DOI: 10.1111/mice.12932

RESEARCH ARTICLE

COMPUTER-AIDED CIVIL AND INFRASTRUCTURE ENGINEERING WILEY

Methodology to integrate augmented reality and pattern recognition for crack detection

Kaveh Malek[1] | Ali Mohammadkhorasani[2] | Fernando Moreu[2]

[1]Department of Mechanical Engineering, University of New Mexico, Albuquerque, New Mexico, USA

[2]Department of Civil, Construction and Environmental Engineering, University of New Mexico, Albuquerque, New Mexico, USA

Correspondence
Fernando Moreu, Department of Civil, Construction and Environmental Engineering, University of New Mexico, 210 University Blvd, NE, MSC01 1070, Albuquerque, NM 87131, USA.
Email: fmoreu@unm.edu

Funding information
Transportation Consortium of South-Central States (Tran-SET), Grant/Award Number: 69A3551747106; National Academy of Science Transportation Research Board (TRB) Rail SAFETY IDEA Program Project RS-43, Grant/Award Number: 163420-0399; Federal Railway Administration (FRA) BAA, Grant/Award Number: FR20RPD34000000006; National Science Foundation, Computer and Information Science and Engineering (CISE), Division Of Information & Intelligent Systems, Grant/Award Number: 2123346

Abstract

In-field visual inspections have inherent challenges associated with humans such as low accuracy, excessive cost and time, and safety. To overcome these barriers, researchers and industry leaders have developed image-based methods for automatic structural crack detection. More recently, researchers have proposed using augmented reality (AR) to interface human visual inspection with automatic image-based crack detection. However, to date, AR crack detection is limited because: (1) it is not available in real time and (2) it requires an external processing device. This paper describes a new AR methodology that addresses both problems enabling a standalone real-time crack detection system for field inspection. A Canny algorithm is transformed into the single-dimensional mathematical environment of the AR headset digital platform. Then, the algorithm is simplified based on the limited headset processing capacity toward lower processing time. The test of the AR crack-detection method eliminates AR image-processing dependence on external processors and has practical real-time image-processing.

1 | INTRODUCTION

Recent computer science advances have made image processing broadly applicable to engineering applications. The application of vision systems for the inspection of the infrastructure has received increasing interest (Martinez et al., 2019). Researchers are adopting image-based structural crack detection as a substitute for traditional non-destructive tests because of its explicitness and its diversity (Mohan & Poobal, 2018). Recent achievements in image-based crack detection include, but are not limited to, faster processing of images (e.g., Miao et al., 2020), near real-time crack identification from video signals (e.g., Piyathilaka et al., 2020), crack detection with unmanned aerial vehicles (e.g., Dorafshan et al., 2019), and methods to detect the low contrast cracks (e.g., Li et al., 2014). Several recent studies utilized different deep learning algorithms for crack detection (Aravind et al., 2021; Çelik & König, 2022; Cha et al., 2017; Chen & He, 2022; Dung & Anh, 2019; Le et al., 2021; C. Liu & Xu, 2022; Miao & Srimahachota, 2021; Ni et al., 2019; Piyathilaka et al., 2020; Żarski et al., 2022; Zhang & Yuen, 2021; Zheng et al., 2022; Zhou et al., 2022; Zou et al., 2022). Alternatively, several studies have focused on pattern recognition crack

[Correction added on 14 November 2022, after first online publication: The funding information and the acknowledgement sections were updated.]

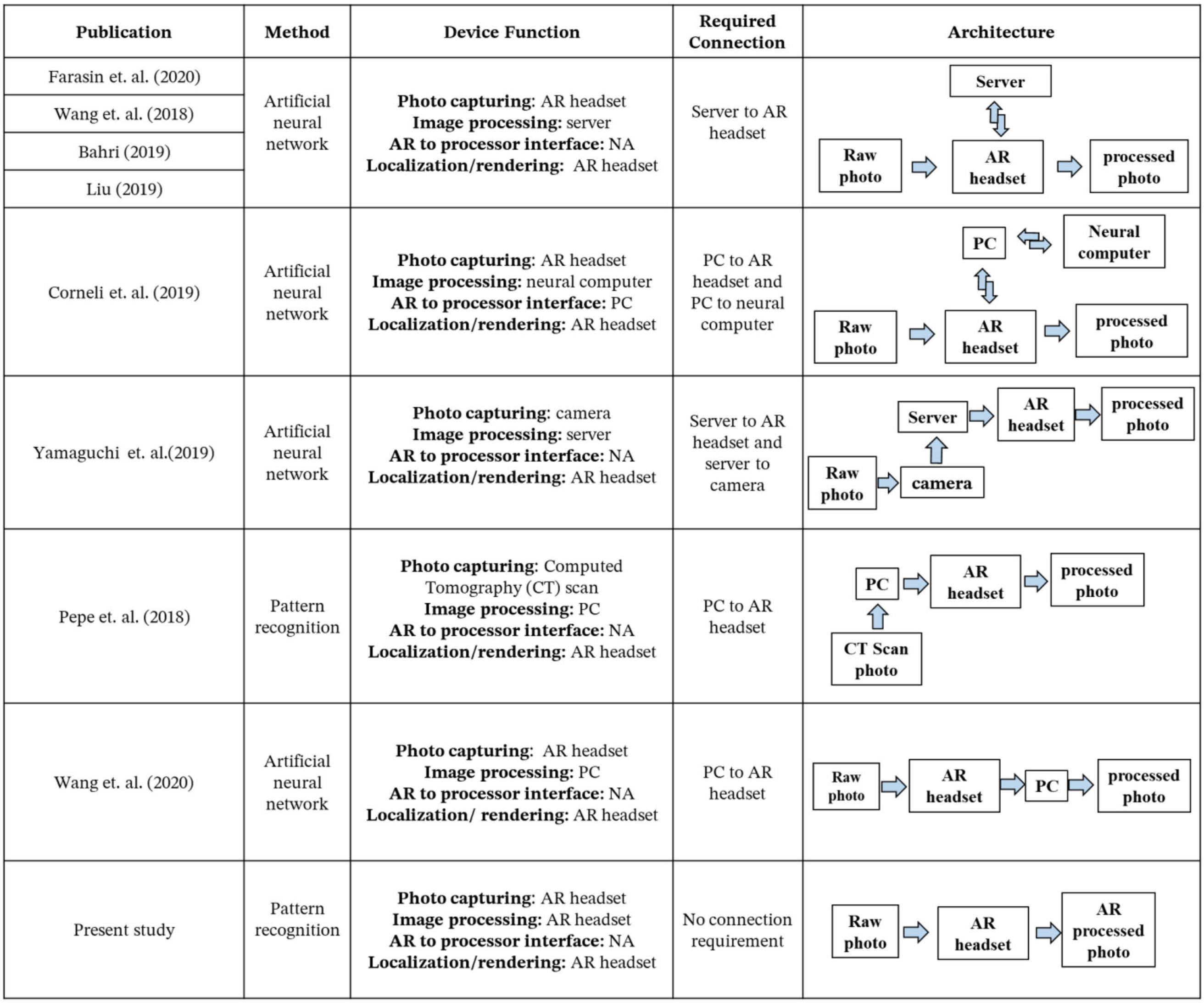
TABLE 1 Comparison of unsupervised pattern recognitions, machine learning (ML) and artificial neural networks (Al-Faris et al., 2020; Georgiou et al., 2020; Salehi & Burgueño, 2018; Wang et al., 2020; Zare et al., 2018)

		ML	
Artificia intelligence method	**Pattern recognition**	**Artificial neural network**	**Other ML methods**
Advantages	–Applicative for traditional structural health monitoring (SHM) systems –Functional for image-based detection and classification SHM systems –Does not obligatorily require large datasets to be trained –Requires lower processing power and time for training and implementation	–Effective for vision-based SHM systems –High efficiency without data pre-processing –High performance for managing big data –Excellent computational performance –Ability to learn features directly from raw data –Enables automatic feature selection from raw data	–Appropriate for traditional and data-driven SHM systems –Optimizable detectability –Does not obligatorily require large datasets to be trained –Excellent computational performance
Disadvantages	–Does not automatically learn from new data –Its performance depends on in-depth data pre-processing, optimal pattern selection, and feature selection algorithm –Includes several ungeneralizable steps (e.g., feature selection)	–Inefficient for traditional SHM systems –Requires large dataset for effective training process –Requires high processing power and time for training	–Not suitable for image-based SHM systems

detection as another approach to image-based methods (Dorafshan et al., 2019; Iyer & Sinha, 2005; Li et al., 2014; Miao et al., 2020; Safaei et al., 2022; W. Wang et al., 2018; Y. Wang et al., 2019). Table 1 shows the benefits and limitations of different artificial intelligence approaches used for structural health monitoring (SHM). The artificial neural network can spontaneously extract features of crack patterns from a training set of images and does not depend on pre-processing to ensure high accuracy. This approach utilizes a set of large data for training the predictive model that necessitates a computationally powerful processor (Zare et al., 2018). Other machine learning (ML) methods have high computational performance and are appropriate for traditional and data-driven SHM systems. On the other hand, unsupervised pattern recognition extracts the features matched with a predefined pattern and needs an efficient predefined pattern and comprehensive pre-processing to enhance detection efficiency. This approach does not necessarily require high computational power to ensure high accuracies.

All of the aforementioned research conducted image-based crack detection with a computer and excluded the user interface for human crack inspection processes (Karaaslan et al., 2019). On the other hand, human visual inspection is still important in the field to enable ad hoc decisions that experts need to conduct in person (Shaohan Wang et al., 2020). Therefore, researchers have recently proposed several alternatives for human-in-the-loop image-based crack detection methods. Researchers and industry leaders are interested to equip human inspectors with a real-time image-processing decision-assistance tool in the field. Specifically, the human inspector needs to instantly validate the imaged-based crack detection with the real crack on the structure. Past studies have proposed implementing image-based crack detection methods inside augmented reality (AR) headsets as the interface of the image-processing and the structural crack inspection (Shaohan Wang et al., 2020). AR headsets can deploy a crack detection/characterization application for humans, enabling hands-free visual inspections while automatically obtaining useful data via computer-environment interaction (Karaaslan et al., 2019). Past studies explored the use of AR technology for object recognition (Bahri et al., 2019; Corneli et al., 2019; Farasin et al., 2020; L. Liu et al., 2019; Pepe et al., 2018; Shuai Wang et al., 2018). Evaluation of infrastructure inspection with AR headsets is among the recent interests in SHM and maintenance methods (Maharjan et al., 2021; Mascareñas et al., 2021). Developing new AR methods for crack detection is also investigated in earlier studies such as Yamaguchi et al. (2019) and Shaohan Wang et al. (2020).

This study develops a standalone real-time system for image-based crack detection that can be instantly deployed in AR headsets. Training of the predictive algorithm for the pattern recognition approach requires less computational power, compared to the artificial neural network (Al-Faris et al., 2020), which implies higher adaptability of pattern recognition with an AR headset platform. Therefore, the proposed system includes a pattern recognition algorithm for image processing. Figure 1 presents the methodology



Publication	Method	Device Function	Required Connection	Architecture
Farasin et. al. (2020)	Artificial neural network	**Photo capturing**: AR headset **Image processing:** server **AR to processor interface:** NA **Localization/rendering:** AR headset	Server to AR headset	Server ⇕ AR headset Raw photo ⇨ AR headset ⇨ processed photo
Wang et. al. (2018)				
Bahri (2019)				
Liu (2019)				
Corneli et. al. (2019)	Artificial neural network	**Photo capturing:** AR headset **Image processing:** neural computer **AR to processor interface:** PC **Localization/rendering:** AR headset	PC to AR headset and PC to neural computer	PC ⇔ Neural computer PC ⇕ AR headset Raw photo ⇨ AR headset ⇨ processed photo
Yamaguchi et. al.(2019)	Artificial neural network	**Photo capturing**: camera **Image processing:** server **AR to processor interface:** NA **Localization/rendering:** AR headset	Server to AR headset and server to camera	Server ⇨ AR headset ⇨ processed photo ⇧ Raw photo ⇨ camera
Pepe et. al. (2018)	Pattern recognition	**Photo capturing**: Computed Tomography (CT) scan **Image processing:** PC **AR to processor interface:** NA **Localization/rendering:** AR headset	PC to AR headset	PC ⇨ AR headset ⇨ processed photo ⇧ CT Scan photo
Wang et. al. (2020)	Artificial neural network	**Photo capturing:** AR headset **Image processing:** PC **AR to processor interface:** NA **Localization/ rendering:** AR headset	PC to AR headset	Raw photo ⇨ AR headset ⇨ PC ⇨ processed photo
Present study	Pattern recognition	**Photo capturing:** AR headset **Image processing:** AR headset **AR to processor interface:** NA **Localization/rendering:** AR headset	No connection requirement	Raw photo ⇨ AR headset ⇨ AR processed photo

FIGURE 1 A brief illustration of the past and present researches. AR, augmented reality

of the mentioned past studies and the proposed research. The first column of Figure 1 shows the significant studies, which have contributed to our understanding of AR systems for image processing. The second column of Figure 1 demonstrates that Pepe et al. (2018) used pattern recognition algorithms for object detection/localization in their methodology. The other past studies applied artificial neural network for their image processing. The third column shows the stationary external processor employed in the past studies i.e., Personal Computer (PC), server, and neural computer. The fourth column explains the required connection between different components of the systems. Specifically, this shows the required connection between the AR headset and the stationary processing device. The interactions between the main components are illustrated in the fifth column. The last row of Figure 1 shows that the present methodology does not use any external processor and does not require any physical or Internet connection. Therefore, this study removes the dependence on AR crack detection methods from external processing devices.

Table 2 shows the benefits and limitations of network-based and standalone AR crack detection. The network-based AR crack detection approach offers high accuracy because it can run complex algorithms. This approach suffers from a latency associated with data transmission between the AR headset and external processor and has mobility limitations corresponding to connection establishment. AR-standalone approach can achieve real-time processing by removing the mentioned latency and has no limitation in mobility. However, this approach is not able to conduct complex algorithms with current standalone processing capabilities unless the algorithms are streamlined for the required levels of complexity.

To implement the proposed methodology, in the first phase, the researchers test the use of a Canny algorithm with a Sobel–Feldman operator to extract cracks' edges

TABLE 2 Benefits and limitations of network-based and standalone augmented reality (AR) crack detection

Existing AR network-based approach (previous works with external processor)	
Advantages	**Limitations**
–Capability to run complex algorithms such as artificial neural network (e.g., Karaaslan et al., 2019; Wang et al., 2021): • Can implement existing efforts from the community without modifications • The fundamental analysis and computation can be instantly validated with existing state of the art results, which should be identical –Efficient crack detection approach with high level of success in the presence of high-quality internet and adequate on-site external processors	–Existing AR-based methods require connecting AR headset to an external processor with two associated drawbacks: • Physical connection of AR headset to external device causes limitation in mobility (Lang et al., 2019) • AR headset connection to external processor through the Internet suffers from access issues if the inspection field is at remote locations (Pourhomayoun et al., 2016) especially in developing countries (Sambuli, 2016) –The latency caused by transferring big datasets (e.g., high quality video stream) between the external processor and the AR headset hinders real-time image-processing (L. Liu et al., 2019)
New AR standalone approach (proposed work)	
Advantages	**Limitations**
–Near real-time processing by removing the latency caused by transferring big datasets between external processor and AR headset –Drops the requirement for availing external processors in inspection fields –Does not need Internet connection to operate: • Full mobility for the inspector who is not restrained by internet or wire connection to external processors • Can be used in remote locations without Internet	–Complex algorithms such as artificial neural network may not be deployable in AR headsets even when the training set is small (Wang et al., 2021) • The AR headset is not able to conduct the existing algorithms with current standalone processing capabilities • Consequently, current algorithms need to be adapted to the AR standalone environment, which requires an optimization that streamlines the algorithm for required levels of complexity. • Optimization is based on some knowledge on the environment to keep low latency and high accuracy

after reducing noise from the crack images using a median filter. During the second phase, the authors transform the algorithm to streamline the detection process and reduce the processing time to deal with the processing capacity limitations of AR headsets. Maximum accuracy drops resulting from the simplification are ~9.5% and ~7.0% for the first and second generations, respectively. The simplifications streamline the algorithm toward real-time processing. Additionally, several experiments were conducted to explore the effect of the following parameters in AR image processing: crack width, pattern recognition thresholds, camera mode, and headset position. In the experiments, researchers tested the crack detection tool using two different AR headsets. This study uses *Recall–Precision* analysis to quantify the effect of the parameters and to detect the *Recall–Precision* direction with the change in the mentioned parameters.

2 | TECHNICAL APPROACH

2.1 | Headset device and software

This study employs a Microsoft HoloLens headset utilizing its capability to perceive the space around it (Brito et al., 2019). To have a wider assessment of the variability of headsets, this study is conducted with two different versions of the HoloLens headset, which assist to evaluate different technical capabilities. Microsoft HoloLens is a see-through AR headset that enables eye tracking, hand gestures and voice commands, and sensor capabilities (Ungureanu et al., 2020). Unreal and Unity engines are two platforms to develop applications for AR headsets with integrated computing capabilities such as Microsoft HoloLens (Santi et al., 2021). Microsoft recommends Unity to create HoloLens applications for the Universal Windows Platform (Evans et al., 2017). In addition, Microsoft implies that expert developers can develop AR app by writing native 3D renderers in OpenXR for HoloLens (Microsoft website, 2022). The AR developers use C# programing language to develop each project.

Table 3 compares the two available generations of Microsoft HoloLens. The second generation of HoloLens is lighter than the first and possesses higher computational capacity. For example, the Central Processing Unit (CPU) speed is 4 and 8 in the first and the second generation, respectively that implies a higher processing speed in the second generation. The higher resolution of the second generation's camera compared to the first generation, as shown in Table 3, provides more photo pixel information for image processing. Table 3 shows that contrary to the first generation, the second generation has the capabilities of eye-tracking and hand-tracking for both hands.

Unity game software and Microsoft Visual Studio are employed for creating a crack detection app. To enable all

TABLE 3 Comparison of two evaluated headsets

Generation	First	Second
CPU cores	4	8
Memory	1 GB	4 GB
Storage	64 GB	64 GB
Hand tracking	One hand	Both hands
Eye-tracking	No	Yes
Horizontal field of view	67°	64.69°
Weight	579 g	566 g
Photo resolution (pixels)	2048 × 1152	3904 × 2196
Video resolution (pixels)	1280 × 720	2272 × 1278

features and interactions, C# is used as the programming language for developing the application.

2.2 | Limitations of AR integration with pattern recognition

There are two technical limitations to integrate the AR headset capability with pattern recognition image processing (1) the computer software and programming languages limitations (Santi et al., 2021); and (2) the scarce processing capability of AR headsets that prevents the implementation of algorithms with high memory complexity (Shaohan Wang et al., 2020).

To overcome these limitations, past studies have connected AR headsets to stationary processing devices using a physical connection or the Internet for implementing their methods (e.g., see Figure 1). In the methodology of this paper, the two mentioned problems are overcome by: (1) making the pattern recognition algorithm compatible with the AR headset digital platform and (2) reducing the memory complexity of the algorithm by streamlining the code. The resulting integrated tool does not require connecting to an external processing device to perform the end-to-end image processing in real time. Section 2.3 outlines the integration of AR and pattern recognition.

2.3 | Outline of the integration

Figure 2 outlines the proposed crack detection system. The key steps of the system include the acquisition of the input image, image processing, localization, and rendering. For feeding the input image, two camera modes are available in AR headsets, that is, photo capturing and webcam (or video) mode as demonstrated in Figure 2. Photo capturing mode captures a photo, and webcam mode records a video from the Hololens web camera. In the present system, a repeated photo capture or webcam video script in the Unity-C# environment produces the input images. The images are then processed inside the AR headset using a streamlined Canny algorithm. The major steps of the image processing algorithm are: (1) graying, (2) applying the simplifications (3) noise filtering, (4) gradient quantification, (5) edge thinning by suppression of non-maximum pixels, (6) double-thresholding, and (7) edge connection using hysteresis. Finally, the processed photos, in which the cracks are overlaid with red indications, are rendered on the concrete surface after the AR headset locates the correct rendering position.

Section 2.4 details the steps in the image processing in the AR headsets platform and describes the approach employed for memory complexity reduction. Afterward, the authors describe the computational challenges involved in AR integration and pattern recognition. Finally, the methodology to overcome those challenges is described.

2.4 | Integration implementation

2.4.1 | Canny algorithm transformation

This study employs the Canny algorithm as the detection technique. Canny edge detection is a pattern recognition technique to recognize the edge pixels in the images (Canny, 1986). Compared to other traditional edge detection methods, the Canny method is superior for its low computational demand, high processing speed, and low-contrast edge detection capabilities (Shan et al., 2016).

Canny is a multi-stage algorithm, and each stage has a special function in the detection process. Figure 3 summarizes the image processing steps. The photos in Figure 3a,h are from the headset view when the AR crack detection app is running, but the other photos in Figure 3 are processed in Python using the same algorithm and are shown for demonstration purposes.

The first step in the image processing is the acquisition of RGB image from the headset camera. The second step involves employing a weighted method to convert a red, green, and blue (RGB) image to a grayscale image. The weighted method computes the grayscale pixel values by weighing R, G, and B according to their wavelengths (Gopinathan & Gayathri, 2016) as shown in Figure 3b. The third step includes storing the grayscale pixels' information in one-dimensional arrays as shown in Figure 3c. Array appears to be the only possible means in the headset for saving big numerical data. Afterward, a noise reduction Kernel reduces the number of false indications in processed photos. Figure 3d ranks the noise performance, processing time, and edge preservation characteristics of several smoothing filters (Moreu & Malek, 2021). As a result, because of its satisfactory characteristic, this study uses a median filter to reduce the noise in images. After the

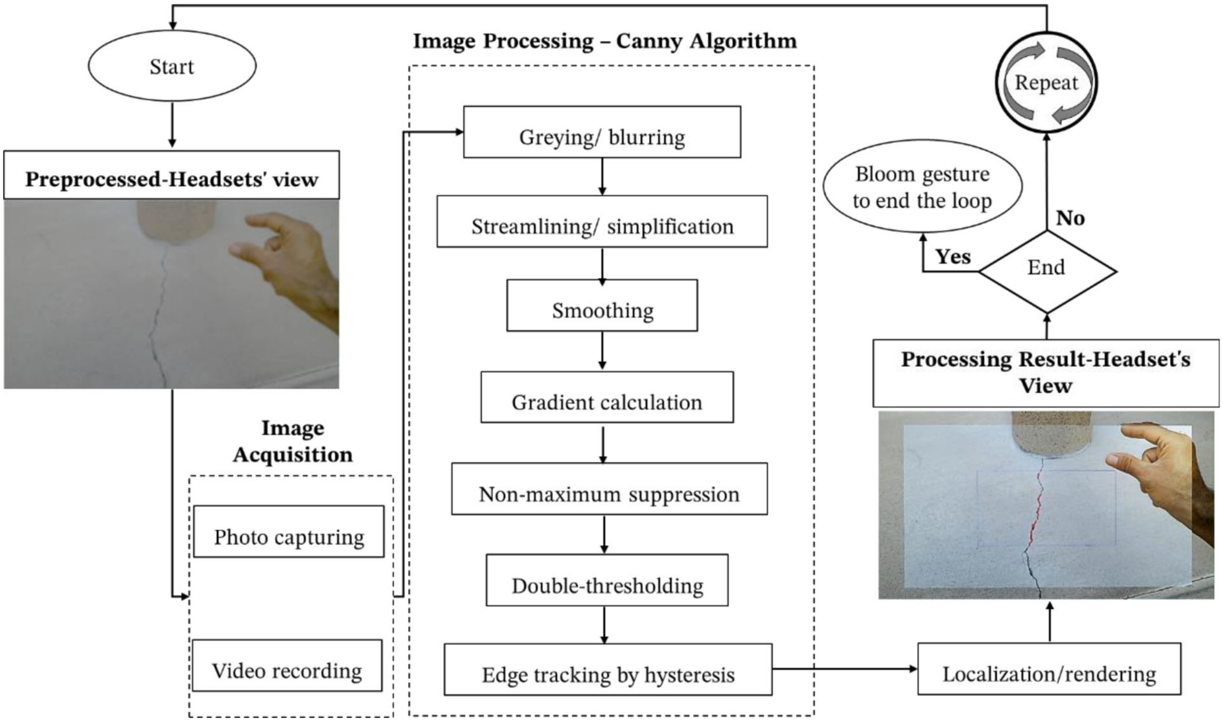
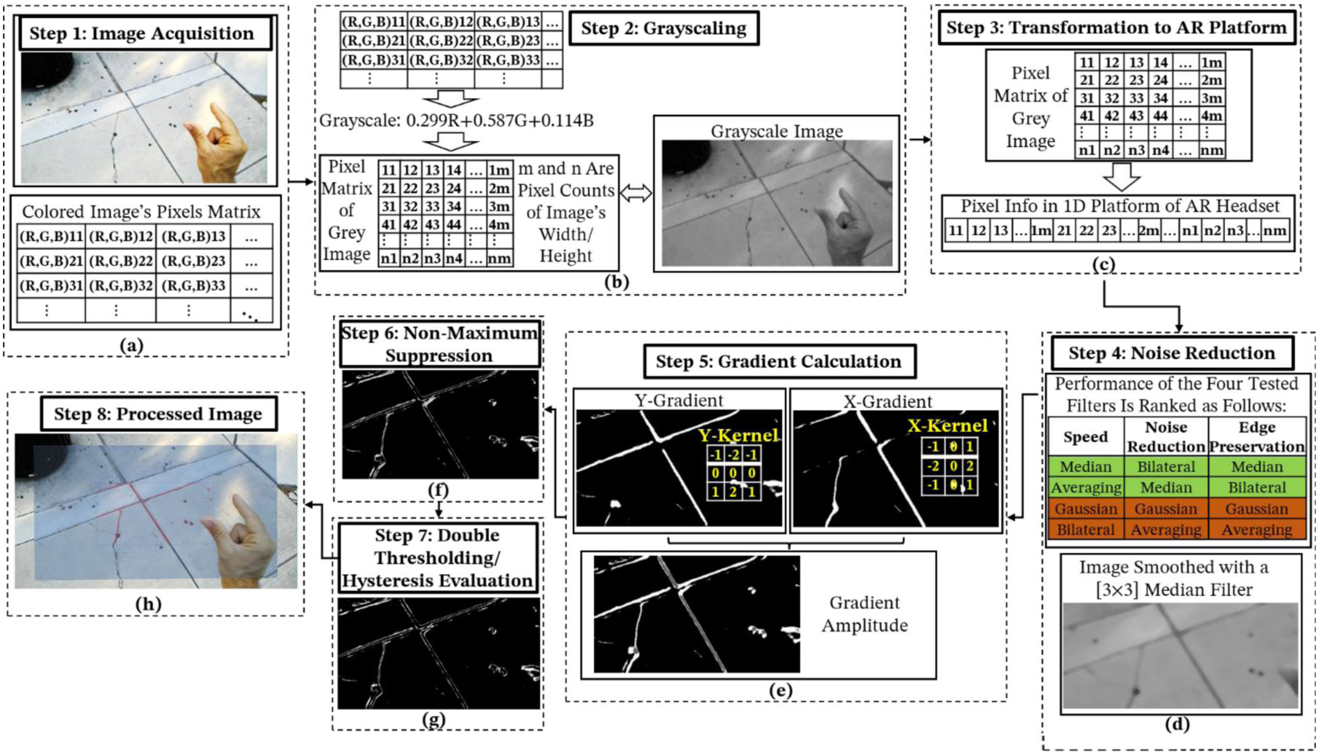

Start
Preprocessed-Headsets' view
Image Acquisition
Photo capturing
Video recording
Image Processing – Canny Algorithm
Greying/ blurring
Streamlining/ simplification
Smoothing
Gradient calculation
Non-maximum suppression
Double-thresholding
Edge tracking by hysteresis
Localization/rendering
Processing Result-Headset's View
End
Yes
No
Bloom gesture to end the loop
Repeat

FIGURE 2 Integration of augmented reality (AR) and pattern recognition for crack detection


Step 1: Image Acquisition
Colored Image's Pixels Matrix
(R,G,B)11 (R,G,B)12 (R,G,B)13 ...
(R,G,B)21 (R,G,B)22 (R,G,B)23 ...
(R,G,B)31 (R,G,B)32 (R,G,B)33 ...
(a)
Step 2: Grayscaling
Grayscale: 0.299R+0.587G+0.114B
Pixel Matrix of Grey Image
11 12 13 14 ... 1m
21 22 23 24 ... 2m
31 32 33 34 ... 3m
41 42 43 44 ... 4m
n1 n2 n3 n4 ... nm
m and n Are Pixel Counts of Image's Width/ Height
Grayscale Image
(b)
Step 3: Transformation to AR Platform
Pixel Matrix of Grey Image
Pixel Info in 1D Platform of AR Headset
11 12 13 ... 1m 21 22 23 ... 2m ... n1 n2 n3 ... nm
(c)
Step 4: Noise Reduction
Image Smoothed with a [3×3] Median Filter
(d)
Step 5: Gradient Calculation
Y-Gradient
Y-Kernel
-1 -2 -1
0 0 0
1 2 1
X-Gradient
X-Kernel
-1 0 1
-2 0 2
-1 0 1
Gradient Amplitude
(e)
Step 6: Non-Maximum Suppression
(f)
Step 7: Double Thresholding/ Hysteresis Evaluation
(g)
Step 8: Processed Image
(h)

Performance of the Four Tested Filters Is Ranked as Follows:

Speed	Noise Reduction	Edge Preservation
Median	Bilateral	Median
Averaging	Median	Bilateral
Gaussian	Gaussian	Gaussian
Bilateral	Averaging	Averaging

FIGURE 3 Image processing steps: (a) image acquisition, (b) saving grayscale pixels' information in one dimensional array, (c) smoothing, (d) Sobel operator for gradient evaluation, (e) suppression of non-maximum pixels, (f) double-thresholding and edge evaluation using hysteresis, (g) streamlining by two simplifications, and (h) processed image

filtering process, the Canny algorithm applies a gradient operator to compute the image gradient. Canny convolves the image with the operators G_x and G_y, which are the first derivatives of a 2D Gaussian G in X and Y directions (Canny, 1986). The integration of other operators such as Sobel gradient into Canny algorithm to improve speed or accuracy is also explored in previous studies (e.g., Luo & Duraiswami, 2008; Ogawa et al., 2010). This study uses Sobel–Feldman operator in X and Y directions as shown in Figure 3e to compute the image gradient in the direction of image width and image height, respectively. Afterward, the algorithm calculates the amplitude and the direction of the resultant gradient. As Figure 3f illustrates, the next step involves the process of non-maximum suppression to remove any irrelevant pixels, that is, the pixels that are less likely to form an edge. The method employs double thresholding and edge-tracking by hysteresis method to detect and connect crack edges as illustrated in Figure 3g. Finally, the algorithm adds red color to detected pixels. Figure 3h shows the processed image captured from the headsets' view.

The headset current computational power is lower than stationary or hand-held computing devices (Santi et al., 2021). Therefore, the Canny algorithm is not implemented in real time inside the AR headsets, and the initial image processing algorithm requires memory complexity reduction to be real time. The researchers make two simplifications to the algorithm that streamline the system and reduce the processing time. Figure 4 describes those simplifications. First, this study assumes that the user's field of attention is limited to a specific part at the center of the camera's field of view. Therefore, the photo off-center (to the extent that it is outside the user's field of attention) is excluded from the image processing as demonstrated in Figure 4a. The AR tool is designed so that the users can adjust the size of the processed part. Second, this study reduces the memory complexity by avoiding pixel-by-pixel processing of images. The researchers apply the Canny algorithm to the pixels in every other row and column as shown in Figure 4b. The unprocessed pixels are evaluated based on the result of adjacent processed pixels as shown in Figure 4c. The algorithm considers the unprocessed pixels on the edge of the crack if at least half of their adjacent processed pixels are on an edge.

2.4.2 | Computational challenge

The research team initially design the algorithm as a C# code for a Unity project in a computer. The proposed methodology is applicable for AR headsets with embedded processing capabilities that can deploy Unity projects. Unity engines is a software to develop applications for AR headsets and is compatible with diverse platforms such as Windows, OS, IOS Linux, Mac, and Android (Santi et al., 2021). Therefore, the methodology for real-time AR crack detection proposed in this study can be implemented in any other AR headset with the following properties: (1) integrated computing ability (2) optical see-through features, and (3) automatic photo capturing/rendering. In addition, this study implies the independence of the proposed method from AR headset by testing the method for two generations of AR headsets.

There are two major software limitation in image processing in the AR Headsets: (1) matrix capability is not available inside the Unity-C# environment to save the photo's information. (2) Unity-C# environment proves deficient in sophisticated image arithmetic functions. A possible solution to overcome the mentioned limitations is to install an auxiliary package inside the Unity project. For example, OpenCVSharp, a library of programming functions, provides facility for the real-time processing of images. Therefore, the researchers initially develop the AR crack-detection tool using OpenCVSharp package and successfully deploy it in a stationary AR device (a PC). However, OpenCVSharp package appears to be incompatible with the AR headsets, and after employing the possible methods to deploy the package, the AR headsets are unable to load its dynamic-link library files. Therefore, this study redesigns the crack detection tool without supplementary packages and saves the pixel values of the photos in the single-dimensional array. This appears to be the only possible means in C#-Unity environment for saving big numeric data. Then, this study utilizes the limited math operators available in C#-Unity to write the detection code and create a new Unity project compatible with the AR headsets platform. The challenge in deploying the computer image-processing project in the AR headset is that several factors such as photo length and width are different in computer and AR headset. In this stage, the researchers set up a correspondence between C# code variables' indexes in the computer and the pixels' indexes in the AR headset photo and finally deploys the crack detection project in the AR headset.

The next section describes the processing time and several potential civil engineering applications of the methodology.

3 | PROPOSED AR-CRACK DETECTION

3.1 | AR standalone crack detection

This study develops an AR decision-assisting tool for human crack inspection in the field. The implemented crack detection methodology is standalone and does not

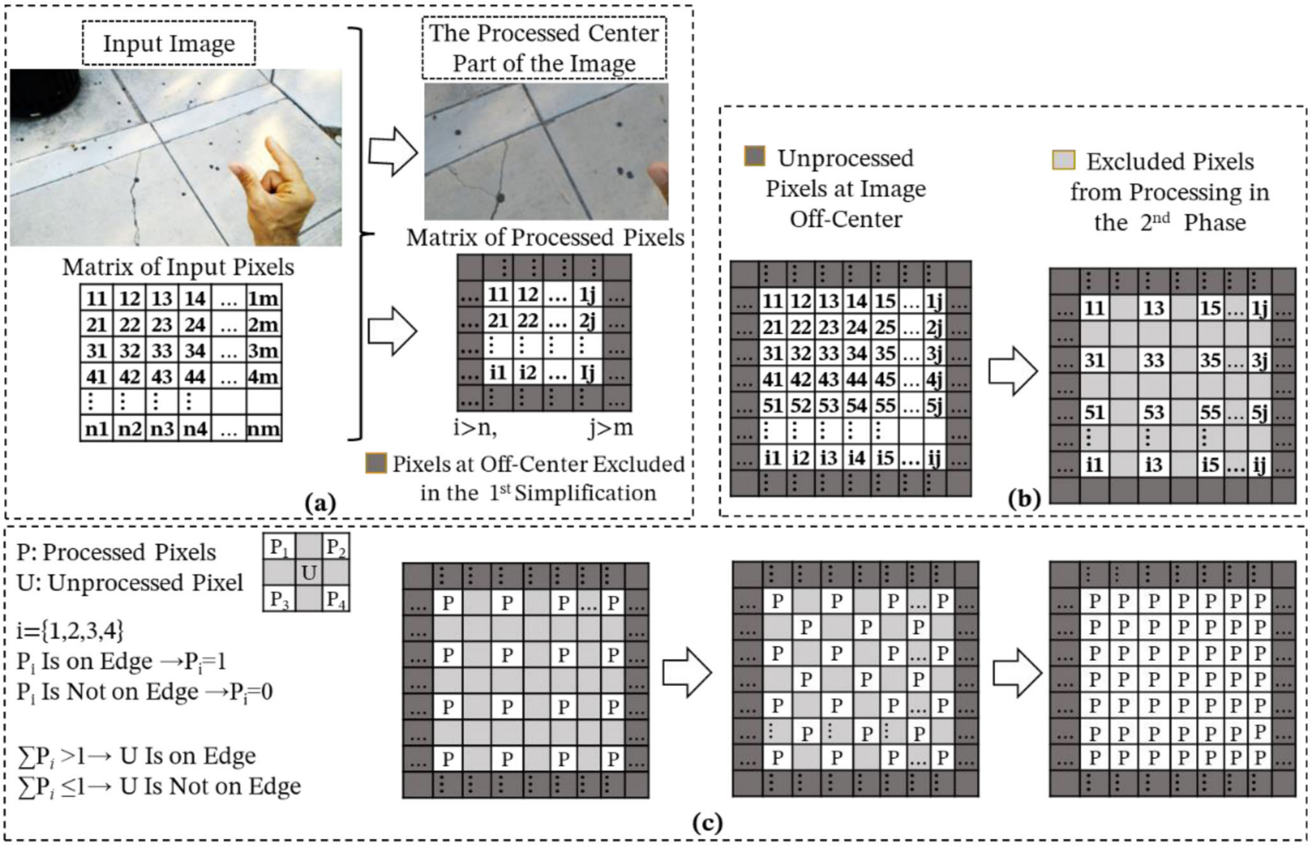


FIGURE 4 Simplifications at pixel level (a) first modification excludes pixels at image off-center, (b) second simplification excludes pixels at every column and rows, and (c) assessment of the unprocessed pixels based on their adjacent processed pixels

require a wire connection to link to external processors. The physical connection of an AR headset to a processing device imposes a limitation in mobility for users (Lang et al., 2019). Therefore, the proposed system eliminates the mobility limitation caused by wire connection to external processors. Additionally, the developed AR app does not require connecting to the Internet. This independence from the Internet connection removes the latency caused by data transfer between the external processor and the AR headset (L. Liu et al., 2019). In addition, this self-dependence resolves the problem caused by Internet limitations in remote locations (Pourhomayoun et al., 2016) especially in developing countries (Sambuli, 2016). Furthermore, the AR headset can connect to external processors only when running on the same wireless network (Jana et al., 2017). Therefore, the self-sufficiency of the proposed system drops the requirement for availing external processors in inspection fields.

The existing AR crack detection methods that utilize external processing units are superior in accuracy because they benefit from higher processing power. Therefore, those methods can run more sophisticated algorithms such as video processing or ML and provide more accurate results. The approach with an external processor, however, involves non-real-time processing because of the latency that exists in the connection establishment between the AR device and the processing unit (L. Liu et al., 2019). For example, Mojidra et al. (2022) developed a system for crack detection in structures using AR. They created a computer vision algorithm to analyze short videos recorded from cracked surfaces and made a database and hotspot connection between the AR headset and the algorithm. They detected and localized cracks with high accuracy and decreased the detection time to approximately 30 s. The following discussion quantifies the processing time of the developed AR tool showing the proposed method can achieve real-time processing.

3.2 | Processing time

The top priority of infrastructure planners is to enable the inspectors to collect/process field data in real time while performing inspections, according to past studies (Byers & Otter, 2006; Maharjan et al., 2021). To reduce the processing time, this study uses the two mentioned simplifications that include: (1) limiting the processing to a rectangular part in the center of the image and (2) applying the

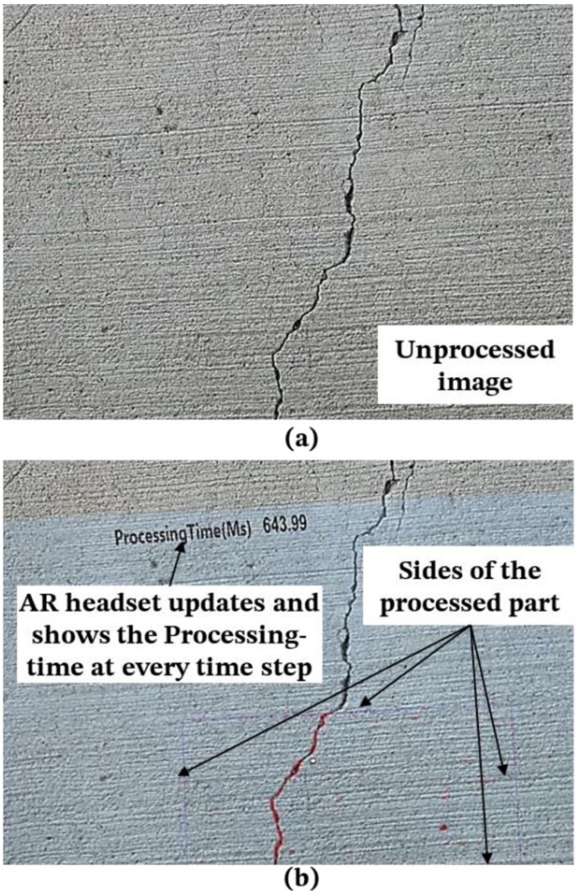

Unprocessed image
(a)
ProcessingTime(Ms) 643.99
AR headset updates and shows the Processing-time at every time step
Sides of the processed part
(b)

FIGURE 5 Processing time computation in AR headset

Canny algorithm to the pixels in every other row and column of the rectangle. Figure 5 explores the processing time computation inside the AR headset and shows how the AR tool displays the processing time at every time step in front of the user. Because the processing time is varying in each processed image, this study uses the average time of 15 sequential images to evaluate the processing speed.

Figure 6 shows the effect of simplifications on the processing time. Y-axis shows the processing time, and X-axis represents the proportion of the processed dimensions to the image's dimensions that is the same for both length and width during the experiments. Figure 6a,b compares the processing time of the simplified and original Canny algorithms, respectively, for the first and second-generation headsets when the image acquisition mode is photo capturing. The result shows that the simplification significantly reduces the processing time in both AR headsets. Figure 6c shows the processing time of the simplified Canny algorithm for the two generations of the AR headset when the image acquisition is through webcam mode. The result shows that image processing in webcam mode is faster than in photo capture mode. However, the webcam mode domain of the application is limited to lighter algorithms. For example, if the rectangle dimensions exceed 40% of the photo dimensions, the only implementable camera mode is photo capturing. Figure 6c demonstrates several examples of the processing time of the webcam mode for the two headsets. To provide more insight into the processing time of the AR tool, the image processing time is separated from the time of photo capturing, localization, and photo rendering. The total processing time includes the time of photo capturing, image processing, localization, and photo rendering. These examples show that the overall processing time of the first generation is more than the second generation for the webcam mode.

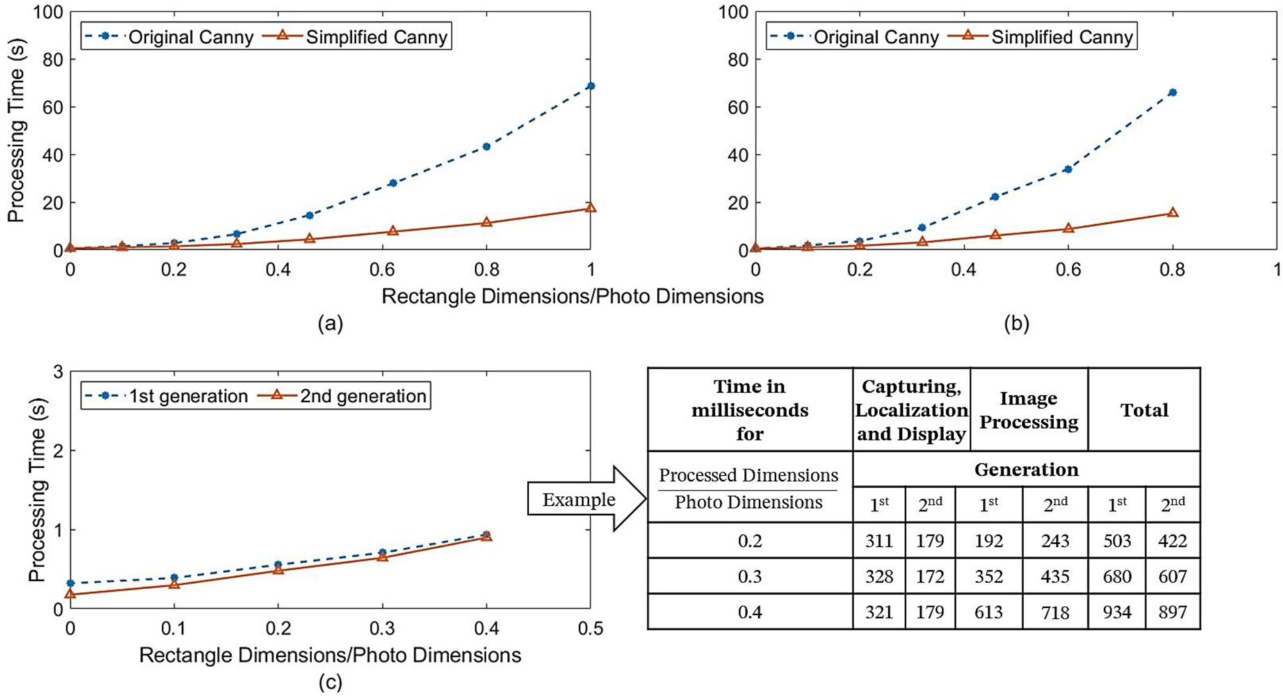
3.3 | Civil engineering applications

The benefits of AR crack detection tools include, but is not limited to (Karaaslan et al., 2019): (1) The inspectors access a real-time image processing tool for crack detection, (2) the system can automatically save the processed image of structures with the cracks marked in the image to facilitate the damage assessment and damage prognosis process, (3) the inspector can detect the cracks if they are distant or in a hard to access location. The AR headset augments the image in those conditions and the inspectors can perform their evaluation using the processed image without needing any access devices such as conveyors or stairs, (4) the inspector can enhance detection efficiency by modifying the image processing parameters based on the environmental conditions during the inspection, and (5) identification of cracks' edges in an image is the first step in crack characterization methodologies (e.g., Y. -F. Liu et al., 2020; Wang et al., 2018). The proposed method can serve as the first step in crack dimensional characterization and condition evaluation with AR headsets. The mentioned objective is one of the ongoing efforts, the research team is focusing on as the future work of the proposed methodology.

The next section explores the experimental setup used to quantify the accuracy of the AR image processing tool.

4 | EXPERIMENT SETUP AND ACCURACY QUANTIFICATION METHOD

The aim of the experiments is to assess the effect of several parameters on detection accuracy. This section will first explain the setting used for the field experiments. Afterward, it explores the method used for quantifying the accuracy of the experimental results.



Time in milliseconds for	Capturing, Localization and Display		Image Processing		Total	
Processed Dimensions / Photo Dimensions	Generation					
	1st	2nd	1st	2nd	1st	2nd
0.2	311	179	192	243	503	422
0.3	328	172	352	435	680	607
0.4	321	179	613	718	934	897

FIGURE 6 Comparing the processing time of (a) simplified and original Canny algorithm for first generation/photo mode, (b) simplified and original Canny algorithm for second generation/photo mode, (c) first and second generations of simplified Canny algorithm for webcam mode

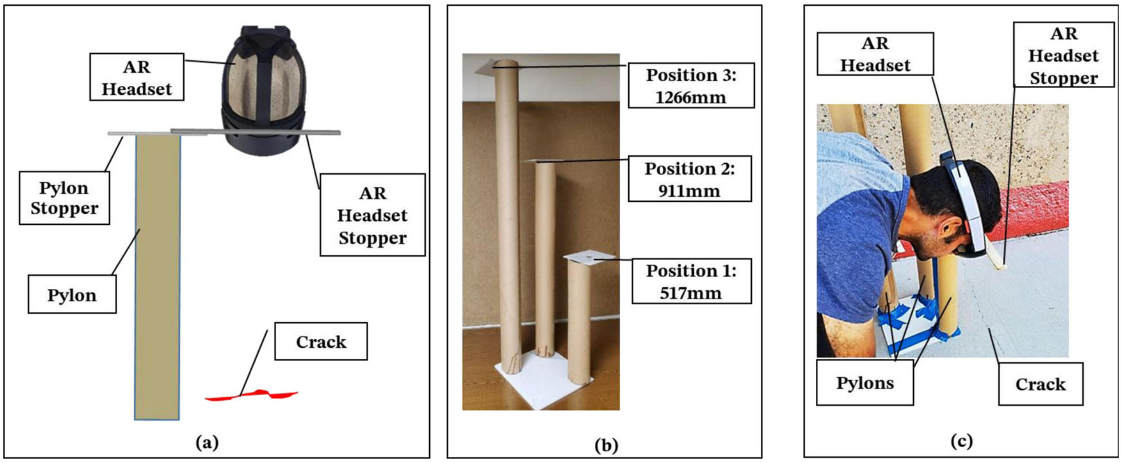


FIGURE 7 Experiment setup: (a) three test seats and their positions, (b) cracks' view and description, and (c) example of the test

4.1 | Experiment setup

Several factors are effective in crack edge extraction with image processing (Mohan & Poobal, 2018). The present experiments analyze the effect of crack size, headset position, camera modes and generation of AR-headset (the first and second generations of HoloLens) on detection accuracy. Additionally, the effect of Canny parameters is briefly evaluated in the experiments.

Figure 7 shows the test setup of the present research. The researchers used a pylon with horizontal stoppers and attached a horizontal bar to the AR headset to adjust and fix the position of the HoloLens camera during the experiments as illustrated in Figure 7a. The pylon contains three stoppers at different positions on which the AR headset remains fixed during the experiments as shown in Figure 7b. Position 1 at 517 mm is within the allowable range of visual inspection codes (e.g., ASME BPVC-Section

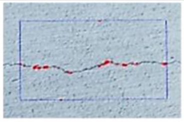
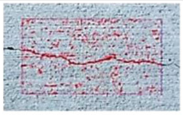
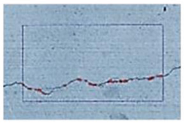
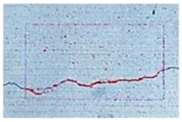
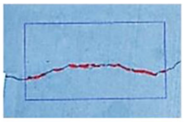
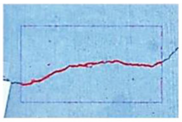
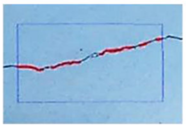
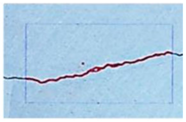
No.	Crack width (mm)	Pylon height (mm)	Result 1st Generation	 2nd Generation
1	0.69	517		
2	1.16	911		
3	1.65	1266		
4	2.44	1266		

FIGURE 8 Experiment results for both headset generations considering different variables

I-T-952 for visual inspection mandates the inspectors' eye to be within 610 mm of the inspected region). The research team selected this position as the benchmark for the highest accuracy, which corresponds to the arm's length of the inspector. The subsequent position at 911 mm was selected using several preliminary tests to make sure the distances between the stoppers cause a meaningful transition in the quality measure values when changing from Position 1 to 2. Position 3 at 1266 mm, which is approximately 250% farther than the arms-length case, corresponds to a distance at which the inspector can miss some of the cracks with unaided eyes and can benefit from the AR tool for identifying those cracks. Figure 7c shows the experiment on a concrete crack and shows how the test setup enabled fixing the distance from the headset to the concrete surface.

The experiments involved testing the app on four concrete cracks of different width as demonstrated in Figure 8. This figure provides examples of processed photos of the four tested cracks associated with eight instances of the experiments. The researchers measured cracks' width manually using a calibrated digital caliper with a resolution of 0.01 mm. Figure 8 also describes the manual measurements of the cracks' average width. There were two manual measurements in each centimeter of crack length.

The performance of the implemented Canny algorithm depends on three parameters; (1) the size of the median filter used for noise reduction, (2) the upper, and (3) the lower thresholds utilized in the edge tracking step. The researchers conducted several preliminary experiments and tested different filter sizes. The result of the preliminary tests implies that the increase in the size of the median Kernel matrix to greater than [3 × 3] does not significantly enhance the detection efficiency. Therefore, to reduce the processing time a median filter of 3 × 3 was applied to the algorithm during the accuracy evaluation experiments.

Canny proposed $\left[\frac{2}{1}\ \frac{3}{1}\right]$ as the effective range of $\frac{upper\ threshold}{lower\ threshold}$ (Canny, 1986). After testing different threshold ratios in preliminary experiments, the research team selected a ratio of $\frac{3}{1}$ for the accuracy evaluation experiments. The normalized values of the upper threshold range from 0 to 1 in the implemented Canny algorithm. In the first phase of the accuracy evaluation experiment, the research team selected the same upper threshold of 0.2 for both generations of the AR headset. This upper threshold achieves optimum efficiency for the first generation at Position 1 (517 mm) and for the second generation at position 3 (1266 mm). In the second phase of the experiments, a different threshold was applied for the two generations. The research team applied an upper threshold of 0.1 and 0.3, respectively, to the first and second generations. Using the mentioned thresholds, the Canny algorithm achieved a high efficiency at Position 3 for the first generation and Position 1 for the second generation.

The experiments took place from 7:00 to 8:00 a.m. during three consecutive sunny days in August 2021 in shadow when lighting conditions did not significantly vary during the experiments.

4.2 | Accuracy quantification

For the evaluation of the exactness of the system, *Precision–Recall* analysis (Fawcett, 2006) is used in this study. Three quality measures, that is, *Recall*, *Precision*, and *Score* quantify the quality of the detection system:

$$Recall = \frac{True\ Positive}{True\ positive\ +\ False\ negative} \tag{1}$$

$$Precision = \frac{True\ Positive}{True\ positive\ +\ False\ positive} \tag{2}$$

$$Score = 2 \times \frac{Precision\ \times\ Recall}{Precision\ +\ Recall} \tag{3}$$

For *Precision–Recall* analysis, this study performs a manual image modification in combination with an RGB-pixel analysis code operating in MATLAB as explained in Figure 9. This figure uses a real crack example to show the steps of the quantification process. Figure 9a,b is, respectively, unprocessed and processed images of the AR headset recorded during an experiment. Figure 9c shows how the processed image is enhanced using ImageJ

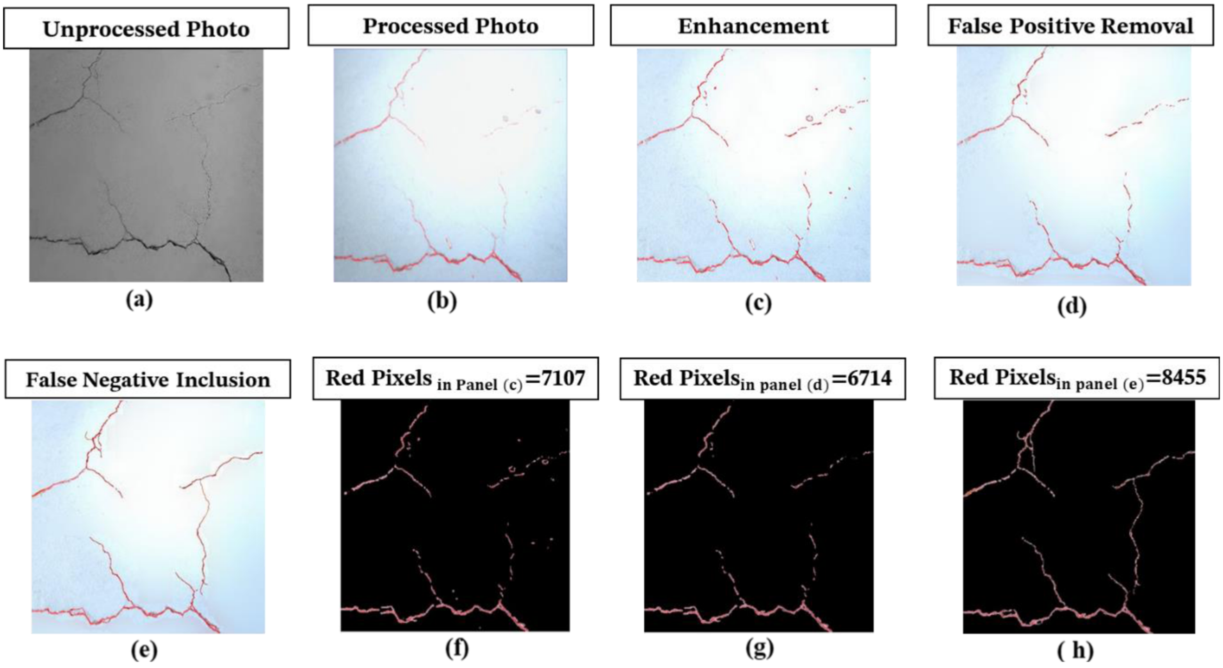


FIGURE 9 Crack pixels analysis steps: (a) crack photo, (b) HoloLens view, (c) enhanced photo, (d) manual removal of false indications, (e) manual completion of the crack, (f) true and false positive pixels, (g) true positive pixels, and (h) true positive and false negative pixels

computer software. The red pixels in Figure 9c comprise the entire positive predictions, that is, true positives plus false positives. In Figure 9d, the researchers manually remove the red indications associated with the false predictions, which means removing the false positives and maintaining the true positives. Figure 9e shows the manual process of adding red color to the false-negative pixels used in *Recall* calculation. The manual process of adding and removing red color to the pixels is conducted using ImageJ software. The thickness of red color (the number of pixels vertical to crack) is kept approximately constant throughout the length of the crack during the manual modifications, and therefore the evaluation is based on the number of pixels along the crack. The researchers evaluate each edge of cracks separately because in some parts of the cracks, one edge is detected, and the other edge is not. Finally, MATLAB processes the photos of Figure 9c–e, counts the red pixels as shown in Figure 9f–h, and calculates the quality measures as follows:

$$Recall = \frac{\text{Red pixels}_{in\ panel\ (d)}}{\text{Red pixels}_{in\ panel\ (e)}} = 0.79$$

$$\text{Precision} \ = \frac{\text{Red pixels}_{in\ panel\ (d)}}{\text{Red pixels}_{in\ panel\ (c)}} = 0.94$$

$$\text{Score} = \frac{2 \times Recall \times Precision}{Recall + Precision} = 0.86$$

The next section discusses the effect of simplifications and several ambient/ algorithmic parameters on the quality measures and describes the real-world applications and the prospects of the proposed methodology.

5 | EXPERIMENTAL TESTS AND ANALYSIS

To reduce the processing time, the present study streamlines the crack detection process using two simplifications. The following section will first explore the effect of these simplifications, on the quality measures, and then will continue with analyzing the effect of several ambient and algorithm parameters on the quality measures. The last part of the present section discusses the real-world applicability and future direction of the proposed methodology.

5.1 | Simplification evaluation

While the first simplification discussed in Section 2.3, does not affect the accuracy within the processed part, the effect of the second modification requires serious scrutiny. For evaluation of the simplification error, the researchers deployed two projects in the AR headsets; the projects are the same in every aspect except that the second simplification is applied just to one of the projects. The researchers performed two similar sets of experiments for the simplified and unmodified projects. Each set included 24 experiments with distinct different parametric combinations. The parameters that changed from one experiment to another included: (I) headset position (three positions), (II) crack width (four cracks) and (III) headset generation (two generations). Therefore, this study conducts 48 different experiments for simplification evaluation.

This study uses two statistical analyses of experimental results to assess the influence of this modification, on the quality measures. More specifically, the authors use linear regression and random forest importance analysis to compare the level of importance of the simplification error and the importance of several parameters, including headset position, crack width, and headset generation. First, the authors briefly describe the mentioned statistical methods.

5.1.1 | Random forest

Random forest (Breiman, 2001) is a famous ML algorithm. Due to its high accuracy and efficiency, researchers widely use the random forest in both classification and regression studies (Leistner et al., 2009). Tree impurity in a random forest classifier or regression model quantifies the effect of each independent variable on the dependent variables. The impurity calculation algorithm is quoted from a reference:

> For the n_{th} case in the data, its margin at the end of a run is the proportion of votes for its true class minus the maximum of the proportion of votes for each of the other classes. The measure of importance of the m_{th} variable is the average lowering of the margin across all cases when the m_{th} variable is randomly permuted. (Breiman, 2002)

This study uses random forest impurity analysis in RStudio software to compare the level of importance of simplification with other mentioned parameters (headset position, crack width, and headset generation) in quality measures.

5.1.2 | Linear regression

Linear regression establishes a linear relationship between the target data of dependent and independent variables (Rodríguez-Barranco et al., 2017). Linear regression's *t*-value of an independent variable shows its level of importance in the dependent variable (Ng et al., 2004):

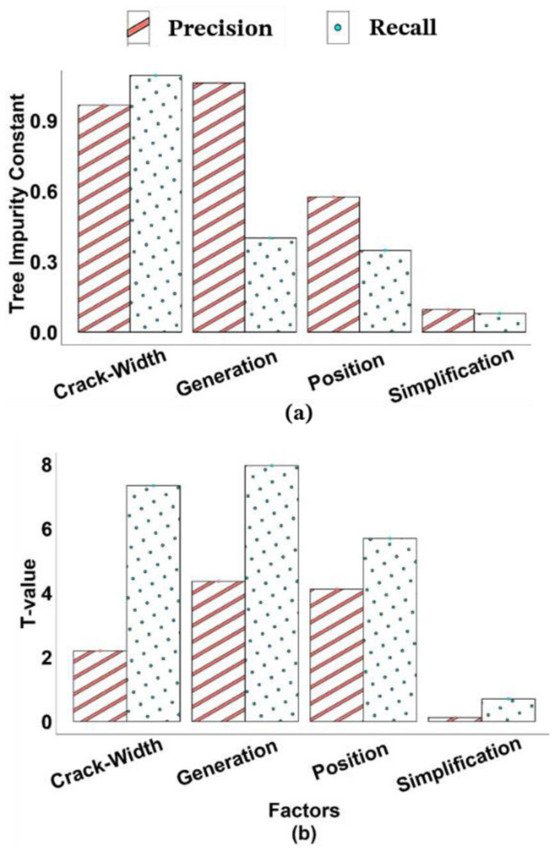


FIGURE 10 Comparison of two quality measures using (a) random forest and (b) linear regression

$$t - \text{value} = \sqrt{\frac{(\text{n} - 1)\,\text{MBE}^2}{\text{RMSE}^2 - \text{MBE}^2}} \tag{4}$$

where *MBE* is the mean bias error, *RMSE* is the root mean square error, and *n* is the number of data. A high *t*-value for an independent variable in a regression model indicates its strong effect on the dependent variable (Malek et al., 2021). This study uses linear regression's *t*-value analysis in RStudio software to compare the effect of simplification with other mentioned parameters on accuracy.

5.1.3 | Results of simplification evaluation

Figure 10 compares the importance level of simplification with the importance of selected effective parameters using the two mentioned statistical methods. Random forest and linear regression indicate a high significance level of crack width, headset version, and headset position for all quality measures, which shows their important effect on detection accuracy. Compared to other factors, simplification does not significantly affect accuracy. The authors provide an

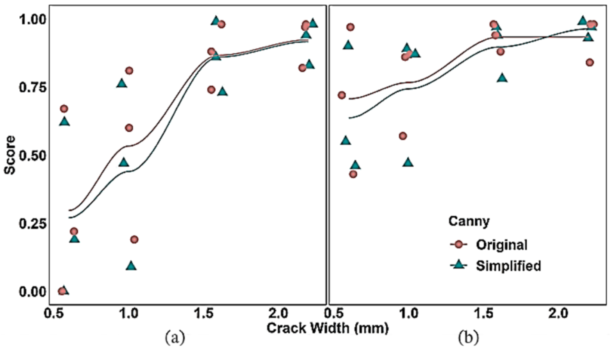


FIGURE 11 Accuracy comparison (a) first generation and (b) second generation

example from Figure 10 in the following to explore how big the effect of the simplifications, compared to other parameters, is. For instance, the *t*-value of the headset version, crack width, crack position, and simplification for *Recall* are 7.96, 7.34, 5.69, and 0.704, respectively. This implies a minor effect of the simplification on the accuracy especially, compared to the effect of the other parameters. Overall, the results of linear regression and random forest find simplification as the least important factor for all quality measures showing that it does not fundamentally affect the accuracy.

Figure 11 compares the accuracy of the simplified and original Canny algorithms. Each graph includes two point plots and two smooth plots. The point plots demonstrate the calculated values of the score as small circles and triangles for the first and second generations, respectively. The smooth plots are the solid curves showing the average score values at each point for the two AR headsets. Figure 11 shows that the average score of the original Canny algorithm is higher than the simplified algorithm used in this study for the first and second generations. Maximum accuracy drops resulting from the second simplification during the experiments are ~9.5% and ~7.0% for the first and second generations, respectively. Additionally, the inaccuracy caused by the simplifications decreases at higher crack widths.

5.2 | Accuracy evaluation experience

5.2.1 | Preliminary experience

The preliminary experiments at the campus of the University of New Mexico included testing a dataset of 15 cracks with different patterns and widths at different distances and light conditions using several Canny thresholds (Moreu & Malek, 2021). The examined patterns included longitudinal, traverse, diagonal, map, craze, corner, and D-crack. Overall, the result shows that the AR tool can detect cracks of different patterns and shapes, and no significant influence of crack type on the accuracy was observed (Moreu & Malek, 2021).

TABLE 4 Experiment combination summary

Parameter	Options	Number
Camera mode	• Webcam • Photo capturing	2
HoloLens version	• First generation • Second generation	2
Headset positions	• 517 mm • 911 mm • 1266 mm	3
Cracks width	• 0.59 mm • 1.04 mm • 1.61 mm • 2.15 mm	4

Note: Total combinations: Camera Mode × Headset Version × Cracks Width × Headset Positions = 2 × 2 × 3 × 4 = 48.

Additionally, several preliminary experiments were conducted to evaluate the sensitivity of the AR tool to shadow and light conditions. The result of those experiments shows that shadow availability in images does not greatly influence the quality measures if the parameters of the Canny algorithm are correctly established (Moreu & Malek, 2021).

In addition, the results of preliminary experiments provide insight into the effect of Canny thresholds on accuracy. The results show that the optimal Canny thresholds change with the distance of the camera from the concrete surface. In other words, optimizing the algorithm requires adaptation of the Canny threshold based on the camera–crack distance. Automatic adaptation of Canny parameters based on distance is a possible future work for this study. Furthermore, the results of preliminary experiments demonstrate that optimal Canny thresholds are different in the two generations of the AR headset.

5.2.2 | Accuracy experiment parameters

After preliminary tests, the research team conducted accuracy evaluation experiments in two phases. The research team conducted 48 new experiments in the first phase and changed the selected effective parameters to explore the trend of the quality measures with the change in the effective factors. Table 4 shows the different combination of parameters in the experiments. In the next phase, the researchers performed several additional experiments to explore the effect of Canny thresholds on accuracy.

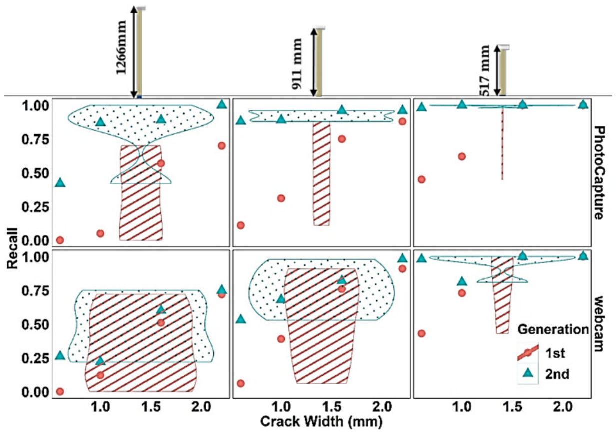


FIGURE 12 Recall's trend of change with effective factors

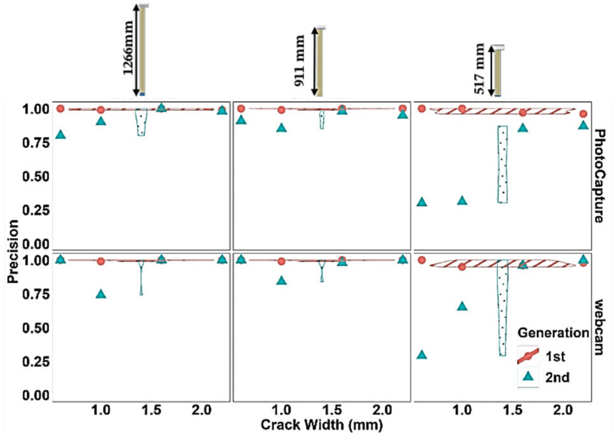


FIGURE 13 Precision's trend of change with effective factors

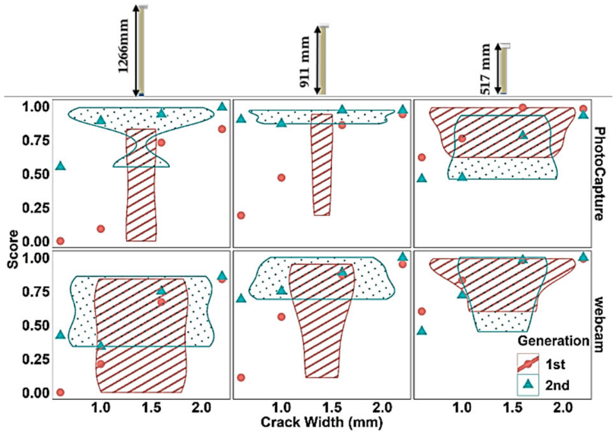


FIGURE 14 Score's trend of change with effective factors

5.2.3 | Results of accuracy evaluation

Figures 12–14 summarize the effect of different factors on the three quality measures: *Recall, Precision, and Score*, respectively. The Y and X-axes show quality measure values and crack width, respectively. Moreover, each figure includes six panels. The three panels in each row represent the three positions of the AR headset, and the two panels in each column represent camera modes (photocapturing and webcam/video modes). Each panel includes a point plot and a violin plot. In the point plots, the calculated quality measure values are plotted as small opaque shapes (circles for the first generation and triangles for the second generation) on a coordinate plane. Dotted and hached pattern blocks correspond to the violin plots. Violin plots use density curves to demonstrate distributions of quality measure data for the two AR headsets (hatched and dotted patterns correspond to the first and second generations, respectively). The widths of the violin curves are approximately proportional to the density of data points in each part on the coordinate plane.

Figure 12 shows the effect of different factors on the *Recalls* of both headsets. This figure shows that there is a direct relationship between the crack size and *Recall*, for both the AR headsets. This means a higher number of correctly predicted pixels (true positives) in wider cracks than the narrower ones. In addition, *Recall* has an inverse relationship with the distance of the headset from the target crack, and the higher and the lower *Recall* values correspond to the closest (517 mm) and the furthest (1266 mm) positions, respectively. The result also demonstrates that the second generation achieves higher *Recall* values than the first. This can be attributed to the higher camera resolution of the second generation as mentioned in Table 3. The effect of camera mode on *Recall* follows different trends in the first and second generations. Camera mode does not seem to have a predictable effect on *Recall* in the first generation. However, using photo-capturing mode in the second generation provides results with higher *Recall* compared to webcam mode.

Figure 13 explains the effect of the effective factors on *Precision*. Proximity to the cracks tends to increase the false positives and the true positives that have a contractive effect on *Precision* (see Equation 2). These contractive effect cause *Precision* not to be affected by the headset position as shown in Figure 13. Moreover, *Precision* tends to increase with the crack size. This corresponds to the rise in true positives in wider cracks. Figure 13 shows the better performance of the first generation over the second generation in *Precision*. The lower performance of the second generation results from the more false positive pixels in the second version's processed photos thanks to the poor fit between the current Canny thresholds and the higher camera resolution of the second generation. Furthermore, *Precision* does not show a significant relation to the camera mode.

Figure 14 demonstrates the effect of the parameters on the score. *Score* is in direct relationship with crack

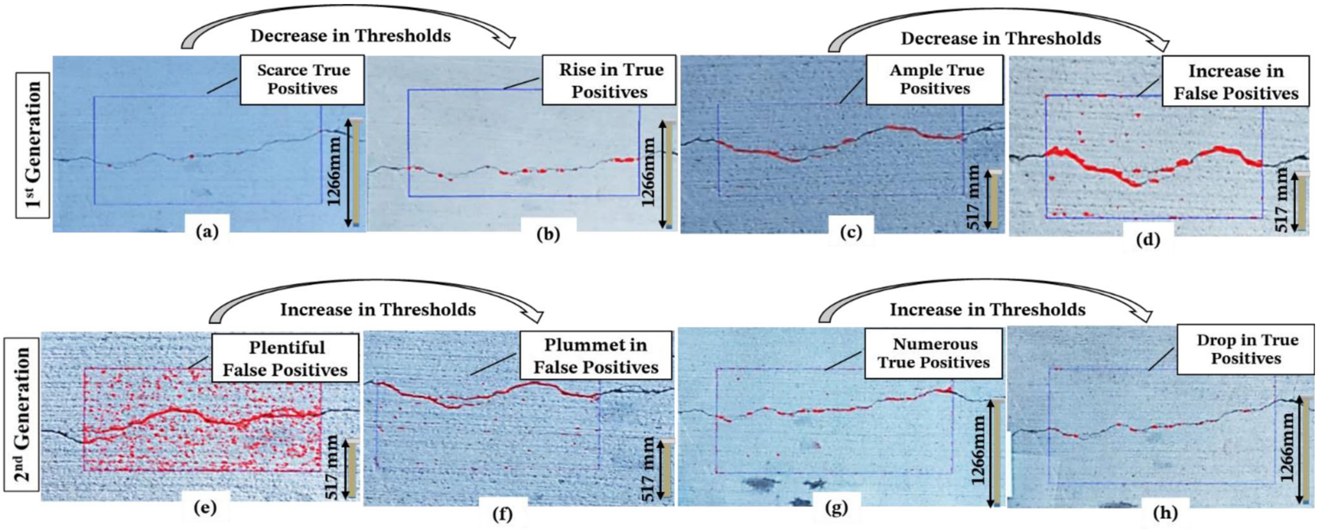


FIGURE 15 Explanatory photos of crack "No 2" with different Canny thresholds

width and in inverse relationship with the headset distance from the cracked surface. Moreover, Figure 14 illustrates that the second version has a higher average *Score* than the first version. However, when the AR headset is as close as 517 mm to the cracked surface, the first generation demonstrates slightly higher *Scores* than the second generation. In addition, the second generation achieves higher *Scores* with photo capturing mode than with webcam mode.

The researchers conducted a new set of experiments to explore the effect of pattern recognition thresholding. The Canny algorithm employs maximum and minimum threshold values so that pixels with a gradient amplitude of more than the maximum threshold are assumed edges and those below minimum thresholds are not. The decision for those pixels between these two thresholds depends on their proximity to previously detected crack pixels(Canny, 1986). Therefore, decreasing the Canny thresholds results in more (true/false) positive predictions (more red indications in photos) and vice versa. Figure 15a shows the *Recalls* of the first generation at 1266 mm in the first set of experiments, which is relatively low due to the scarcity of true positives. To mitigate these low *Recall* values, the researchers repeated the experiments with the first generation and imposed lower thresholds to the system. Figure 15b shows that more red indications overlay the crack with the higher thresholds, leading to higher *Recall* values. Figure 15c,d shows the processed image of the first generation form 517 mm with the thresholds used in the first set of experiments and with the new thresholds, respectively. Lowering the thresholds in the first generation rises the false positives at 517 mm but does change the true positives significantly. Therefore, this new threshold decreases the precision (see Equation 2). Figure 15e shows that the second generation produces numerous false indications at 517 mm. The research team repeated the experiments with the second generation and applied higher thresholds to the system. Applying higher Canny thresholds sharply reduces the false positive predictions at 517 mm as shown in Figure 15f. However, the results show that the new higher thresholds negatively influence the *Recall* at 1266 mm. For example, Figure 15g, which corresponds to the first set of experiments, has fewer correct positive predictions, compared to Figure 15h, that relate to the new experiments with higher thresholds. Therefore, this new thresholding causes a drop in *Recall* values.

In short, the object-detection capability of the AR headsets relates directly to crack width and relates inversely to the headset distance. Photo capturing mode results in higher *Recalls* than webcam mode in the second generation, but in the first generation, camera mode produces a minor effect on the accuracy. The effect of Canny thresholding on the *Recall* and *Precision* appears to be contractive depending on how far the headset is placed from the crack. Two generations of AR headset have different optimal thresholding criteria. However, using the same thresholds, the second generation of the AR headset works more accurately in crack detection from 911 mm and further, while the first generation's functions are more accurate from 511 mm and closer positions.

5.3 | Industry feedback and future work

The research team is now focusing on the industrial applicability and future steps required for the industry implementation of the proposed methodology. The first effort consisted of field test with bridge inspectors and

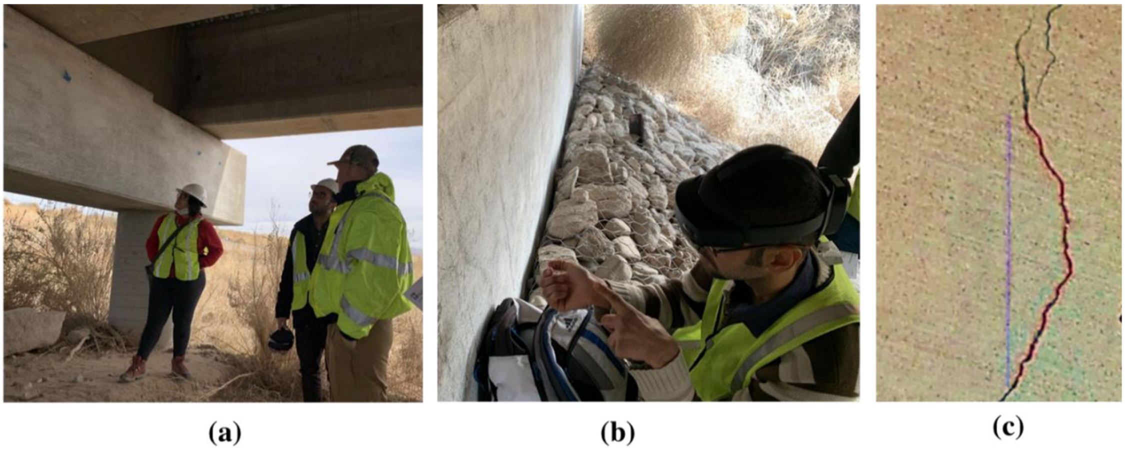


FIGURE 16 Test of industrial applicability with Department of Transportation (DOT) bridge inspectors and managers: (a) the bridge inspectors describing normal bridge inspection process, (b) AR inspector testing the app on a tiny crack on the bridge, and (c) the results of AR inspection

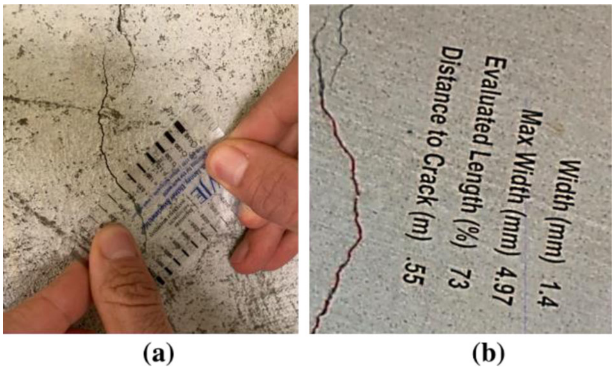


FIGURE 17 Test of preliminary crack characterization tool based on the proposed methodology (a) the traditional crack measurement approach and (b) AR crack characterization approach

managers interested to test new crack detection tools on their own bridge. The feedback from the inspectors and managers is critical to ensure that the work has an impact in the infrastructure industry. Figure 16 summarizes the field test of the AR tool developed in this study with bridge inspectors from the Department of Transportation (DOT). Figure 16a shows the research team learning about the interest of inspectors, bridge inspection requirements, and defect evaluation criteria at a specific bridge. Figure 16b shows the AR inspector conducting the crack inspection in front of the bridge managers; Figure 16c shows the results of the inspection that the bridge managers saw in the field on real time with the help of a monitor that shared the AR inspection results for those not wearing the headset.

DOT inspectors pointed out that the contribution of AR in bridge inspection increases by evolving the AR tool to a crack quantification tool that can measure the cracks and record the measurement data. Dimensional quantification of cracks and keeping a digital record of the crack distributions between inspections can serve as a digital change detector and tracker for inspectors. The inspector can compare the distribution of cracks from the past with any subsequent inspection and digitally score the changes using those data. Using this feedback, the research team is currently adding a new feature to the AR tool that aims to quantify the width of the crack in real time. Figure 17a shows traditional crack width measurement with a crack ruler that is locally measuring the crack width; Figure 17b shows the preliminary measurement tool (Moreu & Malek, 2021) examining a given strip of a surface crack along the crack length.

6 | CONCLUSION

This study deploys a crack detection algorithm based on the Canny method in AR headsets. The proposed methodology for AR crack detection is generalizable for any AR headset with embedded processing capability, which is compatible with Unity software. To reduce the processing time, several modifications have been made to the original algorithm. The experimental evaluation of the modification effect shows a maximum reduction of 9.5% in detection accuracy for the tested cracks. The modifications, however, enable real-time processing in the AR headset by reducing the processing time to 503 and 422 ms in two generations of AR headsets. In addition, several experiments on concrete cracks detail the effect of different ambient and algorithmic parameters such as headset position, camera resolution, Canny thresholds, and crack size on crack detection with AR headsets. Based on the accomplished steps of this study, the following future work is suggested by the authors:

1. The experiments conducted in this study show that the optimal parameters of the Canny algorithm change

with the distance of the camera from the concrete surface. Future work can be coding the distance of the AR headset from the crack using the depth camera capability of AR headsets to optimize the parameters of the Canny algorithm at different crack distances; the new algorithm will automatically optimize the Canny parameters at different distances.

2. The canny algorithm is a classic algorithm, which is superseded by new shape-based pattern recognition algorithms. New algorithms use more realistic assumptions (e.g., Iyer & Sinha, 2005), which can distinguish between cracks edges and irrelevant edges. In the next phase, the authors are going to use more effective algorithms, which are particularly aimed at detecting crack edges as opposed to Canny, which generalizes edges.
3. This study provides enough information for the research team to find the optimal Canny threshold within the distance evaluated in this paper, which is the practical range of crack measurement based on the camera resolution of the AR headset. Therefore, a future step of this research includes using the optimal detection level for developing a new algorithm that enables dimensional characterization of cracks within the distances evaluated in this paper.

The details of the modified Canny algorithm are currently being developed and evaluated by the industry to ensure its impact in a practical context is satisfactory for civil infrastructure inspectors and managers. It is expected that the code can be publicly accessible once industrially validated.

ACKNOWLEDGMENTS

The authors very much appreciate the support of Transportation Consortium of South-Central States (TranSET) for providing a grant for this project (Grant No. 69A3551747106). This research was also supported by the National Academy of Science Transportation Research Board (TRB) Rail SAFETY IDEA Program Project RS-43, project number 163420-0399; and the Federal Railway Administration (FRA) BAA project number FR20RPD34000000006. This research was also supported by the National Academy of Science Transportation Research Board (TRB) Rail SAFETY IDEA Program Project RS-43, project number 163420-0399; National Science Foundation, Computer and Information Science and Engineering (CISE), Division Of Information & Intelligent Systems. Award Number 2123346; and the Federal Railway Administration (FRA) BAA project number FR20RPD34000000006. The authors thank the Center of Advanced Research and Computing (CARC) of the University of New Mexico for providing an AR indoor laboratory to develop the holographic human-structure interface.

REFERENCES

Alam, K. M. R., Siddique, N., & Adeli, H. (2020). A dynamic ensemble learning algorithm for neural networks. *Neural Computing and Applications*, *32*(12), 8675–8690.

Al-Faris, M., Chiverton, J., Ndzi, D., & Ahmed, A. I. (2020). A review on computer vision-based methods for human action recognition. *Journal of Imaging*, *6*(6), 46.

Aravind, N., Nagajothi, S., & Elavenil, S. (2021). Machine learning model for predicting the crack detection and pattern recognition of geopolymer concrete beams. *Construction and Building Materials*, *297*, 123785.

Bahri, H., Krčmařík, D., & Kočí, J. (2019). Accurate object detection system on HoloLens using YOLO algorithm. *2019 International Conference on Control, Artificial Intelligence, Robotics Optimization (ICCAIRO)* (pp. 219–224). Athens, Greece.

Breiman, L. (2001). Random forests. *Machine Learning*, *45*(1), 5–32.

Breiman, L. (2002). *Manual on setting up, using, and understanding random forests v3. 1* (Vol. 1). Statistics Department University of California.

Brito, C., Alves, N., Magalhães, L., & Guevara, M. (2019). Bim mixed reality tool for the inspection of heritage buildings. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *IV-2-W6*, 25–29.

Byers, W. G., & Otter, D. (2006). Reducing the stress state of railway bridges with research. *Railway Track and Structures*, *102*(2), 14–17.

Canny, J. (1986). A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *8*(6), 679–698.

Çelik, F., & König, M. (2022). A sigmoid-optimized encoder–decoder network for crack segmentation with copy-edit-paste transfer learning. *Computer-Aided Civil and Infrastructure Engineering*, *37*(14), 1875–1890.

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

Chen, J., & He, Y. (2022). A novel U-shaped encoder–decoder network with attention mechanism for detection and evaluation of road cracks at the pixel level. *Computer-Aided Civil and Infrastructure Engineering*, *37*(13), 1721–1736.

Corneli, A., Naticchia, B., Cabonari, A., & Bosché, F. (2019). Augmented reality and deep learning towards the management of secondary building assets. *ISARC Proceedings* (pp. 332–339). Banff, Canada.

Dorafshan, S., Thomas, R. J., & Maguire, M. (2019). Benchmarking image processing algorithms for unmanned aerial system-assisted crack detection in concrete structures. *Infrastructures*, *4*(2), 19.

Doulamis, A., Doulamis, N., Protopapadakis, E., & Voulodimos, A. (2018). Combined convolutional neural networks and fuzzy spectral clustering for real time crack detection in tunnels. *2018 25th IEEE International Conference on Image Processing (ICIP)* (pp. 4153–4157). Athens, Greece.

Dung, C. V., & Anh, L. D. (2019). Autonomous concrete crack detection using deep fully convolutional neural network. *Automation in Construction*, *99*, 52–58.

Evans, G., Miller, J., Pena, M. I., MacAllister, A., & Winer, E. (2017). Evaluating the Microsoft HoloLens through an augmented reality assembly application. *Degraded Environments: Sensing, Processing, and Display 2017*, *10197*, 282–297.

Farasin, A., Peciarolo, F., Grangetto, M., Gianaria, E., & Garza, P. (2020). Real-time object detection and tracking in mixed reality using Microsoft HoloLens. *Proceedings of the 15th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications* (pp. 165–172). Valletta, Malta.

Fawcett, T. (2006). An introduction to ROC analysis. *Pattern Recognition Letters*, *27*(8), 861–874.

Georgiou, T., Liu, Y., Chen, W., & Lew, M. (2019). A survey of traditional and deep learning-based feature descriptors for high dimensional data in computer vision. *International Journal of Multimedia Information Retrieval*, *9*(3), 135–170.

Gopinathan, S., & Gayathri, S. (2016). A study on image enhancement techniques using YCbCr color space methods. *International Journal of Advanced Engineering Research and Science (IJAERS)*, *3*(8), 10–112.

Iyer, S., & Sinha, S. K. (2005). A robust approach for automatic detection and segmentation of cracks in underground pipeline images. *Image and Vision Computing*, *23*(10), 921–933.

Jana, A., Sharma, M., & Rao, M. (2017). *HoloLens blueprints*. Packt Publishing Ltd.

Karaaslan, E., Bagci, U., & Catbas, F. N. (2019). Artificial intelligence assisted infrastructure assessment using mixed reality systems. *Transportation Research Record*, *2673*(12), 413–424.

Lang, S., Dastagir Kota, M. S. S., Weigert, D., & Behrendt, F. (2019). Mixed reality in production and logistics: Discussing the application potentials of Microsoft HoloLensTM. *Procedia Computer Science*, *149*, 118–129.

Le, T.-T., Nguyen, V.-H., & Le, M. V. (2021). Development of deep learning model for the recognition of cracks on concrete surfaces. *Applied Computational Intelligence and Soft Computing*, *2021*, e8858545.

Leistner, C., Saffari, A., Santner, J., & Bischof, H. (2009). Semi-Supervised random forests. *2009 IEEE 12th International Conference on Computer Vision* (pp. 506–513). Kyoto, Japan.

Li, X., Jiang, H., & Yin, G. (2014). Detection of surface crack defects on ferrite magnetic tile. *NDT & E International*, *62*, 6–13.

Liu, C., & Xu, B. (2022). A night pavement crack detection method based on image-to-image translation. *Computer-Aided Civil and Infrastructure Engineering*, *37*(13), 1737–1753.

Liu, L., Li, H., & Gruteser, M. (2019). Edge assisted real-time object detection for mobile augmented reality. *The 25th Annual International Conference on Mobile Computing and Networking* (pp. 1–16). Los Cabos, Mexico.

Liu, Y.-F., Nie, X., Fan, J. -S., & Liu, X.-G. (2020). Image-based crack assessment of bridge piers using unmanned aerial vehicles and three-dimensional scene reconstruction. *Computer-Aided Civil and Infrastructure Engineering*, *35*(5), 511–529.

Luo, Y., & Duraiswami, R. (2008). Canny edge detection on NVIDIA CUDA. *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops* (pp. 1–8). Anchorage, AK.

Maharjan, D., Agüero, M., Mascarenas, D., Fierro, R., & Moreu, F. (2021). Enabling human–infrastructure interfaces for inspection using augmented reality. *Structural Health Monitoring*, *20*(4), 1980–1996.

Malek, K., Malek, K., & Khanmohammadi, F. (2021). Response of soil thermal conductivity to various soil properties. *International Communications in Heat and Mass Transfer*, *127*, 105516.

Martinez, P., Al-Hussein, M., & Ahmad, R. (2019). A scientometric analysis and critical review of computer vision applications for construction. *Automation in Construction*, *107*, 102947.

Mascareñas, D. D., Ballor, J. P., McClain, O. L., Mellor, M. A., Shen, C. -Y., Bleck, B., Morales, J., Yeong, L. -M. R., Narushof, B., Shelton, P., Martinez, E., Yang, Y., Cattaneo, A., Harden, T. A., & Moreu, F. (2021). Augmented reality for next generation infrastructure inspections. *Structural Health Monitoring*, *20*(4), 1957–1979.

Miao, P., & Srimahachota, T. (2021). Cost-effective system for detection and quantification of concrete surface cracks by combination of convolutional neural network and image processing techniques. *Construction and Building Materials*, *293*, 123549.

Miao, Y., Jeon, J. Y., & Park, G. (2020). An image processing-based crack detection technique for pressed panel products. *Journal of Manufacturing Systems*, *57*, 287–297.

Microsoft website. (2022). *OpenXR-Mixed reality [Company]*. Microsoft Build. https://docs.microsoft.com/en-us/windows/mixed-reality/develop/native/openxr

Mohan, A., & Poobal, S. (2018). Crack detection using image processing: A critical review and analysis. *Alexandria Engineering Journal*, *57*(2), 787–798.

Mojidra, R., Li, J., Mohammadkhorasani, A., Moreu, F., Collins, W., Bennett, C., & Taher, S. A. (2022). Vision-based inspection of out-of-plane fatigue cracks in steel structures. In D. Zonta, B. Glisic, & Z. Su (Eds.), *Sensors and smart structures technologies for civil, mechanical, and aerospace systems 2022* (pp. 145–151). SPIE Publications.

Moreu, F., & Malek, K. (2021*). Bridge cracks monitoring: Detection, measurement, and comparison using augmented reality. Data*. https://digitalcommons.lsu.edu/transet_data/125

Ng, M. W., Camerlengo, A., Abdul Wahab, A. K., & Harun, S. (2004). Estimation of evaporation and evapotranspiration in Malaysia using Penman and Christiansen methods. *Borneo Science*, *15*, 23–36.

Ni, F., Zhang, J., & Chen, Z. (2019). Zernike-moment measurement of thin-crack width in images enabled by dual-scale deep learning. *Computer-Aided Civil and Infrastructure Engineering*, *34*(5), 367–384.

Ogawa, K., Ito, Y., & Nakano, K. (2010). Efficient canny edge detection using a GPU. *2010 First International Conference on Networking and Computing* (pp. 279–280). Hiroshima Japan.

Pepe, A., Trotta, G. F., Gsaxner, C., Wallner, J., Egger, J., Schmalstieg, D., & Bevilacqua, V. (2018). Pattern recognition and mixed reality for computer-aided maxillofacial surgery and oncological assessment. *2018 11th Biomedical Engineering International Conference (BMEiCON)* (pp. 1–5). Chiang Mai, Thailand.

Pereira, D. R., Piteri, M. A., Souza, A. N., Papa, J. P., & Adeli, H. (2020). FEMa: A finite element machine for fast learning. *Neural Computing and Applications*, *32*(10), 6393–6404.

Piyathilaka, L., Preethichandra, D. M. G., Izhar, U., & Kahandawa, G. (2020). Real-time concrete crack detection and instance segmentation using deep transfer learning. *Engineering Proceedings*, *2*(1), 91.

Pourhomayoun, M., Alshurafa, N., Dabiri, F., Yadav, K., Sideris, C., Tseng, L., Ghasemzadeh, H., Nyamathi, A., & Sarrafzadeh, M. (2016). A robust remote health monitoring and data processing system for rural area with limited internet access. *Proceedings of the 11th EAI International Conference on Body Area Networks* (pp. 26–32). Turin, Italy.

Rafiei, M. H., & Adeli, H. (2017). A new neural dynamic classification algorithm. *IEEE Transactions on Neural Networks and Learning Systems*, *28*(12), 3074–3083.

Rodríguez-Barranco, M., Tobías, A., Redondo, D., Molina-Portillo, E., & Sánchez, M. J. (2017). Standardizing effect size from linear regression models with log-transformed variables for meta-analysis. *BMC Medical Research Methodology*, *17*(1), 44.

Safaei, N., Smadi, O., Masoud, A., & Safaei, B. (2022). An automatic image processing algorithm based on crack pixel density for pavement crack detection and classification. *International Journal of Pavement Research and Technology*, *15*(1), 159–172.

Salehi, H., & Burgueño, R. (2018). Emerging artificial intelligence methods in structural engineering. *Engineering Structures*, *171*, 170–189.

Sambuli, N. (2016). Challenges and opportunities for advancing internet access in developing countries while upholding net neutrality. *Journal of Cyber Policy*, *1*(1), 61–74.

Santi, G. M., Ceruti, A., Liverani, A., & Osti, F. (2021). Augmented reality in industry 4.0 and future innovation programs. *Technologies*, *9*(2), 33.

Shan, B., Zheng, S., & Ou, J. (2016). A stereovision-based crack width detection approach for concrete surface assessment. *KSCE Journal of Civil Engineering*, *20*(2), 803–812.

Ungureanu, D., Bogo, F., Galliani, S., Sama, P., Duan, X., Meekhof, C., Stühmer, J., Cashman, T. J., Tekin, B., Schönberger, J. L., Olszta, P., & Pollefeys, M. (2020). HoloLens 2 research mode as a tool for computer vision research. *ArXiv:2008.11239*.

Wang, S., Guo, R., Wang, H., Ma, Y., & Zong, Z. (2018). Manufacture assembly fault detection method based on deep learning and mixed reality. *2018 IEEE International Conference on Information and Automation (ICIA)* (pp. 808–813). Wuyishan, China.

Wang, S., Sakib Ashraf, Z., & Fuh-Gwo, Y. (2020). Augmented reality for enhanced visual inspection through knowledge-based deep learning. *Structural Health Monitoring*, *20*, 426.

Wang, W., Zhang, A., Wang, K. C. P., Braham, A. F., & Qiu, S. (2018). Pavement crack width measurement based on Laplace's equation for continuity and unambiguity. *Computer-Aided Civil and Infrastructure Engineering*, *33*(2), 110–123.

Wang, Y., Zhang, J. Y., Liu, J. X., Zhang, Y., Chen, Z. P., Li, C. G., He, K., & Yan, R. B. (2019). Research on crack detection algorithm of the concrete bridge based on image processing. *Procedia Computer Science*, *154*, 610–616.

Yamaguchi, T., Shibuya, T., Kanda, M., & Yasojima, A. (2019). Crack inspection support system for concrete structures using head mounted display in mixed reality space. *2019 58th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE)* (pp. 791–796). Hiroshima, Japan.

Zare, M. R., Alebiosu, D. O., & Lee, S. L. (2018). Comparison of handcrafted features and deep learning in classification of medical X-ray images. *2018 Fourth International Conference on Information Retrieval and Knowledge Management (CAMP)* (pp. 1–5). Sabah, Malaysia.

Żarski, M., Wójcik, B., Książek, K., & Miszczak, J. A. (2022). Finicky transfer learning—A method of pruning convolutional neural networks for cracks classification on edge devices. *Computer-Aided Civil and Infrastructure Engineering*, *37*(4), 500–515.

Zhang, Y., & Yuen, K.-V. (2021). Crack detection using fusion features-based broad learning system and image processing. *Computer-Aided Civil and Infrastructure Engineering*, *36*(12), 1568–1584.

Zheng, Y., Gao, Y., Lu, S., & Mosalam, K. M. (2022). Multistage semisupervised active learning framework for crack identification, segmentation, and measurement of bridges. *Computer-Aided Civil and Infrastructure Engineering*, *37*(9), 1089–1108.

Zhou, Z., Zhang, J., & Gong, C. (2022). Automatic detection method of tunnel lining multi-defects via an enhanced you only look once network. *Computer-Aided Civil and Infrastructure Engineering*, *37*(6), 762–780.

Zou, D., Zhang, M., Bai, Z., Liu, T., Zhou, A., Wang, X., Cui, W., & Zhang, S. (2022). Multicategory damage detection and safety assessment of post-earthquake reinforced concrete structures using deep learning. *Computer-Aided Civil and Infrastructure Engineering*, *37*(9), 1188–1204.

How to cite this article: Malek, K., Mohammadkhorasani, A., & Moreu, F. (2023). Methodology to integrate augmented reality and pattern recognition for crack detection. *Computer-Aided Civil and Infrastructure Engineering*, *38,* 1000–1019. https://doi.org/10.1111/mice.12932