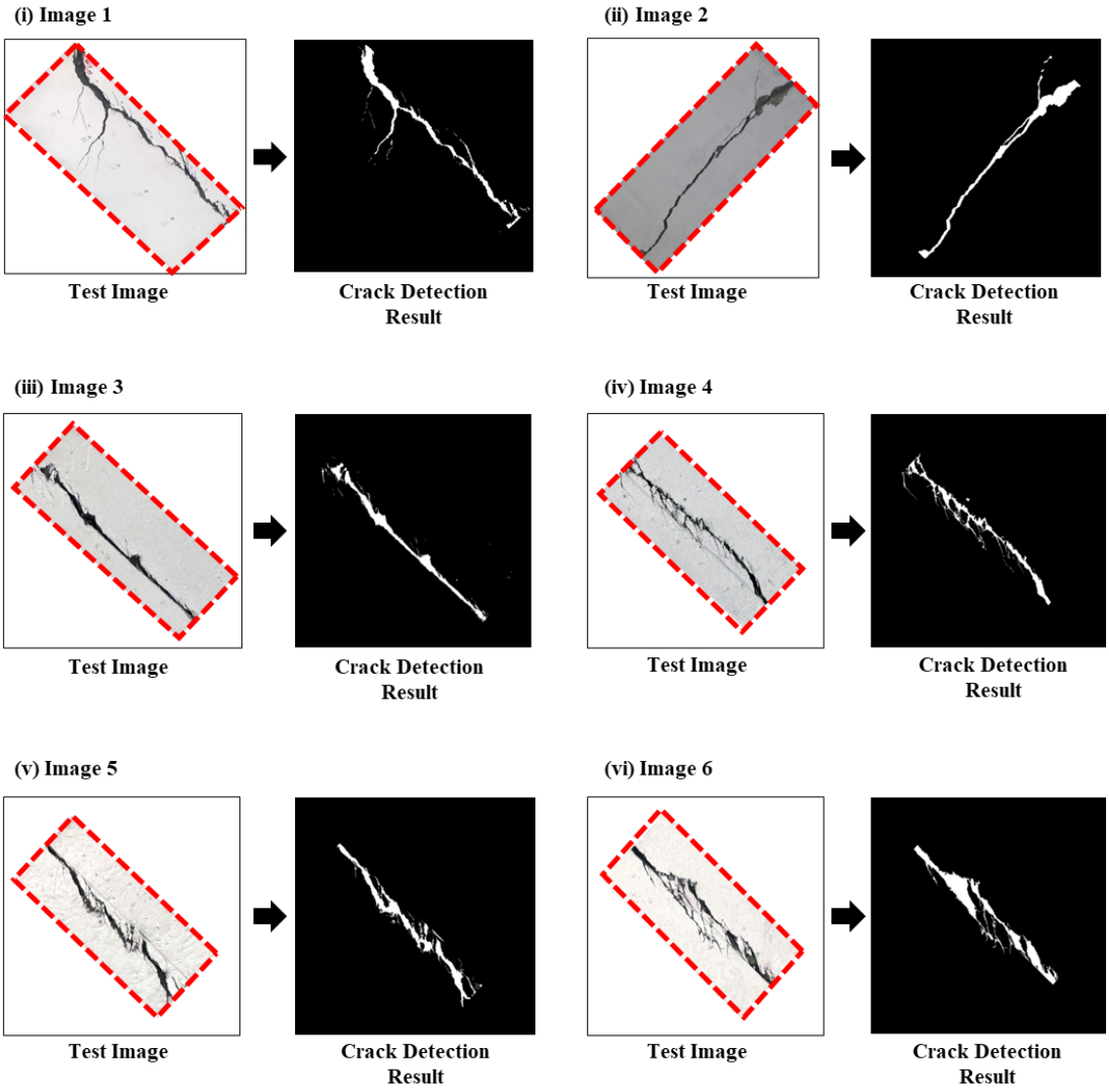

Figure 5.1 Images (Left) and Crack Detection Results (Right)

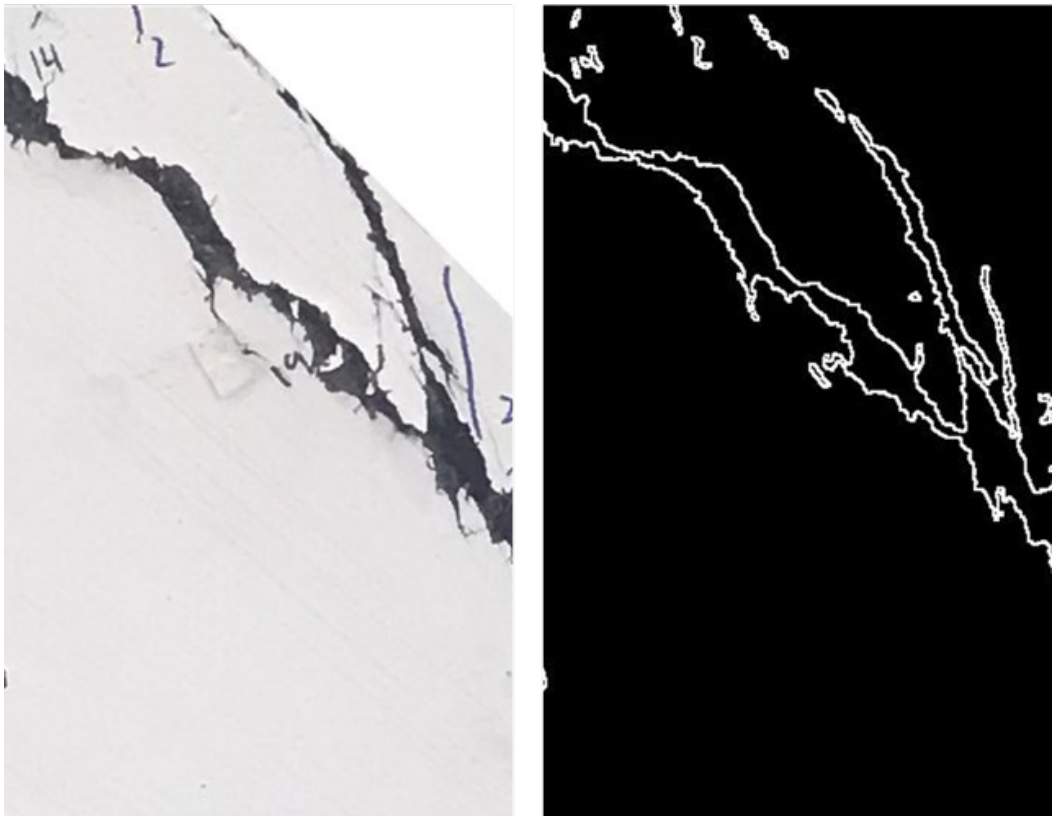

Figure 5.2 Detailed Detection Results from Image 1

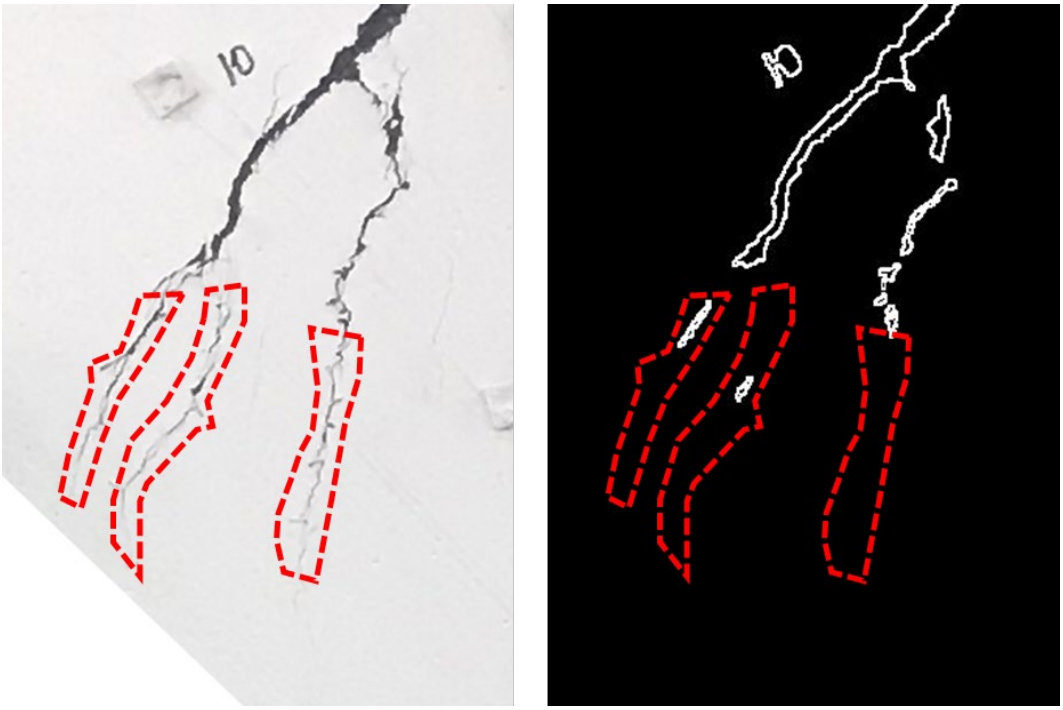

Figure 5.3 Detail of Losses during Crack Detection

5.2. Crack Measurements

In this step, a real-scale conversion factor for each test dataset was calculated to convert pixel-dimensions into metric dimensions (in millimeters). First, the lengths of all four sides of the concrete panels in each image were manually measured in pixel-dimensions, and the average length was obtained. Because the actual length of the test panel equaled 890 mm, the conversion factor for each image was obtained by dividing the actual length with the measured length (mm/pixels). The conversion factors for the six test images were calculated as 0.4904, 0.6445, 0.4457, 0.4353, 0.4353, and 0.4744 mm/pixel, respectively.

In the next step, the performance of the skeletonization and branch filtering algorithms was evaluated by measuring the loss in crack skeletons' length. Following the initial skeletonization, multiple branches along the crack skeleton were detected because the target cracks in the test data had multiple orientations (based on their direction). Accordingly, a branch filtering algorithm was applied to discard any unnecessary branches and to obtain the crack skeleton in 1-pixel thickness. The branch filtering algorithm used was based on a pre-determined threshold value, which is based on the length of each segment along the main skeleton. These filtering steps were also applied to each end of the main skeleton, resulting in crack length loss (Figure 5.4). The total length loss (for a total of ten cracks) due to the branch filtering was 515.20 mm, equal to 51.52 mm per crack on average.

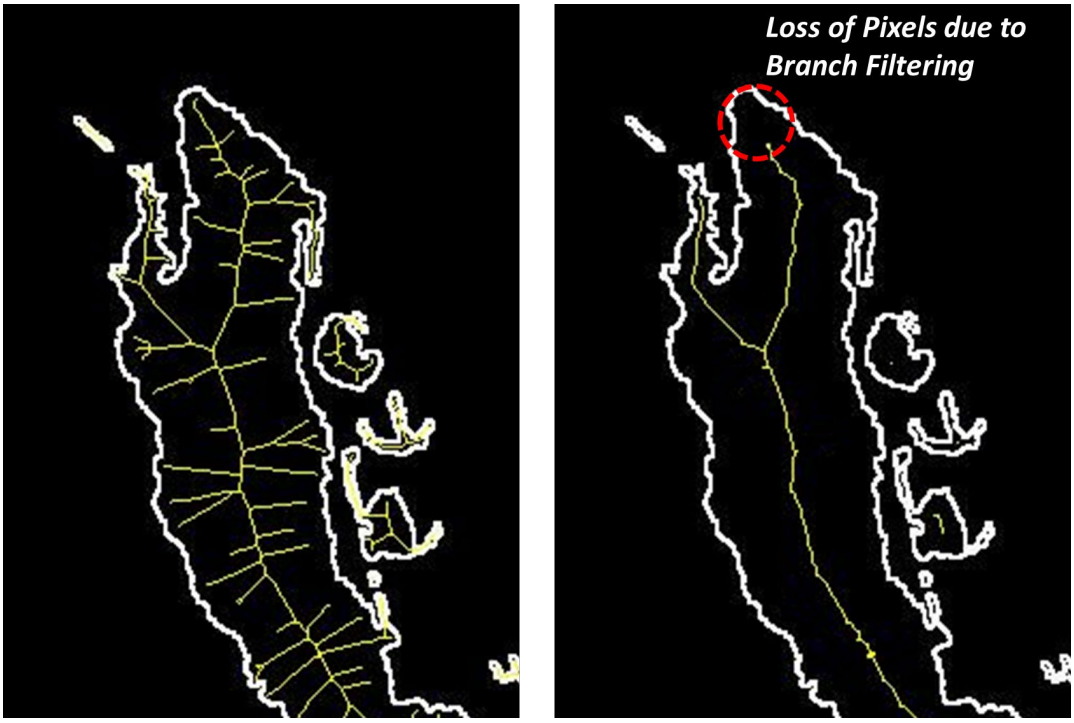

Figure 5.4 Skeletonization Results and Pixel Loss due to Branch Filtering

The length measurements and the amount of loss in length are presented in tables 5.9 and 5.10. The average accuracy of the length measurement was 88.17 percent. In this evaluation, the actual crack length (ground truth) was obtained by dividing each crack into ten segments that shared the same orientation (based on human judgment) and then adding them up.

Table 5.9 Loss of Length from Branch Filtering

Image	Target Crack	Loss of Length (mm)	Ground Truth (mm)	Ratio (%)
1	1-1	101.50	1210.84	8.38
2	2-1	29.99	910.31	3.29
3	3-2	31.45	802.83	3.92
4	4-1	34.94	316.99	11.02
	4-2	78.67	755.51	10.41
	4-3	28.31	334.22	8.47
5	5-1	62.64	881.02	7.11
	5-2	48.69	355.02	13.72
6	6-1	42.81	1063.99	4.02
	6-2	56.21	274.09	20.51
Average Ratio based on Ground Truth				**7.46**

Table 5.10 Length Measurement Results

Image	Target Crack	Proposed method (mm)	Ground Truth (mm)	Accuracy (%)
1	1-1	1087.25	1210.84	89.79
2	2-1	795.34	910.31	87.67
3	3-2	692.57	802.83	86.27
4	4-1	295.14	316.99	93.11
	4-2	680.82	755.51	90.11
	4-3	299.49	334.22	89.61
5	5-1	787.44	881.02	89.38
	5-2	294.69	355.02	83.01
6	6-1	953.00	1063.99	89.57
	6-2	202.08	274.09	73.73
Average Accuracy based on Ground Truth				**88.17**

Finally, the crack width measurement results were evaluated by measuring the widths of five random checkpoints along each test image. Given the coordinates of the selected checkpoints, their width were measured by using the method detailed in Section 3.2. The actual crack widths (ground truths) at each checkpoint were manually measured directly from the original image. The automatic width measurement results (based on orthogonal projection), as well as the ground truth values for all check points, are provided in Table 5.11. As can be seen, the average width measurement accuracy for each image, 1 through 6, was 94.68 percent, 93.48 percent, 93.89 percent, 94.63 percent, 95.31 percent, and 94.39 percent, respectively.

Table 5.11 Width Measurement Results

Image	Target Crack	Checkpoint	Automated Measurement (mm)	Ground Truth (mm)	Accuracy (%)
1	1-1	(583, 276)	29.82	31.21	95.56
		(755, 472)	27.41	28.51	96.15
		(1052, 720)	16.67	17.16	97.14
		(1372, 993)	18.03	19.42	92.86
		(1492, 1175)	15.26	16.65	91.67
				Average - 1	94.68
2	2-1	(314, 1255)	27.71	29.65	93.48
		(538, 928)	10.94	11.85	92.31
		(669, 775)	10.94	11.85	92.31
		(922, 503)	9.23	10.09	91.47
		(1074, 355)	38.91	39.78	97.82
				Average - 2	93.48
3	3-2	(510, 630)	11.34	12.61	90.00
		(660, 847)	20.17	20.80	96.97
		(897, 1079)	11.98	12.61	95.00
		(1114, 1260)	34.76	36.54	95.12
		(1395, 1534)	11.96	12.96	92.31
				Average - 3	93.89
4	4-1	(572, 651)	6.16	6.77	90.91
	4-2	(912, 941)	11.32	11.75	96.30
		(1189, 1144)	17.24	17.85	96.55
	4-3	(1598, 1561)	9.23	9.85	93.75
		(1748, 1830)	13.54	14.16	95.65
				Average - 4	94.63
5	5-1	(444, 481)	18.47	19.08	96.77
		(727, 832)	12.31	13.54	90.91
		(826, 1023)	20.83	22.00	94.69
	5-2	(1379, 1477)	22.78	23.39	97.37
		(1555, 1736)	26.55	27.42	96.83
				Average - 5	95.31
6	6-1	(477, 501)	23.48	24.82	94.59
		(600, 617)	14.64	15.91	92.04
		(1132, 1034)	9.49	10.44	90.91
	6-2	(1302, 1198)	23.48	24.15	97.22
		(1504, 1419)	43.91	45.19	97.18
				Average - 6	94.39
				Average - Total	**94.40**

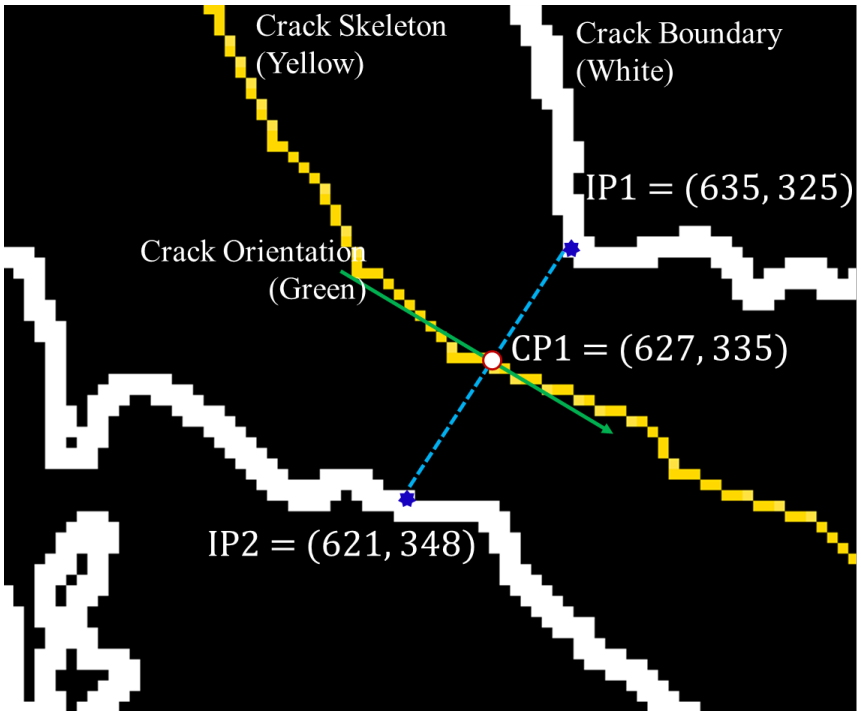

Figure 5.5 Representation of Width Measurement

CHAPTER 6. CONCLUSIONS

Identification and assessment of surface defects on concrete structures are vital to evaluate their condition. Cracks are one of the major surface defects and indicate concrete structures' stability and durability. Because current methods to assess concrete cracks rely on manual identification and assessment, which are time consuming, labor intensive, and prone to errors, recent studies have focused on leveraging computer vision and machine learning techniques to automate this process.

Accordingly, this study proposed a methodology for automated crack detection and measurement. This method combines a CNN-based algorithm and traditional image morphological operations for crack detection with a crack measurement approach based on skeletonization and orthogonal projection algorithms. The overall performance of the proposed methodology was evaluated by using images of concrete panels obtained from shear strength tests on UHPC that were conducted in a controlled laboratory environment. The proposed methodology was tested for crack detection and measurement accuracy by evaluating test results quantitatively. The test results showed that the proposed crack detection method accurately detected crack and non-crack pixels with an average accuracy of 0.9968 and an average F1 score of 0.9290. Although thin cracks that were 1.05 mm (corresponding to 2.18 pixels) or thinner were not detected, this can be considered insignificant in real-life situations because such thin cracks do not have a major impact on the integrity of a structure. The results of the crack measurements that were based on skeletonization and branch filtering algorithms were also evaluated quantitively. For the six test images containing ten cracks, the proposed crack measurement method achieved an average accuracy of 88.17 percent for length measurements, with a 7.46 percent loss in length during the branch filtering process, and an average accuracy of 94.40 percent for width measurements.

As presented in this study, the proposed crack detection and measurement methodology produced promising results. However, the proposed methodology has several shortcomings that must be addressed in future research. First, the ground truths, both for crack detection and measurement, were obtained through manual annotations. Although these ground truths were meant to offer an apple-to-apple comparison, an in-depth evaluation with different sources of ground truths is needed to validate the versatility of the proposed approach. For instance, 3D-

scanned point cloud data or even physically measured results can be utilized as alternative ground truths.

Focusing on the performance of the methodology itself, the branch filtering steps need to be fine-tuned because the loss in crack length due to branch filtering impacts the overall accuracy. Given the average crack length loss results presented in this study, improving the branch filtering steps could ensure better accuracy for crack length measurement. Furthermore, crack length losses were observed at either ends of a crack, and further image processing steps could be implemented to preserve the main skeleton on both ends.

This study used a readily available CNN architecture called DeepCrack, which is open-source, has easy-to-use architecture, and performs very well with a pre-trained model. However, a brand-new CNN-based crack detector with a smaller number of layers (to decrease the time and load required for training) could be developed to make rapid structural assessments more feasible.

The study presented here is intended to be an important component of an overall research framework meant to develop a method for rapid structural assessmentby utilizing UAS images and automatically detecting and measuring the surface defects of large structures such as multi-story buildings or bridges. It is envisioned that, in this overall framework, structural assessment results, i.e., surface defects, would be mapped to building information models (BIM), in which inspection reports from different dates could be integrated with structural drawings and models in a single database.

REFERENCES

American Society of Civil Engineers (ASCE)., 2021. 2021 Report card for America's infrastructure. ASCE, Reston, VA, United States.

Azhari, F., Sennersten, C., Milford, M. and Peynot, T., 2021. "PointCrack3D: Crack detection in unstructured environments using a 3D-point-cloud-based deep neural network," Computing Research Repository (CoRR), 2111.11615.

Bai, X., Zhou, F., and Xue, B., 2012. "Image enhancement using multi scale image features extracted by top-hat transform", Optics & Laser Technology, 44(2):328-336.

Bhowmick, S., Nagarajaiah, S. and Veeraraghavan, A., 2020. "Vision and Deep Learning-Based Algorithms to Detect and Quantify Cracks on Concrete Surfaces from UAV Videos," Sensors, 20:6229.

Calvi, P.M., Proestos, G.T. and Ruggiero, D.M., 2018. "Toward the development of direct crack-based assessment of structures," ACI Structural Journal, 328:9.1–9.20.

Cha, Y.-J., Choi, W. and Buyukozturk, O., 2017. "Deep Learning-Based Crack Damage Detection Using Convolutional Neural Networks," Comput. Aided Civ. Infrastruct. Eng., 32:361–378.

Chen, X., Li, J., Huang, S., Cui, H., Liu, P. and Sun, Q., 2021. "An Automatic Concrete Crack-Detection Method Fusing Point Clouds and Images Based on Improved Otsu's Algorithm," Sensors, 21(5):1581.

Dorafshan, S., Thomas, R.J. and Maguire, M., 2018. "Comparison of deep convolutional neural networks and edge detectors for image-based crack detection in concrete," Construction and Building Materials, 186:1031-1045.

Gopalakrishnan, K., Khaitan, S.K., Choudhary, A. and Agrawal, A., 2017. "Deep Convolutional Neural Networks with transfer learning for computer vision-based data-driven pavement distress detection," Constr. Build. Mater., 157:322–330.

Guan, J., Yang, X., Ding, L., Cheng, X., Lee, V.C.S. and Jin, C., 2021. "Automated pixel-level pavement distress detection based on stereo vision and deep learning," Automation in Construction, 129:103788.

He, K., Zhang, X., Ren, S. and Sun, J., 2016. "Deep residual learning for image recognition," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, Nevada, United States, pp. 770-778.

Islam, M.M.M. and Kim, J.-M., 2019. "Vision-Based Autonomous Crack Detection of Concrete Structures Using a Fully Convolutional Encoder—Decoder Network," Sensors, 19:4251.

Kalfarisi, R., Wu, Z.Y. and Soh, K., 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.

Kanopoulos, N., Vasanthavada, N. and Baker, R.L., 2002. "Design of an image edge detection filter using the Sobel operator," IEEE Journal of Solid State Circuits, 23(2):358-367.

Ko, P., Prieto, S.A. and Garcia de Soto, B., 2021. "ABECIS: an automated building exterior crack inspection system using UAVs, open-source deep learning and photogrammetry," 38th International Symposium on Automation and Robotics in Construction (ISARC 2021), Dubai, UAE, pp. 637-644.

Krizhevsky, A., Sutskever, I. and Hinton, G.E., 2012. "ImageNet classification with deep convolutional neural networks," Advanced in Neural Information Processing Systems 25 (NIPS 2012), Stateline, Nevada, United States.

Lee, J., Kim, H.-S., Kim, N., Ryu, E.-M. and Kang, J.-W., 2019. "Learning to Detect Cracks on Damaged Concrete Surfaces Using Two-Branched Convolutional Neural Network," Sensors, 19:4796.

Li, E.-S., Zhu, S.-L., Zhu, B.-S., Zhao, Y., Xia, C.-G. and Song, L.-H., 2009. "An Adaptive Edge-detection Method Based on the Canny Operator," Proceedings of the International Conference on Environmental Science & Information Application Technology, Wuhan, China, pp. 465-469.

Liu, Y., Yao, J., Lu, X., Xie, R. and Li, L., 2019. "DeepCrack: A deep hierarchical feature learning architecture for crack segmentation," Neurocomputing, 338:139-153.

Maguire, M., Dorafshan, S. and Thomas, R., 2018. "SDNET2018: A Concrete Crack Image Dataset for Machine Learning Applications,", Utah State University, Logan.

Mohan, A. and Poobal, S., 2018. "Crack detection using image processing: A critical review and analysis," Alexandria Engineering Journal, 57:787-798.

Nguyen, T. S., Begot, S., Duculty, F. and Avila, M., 2011. "Free-form anisotropy: A new method for crack detection on pavement surface images," Proceedings of the IEEE International Conference on Image Processing, Brussels, Belgium, pp. 1069-1072.

Ni, T., Zhou, R., Gu, C. and Yang, Y., 2020. "Measurement of concrete crack features with android smartphone APP based on digital image processing techniques," Measurement, 150:107093.

Otsu, N., 1979. "A Threshold Selection Method from Gray-Level Histograms," IEEE Transactions on Systems, Man, and Cybernetics, 9(1):62-66.

Qiu, S., Wang, W., Wang, S. and Wang, K.C.P., 2017. "Methodology for accurate AASHTO PP67-10–based cracking quantification using 1-mm 3D pavement images," Journal of Computing in Civil Engineering, 31(2):04016056.

Safaei, N., Smadi, O., Masoud, A. and Safaei, B., 2022. “An Automatic Image Processing Algorithm Based on Crack Pixel Density for Pavement Crack Detection and Classification,” Int. J. Pavement Res. Technol., 15:159–172.

Simonyan, K. and Zisserman, A., 2015. “Very deep convolutional networks for large-scale image recognition,” Proceedings of the International Conference on Learning Representations (ICLR 2015), San Diego, California, United States.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V. and Rabinovich, A., 2015. “Going deeper with convolutions,” Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, Massachusetts, United States, pp. 1-9.

Voytko, D., Calvi, P.M. and Stanton, J., 2022. “Shear strength of ultra-high-performance concrete,” Engineering Structures, 255:113961.

Wang, D. and Zhou, S., 2008. “Color Image Recognition Method Based on the Prewitt Operator,” Proceedings of the International Conference on Computer Science & Software Engineering, Wuhan, China, pp. 170-173.

Wang, W., Hu, W., Wang, W., Xu, X., Wang, M., Shi, Y., Qiu, S. and Tutumluer, E., 2021. “Automated crack severity level detection and classification for ballastless track slab using deep convolutional neural network,” Automation in Construction, 124:103484.

Xu, Y. and Turkan, Y., 2020. “BrIM and UAS for bridge inspections and management,” Engineering, Construction and Architectural Management, 27(3):785-807.

Yan, Y., Mao, Z., Wu, J., Padir, T. and Hajjar, J. F., 2021. “Towards automated detection and quantification of concrete cracks using integrated images and lidar data from unmanned aerial vehicles,” Structural Control Health Monitoring, 28(8): e2757.

Yang, X., Li, H., Yu, Y., Luo, X., Huang, T. and Yang, X., 2018. “Automatic Pixel-Level Crack Detection and Measurement Using Fully Convolutional Network,” Comput. Aided Civ. Infrastruct. Eng., 33:1090–1109.