Noise Resilient Concrete and Masonry Crack Detection Using Self-Organizing Maps

Xinxin Sun* (ORCID: 0000-0002-9963-2109) and Peter Chang

Department of Civil and Environmental Engineering, University of Maryland, College Park, MD 20742, USA

*Corresponding Author

Email: xinxin68@umd.edu

Abstract

This paper presents a lightweight, training-free framework for automated crack detection in concrete and masonry structures using Self-Organizing Maps (SOMs). By clustering pixel-level features—including grayscale, edge gradients, contrast, hue, and thin-crack indicators—the method performs unsupervised segmentation without labeled data, manual thresholds, or deep models. It demonstrates strong robustness to noise, shadows, and textured surfaces across field-acquired imagery. Feature ablation experiments reveal material-dependent contributions of each input, offering insights into SOM's internal structure and interpretability. In addition, SOM-derived pseudo-labels are used to train CNNs, achieving high segmentation accuracy and generalization across unseen domains. Designed for UAV and mobile SHM platforms, the framework operates with low memory (<100 MB), fast inference (<30 s/image), and minimal human input—offering an interpretable, scalable alternative to heuristic filters and data-intensive networks for real-world infrastructure diagnostics.

***Keywords**:* Self-organizing maps (SOMs); Unsupervised learning; Concrete crack detection; Masonry crack detection; Image-based diagnostics; Structural health monitoring (SHM); Pseudo-labeling; Noise-robust segmentation

1. Introduction

Structural damage in civil infrastructure refers to alterations in material composition, geometry, or system connectivity that degrade performance and reliability [1], [2]. Left unaddressed, such damage can lead to costly repairs, operational disruptions, or safety risks. Early detection has thus become a priority across civil, mechanical, and aerospace engineering [3], [4]. Five related fields—structural health monitoring (SHM), condition monitoring, non-destructive evaluation (NDE), statistical process control, and damage prognosis—collaborate to address this challenge [5], [6], [7], [8], [9], [10]. SHM, in particular, assesses the structural health of materials and systems over time [11], [12], [13], [14], [15], playing a key role in sustainable infrastructure management [16], [17].

Recent SHM advancements utilize large-scale imagery from drones or surveillance systems to remotely monitor surfaces like bridges, tunnels, and façades. This approach enhances efficiency and reduces maintenance costs in modern construction, but manually analyzing vast datasets is impractical [18], necessitating automated, image-based crack detection tools.

Image-based NDE is widely applied to detect surface cracks in concrete and masonry [19], [20], [21]. Since visible cracks may not always signal immediate distress, tracking their evolution over time is critical for maintenance planning. Time-series imagery can reveal crack widening or branching, yet requires efficient, automated analysis tools.

Traditional methods, such as morphological techniques, Canny, and Sobel operators, are widely used but suffer from threshold sensitivity, parameter dependency, and poor adaptability to noise or lighting variations [22], [23], [24]. For instance, the Canny operator achieves high edge localization but remains sensitive to noise and parameter choice [25], while Sobel often yields fragmented results in noisy conditions [26], [27]. Morphological methods reduce noise but demand careful tuning [28], [29], [30], [31], [32], [33], [34], [35], and advanced techniques like Alternating Sequential Filters remain computationally heavy [36], [37]. Despite recent improvements such as adaptive thresholding or multi-scale filtering, these methods remain constrained in field conditions, limiting their use in real-world SHM.

To overcome these challenges, we propose a Self-Organizing Map (SOM)-based algorithm that clusters pixels by visual properties—like intensity, contrast, and edge magnitude—without needing labeled data or extensive tuning. Introduced by Kohonen [38], SOMs map high-dimensional data onto a low-dimensional grid for unsupervised clustering. They achieve this by projecting complex features into an organized lattice, enabling efficient pattern recognition without prior training. In image analysis, they have been applied to tasks such as pattern grouping and texture segmentation, offering a balance of simplicity and adaptability. Unlike prior SOM-based crack detection approaches that operate on image-level or patch-level descriptors and rely on pre-training with labeled data [39], [40], our method applies SOMs directly to per-pixel features—such as grayscale, contrast, and hue—captured from local image neighborhoods. This fully local, training-free formulation enables the algorithm to adapt dynamically to each image's visual structure, regardless of surface material, lighting conditions, or crack morphology. Unlike SOMs used as predictive models in earlier work, we employ them here strictly as interpretable clustering tools, eliminating the need for architectural tuning, patch-level aggregation, or task-specific parameterization. This shift ensures robustness across diverse field conditions and enhances compatibility with lightweight, real-time SHM applications.

Despite the growing popularity of deep learning (DL), especially Convolutional Neural Networks (CNNs), in crack detection tasks due to their powerful feature extraction capabilities [41], [42], such models face practical constraints in real-world SHM deployments. CNNs often require extensive hyperparameter tuning and data augmentation, limiting their adaptability to uncontrolled field conditions. DL techniques typically require large, annotated datasets for training [43], demand significant computational resources[44], and often lack generalizability across varying field conditions. Moreover, CNN-based segmentation methods may miss fine cracks or misclassify noise under lighting and textural distortions [45], which undermines their reliability in uncontrolled environments. In contrast, the proposed SOM framework clusters pixels based on local image properties without requiring training data or task-specific retraining, offering a lightweight, interpretable, and adaptive alternative for field-deployable diagnostics.
This approach excels in longitudinal monitoring, robustly handling shifts in lighting or texture while tracking crack propagation and bifurcation in time-series imagery. It also offers scalability, potentially integrating indicators like moisture streaks for broader assessments.

In summary, this work delivers a robust, unsupervised solution for crack detection in concrete and masonry. By minimizing human intervention and excelling under diverse field conditions, it advances image-based SHM. The proposed approach lays a lightweight, training-free foundation for autonomous systems that enhance infrastructure safety and resilience. Ultimately, it contributes to smarter inspection workflows and more sustainable infrastructure management.

2. Methodology

To achieve robust, unsupervised crack detection at the pixel level—without reliance on manual thresholds or extensive data labeling—we implement a Self-Organizing Map (SOM) approach. SOMs are commonly used to visualize and explore high-dimensional data by reducing complexity through unsupervised learning [46]. The SOM algorithm typically consists of two stages: a competitive stage and a cooperative stage [47], [48]. In the competitive stage, the node closest to the input vector is selected as the Best-Matching Unit (BMU) [49]. A schematic diagram of the SOM algorithm (Fig. 1) illustrates the competition and update process.

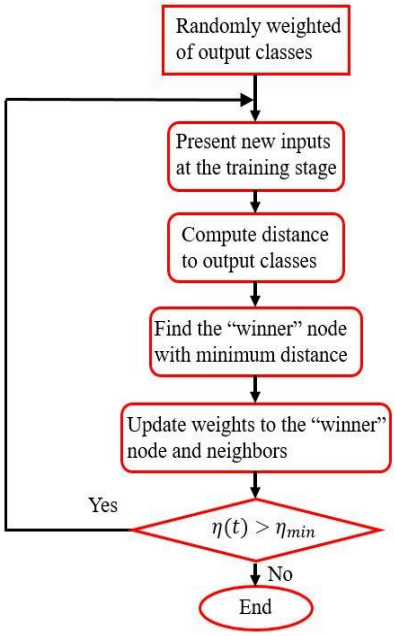


Fig. 1: SOM algorithm

Consider an n-dimensional input vector $p = \{p_1, p_2, \dots, p_n\}$, representing n properties. Each node in the SOM is associated with an n-dimensional weight vector $w_i = \{w_{i1}, w_{i2}, \dots, w_{in}\}$. When a new input vector is presented, the distance d_i between the input and the node i is computed as:

$$d_i = \sqrt{\sum_{j=1}^{n} \left(p_j(t) - w_{ij}(t)\right)^2} \tag{1}$$

Where $p_j(t)$ the value of property j at time t, and $w_{ij}(t)$ is the weight of property j for node i at time t [50]. The node with the smallest d_i is the BMU, denoted by superscript c [51].

After identifying the BMU, the weights of the BMU and its neighbors are updated:

$$w_{ij}(t+1) = w_{ij}(t) + \eta(t) h_{ij}^{c}(t) \left(p_j(t) - w_{ij}(t)\right) \tag{2}$$

Here, $\eta(t)$ is the learning rate, and $h_{ij}^{c}(t)$ is the neighborhood kernel, both non-increasing functions to ensure convergence.

A key feature of our approach is the direct application of SOMs at the pixel level for crack detection, a novel concept within SHM. Unlike traditional methods like Canny edge detection, which rely on predefined thresholds, or other SOM-based techniques requiring extensive pre-training, our method eliminates these needs. Unlike conventional SOM applications in SHM that depend on structure-specific tuning, our approach dynamically adapts to each image, enhancing its versatility. This flexibility allows SOMs to generalize across surface types and environmental conditions, unlike SHM methods constrained by training datasets or structural assumptions. No manual annotations or labeled data are used at any stage of the process, reinforcing the fully unsupervised nature of the framework.

We use a standard linear learning rate:

$$\eta(t) = \eta(0)\left(1 - \frac{t}{T}\right) \tag{3}$$

Where $\eta(0)$ is the initial learning rate, t is the current training iteration, and T is the total training length.

The neighborhood function is defined as [52]:

$$h_{ij}(t) = \begin{cases} 1 & if\ i \in N_c(d) \\ 0 & otherwise \end{cases} \tag{4}$$

Where $N_c(d)$ is the set of nodes within a distance d of the BMU c [53].

$$N_c(d) = \{j \mid d_{cj} \leq d\} \tag{5}$$

The neighborhood radius is initially set based on the map size and dynamically adjusted via a decay schedule, enabling the SOM to transition from global organization to fine local refinement. By avoiding manual thresholds and complex training, our approach efficiently detects cracks across diverse materials.

Input properties for crack detection include grayscale intensity for crack darkness, edge characteristics for boundary detection, local contrast for crack-background separation, and hue for color-varying surfaces like bricks. These properties, combined as a multi-dimensional vector, allow the SOM to cluster pixels based on their intrinsic properties.

After SOM training, each pixel is assigned to one of the output clusters (e.g., 3 for concrete, 5 for brick). To identify the class corresponding to "crack," we used a minimal visual GUI tool for class labeling—selecting the crack class with a single click based on its morphology. This process is distinct from manual annotation: it does not involve tracing boundaries or drawing masks, and typically takes <3 seconds per image. Class selection was based on empirical stability across different images and held fixed for all evaluations (e.g., CNN supervision, ablation analysis). This one-time semantic mapping allows unsupervised clustering to be used as a consistent pseudo-labeling mechanism across the study.

Fig. 2 illustrates how these properties effectively highlight cracks on (a) plain concrete and (b) brick surfaces, demonstrating the method's adaptability.

(a) Cracks on plain concrete (b) Cracks on brick surfaces

Fig. 2: Visual Examples of crack detection properties

Grayscale

Cracks typically appear darker than the undamaged areas in images, as shown in Fig. 2. In the proposed approach, the grayscale intensity of pixels serves as a key feature for clustering. In simple images with uniform backgrounds, such as Fig. 2(a), grayscale alone is sufficient to separate crack regions from non-crack areas. However, in more complex images with varying lighting or textured backgrounds, such as Fig. 2(b), additional features are required to accurately identify cracks. For instance, in such cases, grayscale alone may not suffice to distinguish cracks from non-crack dark regions.

The grayscale value of each pixel corresponds to its light intensity, with darker pixels having lower values [54], [55]. For a given pixel $p(x, y)$, the grayscale intensity $I(x, y)$ is computed using the standard RGB-to-grayscale conversion:

$$I(x, y) = 0.2989 \cdot R(x, y) + 0.5870 \cdot G(x, y) + 0.1140 \cdot B(x, y) \quad (6)$$

This formula emphasizes the green channel's contribution to perceived brightness, which enhances crack visibility. Here, $R(x, y)$, $G(x, y)$, and $B(x, y)$ represent the red, green, and blue channel values of the pixel at position (x, y). Since cracks generally appear darker than the surrounding material, grayscale intensity is a valuable feature for SOM-based crack detection. As shown in Fig. 3, SOM clusters pixels with similar grayscale intensities, effectively grouping darker crack pixels together while separating them from brighter background regions.

While grayscale is a fundamental feature, its integration with additional properties (e.g., edge and contrast) within the SOM framework ensures robust detection across diverse scenarios, as elaborated in subsequent sections.

(a) Cracks on plain concrete (b) Cracks on brick surfaces

Fig. 3: Crack detection using SOM with grayscale intensity as input

Average grayscale of the pixel and its 8 neighbors

Cracks often display subtle variations in grayscale intensity compared to their surrounding material. To improve crack detection accuracy, we introduce a feature based on the average grayscale value of a pixel and its eight neighbors, forming a 3×3 matrix. This 9-pixel average acts as a smoothing filter, reducing noise while emphasizing consistent intensity patterns typical of crack regions.

Unlike edge detection, which also employs a 3×3 matrix but focuses on abrupt intensity changes to highlight structural boundaries, this method targets localized intensity uniformity. Edge detection relies on geometric calculations—such as maximum absolute grayscale differences along diagonals and folds—to identify discontinuities. In contrast, the 9-pixel average assists the SOM in clustering pixels with consistently low intensity, a hallmark of cracks. This distinction enhances the SOM's ability to differentiate crack regions from non-crack areas based on broad, uniform intensity profiles rather than sharp transitions.

As illustrated in Fig. 4, SOM achieves clear and accurate crack classification using only this feature, effectively isolating cracks on both plain concrete and brick surfaces. This underscores the feature's robustness in capturing the distinct intensity characteristics of crack regions.

(a) Cracks on plain concrete (b) Cracks on brick surface

Fig. 4: Crack detection using SOM with only the 9-pixel average grayscale property as input

Average grayscale of the pixel and its 24 neighbors

To refine crack detection, the neighborhood expands from 8 to 24 neighbors (a 5×5 matrix). The 25-pixel average smooths local intensity variations, reducing noise for a cleaner representation. Fig. 5 shows that this method clarifies larger cracks and reduces minor noise, though finer cracks may appear less distinct. However, this larger neighborhood can dilute sharp transitions, potentially obscuring subtle cracks that the 9-pixel method captures more effectively (Fig. 4). The 25-pixel method excels at detecting larger cracks and suppressing noise but may miss early-stage cracks, unlike the more sensitive 9-pixel method. Larger neighborhoods such as 7×7 were also tested, but these produced excessive smoothing, which obscured fine cracks and degraded segmentation precision.

(a) Cracks on plain concrete (b) Cracks on brick surface

Fig. 5: Crack detection using SOM with only the 25-pixel average grayscale property as input

Relative grayscale values

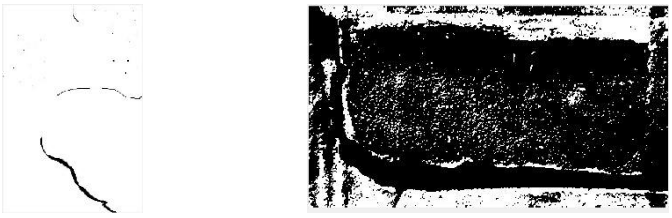
Cracks typically appear darker than their surroundings in grayscale images, creating distinct intensity variations [56]. Plotting pixel positions along a row or column against their grayscale values reveals a valley at crack locations, highlighting their lower intensity [57]. Relative grayscale values enhance the detection of subtle cracks that absolute values might miss [58].

In this approach, each pixel's grayscale value is compared to the median grayscale of its row and column. The relative grayscale properties ρ_{ij} and γ_{ij} are computed as:

$$\rho_{ij} = \frac{p_{ij}}{\mu(p_i)} \times 100 \tag{7}$$

$$\gamma_{ij} = \frac{p_{ij}}{\mu(p_j)} \times 100 \tag{8}$$

Where p_{ij} is the pixel's grayscale value, $\mu(p_i)$ is the median grayscale of pixels in Row i, and $\mu(p_j)$ is the median grayscale of pixels in Column j. As shown in Fig. 6, SOM using only relative grayscale values detects small, thin, and discontinuous cracks effectively, even with some noise, making it ideal for early-stage or fine crack identification.



(a) Cracks on plain concrete (b) Cracks on brick surface

Fig. 6: SOM-based crack detection using only relative grayscale values

Edge detection

Cracks often exhibit distinct edge characteristics, making edge-based features highly effective for identification. Edge detection captures abrupt grayscale changes to separate cracks from backgrounds. Following Sinha and Fieguth [59],

we use a 3x3 neighborhood matrix to compute the maximum grayscale difference across predefined lines and folds (Fig. 7). For a pixel $p(i,j)$, one such absolute difference is:

$$A_{ij}^{n} = \left|\left(p_{i,j-1} + p_{i+1,j-1} + p_{i+1,j}\right) - \left(p_{i,j+1} + p_{i-1,j} + p_{i-1,j+1}\right)\right| \tag{9}$$

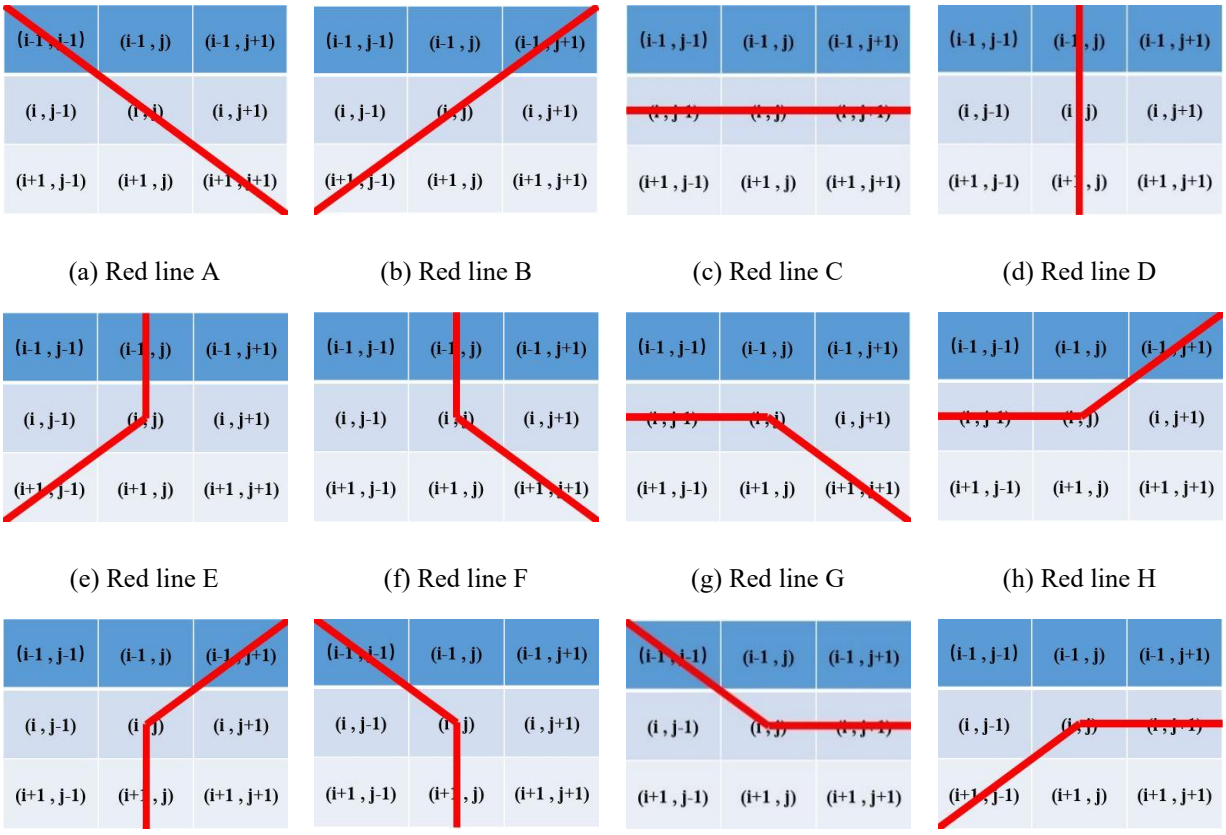
Similarly, for line E:

$$E_{ij}^{n} = \left|2 \times \left(p_{i-1,j-1} + p_{i,j-1}\right) - \left(p_{i-1,j+1} + \mathrm{p}_{i,j+1} + p_{i+1,j+1} + p_{i+1,j}\right)\right| \tag{10}$$

The edge property ϵ_{ij}^{n} is then:

$$\epsilon_{ij}^{n} = \max\left\{A_{ij}^{n}, B_{ij}^{n}, C_{ij}^{n}, D_{ij}^{n}, E_{ij}^{n}, F_{ij}^{n}, G_{ij}^{n}, H_{ij}^{n}, I_{ij}^{n}, J_{ij}^{n}, K_{ij}^{n}, L_{ij}^{n}\right\} \tag{11}$$

Where ϵ_{ij}^{n} represents the normal edge detection property at pixel (i,j).



(a) Red line A (b) Red line B (c) Red line C (d) Red line D

(e) Red line E (f) Red line F (g) Red line G (h) Red line H

(i) Red line I (j) Red line J (k) Red line K (l) Red line L

Fig. 7: Illustration of the edge detection process using a 3x3 matrix.

As shown in Fig. 8, SOM accurately identifies cracks on plain concrete and brick surfaces using only edge features, proving its robustness. Tests with a 5x5 matrix showed no significant improvement, confirming the 3x3 matrix's balance of accuracy and efficiency. Unlike deep learning-based edge detectors that require large datasets, our method leverages simple yet effective edge features, ensuring adaptability and low computational cost.

(a) Cracks on plain concrete (b) Cracks on brick surface

Fig. 8: Crack detection using SOM with edge detection as input

Contrast

Cracks typically exhibit higher contrast than their surroundings, making contrast a key feature for distinguishing crack and non-crack regions. This property enhances segmentation and reduces the impact of noise, improving detection accuracy in complex environments [60].

For a pixel at (i, j), contrast C is computed as:

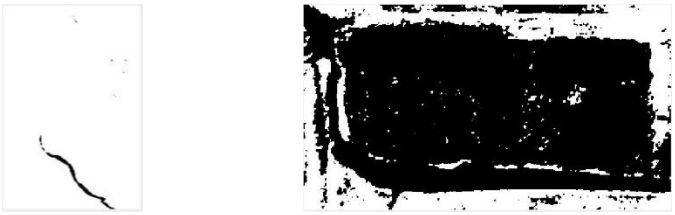
$$C = \max\left(\left|S_{ij} - A_1\right|, \left|S_{ij} - A_2\right|, \ldots, \left|S_{ij} - A_{16}\right|\right) \tag{12}$$

Where S_{ij} is the sum of grayscale values in a 3×3 neighborhood centered at (i, j), defined as:

$$S_{ij} = p_{i-1,j-1} + p_{i-1,j} + p_{i-1,j+1} + p_{i,j-1} + p_{i,j} + p_{i,j+1} + p_{i+1,j-1} + p_{i+1,j} + p_{i+1,j+1} \tag{13}$$

And A_k represents the grayscale values of the 16 pixels surrounding this 3×3 region, located at coordinates ($i-2, j-2$), ($i-2, j-1$), ($i-2, j$), ($i-2, j+1$), ($i-2, j+2$), ($i-1, j-2$), ($i-1, j+2$), (i, j–2), ($i, j+2$), ($i+1, j-2$), ($i+1, j+2$), ($i+2, j-2$), ($i+2, j-1$), ($i+2, j$), ($i+2, j+1$), ($i+2, j+2$).

As illustrated in Fig. 9, the Self-Organizing Map (SOM) effectively detects cracks using only contrast as input, even on complex brick surfaces. While contrast is a straightforward feature, its integration with other properties (e.g., edge and hue) within the SOM framework ensures robust and accurate crack detection across diverse scenarios.



(a) Cracks on plain concrete (b) Cracks on brick surface

Fig. 9: Crack detection using SOM with contrast as input

Thin crack detection

Detecting thin cracks, especially those a single pixel wide, is challenging because they often blend into background noise, evading capture by grayscale or edge detection methods alone. These cracks exhibit subtle intensity variations

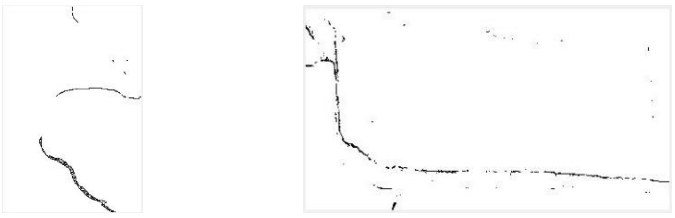
that are difficult to distinguish using conventional approaches. To address this, a specialized algorithm is introduced that enhances thin crack visibility by leveraging localized intensity differences along fine linear structures. For a pixel at (i, j), the algorithm computes a value A_{ij}^t for line A as the ratio of the average grayscale intensity of pixels along the line to that of adjacent pixels, expressed as:

$$A_{ij}^t = \left(\frac{(p_{i-1,j-1} + p_{i,j} + p_{i+1,j+1})/3}{\frac{p_{i,j-1} + p_{i+1,j-1} + p_{i+1,j} + p_{i-1,j} + p_{i-1,j+1} + p_{i.j+1}}{6}} \right) \times 100 \tag{14}$$

This process is repeated for 11 other predefined lines (B through L), and the thin crack property ϵ_{ij}^t is defined as:

$$\epsilon_{ij}^t = \max\{A_{ij}^t, B_{ij}^t, C_{ij}^t, D_{ij}^t, E_{ij}^t, F_{ij}^t, G_{ij}^t, H_{ij}^t, I_{ij}^t, J_{ij}^t, K_{ij}^t, L_{ij}^t\} \tag{15}$$

Here, ϵ_{ij}^t isolates the most prominent linear feature at (i, j), effectively highlighting thin cracks. As shown in Fig. 10, this property successfully detects thin cracks on both plain concrete and brick surfaces, though it is not designed to identify wider or irregularly shaped cracks. By amplifying subtle intensity differences along linear paths, this method enhances the detection of narrow cracks often missed by traditional techniques.



(a) Cracks on plain concrete (b) Cracks on brick surface

Fig. 10: Crack detection using SOM with only the thin crack detection property as input

Hue

While darker regions often indicate cracks, surfaces like bricks or shadows can appear similar, complicating detection [61]. Hue, representing the dominant wavelength of light [62], [63], addresses this challenge. Using the RGB model [40], hue distinguishes cracks from other dark regions based on its higher values. It is calculated as:

$$\text{Hue} = \begin{cases} \dfrac{60(G-B)}{\max(R,G,B) - \min(R,G,B)} \; if\ R\ is\ max \\ \left[120 + \dfrac{60(B-R)}{\max(R,G,B) - \min(R,G,B)}\right] \; if\ G\ is\ max \\ \left[240 + \dfrac{60(R-G)}{\max(R,G,B) - \min(R,G,B)}\right] if\ B\ is\ max \end{cases} \tag{16}$$

Fig. 11 shows SOM-based crack detection using only hue on (a) concrete and (b) brick surfaces, demonstrating its effectiveness. While each feature contributes uniquely, SOM clustering of these properties ensures robust detection in complex scenarios.

(a) Cracks on plain concrete (b) Cracks on brick surface

Fig. 11: Crack detection using SOM with only the hue property as input

In conclusion, while each property—thin crack detection, average grayscale values (9- or 25-pixel), relative grayscale, and hue—contributes valuable information, robust and reliable crack detection in complex structural images is achieved by combining these properties through SOM-based clustering. Each property serves a specific role: thin crack detection highlights narrow crack regions, average grayscale values smooth out noise, relative grayscale captures contrast differences between cracks and their surroundings, and hue variations distinguish cracks from color-based confounding features such as shadows or patterned backgrounds. By integrating these features into SOM-based clustering, the algorithm synthesizes local visual cues into coherent spatial patterns, enabling precise differentiation between crack and non-crack regions—even under complex structural and environmental variations.

3. Experiments

3.1 Example of concrete crack detection

All visualizations in this section were generated directly from full-resolution inputs and exported as lossless PNG files from MATLAB. No image compression or downscaling was applied during manuscript preparation. This ensures visual fidelity for all crack segmentation comparisons. To ensure a controlled and fair benchmarking environment with traditional methods such as Canny and Sobel—known to struggle with noisy or branched features—this study selects images with relatively linear or thin cracks as a baseline. Subsequent studies will examine SOM's behavior on more complex and branched geometries. In this study, a SOM is utilized to classify cracks on concrete surfaces, as illustrated in Fig. 12. To estimate the crack area ratio within the concrete image, three grid resolutions—5×5 pixels, 10×10 pixels, and 15×15 pixels— are overlaid onto the fracture image, as shown in Fig. 13. The crack area ratio r_g is calculated using the following formula:

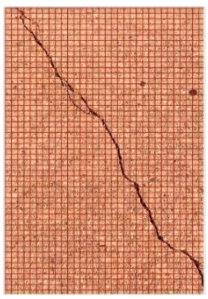
$$r_g = \frac{n_g}{N} \times 100\% \tag{17}$$

Where:

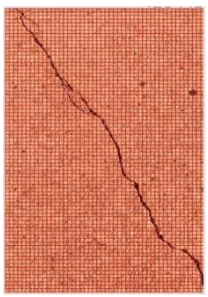
- n_g: Number of grid points intersecting the crack.
- N: Total number of grid points in the image.

Fig. 12: Concrete Crack

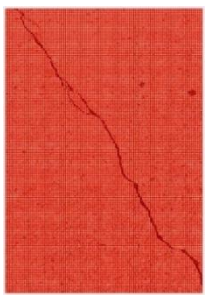
Fig. 13 demonstrates the concrete crack overlaid with these varying grid sizes:



(a) Concrete crack with a 15×15-pixel mask



(b) Concrete crack with a 10×10-pixel mask



(c) Concrete crack with a 5×5-pixel mask

Fig. 13: Concrete crack image overlaid with grid masks of varying resolutions

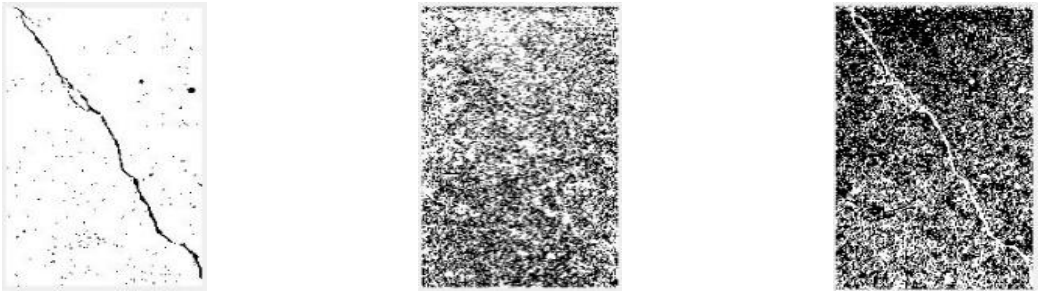
The impact of grid resolution on the accuracy of the crack ratio estimation is summarized in Table 1:

Table 1: Estimation of crack ratios for varying grid dimensions

	15×15-pixel	10×10-pixel	5×5-pixel
n_g	36	60	245
N	1581	3588	14758
r_g	2.28%	1.67%	1.66%

As the grid resolution increases (i.e., smaller pixel sizes), the estimated crack ratio converges toward the true value. The 5×5-pixel grid yields an r_g of 1.66%, with an approximate error of 0.01% compared to finer resolutions or manual estimation. This negligible error suggests that the 5×5 grid provides sufficient granularity for accurate crack area approximation while maintaining computational efficiency. Tests across multiple images with diverse lighting and crack patterns further validate this resolution's robustness.

For crack detection, the SOM leverages a feature set including grayscale, contrast, edge detection, thin crack detection, average grayscale, relative grayscale values, and hue. These properties enable the SOM to cluster pixels into three distinct categories: crack, noise, and concrete, as depicted in Fig. 14:



(a) crack class (b) noise class (c) concrete class

Fig. 14: SOM classification results for concrete crack image

Since concrete fracture images predominantly consist of these three pixel types, this classification ensures precise segmentation. However, initial SOM results may contain small noisy regions. To refine the output, a post-processing step eliminates regions smaller than 1% of the largest detected crack area. This empirically chosen threshold balances noise suppression with the preservation of early-stage or narrow cracks and has proven effective across multiple test conditions. It aligns with conventional SHM practices that prioritize the retention of fine damage patterns for long-

term tracking. Future work will include sensitivity analysis to further validate the robustness of this filtering criterion across diverse surface textures and environmental conditions.

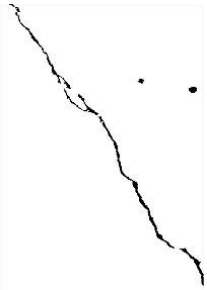
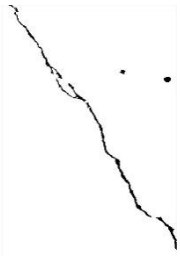
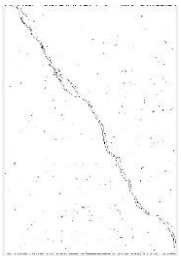
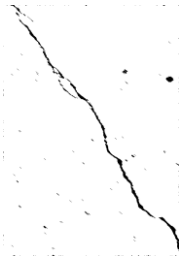


Fig. 15: Crack detection after noise reduction

To assess the SOM-based method's performance, it is benchmarked against traditional image processing techniques: Canny edge detection, Sobel edge detection, and morphological operations. Table 2 compares these methods under their optimal settings. With an error of 5% (crack area ratio: 1.58%), SOM outperforms Canny (8%, 1.80%) and morphological methods (23%, 1.28%), and closely matches Sobel (3%, 1.71%). Notably, SOM achieves this with minimal manual tuning, enhancing its adaptability compared to threshold-dependent methods like Canny and Sobel, which require careful parameter adjustments.

Table 2: Comparison of SOM, Canny, Sobel, and Morphological methods

Methods	SOM	Canny	Sobel	Morphological
Applied Parameter	Noise reduction: <1% of max crack area	Gradient threshold: 0.5	Gradient threshold: 0.25	Binary threshold: 0.5
Detected crack with thresholds				
After manual adjustments	Not applicable			
Crack area	1.58%	1.80%	1.71%	1.28%
Error	5%	8%	3%	23%

Note:

- The true crack area ratio, estimated via Eq. (17) with the 5×5 grid, is 1.66%.
- Relative error is calculated as:

$$Error = \frac{|Estimated\ Ratio - True\ Ratio|}{True\ Ratio} \times 100\% \quad (18)$$

For SOM:

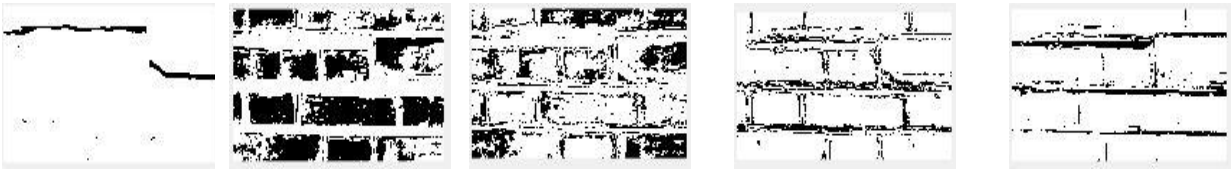
$$Error = \frac{|1.58\% - 1.66\%|}{1.66\%} \times 100\% \approx 5\%$$

3.2 Crack detection on brick surface

Following the methodology from the concrete crack analysis, an image of a brick crack (Fig. 16) is processed to estimate the crack area ratio. A 10×10-pixel grid achieves consistent crack area ratios, ensuring precise fracture mapping with computational efficiency. The final crack ratio is approximately 3.00%, validating this resolution across tests.

Fig. 16: Crack between bricks

For the brick fracture image, the SOM leverages grayscale, contrast, edge detection, thin crack detection, average grayscale, relative grayscale, and hue. The SOM clusters pixels into five interpretable classes: crack, brick, concrete residue, noise, and mortar (Fig. 17). Since SOM does not inherently label classes, the crack category is identified manually via visual inspection. Future research will explore automating this step.



(a) crack class (b) brick class (c) noise class (d) concrete residue class (e) motar class

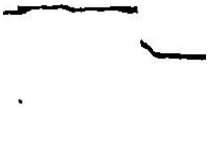
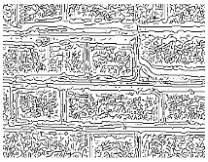
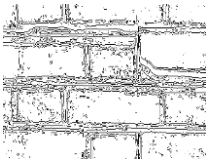
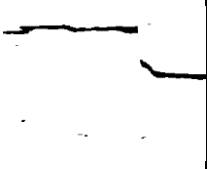
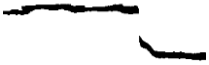
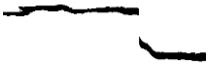
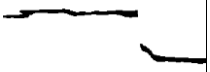
Fig.17: Classification of the brick crack

Post-processing removes regions smaller than 1% of the largest crack to eliminate noise, retaining only significant cracks (Fig. 18). This threshold, previously validated on concrete imagery, was found effective across multiple materials. This minimal intervention preserves SOM's segmentation strengths.

Fig. 18: Crack detection after noise reduction

The SOM's performance is benchmarked against Canny, Sobel, and morphological methods (Table 3). With an error of 2% (crack area ratio: 2.93%), SOM outperforms Canny (15%, 3.46%) and Sobel (12%, 3.37%), and closely matches the morphological method (3%, 2.91%). Notably, SOM achieves this with minimal manual tuning, enhancing its adaptability.

Table 3: Comparison of SOM, Canny, Sobel, and Morphological methods

Methods	SOM	Canny	Sobel	Morphological
Applied Parameter	Noise reduction: <1% of max crack area	Gradient threshold: 0.01	Gradient threshold: 0.05	Binary threshold: 0.25
Detected crack with thresholds				
After manual adjustments	Not applicable			
Crack area	2.93%	3.46%	3.37%	2.91%
Error	2%	15%	12%	3%

Notes:

- True crack ratio: 3.00% (grid-based estimation).

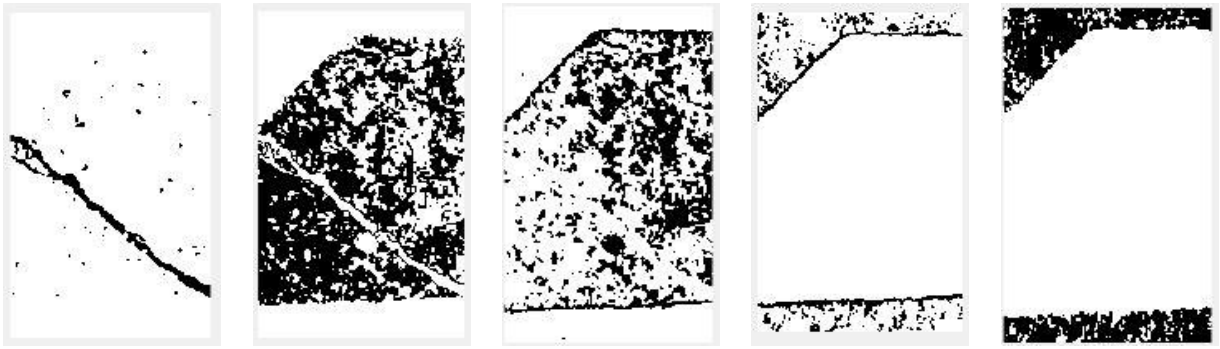
- $Error = \frac{|Estimated\ Ratio - True\ Ratio|}{True\ Ratio} \times 100\%$

3.3 Shadow crack detection

Using the grid-based estimation method from the concrete and brick crack analyses, the shadowed crack (Fig. 19) occupies approximately 3.50% of the image. A 10×10-pixel grid ensures precise fracture mapping with computational efficiency.

Fig. 19: Shadow crack

For classification, the Self-Organizing Map (SOM) uses grayscale, contrast, edge detection, thin crack detection, average grayscale, relative grayscale, and hue to categorize pixels into five classes: crack, shadowed concrete, noise, light edges, and lit concrete (Fig. 20). Due to SOM's unsupervised nature, the crack class is identified post hoc via visual inspection.



(a) crack class (b) shadowed concrete class (c) shadow noise class (d) light concrete noise class (e) lit concrete class

Fig. 20: Classification of the shadow crack

Post-processing removes regions smaller than 1% of the largest crack to eliminate noise, retaining only significant cracks (Fig. 21). This threshold was consistently applied across all examples to ensure reproducibility and preserve small but meaningful crack patterns.

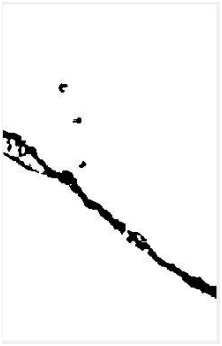
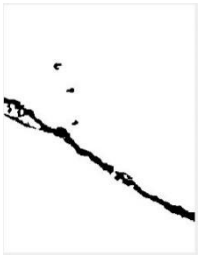
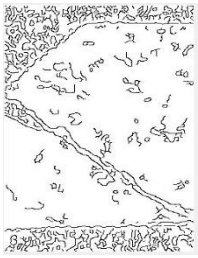
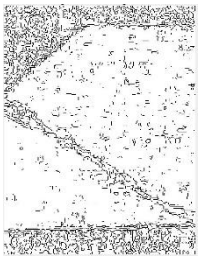


Fig. 21: Crack detection after noise deduction

The SOM's performance is compared with Canny, Sobel, and morphological methods (Table 4). With an error of 4% (crack area ratio: 3.36%), SOM closely matches the true ratio (3.50%), outperforming Canny (6%, 3.29%) and matching morphological methods (4%, 3.37%).

Table 4: Comparison of SOM, Canny, Sobel, and Morphological methods

Methods	SOM	Canny	Sobel	Morphological
Applied Parameter	Noise reduction: <1% of max crack area	Gradient threshold: 0.21	Gradient threshold: 0.03	Binary threshold: 0.22
Detected crack with thresholds				
After manual adjustments	Not applicable			
Crack area	3.36%	3.29%	3.57%	3.37%
Error	4%	6%	2%	4%

Notes:

- True crack ratio: 3.50% (grid-based estimation).
- $Error = \frac{|Estimated\ Ratio - True\ Ratio|}{True\ Ratio} \times 100\%$

- Shadowed areas are excluded from the crack area calculation to avoid lighting artifacts.

3.4 Feature Ablation Analysis for SOM Interpretability

To assess the contribution of each input feature to segmentation quality—and to interpret SOM's internal decision cues—we conducted a series of ablation experiments across three representative crack types: concrete, brick, and shadow. In each experiment, one feature was excluded from the SOM input vector while all other parameters remained constant. The resulting segmentation maps were quantitatively evaluated by comparing crack pixel areas against the full-featured SOM output.

To ensure fair comparison, crack areas were computed after removing noise artifacts smaller than 1% of total segmentation—mirroring practical post-processing conditions. Relative error was defined as the absolute percentage deviation from the full-featured SOM result. For transparency, raw (unfiltered) crack areas are included in Appendix A.

Tables 5a–5c summarize the filtered ablation results. These error-based comparisons offer a coarse but interpretable measure of each feature's structural contribution, which is further visualized in the radar plots of Figures 22a–22c.

Table 5a: Feature Ablation on SOM-Based Concrete Crack Segmentation (Filtered)

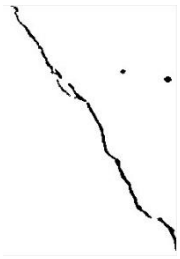
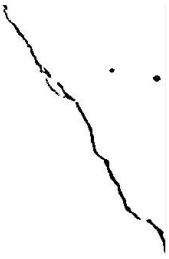
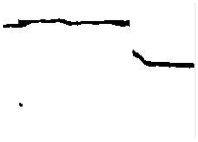
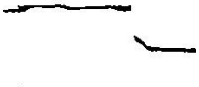
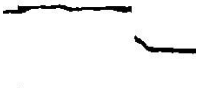
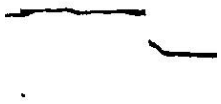
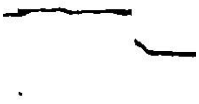
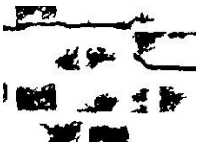
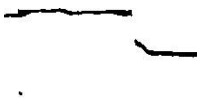
Feature Ablation	Without Grayscale	Without Contrast	Without Edge	Without Thin	Without Avg (3×3)
Image					
Crack area	1.59%	1.89%	1.33%	1.59%	1.59%
Error	4%	14%	20%	4%	4%
Feature Ablation	Without Avg (5×5)	Without ρ_{ij}	Without γ_{ij}	Without Hue	With All

Image					
Crack area	1.58%	1.57%	1.59%	1.57%	1.58%
Error	5%	5%	4%	5%	5%

Notes:

- True crack ratio: 1.66% (grid-based estimation).
- $Error = \frac{|Estimated\ Ratio - True\ Ratio|}{True\ Ratio} \times 100\%$

Table 5b. Feature Ablation on SOM-Based Brick Crack Segmentation (Filtered)

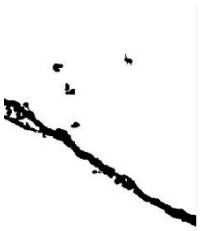
Feature Ablation	Without Grayscale	Without Contrast	Without Edge	Without Thin	Without Avg (3×3)
Image					
Crack area	2.93%	10.48%	2.48%	2.94%	2.93%
Error	2%	249%	17%	2%	2%
Feature Ablation	Without Avg (5×5)	Without ρ_{ij}	Without γ_{ij}	Without Hue	With All
Image					

Crack area	2.93	2.96%	2.92%	2.25%	2.93%
Error	2%	1%	3%	25%	2%

Notes:

- True crack ratio: 3.00% (grid-based estimation).
- $Error = \frac{|Estimated\ Ratio - True\ Ratio|}{True\ Ratio} \times 100\%$

Table 5c. Feature Ablation on SOM-Based Shadow Crack Segmentation (Filtered)

Feature Ablation	Without Grayscale	Without Contrast	Without Edge	Without Thin	Without Avg (3×3)
Image					
Crack area	3.37%	5.52%	3.32%	3.38%	3.36%
Error	4%	58%	5%	3%	4%
Feature Ablation	Without Avg (5×5)	Without ρ_{ij}	Without γ_{ij}	Without Hue	With All
Image					
Crack area	3.37%	3.37%	3.39%	3.37%	3.36%

Error	4%	4%	3%	4%	4%

Notes:

- True crack ratio: 3.50% (grid-based estimation).
- $Error = \frac{|Estimated\ Ratio - True\ Ratio|}{True\ Ratio} \times 100\%$

Across the three materials, the ablation results highlight distinct dependencies:

- Concrete cracks were most sensitive to edge removal (20% error), followed by contrast (14%), confirming the importance of sharp geometric transitions for fine surface discontinuities.
- Brick cracks relied heavily on contrast (249% error), with hue (25%) and edge (17%) also playing key roles—reflecting the difficulty of distinguishing cracks in high-texture masonry.
- Shadow cracks showed strong reliance on contrast (58%), consistent with the need for luminance variation when spatial geometry is obscured.

These patterns confirm that feature influence is material-dependent. While contrast, edge, and hue emerge as key drivers, other features provide complementary benefits—e.g., thinness enhances detection of hairline cracks, and local averaging mitigates noise from uneven backgrounds. Although the benchmark datasets do not include ultra-fine cracks, retaining the full feature set ensures adaptability across diverse SHM contexts.

To complement the ablation tables, radar plots were generated to visualize each feature's contribution under the SOM clustering mechanism. These charts capture SOM's unsupervised internal response to feature removal—providing a training-free perspective on interpretability beyond scalar metrics.

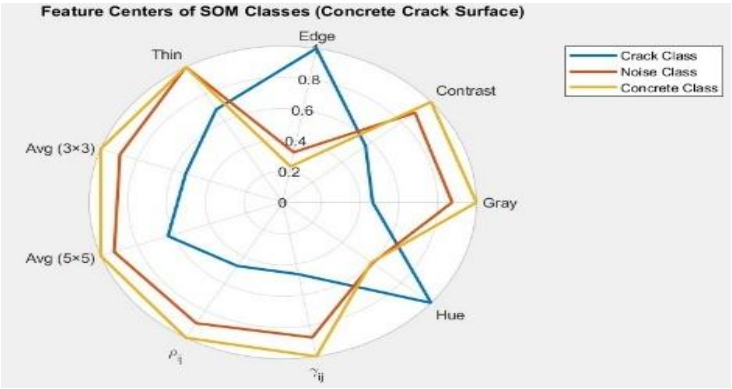


Fig 22a: Normalized crack area retention under feature ablation for concrete cracks

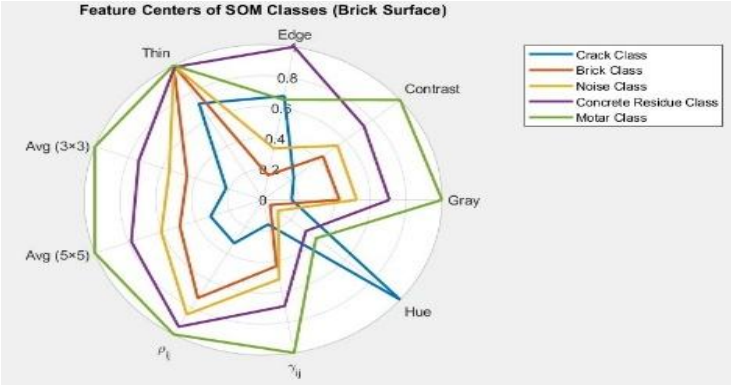


Fig 22b: Normalized crack area retention under feature ablation for brick cracks

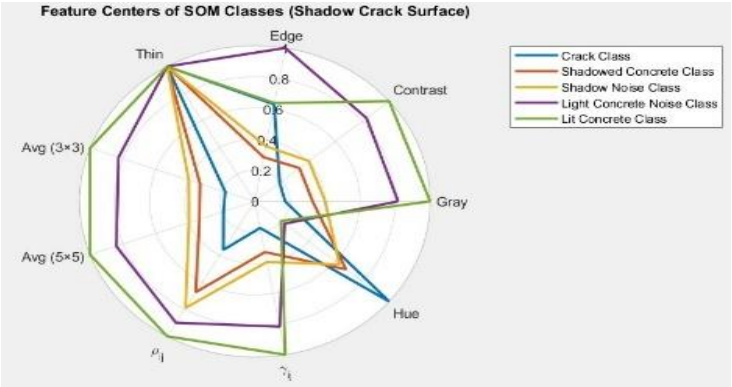


Fig 22c: Normalized crack area retention under feature ablation for shadow cracks

Figures 22a–22c confirm that:

- [Concrete] Geometry-delineating edge features dominate segmentation stability.
- [Brick] Texture suppression via contrast is crucial to prevent false positives.
- [Shadow] Luminance cues carried by contrast drive crack visibility when shape is weak.

Notably, hue consistently shows the highest normalized activation across all crack classes, indicating that color-based discontinuities form a universal visual signature of surface damage—even in near-grayscale conditions.

These observations also reinforce SOM's interpretable design: its clustering relies on both activating features (hue, edge) and suppressive signals (contrast, grayscale), balancing sharp cues with spatial context. The absence of any single dominant feature underscores the value of a multi-feature ensemble, particularly in training-free, domain-adaptive SHM systems.

3.5 Discussion

The SOM-based approach excels in real-world scenarios, such as textured brick surfaces and shadowed concrete, where traditional edge detectors and CNNs often struggle due to noise, lighting variations, or complex surface patterns. Tests on textured brick and shadowed concrete achieved errors as low as 2%–4% (Tables 3–4), compared to CNN

errors up to 10% in similar conditions [45]. For instance, rule-based classifiers leveraging crack geometry or semi-supervised learning with minimal labeled samples will be explored. The experimental results highlight fundamental differences among Sobel, Canny, morphological operations, and the proposed SOM-based approach for crack detection across concrete, brick, and shadowed surfaces. While Sobel and Canny effectively detect edges, they often misclassify shadows, stains, or surface textures as cracks, and require extensive manual threshold tuning. Morphological methods, although capable of reducing noise, are highly sensitive to parameter settings and often struggle to generalize across different materials and environmental conditions.

In contrast, the SOM-based method dynamically adapts to each image's visual structure through unsupervised clustering of pixel-level features—such as grayscale gradients, contrast, and hue—eliminating the need for predefined thresholds, feature engineering, or labeled training data. This adaptability minimizes human intervention: only a lightweight post-clustering step is required to identify the "crack" class. In practice, this step involves visually inspecting a few output classes and selecting the correct one using a GUI-supported labeling tool, typically taking less than 5 minutes per image—negligible compared to the hundreds of hours required for manual pixel-wise annotation in supervised learning workflows.

Compared to deep learning methods, the SOM framework offers a lightweight, training-free, and interpretable alternative. CNN-based segmentation models such as U-Net and Mask R-CNN have demonstrated high F1-scores (>0.9) on curated datasets with over 10^5 labeled training samples [41]. However, these models require significant GPU resources for both training and inference, and their performance often degrades under real-world conditions such as shadow interference, heterogeneous surfaces, and fine crack patterns [44], [45].

Our SOM-based method achieves comparable accuracy (≤5% error) without any labeled data or model retraining. It operates efficiently on standard CPUs with minimal memory usage (~100 MB), enabling practical deployment on embedded systems. A 512×512 image can be processed in under 1 minute, while high-resolution images (e.g., 3024×4032 JPEGs) can be analyzed within ~10 minutes without retraining, demonstrating strong scalability for large-format field imagery.

While CNN inference can be accelerated to ~5 seconds per image on GPUs, this speed is achievable only after extensive training involving 10^6+ FLOPs. In contrast, our SOM clustering requires approximately 10^4 FLOPs per image and operates in a fully training-free manner, making it ideal for deployment on UAVs, edge devices, and other compute-constrained platforms.

While SOM performs well across diverse materials, occasional misclassifications may still occur in visually ambiguous regions—such as mortar joints, surface stains, or deep shadows—where pixel features resemble true cracks.

SOM's current application focuses on simpler crack patterns to ensure fair benchmarking against Canny and Sobel, which are sensitive to complex morphologies. However, the multi-dimensional feature integration (e.g., edge gradient, thin crack structure) equips the SOM framework to handle branched, networked, or hairline cracks as well. Preliminary tests on more complex cracks have shown promising adaptability, and future work will expand validation across these more challenging scenarios.

By combining zero-training operation, adaptability to real-world artifacts, low hardware demand, and alignment with SHM practices emphasizing visual interpretability, the proposed SOM framework provides a robust foundation for scalable, autonomous crack detection in dynamic field environments.

To summarize the trade-offs among traditional, deep learning-based, and SOM-based approaches, Table 6 provides a comparative summary across key criteria:

Table 6: Comparative summary of crack detection methods for SHM deployment

Method	Data Requirement	Training	Accuracy	Explainable	Field Deployable
SOM (ours)	None	×	<5% error	✓	✓
CNN (e.g., U-Net)	10^{5+} samples	✓	~4% error [41]	×	×
Sobel	None	×	~3–5% error	✓	✓

Note：CNN accuracy assumes curated datasets with $>10^5$ labeled samples and GPU resources, whereas SOM operates on standard CPUs under real-world imagery.

To illustrate SOM's adaptability to non-linear and networked crack morphologies, Figure 23 shows preliminary clustering results on a complex fracture scenario. Despite the presence of intersecting and overlapping crack features, the SOM framework robustly distinguishes crack paths without requiring model retraining or parameter tuning.

Fig. 23: Complex crack detection by SOM

The complex crack pattern was derived from real-world imagery with intersecting and overlapping features, demonstrating SOM's robustness.

3.6 CNN Supervision with SOM-Derived Labels: A Study of Generalization and Interpretability

To evaluate the structural consistency and labeling reliability of the proposed SOM-based segmentation framework, we designed a series of experiments in which SOM-generated crack masks were directly employed as pseudo-labels

to supervise a lightweight U-Net CNN. These experiments were not intended to demonstrate the superiority of CNNs over unsupervised methods, but rather to assess whether SOM-derived masks can serve as stable, transferable supervisory targets. If a CNN trained on such weak labels can replicate the segmentation quality of SOM, it would further validate the utility of SOM as a self-contained, training-free pseudo-labeling engine for SHM tasks in noisy, real-world environments.

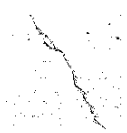
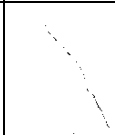
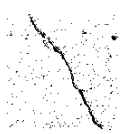
Four training regimes were tested by varying both the quality of SOM-generated labels and the semantic diversity of training images:

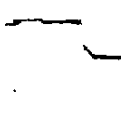
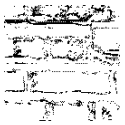
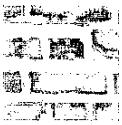
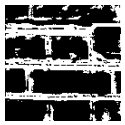
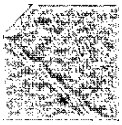
- Experiment 1: Clean SOM masks (with 1% noise removal) for all three crack types: concrete, brick, and shadow
- Experiment 2: Raw (unfiltered) SOM segmentations for the same three crack types
- Experiment 3: Clean SOM mask for concrete only; brick and shadow images were labeled as crack-free backgrounds
- Experiment 4: Same as Experiment 3, but using raw SOM mask for the concrete crack

Table 7 presents a visual and qualitative summary of CNN performance across the four configurations. Among them, Experiment 3—featuring a minimal yet semantically clear supervision setup—achieved the most accurate and noise-free segmentation for concrete cracks, while avoiding false detections in the brick and shadow domains. In contrast, Experiments 1 and 2 exhibited over-segmentation, misclassification of background textures, and degraded boundary clarity due to noisy or overly diverse training labels.

These results demonstrate that CNNs can internalize crack-relevant representations even under sparse supervision—provided the labels are of high semantic quality. Conversely, the results also expose CNNs' fragility when applied to surface domains not encountered during training, highlighting SOM's robustness across unseen material types.

Table 7: CNN Performance Under Four SOM-Guided Supervision Regimes

Experiment	Clean SOM (All Types)		Raw SOM (All Types)		Clean SOM (Concrete Only)		Raw SOM (Concrete Only)	
Training Labels	Clean SOM: Concrete, Brick, Shadow		Raw SOM: Concrete, Brick, Shadow		Clean SOM: Concrete; Birck/Shadow as Background		Raw SOM: Concrete; Birck/Shadow as Background	
	Label	Output	Label	Output	Label	Output	Label	Output
Concret Crack								

Brick Crack								
Shadow Crack								
Time	42s		42s		30s		36s	
CNN Output Summary	Concrete: Crack detected but noisy Brick: Texture misclassified as cracks Shadow: Crack not detected		Concrete: Discontinuous cracks Brick: Partial crack + brick misclassified Shadow: Crack and shadow merged		Concrete: Clear crack with minimal noise Brick/Shadow: No crack detected		Concrete: Crack detected but noisy Brick/Shadow: No crack detected	
Qualitative Observations	Misinterpretation of shadow–background contrast; over-segmentation on brick		Label noise led to boundary instability; false positives		Best segmentation among all; strong boundary formation with no generalization artifacts		Raw label noise impaired segmentation quality vs. Exp. 3	

To further evaluate the structural fidelity of SOM outputs, we compared the best-performing CNN (Experiment 3) to its original SOM pseudo-labels for each crack type. This comparison is detailed in Tables 8a and 8b. While CNN showed promising performance for supervised domains (e.g., concrete), it failed to generalize to unseen materials like brick and shadow. SOM, in contrast, preserved crack-relevant structures across all domains, underscoring its value as a training-free segmentation method with strong domain generalization.

Table 8a: Visual Comparison Between SOM and CNN Outputs

Crack Type	Input	SOM Output	CNN Output
Concrete Crack			

Brick Crack			
Shadow Crack			

483 Table 8b. Structural and Practical Trade-offs Between SOM (Unsupervised) and CNN (Supervised via SOM)

Criteria	SOM (Unsupervised)	CNN (Clean Concrete Only)
Inference Time (3 images)	25s (5s+8s+12s)	30s total
Training Requirement	None	Requires pseudo-labels (SOM supervision)
Material Transferability	✓ Robust across unseen textures (e.g., brick, shadow)	× Overfitting to trained material; fails to generalize
Robustness to Artifacts	✓ Stable under texture, lighting, and shadow variation	× Sensitive to non-crack textures; prone to false positives
Labeling Interpretability	✓ Directly interpretable clustering results	× Learned representation; difficult to interpret or trace decision boundaries
Deployment Readiness	✓ Lightweight, training-free, suitable for low-resource, real-time SHM applications	× Requires training, less robust to unseen conditions

484 Interpretability-Driven Lessons from CNN-SOM Supervision Experiments:

485 1. High-quality pseudo-labels trump noisy abundance.

CNNs trained on fewer but semantically cleaner SOM labels outperformed models trained on noisy or overly diverse data, confirming that quality outweighs quantity in supervision.

2. SOM labels enable transferable learning across unseen domains.

   Unlike CNNs, which struggled to generalize to untrained textures (e.g., bricks, shadows), SOM preserved semantic segmentation robustness across all materials without supervision.

3. Training-free SOMs outperform supervised CNNs in robustness and deployability.

   SOM offered faster inference (25s vs. 30s), zero retraining cost, and structural generalizability—making it highly suitable for real-time SHM scenarios in low-resource settings.

4 Conclusion

This study presents a lightweight, training-free framework for vision-based structural health monitoring (SHM) using Self-Organizing Maps (SOM). Unlike traditional approaches such as Canny, Sobel, and morphological operators—which rely on handcrafted thresholds and often fail under noise or texture interference—SOM clusters pixel-level features including grayscale, edge gradients, contrast, hue, and thin-crack indicators to segment damage regions without requiring any labeled data or parameter tuning. This unsupervised design consistently delivers robust segmentation across concrete, brick, and shadowed surfaces under real-world field conditions.

Compared to rule-based filters, SOM eliminates heuristic dependency while offering strong resilience in low-contrast and cluttered scenes. Its low computational cost (typically under 30 seconds per image on standard hardware) and compact memory footprint (<100 MB) make it well-suited for deployment on edge platforms such as UAVs and mobile SHM devices. Feature ablation experiments revealed material-specific dependencies—e.g., edge features produced 20% error on concrete, while contrast caused 249% error on brick. Radar-based visualizations further confirmed hue as a dominant contributor, offering insight into SOM's internal encoding dynamics.

CNNs trained on SOM-derived pseudo-labels achieved performance comparable to fully supervised models, validating the structural fidelity of SOM outputs. However, these CNNs failed to generalize across unseen domains, underscoring SOM's unique ability to serve as a label-free, transferable segmentation engine across diverse materials and environmental conditions—especially where data-driven models struggle to extrapolate.

Despite these advantages, limitations remain. The framework's reliance on well-crafted pixel features may reduce segmentation accuracy under extreme noise or in visually ambiguous textures—such as shadows, stains, or mortar lines that resemble true cracks. Future work will address these constraints through adaptive feature weighting, spatial encoding, and hybrid SOM–deep learning architectures. Extensions to UAV-based imaging—including compensation for motion blur and correction of camera drift—as well as integration into IoT-linked inspection pipelines will further enhance scalability and field automation. While SOM parameters were fixed in this study for consistency, dynamic tuning strategies will also be explored to enable adaptive clustering under ultra-low-contrast or in-flight conditions.

In conclusion, this work redefines SOM not as a legacy clustering algorithm, but as a next-generation, field-deployable intelligence engine for infrastructure diagnostics—offering a rare blend of interpretability, adaptability, and zero training cost. It lays a foundation for robust, explainable, and scalable SHM systems capable of operating under real-world constraints. Looking ahead, we envision fully autonomous crack segmentation workflows that integrate interpretable AI, statistical clustering dynamics, and real-time visual reasoning—paving the way for resilient, low-footprint diagnostic intelligence throughout the built environment.

Funding

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Declaration of Generative AI and AI-Assisted Technologies in the Writing Process

During the preparation of this article, the author utilized ChatGPT to assist with language refinement and editing. Following the use of this tool, the author reviewed and edited the content independently and takes full responsibility for the final version of the work.

Data Availability

The data supporting the findings of this study are available from the corresponding author upon reasonable request.

Submission Declaration

This article is an original work that has not been published previously and is not under consideration for publication elsewhere.

Acknowledgements

The author would like to thank Professor Peter Chang for his valuable discussions and constructive feedback throughout this work.

Reference

[1] P. Padmapoorani, S. Senthilkumar, and R. Mohanraj, "Machine Learning Techniques for Structural Health Monitoring of Concrete Structures: A Systematic Review," *Iranian Journal of Science and Technology, Transactions of Civil Engineering*, pp. 1–13, 2023, doi: 10.1007/s40996-023-01054-5.

[2] D. A. Tibaduiza Burgos, R. C. Gomez Vargas, C. Pedraza, D. Agis, and F. Pozo, "Damage identification in structural health monitoring: A brief review from its implementation to the use of data-driven applications," *Sensors*, vol. 20, no. 3, p. 733, 2020, doi: https://doi.org/10.3390/s20030733.

[3] M. Kohm, L. Stempniewski, and A. Stark, "Influence of vehicle traffic on modal-based bridge monitoring," *Journal of Civil Structural Health Monitoring*, pp. 1–16, 2022, doi: https://doi.org/10.1007/s13349-022-00630-z.

[4] C. Zhang, A. A. Mousavi, S. F. Masri, G. Gholipour, K. Yan, and X. Li, "Vibration feature extraction using signal processing techniques for structural health monitoring: A review," *Mechanical Systems and Signal Processing*, vol. 177, p. 109175, 2022, doi: https://doi.org/10.1016/j.ymssp.2022.109175.

[5] D. E. Bentley and C. T. Hatch, "Fundamentals of rotating machinery diagnostics," *Mechanical Engineering-CIME*, vol. 125, no. 10, pp. 61–62, 2003.

[6] C. R. Farrar *et al.*, "Damage prognosis: current status and future needs," *Los Alamos National Laboratory, LA*, vol. 176, pp. 177–178, 2003.

[7] C. R. Farrar and N. A. Lieven, "Damage prognosis: the future of structural health monitoring," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 365, no. 1851, pp. 623–632, 2007, doi: https://doi.org/10.1098/rsta.2006.1927.

[8] C. R. Farrar and K. Worden, "An introduction to structural health monitoring," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 365, no. 1851, pp. 303–315, 2007, doi: Https://doi.org/10.1098/rsta.2006.1928.

[9] D. C. Montgomery, *Introduction to statistical quality control*. John Wiley & Sons, 2007.

[10] P. J. Shull, *Nondestructive evaluation: theory, techniques, and applications*. CRC press, 2002.

[11] D. Balageas, C.-P. Fritzen, and A. Güemes, *Structural health monitoring*, vol. 90. John Wiley & Sons, 2010.

[12] C. R. Farrar, H. Sohn, and A. N. Robertson, "Applications of nonlinear system identification to structural health monitoring.," Los Alamos National Lab.(LANL), Los Alamos, NM (United States), 2004.

[13] F. J. Pallarés, M. Betti, G. Bartoli, and L. Pallarés, "Structural health monitoring (SHM) and Nondestructive testing (NDT) of slender masonry structures: A practical review," *Construction and Building Materials*, vol. 297, p. 123768, Aug. 2021, doi: 10.1016/j.conbuildmat.2021.123768.

[14] H. Sohn *et al.*, "A review of structural health monitoring literature: 1996–2001," *Los Alamos National Laboratory, USA*, vol. 1, 2003.

[15] K. Worden, C. R. Farrar, J. Haywood, and M. Todd, "A review of nonlinear dynamics applications to structural health monitoring," *Structural Control and Health Monitoring: The Official Journal of the International Association for Structural Control and Monitoring and of the European Association for the Control of Structures*, vol. 15, no. 4, pp. 540–567, 2008, doi: https://doi.org/10.1002/stc.215.

[16] F.-K. Chang, *Structural health monitoring 2000*. CRC Press, 1999.

[17] D. Feng and M. Q. Feng, "Computer vision for SHM of civil infrastructure: From dynamic response measurement to damage detection – A review," *Engineering Structures*, vol. 156, pp. 105–117, Feb. 2018, doi: 10.1016/j.engstruct.2017.11.018.

[18] A. Sharma and N. Mehta, "Structural health monitoring using image processing techniques-a review," *International Journal of Modern Computer Science*, vol. 4, no. 4, 2016.

[19] P. Broberg, "Surface crack detection in welds using thermography," *NDT & E International*, vol. 57, pp. 69–73, 2013, doi: https://doi.org/10.1016/j.ndteint.2013.03.008.

[20] A. Mohan and S. Poobal, "Crack detection using image processing: A critical review and analysis," *Alexandria Engineering Journal*, vol. 57, no. 2, pp. 787–798, 2018, doi: https://doi.org/10.1016/j.aej.2017.01.020.

[21] J. Yang, S. Li, Z. Wang, H. Dong, J. Wang, and S. Tang, "Using deep learning to detect defects in manufacturing: a comprehensive survey and current challenges," *Materials*, vol. 13, no. 24, p. 5755, 2020, doi: 10.3390/ma13245755.

[22] J. Canny, "A computational approach to edge detection," *IEEE Transactions on pattern analysis and machine intelligence*, no. 6, pp. 679–698, 1986, doi: 10.1109/TPAMI.1986.4767851.

[23] J. Serra and P. Soille, *Mathematical morphology and its applications to image processing*, vol. 2. Springer Science & Business Media, 2012.

[24] I. Sobel and G. Feldman, "A 3x3 isotropic gradient operator for image processing," *a talk at the Stanford Artificial Project in*, pp. 271–272, 1968.

[25] S. Agaian, A. Almuntashri, and A. T. Papagiannakis, "An improved canny edge detection application for asphalt concrete," in *2009 IEEE International Conference on Systems, Man and Cybernetics*, IEEE, 2009, pp. 3683–3687. doi: 10.1109/ICSMC.2009.5346873.

[26] A. S. Ahmed, "Comparative study among Sobel, Prewitt and Canny edge detection operators used in image processing," *J. Theor. Appl. Inf. Technol*, vol. 96, no. 19, pp. 6517–6525, 2018.

[27] R. Gupta, A. Tiwari, and P. Khatri, "Analysis of Satellite Image Filter for RISAT: A Review," *International Journal of Grid and Distributed Computing*, vol. 8, no. 5, pp. 111–116, 2015.

[28] E. R. Dougherty and R. A. Lotufo, *Hands-on morphological image processing. SPIE The International Society for Optical Engineering, Bellingham*. Washington USA, 2003.

[29] M. Goyal, "Morphological image processing," *IJCST*, vol. 2, no. 4, p. 59, 2011, doi: https://doi.org/10.1002/9781118093467.ch13.

[30] M. Hÿtch and P. W. Hawkes, *Morphological image operators*. Academic Press, 2020.

[31] M. Jankowski, "Erosion, dilation and related operators," in *8th International Mathematica Symposium*, 2006, pp. 1–10.

[32] Y. Li, S. Dhakal, and Y. Peng, "A machine vision system for identification of micro-crack in egg shell," *Journal of Food Engineering*, vol. 109, no. 1, pp. 127–134, 2012.

[33] M. Roushdy, "Comparative study of edge detection algorithms applying on the grayscale noisy image using morphological filter," *GVIP journal*, vol. 6, no. 4, pp. 17–23, 2006.

[34] L. Vincent, “Morphological area openings and closings for grey-scale images,” in *Shape in Picture*, Springer, 1994, pp. 197–208.

[35] D. Ze-Feng, Y. Zhou-Ping, and X. You-Lun, “High probability impulse noise-removing algorithm based on mathematical morphology,” *IEEE signal processing Letters*, vol. 14, no. 1, pp. 31–34, 2006, doi: https://doi.org/10.1109/lsp.2006.881524.

[36] W. Rong, Z. Li, W. Zhang, and L. Sun, “An improved CANNY edge detection algorithm,” in *2014 IEEE international conference on mechatronics and automation*, IEEE, 2014, pp. 577–582. doi: 10.1109/ICMA.2014.6885761.

[37] X. Tang, X. Wang, J. Hou, H. Wu, and D. Liu, “An improved sobel face gray image edge detection algorithm,” in *2020 39th chinese control conference (CCC)*, IEEE, 2020, pp. 6639–6643. doi: 10.23919/CCC50068.2020.9189302.

[38] T. Kohonen, “The self-organizing map,” *Proceedings of the IEEE*, vol. 78, no. 9, pp. 1464–1480, 1990, doi: 10.1109/5.58325.

[39] J.-H. Chen, M.-C. Su, R. Cao, S.-C. Hsu, and J.-C. Lu, “A self organizing map optimization based image recognition and processing model for bridge crack inspection,” *Automation in Construction*, vol. 73, pp. 58–66, 2017, doi: https://doi.org/10.1016/j.autcon.2016.08.033.

[40] S. Mathavan, M. Rahman, and K. Kamal, “Use of a self-organizing map for crack detection in highly textured pavement images,” *Journal of Infrastructure Systems*, vol. 21, no. 3, pp. 1–11, 2015, doi: https://doi.org/10.1061/(asce)is.1943-555x.0000237.

[41] Y. Hamishebahar, H. Guan, S. So, and J. Jo, “A Comprehensive Review of Deep Learning-Based Crack Detection Approaches,” *Applied Sciences*, vol. 12, no. 3, 2022, doi: 10.3390/app12031374.

[42] Hsieh Yung-An and Tsai Yichang James, “Machine Learning for Crack Detection: Review and Model Performance Comparison,” *Journal of Computing in Civil Engineering*, vol. 34, no. 5, p. 04020038, Sep. 2020, doi: 10.1061/(ASCE)CP.1943-5487.0000918.

[43] J. Li, X. Li, K. Liu, and Z. Yao, “Crack Identification for Bridge Structures Using an Unmanned Aerial Vehicle (UAV) Incorporating Image Geometric Correction,” *Buildings*, vol. 12, no. 11, Art. no. 11, Nov. 2022, doi: 10.3390/buildings12111869.

[44] S. D. Nguyen, T. S. Tran, V. P. Tran, H. J. Lee, M. J. Piran, and V. P. Le, “Deep learning-based crack detection: A survey,” *International Journal of Pavement Research and Technology*, vol. 16, no. 4, pp. 943–967, 2023.

[45] K. Bhalaji Kharthik *et al.*, “Transfer learned deep feature based crack detection using support vector machine: a comparative study,” *Scientific Reports*, vol. 14, no. 1, p. 14517, 2024.

[46] J. Vesanto and E. Alhoniemi, “Clustering of the self-organizing map,” *IEEE Transactions on neural networks*, vol. 11, no. 3, pp. 586–600, 2000, doi: https://doi.org/10.1109/72.846731.

[47] G. Rozenberg, T. Bäck, and J. N. Kok, *Handbook of natural computing*. Springer, 2012.

[48] R. Sathya and A. Abraham, “Comparison of supervised and unsupervised learning algorithms for pattern classification,” *International Journal of Advanced Research in Artificial Intelligence*, vol. 2, no. 2, pp. 34–38, 2013, doi: https://doi.org/10.14569/ijarai.2013.020206.

[49] M. Ghaseminezhad and A. Karami, “A novel self-organizing map (SOM) neural network for discrete groups of data clustering,” *Applied Soft Computing*, vol. 11, no. 4, pp. 3771–3778, 2011, doi: https://doi.org/10.1016/j.asoc.2011.02.009.

[50] F. Rossi, B. Conan-Guez, and A. El Golli, “Clustering functional data with the SOM algorithm.,” presented at the ESANN, Citeseer, 2004, pp. 305–312.

[51] T. Furukawa, “SOM of SOMs,” *Neural Networks*, vol. 22, no. 4, pp. 463–478, 2009, doi: https://doi.org/10.1016/j.neunet.2009.01.012.

[52] E. Berglund and J. Sitte, “The parameterless self-organizing map algorithm,” *IEEE Transactions on neural networks*, vol. 17, no. 2, pp. 305–316, 2006.

[53] J. Vesanto, J. Himberg, E. Alhoniemi, and J. Parhankangas, “Self-organizing map in Matlab: the SOM Toolbox,” presented at the Proceedings of the Matlab DSP conference, Espoo, 1999, pp. 16–17.

[54] P. F. Van Kessel, L. J. Hornbeck, R. E. Meier, and M. R. Douglass, “A MEMS-based projection display,” *Proceedings of the IEEE*, vol. 86, no. 8, pp. 1687–1704, 1998, doi: https://doi.org/10.1109/5.704274.

[55] L. Vincent, “Grayscale area openings and closings, their efficient implementation and applications,” in *First Workshop on Mathematical Morphology and its Applications to Signal Processing*, 1993, pp. 22–27.

[56] W. Zhang, Z. Zhang, D. Qi, and Y. Liu, “Automatic crack detection and classification method for subway tunnel safety monitoring,” *Sensors*, vol. 14, no. 10, pp. 19307–19328, 2014, doi: https://doi.org/10.3390/s141019307.

[57] S. Nagabhushana, *Computer vision and image processing*. New Age International, 2005.

[58] M. S. Kaseko and S. G. Ritchie, “A neural network-based methodology for pavement crack detection and classification,” *Transportation Research Part C: Emerging Technologies*, vol. 1, no. 4, pp. 275–291, 1993, doi: https://doi.org/10.1016/0968-090x(93)90002-w.

[59] S. K. Sinha and P. W. Fieguth, “Automated detection of cracks in buried concrete pipe images,” *Automation in construction*, vol. 15, no. 1, pp. 58–72, 2006.

[60] K. Gu, D. Tao, J.-F. Qiao, and W. Lin, “Learning a no-reference quality assessment model of enhanced images with big data,” *IEEE transactions on neural networks and learning systems*, vol. 29, no. 4, pp. 1301–1313, 2017, doi: https://doi.org/10.1109/tnnls.2017.2649101.

[61] D. Phillips, “Image processing in C- Analyzing and enhancing digital images(Book),” *Lawrence, KS: R & D Publications, Inc, 1994.*, 1994.

[62] T. Gevers and A. W. Smeulders, “Color-based object recognition,” *Pattern recognition*, vol. 32, no. 3, pp. 453–464, 1999, doi: https://doi.org/10.1016/s0031-3203(98)00036-3.

[63] J. Zou and H. Kim, “Using hue, saturation, and value color space for hydraulic excavator idle time analysis,” *Journal of computing in civil engineering*, vol. 21, no. 4, pp. 238–246, 2007, doi: https://doi.org/10.1061/(asce)0887-3801(2007)21:4(238).

Appendix A: Raw Feature Ablation Results for SOM-Based Crack Segmentation

This appendix presents raw segmentation results under systematic feature ablation for each image type. Each row represents the SOM output when a specific descriptor is removed from the input vector.

Table A1: Concrete: SOM Ablation Results by Feature Removal

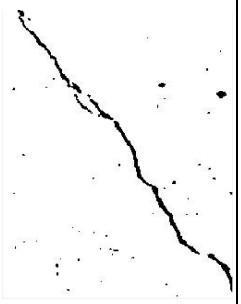
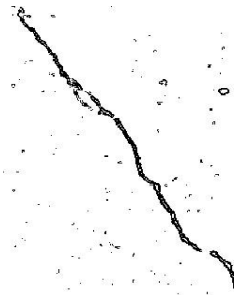
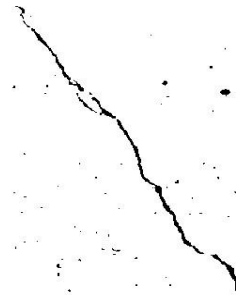
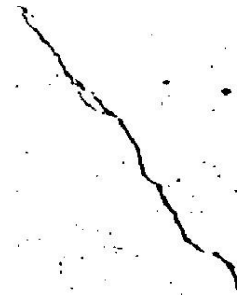
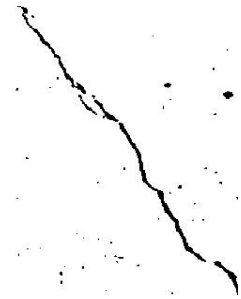
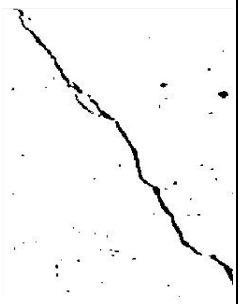
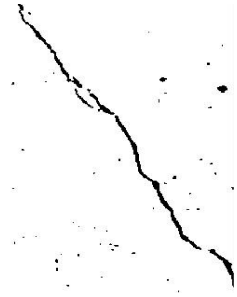
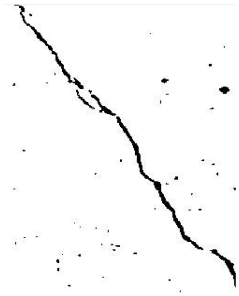
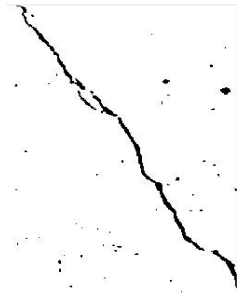
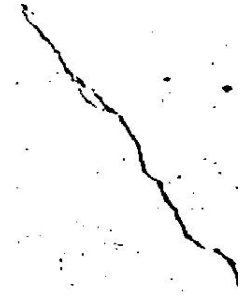
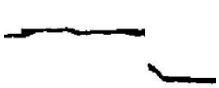
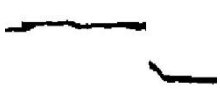
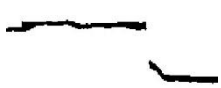
Without Grayscale	Without Contrast	Without Edge	Without Thin	Without Avg (3×3)
				
Without Avg (5×5)	Without ρ_{ij}	Without γ_{ij}	Without Hue	With All Features
				

Table A2: Brick: SOM Ablation Results by Feature Removal

Without Grayscale	Without Contrast	Without Edge	Without Thin	Without Avg (3×3)
				
Without Avg (5×5)	Without ρ_{ij}	Without γ_{ij}	Without Hue	With All Features

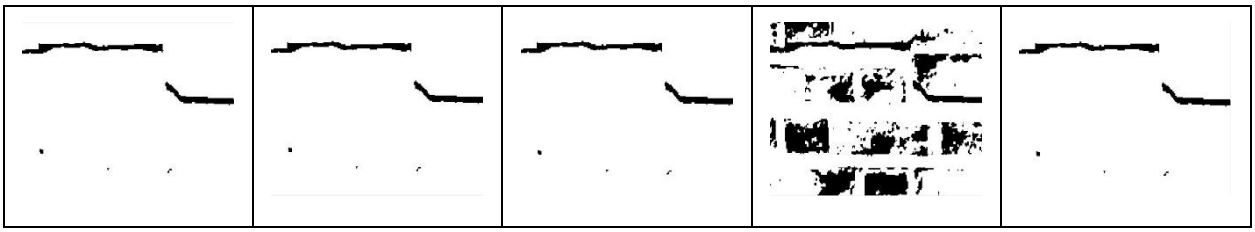
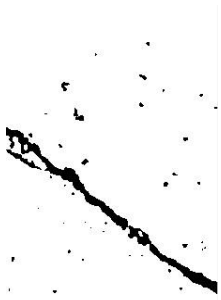
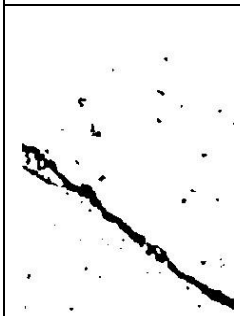
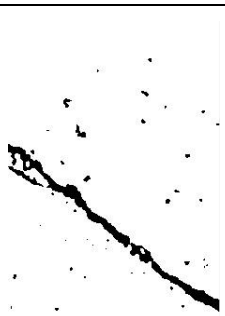


Table A3: Shadow: SOM Ablation Results by Feature Removal

Without Grayscale	Without Contrast	Without Edge	Without Thin	Without Avg (3×3)
				
Without Avg (5×5)	Without ρ_{ij}	Without γ_{ij}	Without Hue	With All Features
				

Appendix B: Image Dimensions, SOM Settings, and Feature Composition

1. Image Dataset and Dimensions

Three representative crack images—capturing concrete, brick, and shadowed surfaces—were used consistently across all experiments to ensure a unified basis for method comparison. These images were employed in:

- SOM-based unsupervised crack segmentation, including class-wise clustering and post-processing
- Baseline method comparison, including Canny, Sobel, and morphological operations
- Feature ablation experiments to assess the impact of individual visual descriptors
- CNN evaluation, using SOM outputs as pseudo labels
- Grid-based crack area estimation, used to approximate ground-truth ratios for accuracy evaluation

The grid-derived crack area ratios served as reference baselines to evaluate the segmentation accuracy of all methods, as reported in Tables 2–4 of the main manuscript.

Table B1: Crack Image Dataset with Resolutions and Experimental Usage

Material	Original Resolution (pixels)	Used In
Concrete	796 × 481	SOM, Canny, Sobel, Morphological, Ablation, CNN, Crack Ratio Estimation
Brick	800 × 600	SOM, Canny, Sobel, Morphological, Ablation, CNN, Crack Ratio Estimation
Shadow	276 × 434	SOM, Canny, Sobel, Morphological, Ablation, CNN, Crack Ratio Estimation

2. SOM Training Configuration

All SOM networks were initialized using MATLAB's `selforgmap` function and trained using the default `train` routine, which internally applies `trainr` — a random-order incremental update algorithm. Unless otherwise specified, all parameters remained at default values.

To accommodate varying material and visual complexity, the SOM output map size was adapted for each dataset:

Table B2: SOM Map Size and Output Classes per Dataset

Image Type	SOM Map Size	Output Classes
Concrete	1 × 3	Concrete, Crack, Noise

Brick	1 × 5	Crack, Brick, Noise, Concrete Residue, Motar
Shadow	1 × 5	Crack, Shadowed Concrete, Shadow Noise, Light Concrete Noise, Lit Concrete

The following parameters were consistent across all SOM models:

Table B3: Common Training Parameters for All SOM Models

Parameter	Value / Description
Training function	`trainr` (random-order incremental update)
Topology function	`hextop`
Distance function	`linkdist`
Initial neighborhood size	3
Training epochs	200
Learning rate	Linearly decayed from 0.9 to 0.02