Architectural Engineering -- Faculty Publications

Architectural Engineering and Construction, Durham School of

Benchmarking Image Processing Algorithms for Unmanned Aerial System-Assisted Crack Detection in Concrete Structures

Sattar Dorafshan
Turner-Fairbank Highway Research Center, sattar.dorafshan.ctr@dot.gov

Robert J. Thomas
Clarkson University, rthomas@clarkson.edu

Marc Maguire
Utah State University, m.maguire@usu.edu

Article

Benchmarking Image Processing Algorithms for Unmanned Aerial System-Assisted Crack Detection in Concrete Structures

Sattar Dorafshan [1,*]**, Robert J. Thomas** [2] **and Marc Maguire** [3]

1   Office of Infrastructure Research and Development, Turner-Fairbank Highway Research Center, Federal Highway Administration, Sterling, VA 22101, USA

2   Department of Civil and Environmental Engineering, Clarkson University, Potsdam, NY 13699-5710, USA; rthomas@clarkson.edu

3   Department of Civil and Environmental Engineering, Utah State University, Logan, UT 84322-4110, USA; m.maguire@usu.edu

*   Correspondence: sattar.dorafshan.ctr@dot.gov

check for updates

Abstract: This paper summarizes the results of traditional image processing algorithms for detection of defects in concrete using images taken by Unmanned Aerial Systems (UASs). Such algorithms are useful for improving the accuracy of crack detection during autonomous inspection of bridges and other structures, and they have yet to be compared and evaluated on a dataset of concrete images taken by UAS. The authors created a generic image processing algorithm for crack detection, which included the major steps of filter design, edge detection, image enhancement, and segmentation, designed to uniformly compare different edge detectors. Edge detection was carried out by six filters in the spatial (Roberts, Prewitt, Sobel, and Laplacian of Gaussian) and frequency (Butterworth and Gaussian) domains. These algorithms were applied to fifty images each of defected and sound concrete. Performances of the six filters were compared in terms of accuracy, precision, minimum detectable crack width, computational time, and noise-to-signal ratio. In general, frequency domain techniques were slower than spatial domain methods because of the computational intensity of the Fourier and inverse Fourier transformations used to move between spatial and frequency domains. Frequency domain methods also produced noisier images than spatial domain methods. Crack detection in the spatial domain using the Laplacian of Gaussian filter proved to be the fastest, most accurate, and most precise method, and it resulted in the finest detectable crack width. The Laplacian of Gaussian filter in spatial domain is recommended for future applications of real-time crack detection using UAS.

Keywords: structural condition assessment; concrete structures; unmanned aerial systems; crack detection; image processing; noncontact methods

1. Introduction

The United States is home to more than 600,000 bridges, more than one-third of which include a concrete superstructure or wear surface [1]. These bridges require a variety of periodic inspections in accordance with federal regulations. The most common inspection type is routine inspection, wherein the inspector scans the bridge deck to identify surface degradation or surface cracking. Such inspections are costly, time-consuming, and labor-intensive [2,3]. Autonomous inspection could be a cost-effective solution to these problems if the accuracy of human inspection can be matched [3–9]. Image-based inspection of infrastructure for concrete delamination [10–13], cracks [14–17], and spalls [18,19] using unmanned aerial systems (UASs) have been proven effective based on previous literature [20].

Image-based autonomous inspections still require human inspectors to review images. The number of images collected depends on a number of factors, but it is commonly in the several thousands. For instance, SDNET2018 [21], with more than 56,000 labeled images of concrete structures, covers three small lab-made bridge decks, walls of a building, and several paved sidewalks, which are significantly smaller than common inspected infrastructures in practice. Manual identification of flaws in such large image sets is time consuming and prone to inaccuracy because of inspector fatigue or human error [22–26]. Image processing algorithms can improve the accuracy and efficiency of autonomous inspections by either (a) enhancing images to improve ease of human detection of defects or (b) autonomously identifying defects. Additionally, edge detectors are used in combination with more contemporary techniques such as deep learning convolutional neural networks for UAS applications [27], reducing false positive cases by 20 times compared to sole use of edge detectors [28].

Cracks in a two-dimensional (2D) image are classified as edges, and, thus, existing edge detection algorithms are likely candidates for crack identification. 2D images are represented mathematically by matrices (one matrix, in the case of greyscale images, or three matrices in the case of red/green/blue color images). An ideal edge is defined as a discontinuity in the greyscale intensity field. Crack detection algorithms can emphasize edges by applying filters in either the spatial or frequency domain. Even though use of edge detectors for crack detection goes back to the early 2000s [29], these methods have been used in the past and are still being used in recent studies because of their simplicity and pixel-based detection of cracks [15,28,30–42]. Even in emerging applications of supervised machine learning methods, edge detectors are still considered in practice since they do not require expensive annotated training datasets. Using edge detectors for crack detection is computationally fast, making it an appealing option for real-time crack detection during UAS inspections; they have been implemented in ground-based robotic inspections in the past [4,18,38]. Edge detectors can have different sizes, shapes, and values. There are limited guidelines for researchers and practitioners in choosing a proper edge detector for their applications, especially for UAS collected data.

Save two noteworthy exceptions, most research focuses on developing new methods for crack detection rather than comparing the performance of existing methods. Abdel-Qader et al. [29] compared the performance of the fast Haar transform, Fourier transform, Sobel filter, and Canny filter for crack detection in 25 images of defected concrete and 25 images of sound concrete. The fast Haar transform was the most accurate method, with an overall accuracy of 86%, followed by the Canny filter (76%), Sobel filter (68%), and the Fourier transform (64%). Processing time was not considered, and no actual definition of accuracy was presented. Mohan and Poobal [43] reviewed a number of edge detection techniques for visual, thermal, and ultrasonic images, but the information presented was from several studies that considered vastly different data sets, and so the results were not directly comparable. This paper presents the results of a direct comparison of four common edge detection methods in the spatial domain (Roberts, Prewitt, Sobel, and Laplacian of Gaussian) and two in the frequency domain (Butterworth and Gaussian) by applying them to a dataset of 50 sound and 50 defected images of concrete collected from UAS. The goal of this study is to determine the most efficient edge detectors to use in aiding UAS condition assessments of concrete structures when all the other parameters are kept the same. Prior to implementation, as an emerging technology in bridge inspections, UAS need a rigorous investigation as part of a process model to determine whether they can be used in lieu of hands-on inspections. While this is not a part of this study, such an approach would involve investigating the reliability and effectiveness of UAS inspections through a generic decision making tool that includes a multivariable analysis of the inspection accuracy, cost, time, and hazard.

2. Analytical Program

Figure 1 shows a generic image analysis algorithm developed for this study. The generic algorithm included three main steps: Edge detection, edge image enhancement, and segmentation. Edge detection in the spatial domain involved greyscale conversion and application of a filter. Edge detection in the frequency domain required additional steps to transform the image from the spatial domain to the

frequency domain before application of the filter. The inverse operation to transform the filtered image back to the spatial domain was also an additional step. This section includes the particulars of each step in the generic image processing algorithm.

(a)

(b)
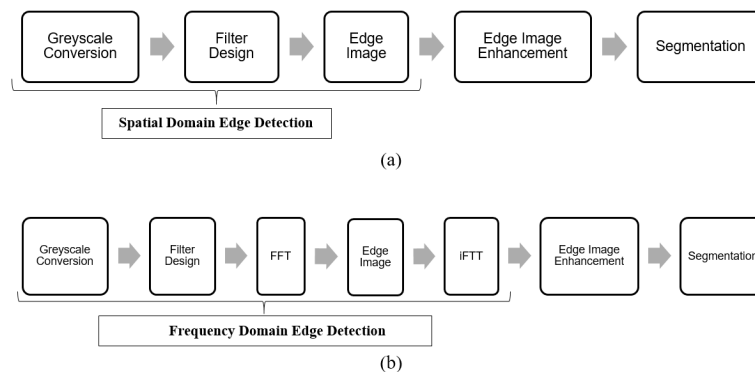
Figure 1. The steps in the proposed crack detection algorithm. (**a**) The spatial domain and (**b**) the frequency domain.

2.1. Greyscale Conversion

Edge detection algorithms perform best with greyscale images [44], so the first step in the image analysis procedure was greyscale conversion of color images. The original color image comprised a matrix of pixels, each with a defined red, green, and blue intensity. Greyscale conversion followed Equation (1), where $I(x, y)$ is the grayscale intensity of pixel (x, y), and $R(x, y)$, $G(x, y)$, and $B(x, y)$ are the red, green, and blue pixel intensities of the same, respectively.

$$I(x, y) = 0.2989R(x, y) + 0.5870G(x, y) + 0.1140B(x, y). \tag{1}$$

2.2. Edge Detection in the Spatial Domain

In general, edge detection in images requires filtering by one of several common methods, which are discussed in detail below. Filters are applied as a small matrix of values (called a kernel) through a mathematical operation known as convolution. In general form, the convoluted image **O** is the sum of the element-by-element products of the image intensity matrix **I** and the kernel **K** in every position in which **K** fits fully inside **I**. Equation (2) describes this in plainer terms for image size $M \times N$ and kernel size $m \times n$.

$$O(i, j) = \sum_{k=1}^{m} \sum_{\ell=1}^{n} I(i + k - 1, j + \ell - 1)K(k, \ell), \tag{2}$$

The convoluted image **O** will be of size $(M - m + 1) \times (N - n + 1)$. The kernel typically includes both x and y components; the convoluted images E_x and E_y obtained from the x and y components of the filter emphasize vertical and horizontal edges, respectively. The final edge image E is the square root of the sum of the squared component images, i.e.,

$$O(i, j) = \sum_{k=1}^{m} \sum_{\ell=1}^{n} I(i + k - 1, j + \ell - 1)K(k, \ell). \tag{3}$$

Common edge detecting filters in the spatial domain include Roberts, Prewitt, and Sobel. Equations (4)–(6) give the x and y kernels for the Roberts (R_x and R_y), Prewitt (P_x and P_y), and Sobel (S_x and S_y) filters. These filters compute the gradient between neighboring pixels in the x and y directions and intensify areas of high gradient (i.e., edges). Filters are constructed such that the components are of opposite sign and the sum of all components is zero. The Roberts filter (Equation (4)) is a compact kernel, which could lead to very fast processing times. The Prewitt (Equation (5)) and Sobel (Equation (6)) filters use larger 3×3 kernels and are therefore more powerful but require

extended computation times. The Prewitt is a first-order filter (the largest magnitude component is one); the second-order Sobel filter will likely produce an image with more intensified edges.

$$R_x = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, R_y = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}, \tag{4}$$

$$OP_x = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}, P_y = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}, \tag{5}$$

$$S_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, S_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}, \tag{6}$$

Another popular edge detection method in the spatial domain is the Laplacian of Gaussian (LoG) function. When applied to an image with intensities $I(x, y)$, the Laplacian operator $\nabla^2 = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2}$ emphasizes both edges and noise or artifact. The influence of noise can be reduced by first applying the Gaussian smoothing filter given by Equation (7), where x and y are the spatial coordinates within the Gaussian kernel and σ is the standard deviation

$$G(x, y) = \frac{1}{\sqrt{2\pi\sigma^2}} exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right). \tag{7}$$

Equation (8) gives the Laplacian of the Gaussian, which can be pre-allocated for a given filter size m × n and standard deviation σ.

$$LoG = \nabla^2(G(x, y)) = \frac{x^2 + y^2 - 2\sigma^2}{4\sigma^4} exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right). \tag{8}$$

Iterative optimization of the parameters m, n, and σ is possible on an image-by-image basis, but it is convenient to predefine both the size and standard deviation. For the purposes of this study, the LoG kernel was defined as a square matrix with size equal to 0.5% of the maximum image dimension, and the standard deviation was defined as one-fourth the maximum image dimension. At first glance, it would appear that the larger 13 × 13 LoG filter would be more computationally intensive than the smaller Roberts, Prewitt, and Sobel filters discussed previously. However, the LoG filter does not include x and y component kernels. Thus, only one convolution operation (Equation (2)) was required, and there was no need for the component transformation (Equation (3)).

2.3. Edge Detection in the Frequency Domain

Edge detection in the frequency domain requires transformation from the spatial domain to the frequency domain. This is quickly accomplished using the fast Fourier transform (FFT), which transforms the greyscale image intensities $I(x, y)$ into the frequency components $F(u, v)$. Unlike in the spatial domain, where the filter kernel is of arbitrary size, the filter kernel in the frequency domain is the same size as the image. The edge image $E(u, v)$ in the frequency domain is the element-by-element product of the filter kernel $K(u, v)$ and the frequency domain image $F(u, v)$, i.e.,

$$E(u, v) = K(u, v) \odot F(u, v), \tag{9}$$

where $\odot$ denotes element-wise multiplication. Inverse fast Fourier transformation (iFFT) of the frequency domain edge image $E(u, v)$ gives the edge image in the spatial domain $E(x, y)$.

The two most common frequency domain edge detection filters include Butterworth [45] and Gaussian [45–47] high pass filters. High pass filters attenuate frequencies above some defined cutoff frequency D_0. Equation (10) gives the general form of the nth-order Butterworth filter kernel $K_B(u, v)$,

where $D(u, v)$ is the distance between the pixel (u, v) and the origin of the frequency (the center of the $M \times N$ image), as defined by Equation (11).

$$K_B(u, v) = 1 - \frac{1}{1 + \left[\frac{D(u,v)}{D_0} \right]^{2n}} \quad , \tag{10}$$

$$D(u, v) = \sqrt{\left[u - \left(\frac{M}{2} + 1 \right) \right]^2 + \left[v - \left(\frac{N}{2} + 1 \right) \right]^2}. \tag{11}$$

Similarly, Equation (12) gives the general form of the Gaussian high pass filter kernel $K_G(u, v)$, where $D(u,v)$ is again the distance between the pixel (u, v) and the frequency origin, and σ is the assumed standard deviation of the frequency distribution.

$$K_G(u, v) = 1 - e^{\frac{-D^2(u,v)}{2\sigma^2}} \quad . \tag{12}$$

For the purposes of this study, a fourth order (n = 4)Butterworth filter was constructed with cutoff frequency $D_0 = M/10$ The Guassian filter was constructed with standard deviation $\sigma = M/10$ Figure 2 presents a graphical representation of the Butterworth and Gaussian filters.

(a)   (b)
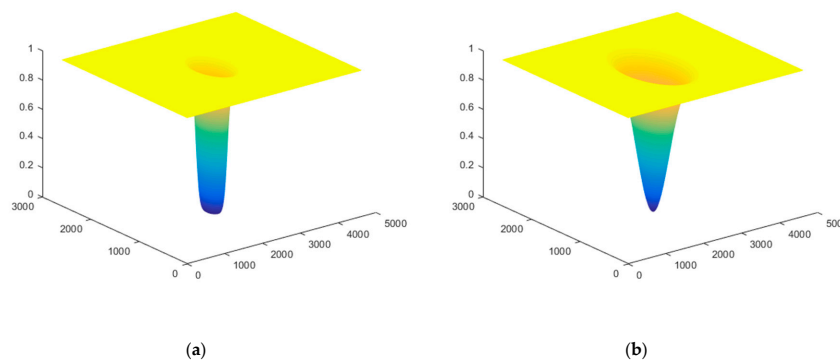
Figure 2. (**a**) Butterworth ($n = 4$, $D_0 = 259$) and (**b**) Gaussian ($\sigma = 259$) filters for edge detection in the frequency domain.

2.4. Edge Image Enhancement

Edge images $E(x, y)$ resulting from spatial or frequency domain edge detection filters contain a range of pixel intensities that require scaling. The scaling function given by Equation (13) converts edge image pixel intensities $E(x, y)$ to linearly scaled edge image pixel intensities $E_{sc}(x, y)$ such that $0 \leq I_{sc}(x, y) \leq 1$.

$$E_{sc}(x, y) = [E(x, y) - min(E)] \left[\frac{1}{max(E) - min(E)} \right]. \tag{13}$$

The scaled edge image $E_{sc}(x, y)$ requires contrast adjustment to improve edge clarity. Equation (14) transforms the scaled edge image $E_{sc}(x, y)$ into the enhanced edge image $E_e(x, y)$, where $\mu_{E_{sc}}$ and $\sigma_{E_{sc}}$ are the mean and standard deviation of the scaled edge image pixel intensities, respectively.

$$E_e(x, y) = [E_{sc}(x, y) - min(E_{sc})] \left[\frac{2\sigma_{E_{sc}}}{max(E_{sc}) - min(E_{sc})} \right] + \mu_{E_{sc}} \quad . \tag{14}$$

2.5. Segmentation

Segmentation was the final step in the proposed image analysis algorithm. This process converts the edge image to the binary image, in which pixels belonging to a crack take an intensity value of one, and the remaining pixels take an intensity value of zero. Selection of an appropriate threshold

intensity—above which a pixel is classified as a crack and below which it is not—is critical. If the threshold intensity is too high, cracks go undetected. If it is too low, the image becomes noisy, and it is difficult to differentiate cracks from noise. This work considered two threshold operations for segmentation: pixel threshold and area threshold.

The pixel threshold operation follows Equation (15), where $B_1(x, y)$ is the first-level binary image, and T_1 is the pixel threshold value.

$$B_1(x, y) = \begin{cases} 0, & E_e(x, y) < T_1 \\ 1, & E_e(x, y) \geq T_1 \end{cases}.$$ (15)

T_1 can be selected using Otsu's method [47] or other intuitive/adaptive approaches [31]. In this study, T_1 was selected based on the statistical properties of pixel intensities in the enhanced edge image $E_e(x, y)$. Equation (16) defines T_1, where μ_{E_e} and σ_{E_e} are the mean and standard deviation of the enhanced edge image pixel intensities.

$$T_1 = \mu_{E_e} + 3\sigma_{E_e}.$$ (16)

Similarly, the area threshold operation follows Equation (17), where $B_2(x, y)$ is the second-level binary image and T_2 is the area threshold value.

$$B_2(x, y) = \begin{cases} 0, & B_1(x, y) < T_2 \\ 1, & B_1(x, y) \geq T_2 \end{cases}.$$ (17)

Equation (18) defines T_2 according to the area of each connected component A_{cc}, where $\sigma_{A_{cc}}$ is the standard deviation of the areas of connected components in B_1.

$$T_2 = \sigma_{A_{cc}}.$$ (18)

The area of connected components A_{cc} is determined according to eight-neighbor connectivity, which considers pixel connectivity in the vertical, horizontal, or diagonal directions, such that pixel (x, y) is connected to all pixels $(x \pm 1, y \pm 1)$. A_{cc} could alternatively be defined according to four-neighbor connectivity, which is a stricter definition that only considers connectivity in the vertical and horizontal directions, such that pixel (x, y) is connected to pixels $(x \pm 1, y)$ and $(x, y \pm 1)$. For the purposes of this research, the more relaxed eight-neighbor definition of connectivity was adopted.

The second-level binary image B_2 is the final product of the proposed crack detection algorithm.

3. Experimental Program

In order to test the crack detection algorithm discussed above, the researchers gathered 50 images of sound concrete and 50 images of cracked concrete from several previously tested concrete panels at the Systems, Materials, and Structural Health Laboratory (SMASH Lab) at Utah State University. Images were taken with a 12 MP digital camera with focal length of 35 mm. The distance between the lens and the surface was approximately 0.3 m given the ability of the UAS to hold its position. The surface illumination, as verified by a Digi-Sense data logging light meter with NIST traceable calibration, was 1500–5000 lx. The image resolution was 2592 × 4608 px, and the approximate field size was 1.0 × 1.2 m. RGB images were saved in JPEG format. Image processing was performed in MATLAB on a 64-bit operating system with 32 GB memory and 3.40 GHz processor. Figure 3 shows representative images of defected and sound concrete. Images were processed in six iterations, corresponding to the four spatial domain edge detectors and two frequency domain edge detectors.

(**a**)　　　　　　　　　　　　　　　　　　　　　　　　　　　(**b**)

Figure 3. Representative images of (**a**) cracked concrete and (**b**) sound concrete.

Following image processing, an inspector reviewed the second-level binary images resulting from each of the six iterations in random order and classified each image as cracked or sound. The inspector reviewed only the second-level binary images; they did not review the original images or images from intermediate steps in the crack detection algorithm. The same inspector reviewed all of the images. The team then compared the results of each inspection to the ground truth, i.e., the known classification of each image as defected or sound based on physical inspection of the concrete surface aided by a crack microscope. The team then recorded the number of true positives (TPs), true negatives (TNs), false positives (FPs), and false negatives (FNs) for each iteration of the crack detection algorithm. A TP was a defected image in which the inspector accurately identified the defect. A TN was a sound image that the inspector accurately identified as sound. An FP was a sound image within which the inspector inaccurately identified a defect. An FN was a defected image that the inspector inaccurately identified as sound. A hit required the inspector to identify at least half of the actual crack length in a defected image. FP occurred when the inspector identified a crack in the noise or artifact of the second-level binary image. The performances of each approach of crack detection were evaluated in terms of accuracy, precision, processing time, and missed crack width (MCW). Accuracy, Ac., and precision, Pr., were calculated according to the following equations:

$$Ac. = \frac{TP + TN}{TP + FP + TN + FN} \; ; \tag{19}$$

$$Pr = \frac{TP}{TP + FP} \tag{20}$$

To obtain the processing time, each method of crack detection was run ten times on the same desktop and the same dataset, and the mean of the processing times was reported as each method's processing time. In order to find MCW, the missed cracks by each crack detection method were identified and then measured using a crack microscope with 0.02 mm resolution. The algorithms were also compared in terms of the pixel intensity range in the enhanced edge images and the noise-to-signal ratio (N/S). A wider range of pixel intensities suggested a sharper contrast between defects and sound regions. N/S described the level of noise or artifact in the image and was defined as the ratio of lit pixels (ones) to the total number of pixels in the second-level binary image B$_2$. N/S was only computed for the sound dataset because any lit pixels were known to be noise and not defects. A lower N/S was greatly preferred because defects became more difficult to resolve when the image was noisy.

4. Results

Table 1 summarizes the results of the six iterations of the proposed crack detection algorithm. The TP rate (TPR), TN rate (TNR), FP rate (FPR), and FN rate (FNR) in the table are the percentages of TP, TN, FP, and FN reports. Note that the reported metrics in this paper were inclusive to the defined

parameters in experimental and analytical procedures, and the authors were not suggesting that one could get similar results in practice with less-controlled situations.

Table 1. Performance of different edge detectors in the proposed crack detection algorithm.

Domain	Edge Detector	TPR[1] (%)	TNR[2] (%)	FPR[3] (%)	FNR[4] (%)	Ac.[5] (%)	Pr.[6] (%)	MCW[7] (mm)	Time (s)
Spatial	Roberts	64	90	10	36	77	86	0.4	1.67
Spatial	Prewitt	82	82	18	18	82	82	0.2	1.4
Spatial	Sobel	86	84	16	14	85	84	0.2	1.4
Spatial	Laplacian of Gaussian (LoG)	98	86	14	2	92	88	0.1	1.18
Frequency	Butterworth	80	86	14	20	83	85	0.2	1.81
Frequency	Gaussian	80	88	12	20	84	87	0.2	1.92

[1] Ture Positive Rate, [2] Ture Negative Rate, [3] False Positive Rate, [4] False Negative Rate, [5] Accuracy, [6] Precision, and [7] Missed Crack Width.

In order to gain perspective about the numbers in this table, the accuracy of visual inspections should be determined. Washer et al. investigated the quality of element-level bridge inspection data. A required accuracy was arbitrary in visual inspections because the achieved accuracy was tied to the inspection process itself and was unaffected by the requirement [48]. Different inspectors responded very differently when inspecting the same structure or defect. The coefficients of variation of a rated bridge deck by different inspectors in visual inspections were between 57% and 96%. Detection of concrete deck cracks can be considered a "typical" decision in bridge inspections, and considering a normal distribution for all the calls, it is acceptable to have 67.5% of correct calls, i.e., ± one standard deviation, [23,26,48]. This meant each of the edge detectors, except for Roberts, had surpassed the acceptable accuracy of the visual inspections. A bridge inspector reviewed the images in this study and was able to label all of the images correctly into sound and defected images (100% accuracy). Unlike the edge detectors, the inspector did not localize the pixels associated with cracks in the images, since it was very time-consuming, to provide a pixel-segmented ground truth [28]. This gave the edge detectors an advantage over human inspectors.

4.1. Spatial Domain, Roberts Filter

Crack detection in the spatial domain using the Roberts filter resulted in the lowest number of TPs (32), but it also resulted in the lowest FP (5). Thus, while the Roberts filter was the least accurate (77%), its precision (86%) was among the highest. The minimum detectable crack width was 0.4 mm and was the largest of the six edge detectors evaluated. The processing time (1.67 s per image) was near the median of the six methods evaluated. Figure 4 shows representative enhanced edge, first-level binary, and second-level binary images from spatial domain edge detection of an image from the defected set (Figure 3a) using the Roberts filter.

| (**a**) | (**b**) | (**c**) |
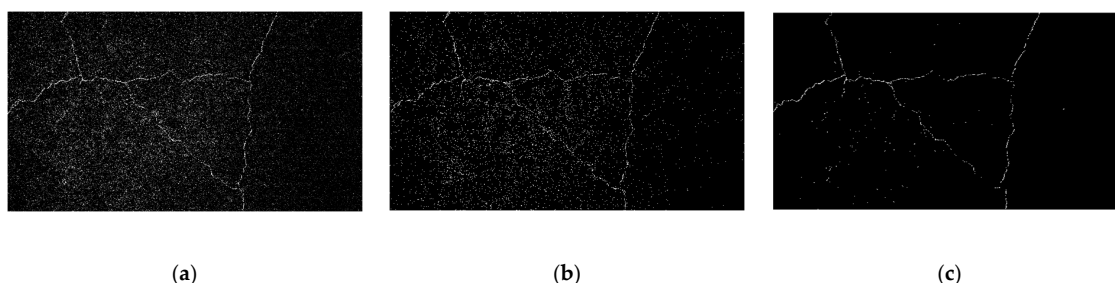
Figure 4. (**a**) Enhanced edge, (**b**) first-level binary, and (**c**) second-level binary images; defected dataset, spatial domain, and Roberts filter.

4.2. Spatial Domain, Prewitt Filter

Crack detection in the spatial domain using the Prewitt filter resulted in the second lowest number of TP (41) and the highest FP (9). The Prewitt filter was the second least accurate and the least precise of the six methods evaluated. The minimum detectable crack width was 0.2 mm, which was comparable to four of the six methods. The processing time (1.40 s per image) was among the shortest. Figure 5 shows representative enhanced edge, first-level binary, and second-level binary images from spatial domain edge detection of an image from the defected set using the Prewitt filter.

　　　　　　(**a**)　　　　　　　　　　　　　　　　(**b**)　　　　　　　　　　　　　　　　(**c**)
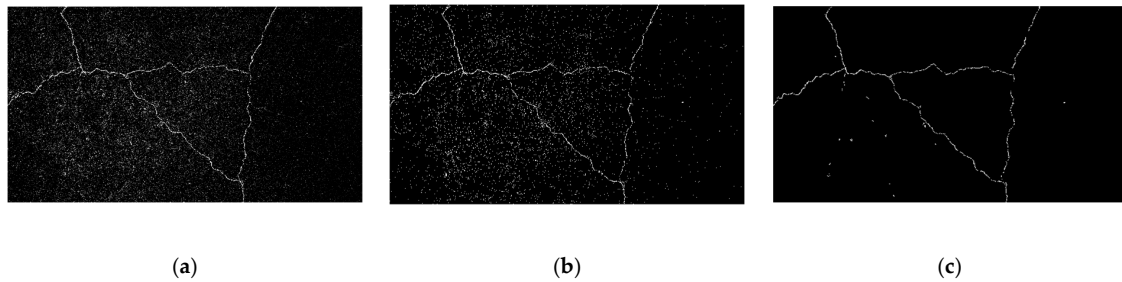
Figure 5. (**a**) Enhanced edge, (**b**) first-level binary, and (**c**) second-level binary images; defected dataset, spatial domain, and Prewitt filter.

4.3. Spatial Domain, Sobel Filter

Crack detection in the spatial domain using the Sobel filter resulted in the second highest number of TPs (43) and the second highest FP (8). Thus, while the Sobel filter was among the most accurate (85%), it was also among the least precise (84%). The minimum detectable crack width was 0.2 mm, and the processing time (1.4 s per image) was among the shortest. Figure 6 shows representative enhanced edge, first-level binary, and second-level binary images from spatial domain edge detection of an image from the defected set using the Sobel filter.

　　　　　　(**a**)　　　　　　　　　　　　　　　　(**b**)　　　　　　　　　　　　　　　　(**c**)
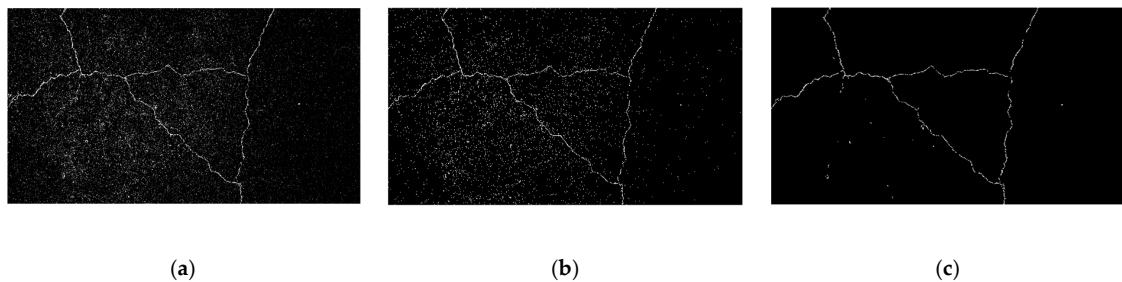
Figure 6. (**a**) Enhanced edge, (**b**) first-level binary, and (**c**) second-level binary images; defected dataset, spatial domain, and Sobel filter.

4.4. Spatial Domain, Laplacian of Gaussian (LoG) Filter

Crack detection in the spatial domain using the LoG filter resulted in the highest number of TPs (49), with only one miss in 50 defected images. The FP (7) was near the median for the six methods evaluated. Nevertheless, the LoG filter was the most accurate (92%) and the most precise (88%). Furthermore, the LoG method had the narrowest minimum detectable crack width (0.1 mm) and the shortest processing time (1.18 s per image). Figure 7 shows representative enhanced edge, first-level binary, and second-level binary images from spatial domain edge detection of an image from the defected set using the LoG filter.

(**a**)   (**b**)   (**c**)
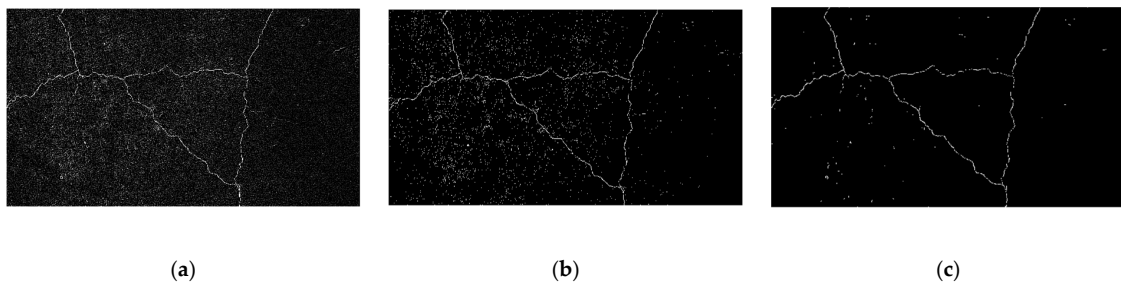
Figure 7. (**a**) Enhanced edge, (**b**) first-level binary, and (**c**) second-level binary images; defected dataset, spatial domain, and LoG filter.

## 4.5. Frequency Domain, Butterworth Filter

Crack detection in the frequency domain using the Butterworth filter resulted in the median number of TPs (40) and the median FP (7). The accuracy (83%) and precision (85%) were also near the median of the six methods evaluated. The minimum detectable crack width was again 0.2 mm, and the processing time (1.81 s per image) was the second longest of the six methods. Figure 8 shows representative enhanced edge, first-level binary, and second-level binary images from frequency domain edge detection of an image from the defected set using the Butterworth filter.

(**a**)   (**b**)   (**c**)
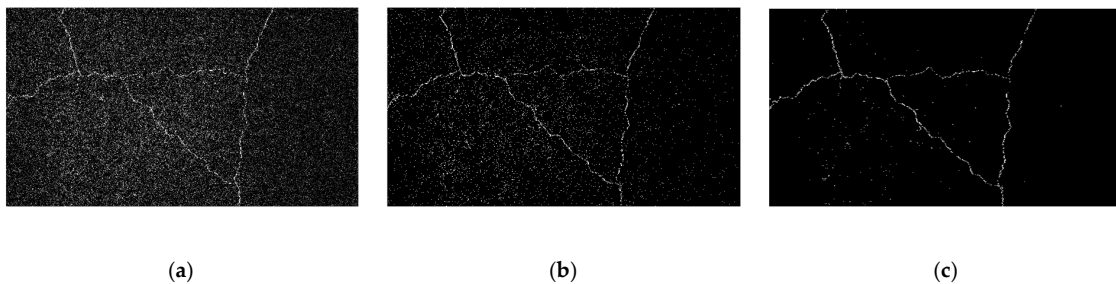
Figure 8. (**a**) Enhanced edge, (**b**) first-level binary, and (**c**) second-level binary images; defected dataset, frequency domain, and Butterworth filter.

4.6. Frequency Domain, Gaussian Filter

Crack detection in the frequency domain using the Gaussian filter resulted in the median number of TPs (40) and the second lowest FP (12%). The accuracy (84%) was also near the median value, but the precision (87%) was the second highest. The minimum detectable crack width was again 0.2 mm. The processing time (1.92 s per image) was the longest of the six methods evaluated. Figure 9 shows representative enhanced edge, first-level binary, and second-level binary images from frequency domain edge detection of an image from the defected set using the Gaussian filter.

(**a**)   (**b**)   (**c**)
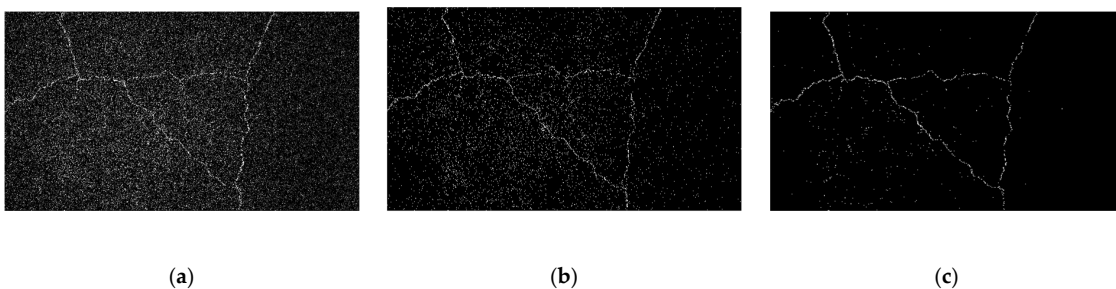
Figure 9. (**a**) Enhanced edge, (**b**) first-level binary, and (**c**) second-level binary images; defected dataset, spatial domain, and Gaussian filter.

4.7. Comparison

Table 2 presents a comparison of the range of pixel intensities in the enhanced edge image E_e, the pixel thresholds T_1 and T_2 used for construction of the first- and second-level binary images B_1 and B_2, and the noise-to-signal ratio N/S observed in sound dataset using the six edge detection methods. Figure 10 presents a direct comparison of superimposed second-level binary images to the original image from analysis of the image in Figure 3a, a member of the defected dataset. Similarly, Figure 11 shows a direct comparison of the second-level binary images from analysis of the image in Figure 3b, a member of the sound dataset (only LoG and Gaussian filter results were illustrated for brevity).

Table 2. The average range and threshold value for each method in defected and sound datasets.

Edge Detector	Defected Dataset				Sound Dataset				Average N/S (%)
	J_1	J_2	T_1	T_2	J_1	J_2	T_1	T_2	
Roberts	0.204	0.251	0.70	25	0.21	0.25	0.64	21	0.41
Prewitt	0.232	0.290	0.66	76	0.23	0.29	0.59	53	0.32
Sobel	0.230	0.291	0.67	75	0.23	0.29	0.59	53	0.33
LoG	0.534	0.590	0.71	58	0.62	0.69	0.63	32	0.90
Butterworth	0.581	0.631	0.89	10	0.57	0.64	0.93	6	1.74
Gaussian	0.594	0.640	0.89	8	0.58	0.64	0.93	5	1.76

(**a**)   (**b**)   (**c**)
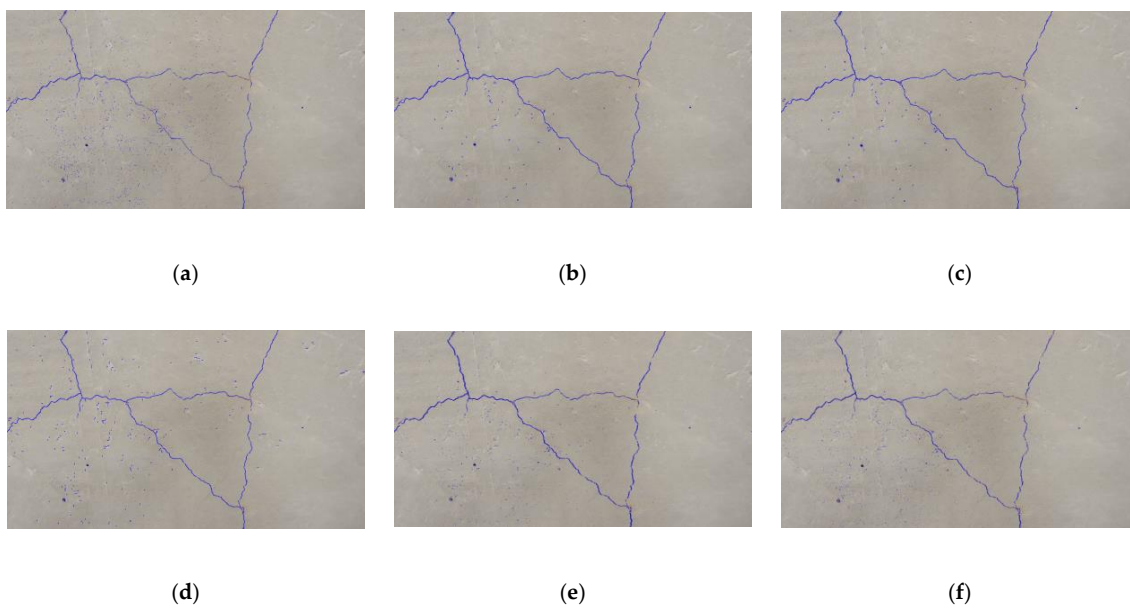

(**d**)   (**e**)   (**f**)

Figure 10. Superimposed second-level binary images on the original image using (**a**) Roberts, (**b**) Prewitt, (**c**) Sobel, (**d**) LoG, (**e**) Butterworth, and (**f**) Gaussian filters.
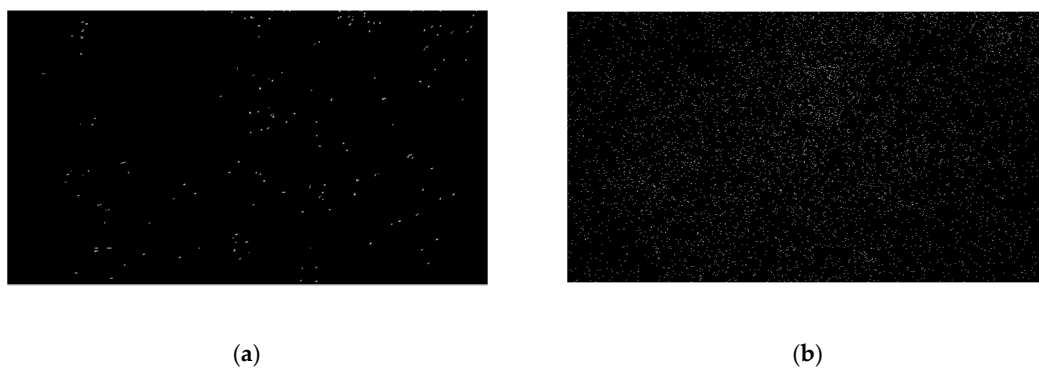

(**a**)   (**b**)

Figure 11. Second-level binary images from sound dataset obtained by crack detection using (**a**) LoG and (**b**) Gaussian filters.

Edge detection in the spatial domain using the LoG filter was the fastest of the six crack detection methods evaluated. Even though the difference between computational times was not very significant for one image (0.74 s between LoG and Gaussian), and considering it could take more than 1000 images to cover all areas of an infrastructure, using LoG would be roughly 10 min faster over all images than Gaussian, which was definitely significant. For example, current inspection procedures may only take 10 min of total overall inspection time for a routine inspection [9,49]. The effect of this time difference on an automated or semi-automated inspection is yet unknown, as these are not yet possible and could have very different processes; however, it could result in less on-site time or changes to the inspection procedure.

Frequency domain methods were expected to be the fastest because the element-wise product (Equation (9)) required far fewer floating-point operations than the iterative convolution operation (Equation (2)). However, the computational intensity of the Fourier and inverse Fourier transformations used to move between the spatial and frequency domains greatly increased processing time. The frequency domain methods took an average of 1.87 s per image, while the spatial domain methods took an average of 1.41 s per image. The LoG filter was expected to be computationally efficient compared to the other methods despite its comparatively large size (13×13). The computational efficiency of this method resulted from the fact that LoG used only one kernel, as opposed to x and y component kernels in the other spatial domain methods. This reduced the number of convolution operations (Equation (2)) from two to one, which obviated the use of Equation (3). Computational efficiencies of the other spatial domain methods did not follow the expected trend. It was expected that processing time would increase with kernel size, and that the 3×3 Prewitt and Sobel filters would require longer computational times than the 2×2 Roberts filter. In fact, the opposite was true. Processing time for the Roberts filter was 20% longer than for the Prewitt or Sobel. The reader will recall the output image from Equation (2) is of dimension $(M - m + 1) \times (N - n + 1)$ for image size $M \times N$ and kernel size $m \times n$. Thus, a smaller kernel produced a larger edge image. This explained, at least in part, the increased computational time for the smaller Roberts filter. The LoG filter was also both the most accurate and the most precise of the six methods tested. The LoG method resulted in 98% TPs with only one miss among the fifty images in the defected dataset. The next most accurate method recorded seven misses. The remaining methods all recorded ten or more misses. Thus, the accuracy of LoG (92%) was significantly higher than the other five methods (77%–85%). The precision of LoG (88%), which also considered FP, was much closer to that of the other five methods (82%–87%). The LoG method recorded seven false positives in the 50 images in the sound dataset. The Roberts filter, with 18 misses, was by far the least accurate (77%). However, with only five false positives, Roberts was among the most precise (86%). Prewitt was the least precise with nine misses, nine false positives, and 82% precision.

The LoG filter resolved the finest cracks with an MCW of 0.1 mm. Most of the other methods were only able to resolve cracks 0.2 mm or wider. The Roberts filter could only detect cracks 0.4 mm or wider. Considering the image size used in this study, one pixel was equivalent to 0.2 mm. Thus, the LoG filter was useful in detecting cracks that were about one pixel wide, while Roberts could only resolve cracks that were two pixels wide.

The contrast adjustment ranges, segmentation thresholds T_1 and T_2, and noise-to-signal ratios N/S listed in Table 1 gave some context to the performance metrics discussed above. The contrast adjustment ranges, J_1 and J_2, represented the range of pixel intensities in the enhanced edge image E_e. A wider range of contrast values $(J_2–J_1)$ corresponded to more intensified edges within the image. Thus, cracks should be more easily detected when the contrast adjustment range was large. The Roberts filter, which performed poorly according to the performance metrics discussed above, exhibited the smallest range. The LoG filter, which arguably exhibited the best performance, had one of the widest contrast adjustment ranges. Furthermore, the contrast adjustment range for the LoG filter was quite different between the defected and sound datasets. This resulted from a large number of pixels with high intensities in the defected image.

The noise-to-signal ratio N/S was evaluated only for the sound dataset for the simple reason that the noise in sound images was more well-defined. In the perfectly ideal case, no pixels should be lit in the second-level binary image from the sound dataset. Thus, any lit pixels were noise by default. In the defected dataset, distinction between signal and noise was ill-defined. In general, spatial domain methods exhibited lower N/S than frequency domain methods. The lowest N/S ratios were observed for the Prewitt and Sobel filters, with N/S ratios of 0.32 and 0.33, respectively. The Roberts filter exhibited only slightly more noise (N/S = 0.41). In comparison, the LoG filter produced a fairly noisy edge image (N/S = 0.90).

Increased noise in the frequency domain manifested as an increase in the standard deviation σ_{E_e} of the pixel intensities of the enhanced edge image E_e. Following Equation (18), this caused an increase in the pixel threshold T_1. While pixel thresholds were higher in the frequency domain, the area thresholds were lower. This resulted from reduced continuity of cracks in the frequency domain.

It was expected that the LoG method, which was the most successful in terms of accuracy, precision, MCW, and processing time, would also exhibit the least noise. Instead, the noise-to-signal ratio in the LoG images was among the highest observed. This can be explained in part by the images shown in Figures 10 and 11. The presence of cracks in even the noisiest of images in Figure 10e and f was clear. Similarly, even in the sound images with the highest N/S (Figure 11e,f), it was easy to see that no cracks were present. Despite the large number of lit pixels, no pattern of connectivity was apparent, thus, the inspector could reasonably conclude that he or she was observing noise and not a defect. These images represented only a single data point for each method from defected and sound datasets. However, they suggested that the level of noise in the binary image was not the only factor affecting the inspector's ability to detect cracks. Continuity of cracks in the binary image was also important, especially considering that the inspector needed to identify at least half of the crack in order to register a hit.

The value of area threshold T_2 gave some idea of the continuity of cracks in the defected images. T_2 was defined in Equation (18) as the standard deviation $\sigma_{A_{cc}}$ of the areas of connected components A_{cc}. When the continuity of cracks in the binary image was poor (i.e., the cracks were discontinuous), $\sigma_{A_{cc}}$ was small. Conversely, when the cracks in the binary image were highly continuous, $\sigma_{A_{cc}}$ increased. Thus, higher values of T_2 implied a higher degree of continuity of cracks in the binary image. Additionally, when the cracks were highly continuous in binary images from the defected dataset, the value of T_2 would be much higher for the defected dataset than for the sound dataset. Such was the case for the Prewitt, Sobel, and LoG filters. The same was also true, but to a lesser degree, for the Butterworth and Gaussian filters. The values of T_2 for the sound and defected datasets using the Roberts filter were similar. This suggested poor continuity of cracks in the binary images, which was confirmed in Figure 10a. Considering that the Roberts filter was among the worst methods tested here, this result was not at all surprising. The cracks in the rest of the binary images from the defected dataset (Figure 10b–f) were visibly more continuous.

The results presented here have some significant implications for future work in the realm of automated detection without human inspectors. For all of the evaluated methods, the pixel segmentation threshold T_1 was higher for the defected dataset than for the sound dataset. The same was true for the area segmentation threshold T_2. For the LoG method, the contrast adjustment ranges were also much different for the defected dataset than for the sound dataset. Future research can consider these differences as indicators of the likelihood that a particular processed image includes a defect.

5. Conclusions

The literature contains few investigations comparing different edge detection algorithms for accuracy, none of which are on UAS captured images. This study investigated a generic image-processing algorithm designed to objectively compare different edge detection algorithms on detection of defects in concrete. The algorithm involved edge detection, edge image enhancement,

and segmentation. Edge detection was completed in the spatial domain using Roberts, Prewitt, Sobel, and LoG filters, and in the frequency domain using Butterworth and Gaussian filters. Fifty images of defected concrete and 50 of sound concrete were analyzed by the proposed algorithm in six iterations (making use of the six aforementioned edge detection strategies). An inspector reviewed the resulting binary images from each iteration to determine if the detector found a crack. The inspection results were compared to the ground truth, and the six edge detection methods were compared based on accuracy, precision, minimum detectable crack width, and processing time per image. Edge detection in the spatial domain using the LoG filter yielded the highest accuracy (92%) and precision (88%), the finest minimum detectable crack width, and the fastest processing time (1.18 s per image). All but one of the remaining methods (edge detection in the spatial domain using the Roberts filter) yielded greater than 80% accuracy, and they were able to detect cracks as fine as 0.2 mm. While crack detection in the spatial domain using the Roberts filter yielded the lowest accuracy (77%), it also yielded the fewest false positives (10%), and its precision (86%) was among the highest. In general, the processing time was longer for crack detection in the frequency domain (1.8–1.9 s per image) than in the spatial domain (1.2–1.7 s per image). Additionally, the second-level binary images (the final product of the image processing algorithm) were much noisier in the frequency domain. According to these results, crack detection in the spatial domain using the LoG filter yields the best and fastest results for detecting defects in concrete structures. Therefore, the proposed crack detection algorithm using LoG filter is recommended for real-time crack detection of concrete structures using UASs.

Author Contributions: Conceptualization, S.D.; methodology, S.D., and M.M.; software, S.D.; validation, S.D., R.T., and M.M.; formal analysis, S.D.; investigation, S.D.; resources, M.M.; data curation, S.D.; writing—original draft preparation, S.D.; writing—review and editing, S.D., R.T., and M.M.; visualization, S.D., R.T., and M.M.; supervision, M.M.; project administration, M.M.; funding acquisition, M.M.

References

1. FHWA (Federal Highway Administration). Tables of Frequently Requested NBI Information. Available online: https://www.fhwa.dot.gov/bridge/britab.cfm (accessed on 16 January 2018).
2. Dorafshan, S.; Maguire, M.; Hoffer, N.V.; Coopmans, C. Challenges in bridge inspection using small unmanned aerial systems: Results and lessons learned. In Proceedings of the 2017 International Conference on Unmanned Aircraft Systems (ICUAS), Miami, FL, USA, 13–16 June 2017; pp. 1722–1730.
3. Dorafshan, S.; Maguire, M. Bridge inspection: Human performance, unmanned aerial systems and automation. *J. Civ. Struct. Health Monit.* **2018**, *8*, 443–476. [CrossRef]
4. Gucunski, N.; Boone, S.D.; Zobel, R.; Ghasemi, H.; Parvardeh, H.; Kee, S.H. Nondestructive evaluation inspection of the Arlington Memorial Bridge using a robotic assisted bridge inspection tool (RABIT). In *Nondestructive Characterization for Composite Materials, Aerospace Engineering, Civil Infrastructure, and Homeland Security 2014*; International Society for Optics and Photonics: Bellingham, WA, USA, 2014; Volume 9063, p. 90630N.
5. Lattanzi, D.; Miller, G. Review of robotic infrastructure inspection systems. *J. Infrastruct. Syst.* **2017**, *23*, 04017004. [CrossRef]
6. Le, T.; Gibb, S.; Pham, N.; La, H.M.; Falk, L.; Berendsen, T. Autonomous robotic system using non-destructive evaluation methods for bridge deck inspection. In Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 29 May–3 June 2017; pp. 3672–3677.
7. Dorafshan, S.; Thomas, R.J.; Maguire, M. Fatigue crack detection using unmanned aerial systems in fracture critical inspection of steel bridges. *J. Bridge Eng.* **2018**, *23*, 04018078. [CrossRef]
8. Khaloo, A.; Lattanzi, D.; Cunningham, K.; Dell'Andrea, R.; Riley, M. Unmanned aerial vehicle inspection of the Placer River Trail Bridge through image-based 3D modelling. *Struct. Infrastruct. Eng.* **2018**, *14*, 124–136. [CrossRef]

9.   Dorafshan, S.; Maguire, M.; Hoffer, N.; Coopmans, C. *Fatigue Crack Detection Using Unmanned Aerial Systems in Under-Bridge Inspection*; RP 256; Final Report; Idaho Department of Transportation: Boise, ID, USA, 2017. Available online: http://apps.itd.idaho.gov/apps/research/Completed/RP256.pdf (accessed on 30 January 2019).

10.   Omar, T.; Nehdi, M.L. Remote sensing of concrete bridge decks using unmanned aerial vehicle infrared thermography. *Autom. Constr.* **2017**, *83*, 360–371. [CrossRef]

11.   Omar, T.; Nehdi, M.L.; Zayed, T. Infrared thermography model for automated detection of delamination in RC bridge decks. *Constr. Build. Mater.* **2018**, *168*, 313–327. [CrossRef]

12.   Sony, S.; Laventure, S.; Sadhu, A. A literature review of next-generation smart sensing technology in structural health monitoring. *Struct. Control. Health Monit.* **2019**, *26*, e2321. [CrossRef]

13.   Hiasa, S.; Karaaslan, E. Infrared and High-definition Image-based Bridge Scanning Using UAVs without Traffic Control. In Proceedings of the 27th ASNT Research Symposium, Orlando, FI, USA, 26 March 2018; pp. 116–127.

14.   Farhidzadeh, A.; Ebrahimkhanlou, A.; Salamone, S. A vision-based technique for damage assessment of reinforced concrete structures. In *Health Monitoring of Structural and Biological Systems 2014*; International Society for Optics and Photonics: Bellingham, WA, USA, 2014; Volume 9064, p. 90642H.

15.   Dorafshan, S.; Maguire, M.; Qi, X. *Automatic Surface Crack Detection in Concrete Structures Using OTSU Thresholding and Morphological Operations*; Paper 1234; Civil and Environmental Engineering Faculty Publications: Logan, UT, USA, 2016. Available online: https://digitalcommons.usu.edu/cee_facpub/1234 (accessed on 30 January 2019).

16.   Liu, X.; Ai, Y.; Scherer, S. Robust image-based crack detection in concrete structure using multi-scale enhancement and visual features. In Proceedings of the 2017 IEEE International Conference on Image Processing (ICIP), Beijing, China, 17–20 September 2017; pp. 2304–2308.

17.   Gopalakrishnan, K.; Gholami, H.; Vidyadharan, A.; Choudhary, A.; Agrawal, A. Crack Damage Detection in Unmanned Aerial Vehicle Images of Civil Infrastructure Using Pre-Trained Deep Learning Model. *Int. J. Traffic Transp. Eng.* **2018**, *8*, 1–14.

18.   Yang, L.; Li, B.; Li, W.; Liu, Z.; Yang, G.; Xiao, J. A robotic system towards concrete structure spalling and crack database. In Proceedings of the 2017 IEEE International Conference on Robotics and Biomimetics (ROBIO), Macau, China, 5–8 December 2017; pp. 1276–1281.

19.   Vaghefi, K.; Ahlborn, T.M.; Harris, D.K.; Brooks, C.N. Combined imaging technologies for concrete bridge deck condition assessment. *J. Perform. Constr. Facil.* **2013**, *29*, 04014102. [CrossRef]

20.   Ellenberg, A.; Kontsos, A.; Moon, F.; Bartoli, I. Bridge related damage quantification using unmanned aerial vehicle imagery. *Struct. Control Health Monit.* **2016**, *23*, 1168–1179. [CrossRef]

21.   Dorafshan, S.; Thomas, R.J.; Maguire, M. SDNET2018: An annotated image dataset for non-contact concrete crack detection using deep convolutional neural networks. *Data Brief* **2018**, *21*, 1664–1668. [CrossRef] [PubMed]

22.   Ebrahimkhanlou, A.; Farhidzadeh, A.; Salamone, S. Multifractal analysis of crack patterns in reinforced concrete shear walls. *Struct. Health Monit.* **2016**, *15*, 81–92. [CrossRef]

23.   Graybeal, B.A.; Phares, B.M.; Rolander, D.D.; Moore, M.; Washer, G. Visual inspection of highway bridges. *J. Nondestruct. Eval.* **2002**, *21*, 67–83. [CrossRef]

24.   Phares, B.M.; Washer, G.A.; Rolander, D.D.; Graybeal, B.A.; Moore, M. Routine highway bridge inspection condition documentation accuracy and reliability. *J. Bridge Eng.* **2004**, *9*, 403–413. [CrossRef]

25.   Agdas, D.; Rice, J.A.; Martinez, J.R.; Lasa, I.R. Comparison of visual inspection and structural-health monitoring as bridge condition assessment methods. *J. Perform. Constr. Facil.* **2015**, *30*, 04015049. [CrossRef]

26.   Frangopol, D.M.; Liu, M. Maintenance and management of civil infrastructure based on condition, safety, optimization, and life-cycle cost∗. *Struct. Infrastruct. Eng.* **2007**, *3*, 29–41. [CrossRef]

27.   Dorafshan, S.; Thomas, R.J.; Coopmans, C.; Maguire, M. Deep learning neural networks for sUAS-assisted structural inspections: Feasibility and application. In Proceedings of the 2018 International Conference on Unmanned Aircraft Systems (ICUAS), Dallas, TX, USA, 12–15 June 2018; pp. 874–882.

28.   Dorafshan, S.; Thomas, R.J.; Maguire, M. Comparison of deep convolutional neural networks and edge detectors for image-based crack detection in concrete. *Constr. Build. Mater.* **2018**, *186*, 1031–1045. [CrossRef]

29.   Abdel-Qader, I.; Abudayyeh, O.; Kelly, M.E. Analysis of edge-detection techniques for crack identification in bridges. *J. Comput. Civ. Eng.* **2003**, *17*, 255–263. [CrossRef]

30. Kim, H.; Ahn, E.; Cho, S.; Shin, M.; Sim, S. Comparative analysis of image binarization methods for crack identification in concrete structures. *Cem. Concr. Res.* **2017**, *99*, 53–61. [CrossRef]

31. Dorafshan, S.; Maguire, M. Autonomous detection of concrete cracks on bridge decks and fatigue cracks on steel members. In Proceedings of the Digital Imaging 2017, Mashantucket, CT, USA, 26–28 June 2017; pp. 33–44.

32. Hoang, N.-D. Detection of surface crack in building structures using image processing technique with an improved Otsu method for image thresholding. *Adv. Civ. Eng.* **2018**, *195*, 3924120. [CrossRef]

33. Luo, Q.; Ge, B.; Tian, Q. A fast adaptive crack detection algorithm based on a double-edge extraction operator of FSM. *Constr. Build. Mater.* **2019**, *204*, 244–254. [CrossRef]

34. Hoang, N.-D.; Nguyen, Q.-L. Metaheuristic Optimized Edge Detection for Recognition of Concrete Wall Cracks: A Comparative Study on the Performances of Roberts, Prewitt, Canny, and Sobel Algorithms. *Adv. Civ. Eng.* **2018**, *2018*, 7163580. [CrossRef]

35. Yamaguchi, T.; Nakamura, S.; Saegusa, R.; Hashimoto, S. Image-based crack detection for real concrete surfaces. *IEEJ. Trans. Electr. Electron. Eng.* **2008**, *3*, 128–135. [CrossRef]

36. Yang, C.; Geng, M. The crack detection algorithm of pavement image based on edge information. *AIP Conf. Proc.* **2018**, *1967*, 040023.

37. Li, Y.; Liu, Y. High-accuracy crack detection for concrete bridge based on sub-pixel. In Proceedings of the 2017 IEEE International Conference on Real-time Computing and Robotics (RCAR), Okinawa, Japan, 14–18 July 2017; pp. 234–239.

38. Oh, J.; Jang, G.; Oh, S.; Lee, J.H.; Yi, By.; Moon, Y.S.; Lee, J.S.; Choi, Y. Bridge inspection robot system with machine vision. *Autom. Constr.* **2009**, *18*, 929–941. [CrossRef]

39. Ali, R.; Gopal, D.L.; Cha, Y. Vision-based concrete crack detection technique using cascade features. In *Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems 2018*; International Society for Optics and Photonics: Bellingham, WA, USA, 2018; Volume 10598, p. 105980L.

40. Van Nguyen, L.; Gibb, S.; Pham, H.X.; La, H.M. A Mobile Robot for Automated Civil Infrastructure Inspection and Evaluation. In Proceedings of the 2018 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR), Philadelphia, PA, USA, 6–8 August 2018; pp. 1–6.

41. Rimkus, A.; Podviezko, A.; Gribniak, V. Processing digital images for crack localization in reinforced concrete members. *Procedia Eng.* **2015**, *122*, 239–243. [CrossRef]

42. Talab, A.M.A.; Huang, Z.; Xi, F.; HaiMing, L. Detection crack in image using Otsu method and multiple filtering in image processing techniques. *Opt. Int. J. Light Electron Opt.* **2016**, *127*, 1030–1033. [CrossRef]

43. Mohan, A.; Poobal, S. Crack detection using image processing: A critical review and analysis. *Alex. Eng. J.* **2018**, *57*, 787–798. [CrossRef]

44. Gonzalez, R.C.; Wintz, P. Digital image processing. In *Applied Mathematics and Computation 1997*; Addison-Wesley Publishing Co., Inc.: Reading, MA, USA, 1977; Volume 13, p. 451.

45. Butterworth, S. On the theory of filter amplifiers. *Wirel. Eng.* **1930**, *7*, 536–541.

46. Blinchikoff, H.; Krause, H. *Filtering in the Time and Frequency Domains*; The Institution of Engineering and Technology: Stevenage, UK, 2001.

47. Otsu, N. A threshold selection method from gray-level histograms. *IEEE Trans. Syst. Manand Cybern.* **1979**, *9*, 62–66. [CrossRef]

48. Washer, G.; Hammed, M.; Brown, H.; Connor, R.; Jensen, P.; Fogg, J.; Salazar, J.; Leshko, B.; Koonce, J.; Karper, C. *Guidelines to Improve the Quality of Element-Level Bridge Inspection Data*; No. NCHRP Project 12-104; NCHRP: Washington, DC, USA, 2018.

49. Dorafshan, S.; Maguire, M.; Hoffer, N.V.; Coopmans, C.; Thomas, R.J. *Unmanned Aerial Vehicle Augmented Bridge Inspection Feasibility Study*; No. CAIT-UTC-NC31; Rutgers University: Piscataway, NJ, USA, 2017.