From Semi-Automatic to Fully Automatic: An Auditable SOM Pipeline for Label-Free Crack Segmentation

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

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

*Corresponding Author

Email: xinxin68@terpmail.umd.edu

Abstract

Self-Organizing Maps (SOMs) enable unsupervised, label-free, and interpretable crack segmentation, yet practical deployment is often limited by manual choices of feature-space dimension, cluster number, and visual crack-class confirmation. This paper presents a fully automatic and auditable SOM-based pipeline that removes these expert-dependent degrees of freedom while preserving interpretable pixel descriptors. The method forms a deterministic decision chain integrating PCA-based automatic dimension selection, elbow-based cluster-number determination, an intensity-driven crack-class selector with signature-based auditing evidence (radar and bar summaries), and fixed post-processing to produce a binary crack mask without any pixel-wise labels. An optional per-image CNN refinement stage, trained only on SOM pseudo-labels, serves as a lightweight spatial regularizer to refine boundary coherence. Experiments on three representative field conditions—aggregate-textured concrete, texture-dominated masonry, and shadow-affected surfaces—demonstrate robust end-to-end automation, high SOM–CNN internal consistency (Dice 0.86–0.91), and practical CPU-only runtimes (4–14 min per image). The results position fully automatic, interpretable clustering as a viable crack segmentation pathway for edge-oriented inspection workflows under realistic constraints and distribution shifts.

Keywords

Structural health monitoring; crack segmentation; Self-Organizing Maps (SOM); unsupervised learning; interpretable clustering; label-free; auditable AI

1. Introduction

Crack detection is a fundamental task in vision-based structural health monitoring (SHM), providing direct evidence of surface damage, material degradation, and structural distress that can inform maintenance decisions and safety-relevant assessments [1], [2], [3]. Although supervised deep learning has advanced crack segmentation performance on curated datasets [4], [5], practical deployment in infrastructure inspection often operates under different constraints [6]. Field imagery can vary substantially across materials, textures, surface states, and illumination conditions [7], [8]; pixel-level crack masks are costly and rarely available at scale [9]; and model maintenance may require repeated adaptation as inspection conditions shift [10]. In parallel, engineering workflows frequently prefer decision processes that are transparent and auditable, particularly when outputs inform safety-related decisions [11], [12], [13]. These considerations motivate training-free or label-free methods that can be deployed immediately on new surfaces without adaptation or retraining, while retaining a clear decision trail to support auditable engineering use and operator trust [14].

Training-free baselines such as classical edge and ridge detectors are attractive for their simplicity, yet they are highly sensitive to heterogeneous textures and illumination artifacts and can produce elevated false positives on real structural surfaces [15]. More recent label-free paradigms reduce manual annotation requirements, but may still involve nontrivial model training [16], [17], [18], [19], large image collections, or substantial compute, and their decision logic can remain difficult to interpret [20]. Clustering-based segmentation offers a complementary pathway: by grouping

pixels in an explicitly defined feature space, clustering can expose interpretable structure while avoiding dependence on annotated masks [21]. In particular, Self-Organizing Maps (SOMs) provide a practical and computationally lightweight mechanism for organizing high-dimensional descriptors while preserving topological relationships in the feature space, and have shown promise for material-agnostic segmentation [22], [23], [24]. However, prior SOM-style crack workflows often retain human-in-the-loop steps, such as manual choices of feature dimensionality, cluster number, or visual confirmation of which cluster corresponds to cracks, limiting scalability and autonomy in large inspection campaigns and edge-oriented deployments [25].

In our prior study [26], we demonstrated that pixel-level SOM clustering using an interpretable, physics-informed feature set can achieve robust crack region segmentation without labeled training data, pretrained models, or GPU resources. The method's appeal lies not only in its training-free nature, but also in its intrinsic explainability: clustering outcomes can be inspected through feature signatures that align with crack-like morphology. Nevertheless, the earlier framework still required minimal operator interaction to finalize segmentation, including selecting the number of clusters based on perceived surface complexity and identifying the crack cluster through visual inspection. While these steps are fast on a single image, they introduce practical barriers at scale by reducing throughput, introducing operator-to-operator variability over long sessions, and preventing autonomous operation when real-time operator access is unavailable [27], [28], [29].

This paper removes that final manual bottleneck by presenting a fully automatic extension of the SOM framework that preserves interpretability while eliminating subjective parameter selection and cluster confirmation. The proposed pipeline forms an end-to-end, deterministic decision chain consisting of: (i) PCA-based dimensionality reduction with automatic retention of informative components, (ii) elbow-based automatic selection of the number of clusters, (iii) automatic crack-

class identification using a physically motivated intensity criterion, and (iv) fixed post-processing to produce a final crack mask. While the selection rule leverages local contrast as a practical cue, the pipeline retains an intrinsic explainability layer via cluster signatures (e.g., radar and bar summaries) that verify the automated decision and anchor it in multidimensional feature evidence, improving robustness when intensity alone is ambiguous. Finally, we introduce an optional per-image CNN refinement stage trained purely on SOM pseudo-labels. This CNN is not positioned as a semantic crack detector; instead, it functions as a spatial regularizer that smooths boundaries while remaining anchored to SOM-generated semantics. Because the refinement is trained per image, the CNN primarily injects a local smoothness and connectivity prior, which can suppress isolated speckles or fragmented artifacts that occasionally arise in pixel-wise clustering without altering the SOM-derived class assignment.

The contributions of this work are summarized as follows:

(1) Fully automatic SOM model selection: a training-free mechanism that determines feature-space dimension and cluster number without manual tuning, consistent with deployment scenarios where hyperparameter search is infeasible.

(2) Automatic crack-class identification with intrinsic auditability: a deterministic crack-selection rule paired with cluster signature evidence, enabling transparent verification of the automated decision.

(3) Optional pseudo-label CNN refinement as spatial regularization and coherence checking: per-image training that improves mask regularity without ground-truth labels or pretrained weights, while providing an additional check on SOM pseudo-label spatial coherence through SOM–CNN agreement.

(4) Empirical validation under representative field nuisances: evaluation on heterogeneous surface conditions (aggregate-textured concrete under clean lighting, texture-dominated masonry, and shadow-affected imagery) in a strictly label-free setting, demonstrating that full automation preserves segmentation coherence while eliminating human interaction, together with component-wise analysis that clarifies automation impact and internal consistency.

Overall, the proposed system targets a distinct and practically important operating regime for infrastructure inspection: immediate deployment without labeled datasets, without GPUs, and with decisions that remain interpretable by construction. Rather than optimizing peak benchmark performance, this work focuses on removing the last expert-dependent step from a proven unsupervised framework, enabling scalable processing and edge-oriented use under realistic constraints, including limited compute (CPU-only), absence of labeled data, and the need for interpretable, auditable decisions.

The remainder of this paper is organized as follows. Section 2 describes the fully automated pipeline, including PCA-based dimensionality selection, elbow-based cluster number determination, automatic crack-class identification with signature-based verification, and optional CNN refinement. Section 3 presents experimental validation on representative field conditions. Section 4 concludes with a discussion of practical considerations, limitations, and future directions. The contribution is not a new clustering algorithm, but a deterministic, auditable decision chain that removes expert-dependent degrees of freedom in deployment.

2. Methodology

An overview of the proposed methodology is provided in Figure 1.

A fully automated, label-free, and explainable SOM-centered pipeline is proposed as an extension of the prior unsupervised crack-segmentation framework, targeting field variability and

eliminating manual decisions in the SOM workflow. The methodology contains four stages: (1) unsupervised feature-space compression via PCA with automatic dimension selection, (2) SOM clustering with elbow-based automatic selection of the number of classes, (3) explainable crack-class identification using interpretable cluster signatures (intensity/density rule supported by radar/bar visualizations), and (4) an optional CNN refinement trained only on SOM pseudo-labels to smooth boundaries and provide an independent consistency check. The entire pipeline requires no human ground truth, no manual threshold tuning, and no pretrained deep models.

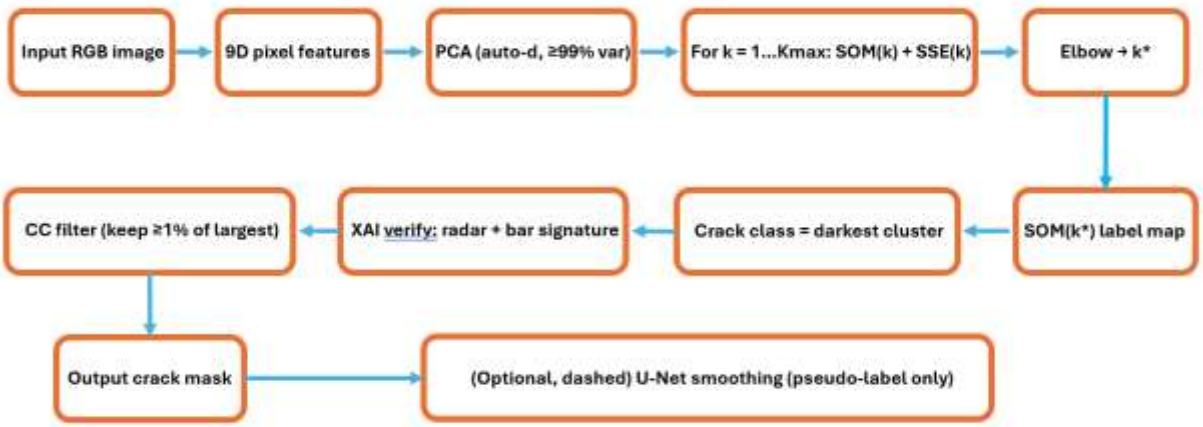


Figure 1. Fully automated and explainable SOM-based crack segmentation pipeline with optional U-Net refinement

2.1 Feature representation

Each pixel is represented using the same nine-dimensional, interpretable descriptor previously introduced for crack segmentation [26]. The descriptor combines complementary cues capturing crack-like morphology (thin/line/edge responses), local contrast statistics, multi-scale neighborhood intensity structure, and hue information, enabling robust unsupervised clustering under heterogeneous textures and illumination. All feature channels are computed and normalized using the identical preprocessing protocol described in that work prior to SOM-based clustering.

2.2 PCA for explainable, automatic dimensionality selection

Principal component analysis (PCA) is a deterministic, unsupervised linear transformation that re-expresses correlated feature channels into orthogonal latent axes ordered by explained variance [30], [31], [32], [33]. In this pipeline, PCA is used to enable explainable, automatic selection of the retained dimension: different surfaces (clean concrete vs. highly textured masonry or shadowed regions) exhibit different effective complexity, so the number of latent factors required for stable clustering should be inferred from the data rather than fixed manually. The cumulative explained-variance curve provides a direct, human-interpretable justification for the selected dimension.

Let the raw per-pixel feature matrix be $X \in \mathbb{R}^{N\times 9}$, where each row corresponds to one valid pixel. For numerical robustness, non-finite entries are replaced by zeros. PCA is performed on the z-score standardized matrix Z:

$$Z_{:,j} = \frac{X_{:,j} - \mu_j}{\sigma_j}, j = 1, \dots, 9, \tag{1}$$

Here σ_j is safeguarded to avoid division-by-zero for near-constant channels. PCA yields eigenvalues $\{\lambda_i\}$and explained-variance ratios $\{e_i\}$(sorted in descending order):

$$e_i = \frac{\lambda_i}{\sum_{m=1}^{9} \lambda_m} \tag{2}$$

The retained dimension is selected fully automatically by a fixed, non-tuned criterion:

$$d = \min\{\, k \;\mid\; \textstyle\sum_{i=1}^{k} e_i \geq 99\% \,\} \tag{3}$$

Only the first d principal component scores are retained and used as the SOM clustering input, ensuring a compact yet information-preserving embedding without manual dimensionality choices.

2.3 SOM clustering with elbow-based automatic selection of class number

SOM performs unsupervised, topology-preserving clustering by mapping high-dimensional samples onto a low-dimensional lattice while maintaining neighborhood relationships. In this study,

SOM is used to organize pixel embeddings into a small number of interpretable prototype-like groups, which is particularly suitable for subsequent rule-based (explainable) crack-class identification [34], [35].

A key practical challenge is that the appropriate number of clusters varies with material texture, surface heterogeneity, and illumination. Rather than manually specifying the class number, an elbow criterion is adopted as a deterministic, data-driven model-complexity selection rule: it identifies the point beyond which increasing the number of clusters yields diminishing returns in within-cluster compactness [36], [37], [38], [39], [39], [40].

After PCA compression, SOM clustering is performed in the reduced space. Candidate class numbers $k \in \{2, \dots, K_{\max}\}$ are evaluated, where $K_{\max}$ acts only as a search upper bound rather than the final class count. For each k, SOM assigns a label $c(t) \in \{1, \dots, k\}$ to each pixel embedding $y_t \in \mathbb{R}^d$(PCA score). The within-cluster sum-of-squares (SSE) is computed in the PCA space as:

$$\mathrm{SSE}(k) = \sum_{t=1}^{N} \| y_t - \mu_{c(t)} \|_2^2 \tag{4}$$

where $\mu_c$denotes the centroid of class $c$in the PCA space. The final class number $k^*$is selected by detecting the largest curvature of the SSE–k curve using a second-order change criterion (discrete second difference):

$$k^* = \arg \max_{k \in \{3, \dots, K_{\max}-1\}} (\Delta^2 \mathrm{SSE}(k)), \Delta^2 \mathrm{SSE}(k) = (\mathrm{SSE}(k-1) - \mathrm{SSE}(k)) - (\mathrm{SSE}(k) - \mathrm{SSE}(k+1))$$

This yields a fully automated, material-adaptive selection of k without manual tuning.

2.4 Explainable crack-class evidence from cluster signatures (radar + bar validation)

After SOM clustering, each class is interpreted through an explicit *cluster signature* in the original nine-dimensional feature space (pre-PCA). Specifically, for class c, the class-wise mean descriptor is computed from all pixels assigned to that class:

$$\bar{\mathbf{f}}^{(c)} = \frac{1}{\mid \Omega_c \mid} \sum_{p \in \Omega_c} \mathbf{f}_p \in \mathbb{R}^9 \tag{6}$$

where $\mathbf{f}_p$is the raw nine-dimensional feature vector of pixel p, and $\Omega_c$denotes the set of pixels with SOM label c. These class-wise means directly correspond to the per-class feature averages used to generate the radar and bar plots.

Because the nine features have heterogeneous units and numerical ranges, visualization is performed using a feature-wise normalization identical to the implementation in the code: each feature channel is scaled by the maximum class mean of that channel across all classes,

$$\tilde{f}_j^{(c)} = \frac{\bar{f}_j^{(c)}}{\max\limits_{c'} \bar{f}_j^{(c')}}, j = 1, \dots, 9, \tag{7}$$

yielding $\tilde{f}_j^{(c)} \in [0,1]$for consistent cross-feature comparison. The normalized class signatures are then visualized by (i) a radar plot that emphasizes the *shape* of the multi-feature profile and (ii) a grouped bar chart that makes the *per-feature ranking* across classes explicit. Importantly, both plots are derived from the same normalized statistics, providing redundant, cross-checkable evidence rather than a single subjective visualization.

Crack pixels exhibit a physically meaningful signature: they are typically darker than surrounding background and form thin, line-like structures that induce strong local discontinuities. Accordingly, the crack class is expected to show (i) low intensity-related cues, reflected by a lower Gray feature and reduced local mean features (Avg 3×3, Avg 5×5), together with (ii) high line/edge-related cues, reflected by elevated Edge (and often Contrast and thinness-related responses). In contrast, non-crack classes tend to display more uniform profiles corresponding to background texture, aggregate/brick patterns, or color-driven regions (Hue-dominant), without simultaneously exhibiting the "dark + line-like" combination.

In practice, the radar plot provides a quick visual identification of the candidate crack class as the one whose profile simultaneously dips on Gray/local-mean axes while peaking on the Edge axis, producing a distinctly anisotropic signature compared with other classes. The bar chart serves as a second, stricter validation by showing the same evidence feature-by-feature: the candidate crack class should remain among the lowest in Gray-related features while ranking among the highest in Edge/line-strength. This radar–bar agreement forms an explainable evidence layer that makes the crack-class decision transparent and verifiable, before the fully automatic decision rule is applied in the next step.

2.5 Explainable crack-class identification via an automatic intensity rule with signature-based verification

Once SOM yields a pixel-wise partition into k^* classes, the remaining unlabeled decision is to determine which class corresponds to cracks. In the current implementation, this decision is made fully automatically using a deterministic intensity rule computed in the original image domain.

For each class $c \in \{1, \dots, k^*\}$, the mean grayscale intensity is computed over its pixel support Ω_c:

$$\bar{I}(c) = \frac{1}{|\,\Omega_c\,|} \sum_{p \in \Omega_c} I(p) \tag{8}$$

where $I(p)$ denotes the grayscale intensity at pixel p. The crack class is selected as the class with the minimum mean intensity:

$$c_{\text{crack}} = \arg \min_{c \in \{1, \dots, k^*\}} \bar{I}(c) \tag{9}$$

This rule is physically motivated and reproducible: for many field surfaces, cracks appear as thin, dark discontinuities relative to the surrounding material, making the "darkest-cluster" criterion a stable selector without manual thresholds or labels. In practice, this intensity-based selector is cross-validated by the cluster signatures (Section 2.4), ensuring that the selected class also exhibits

crack-consistent edge and thin-structure responses rather than merely low intensity. In cases where cracks are not the darkest regions (e.g., due to efflorescence, dust filling, or overexposure), the same framework can be retained by replacing the intensity-only selector with a multi-criterion crackness score that re-weights interpretable cues already available in the feature set (e.g., edge/line response and thinness), while keeping the overall pipeline fully automatic.

Dual verification (XAI-critical). Eq. (9) provides the primary automatic selector, while the cluster-signature evidence in Section 2.4 (radar + bar diagnostics) provides a secondary, interpretable verification: the selected class should also exhibit the crack-consistent multi-feature profile (low Gray/local-mean cues together with strong line/edge-related cues). Agreement between the automatic selector and the signature diagnostics forms a dual-check mechanism that makes the crack-class decision auditable rather than heuristic.

2.6 SOM-mask post-processing via connected-component area filtering

To suppress scattered fragments while preserving **major crack components**, connected-component filtering is applied to the selected crack mask using 8-connectivity. Let the connected components have areas $\{A_j\}$ and define $A_{\max} = \max_j A_j$. Components are retained if

$$A_j \geq 0.01\, A_{\max} \tag{10}$$

This relative-area rule removes isolated speckles induced by surface texture and illumination while preserving all crack-related components above a fixed fraction of the largest component. The threshold is fixed a priori and applied uniformly across all cases, providing a parameter-light and reproducible post-processing step.

2.7 CNN refinement as a pseudo-label–only spatial regularizer (optional smoothing stage)

As an optional secondary stage, a compact U-Net–style model is trained to regularize the spatial coherence of the SOM crack mask. The CNN is not used as a supervised crack detector and does

not rely on any human-annotated ground truth. Instead, it learns a local smoothing prior anchored to the SOM-derived pseudo-labels, improving boundary regularity and suppressing small discontinuities while preserving the unsupervised crack semantics. The U-Net is trained exclusively on pseudo-labels generated by the preceding SOM pipeline and does not rely on any human annotation or external ground truth.

The CNN module is intentionally under-parameterized and trained per-image on a single pseudo-labeled sample, preventing it from acting as an independent semantic classifier and restricting its role to spatial regularization.

The input is the grayscale image. The target pseudo-label is the cleaned SOM crack mask (binary, stored as $\{0, 1\}$or $\{0, 255\}$and binarized automatically). Training is patch-based using a sliding-window dataset with data augmentation (random flips/rotations and mild photometric perturbations). Class imbalance is handled by a positive-class weight in the binary cross-entropy loss. The optimization objective is

$$\mathcal{L} = \mathcal{L}_{\mathrm{BCE}}(\mathrm{logits}, M_{\mathrm{SOM}}) \; + \; w\,\mathcal{L}_{\mathrm{Dice}}(\mathrm{logits}, M_{\mathrm{SOM}}) \tag{11}$$

where $w$controls the Dice term weight.

During training, performance is monitored only by agreement with the SOM pseudo-label rather than supervised accuracy: after each epoch, a full-image probability map is obtained by sliding-window inference with overlap-averaging. A small set of candidate thresholds is searched automatically to maximize a selected agreement metric (Dice/IoU/F1) between the binarized CNN output and the SOM pseudo-label. The binarized CNN output is further post-processed using the same connected-component area filtering rule as in the SOM stage (retain components ≥ 1% of the largest component), ensuring consistent cleaning across stages. The best checkpoint is selected by

the chosen pseudo-label agreement metric, and final outputs include the probability map, the post-processed binary mask, and boundary overlays for visual auditing.

The CNN stage is intentionally not used as an independent validator of SOM correctness. Instead, CNN–SOM agreement is reported solely as an indicator of spatial coherence and refinement stability, answering whether the unsupervised SOM output is learnable and reproducible by a spatial model under pseudo-supervision.

3. Experiment

All experiments were conducted in a fully label-free setting. Three representative field images were selected (consistent with our prior study): (i) a concrete surface with well-defined cracks (477×447 pixels), (ii) a masonry brick wall with strong texture interference (551×1055 pixels), and (iii) a shadow-affected surface that produces illumination-induced "false crack" patterns (323×275 pixels). Together, these cases cover clean crack morphology, texture-dominated clutter, and lighting-induced ambiguity commonly encountered in field inspections. No human-annotated ground truth, manual threshold tuning, or pretrained deep models were used at any stage.

Feature extraction and SOM clustering were implemented in MATLAB R2023b, while the U-Net-based refinement module was implemented in Python 3.13.2 using PyTorch 2.9.0. All experiments were run on a standard workstation (Intel i7-13700H CPU, 16 GB RAM), and the CNN was trained on CPU.

Because the CNN is trained using SOM-generated pseudo-labels (rather than human ground truth), we do not report supervised accuracy. Instead, we monitor training stability using soft-agreement metrics—Dice coefficient and Intersection-over-Union (IoU)—computed between the CNN predictions and the SOM pseudo-labels.

3.1 Concrete Surface: SOM-Based Crack Segmentation with CNN Smoothing

This subsection presents the concrete surface example to illustrate the complete SOM–CNN workflow, including pixel-level unsupervised clustering, automatic crack-class identification, and CNN-based spatial refinement.

Figure 2 shows the original concrete surface image together with the SOM-derived pseudo-label boundary overlay. The SOM operates directly on pixel-wise visual descriptors using the nine-dimensional feature representation defined in Section 2, without any manual annotation, threshold tuning, or pretrained models.

Figure 2. Original concrete crack image

Prior to clustering, PCA is applied to the standardized feature space to inspect intrinsic dimensionality. The cumulative explained variance exceeds 99% using five principal components, indicating a compact yet informative embedding for clustering. The number of SOM clusters is then selected via the elbow method applied to the within-cluster sum-of-squares (Figure 4).

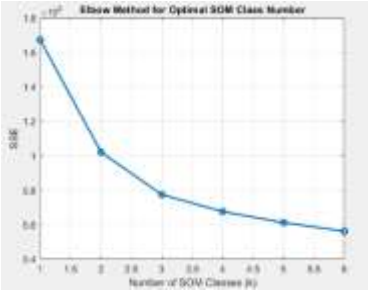


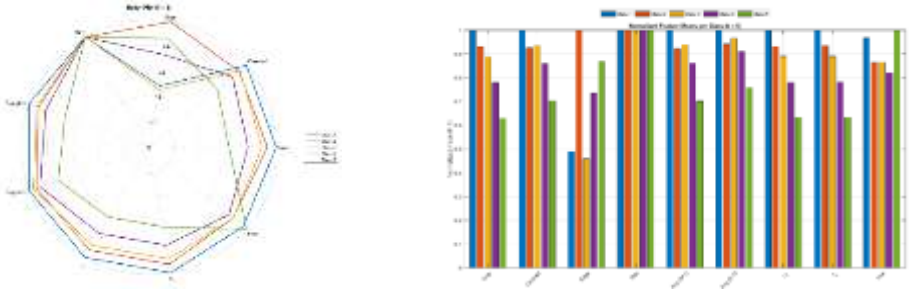
Figure 3. PCA dimensionality inspection and elbow-based selection of SOM class number for the concrete surface

With k = 5, the SOM produces a dense, pixel-wise semantic partition of the image. Figure 4(a) shows the SOM classification map, and Figure 4(b) displays the corresponding per-class binary maps, where one class forms a spatially coherent crack-like structure.

(a) SOM classification map (b) Binary maps for each SOM class

Figure 4. SOM clustering results on the concrete surface (k = 5)

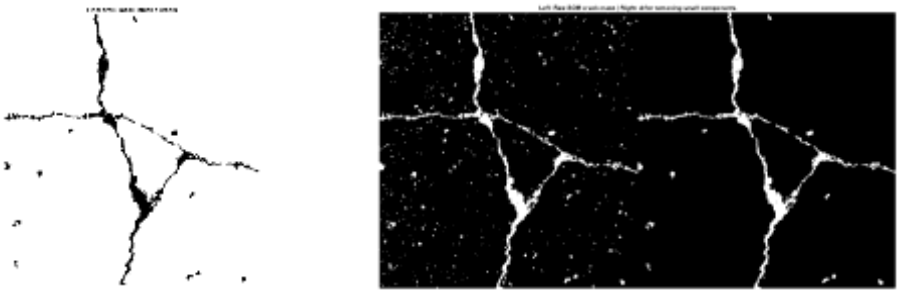
To support explainable crack-class identification, cluster-level feature responses are summarized using both a radar plot and a grouped bar chart, where cluster means are normalized by the feature-wise maxima across clusters (Figure 5). The crack-related cluster exhibits a distinctive structural signature, characterized by strong responses in edge/line-strength and thin-structure features, along with comparatively reduced grayscale intensity and characteristic neighborhood statistics—consistent with the morphology of surface cracks.



(a) Radar plot of normalized feature means for each SOM cluster (b) Grouped bar chart of normalized feature means (same normalization as the radar plot)

Figure 5. Interpretable cluster-level feature responses (k = 5)

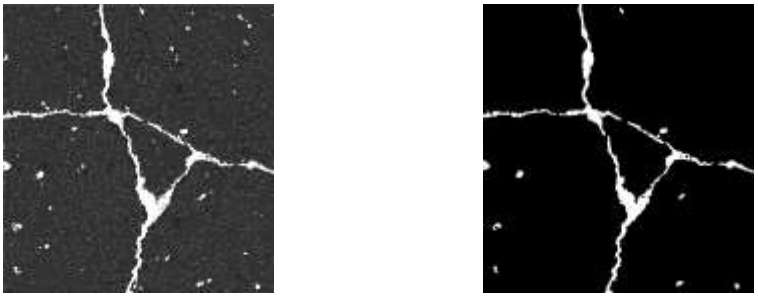
The crack class is automatically initialized using a minimum-intensity criterion (darkest cluster), and subsequently audited through multi-feature cluster signatures (radar and bar plots) to verify crack-consistent morphology (edge/line dominance). In this example, the selected crack class is Class 5 (Figure 6(a)). To suppress isolated noise while preserving spatial continuity, we apply connected-component filtering to the SOM crack mask: components smaller than 1% of the largest connected crack region are removed (Figure 6(b)).



(a) Automatically selected crack class (Class 5) (b) Raw SOM crack mask (left) and after removing small components (right), using a 1% threshold of the largest connected component

Figure 6. Automatic crack-class identification and SOM mask post-processing

To further improve spatial continuity and produce a smoother crack boundary, a lightweight U-Net is trained using the SOM-refined mask as pseudo-labels. Importantly, the CNN is not trained on any human-annotated ground truth; it only learns from SOM-generated pseudo-labels, serving as a smoothing/refinement module rather than a supervised detector. Figure 7(a) shows the CNN-predicted crack probability map, and Figure 7(b) shows the final CNN-refined crack mask after thresholding and connected-component area filtering (retain components $\geq$ 1% of the largest component).



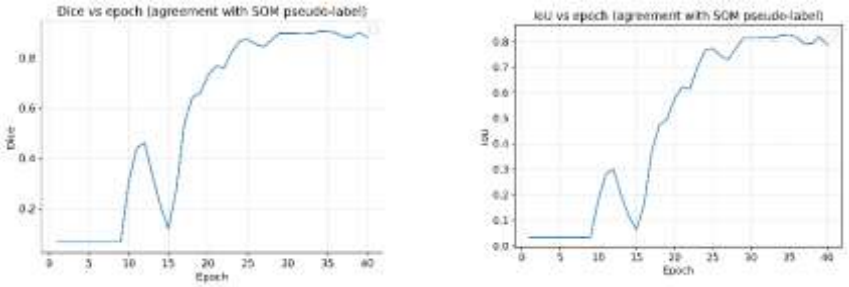
(a) CNN-predicted crack probability map learned from SOM pseudo-labels (b) Final CNN-refined crack mask (connected-component area filtering, retain ≥ 1% of the largest component)

Figure 7. SOM-guided CNN smoothing results (no manual labels)

For direct visual comparison, Figure 8 overlays the SOM boundary (green) and CNN boundary (red) on the original RGB image. The CNN output preserves the overall crack topology indicated by the SOM while reducing local jaggedness and suppressing isolated artifacts.

Figure 8. Boundary overlay comparison: SOM (green) vs CNN (red)

Finally, Figure 9 reports the CNN training dynamics measured against the SOM pseudo-labels (Dice and IoU), demonstrating stable convergence under this pseudo-supervised setting.



(a) Dice vs epoch (b) IoU vs epoch

Figure 9. CNN training curves under SOM pseudo-supervision: Dice and IoU vs epoch.

In terms of computational cost, the SOM stage took 54.39 s and the CNN refinement stage took 206.59 s for this concrete case (CPU), with an automatically selected PCA dimension of 5 and an elbow-selected SOM class number of k = 5.

3.2 Masonry Brick Wall: Robust Crack Isolation under Strong Texture Interference

This subsection evaluates the proposed SOM–CNN framework on a masonry brick wall image (551×1055 pixels), which represents a more challenging field scenario dominated by strong repetitive texture and mortar–brick contrast. Such texture interference frequently induces false positives for traditional edge- or threshold-based methods. Here, the goal is to test whether the unsupervised clustering stage can separate crack-like structures from texture-dominated patterns, and whether the SOM-guided CNN can further enhance spatial continuity without any manual labels or parameter tuning.

Figure 10. Original brick crack image

Figure 10 shows the original brick wall image together with the SOM-derived pseudo-label boundary overlay. As in the concrete case, the SOM operates directly on pixel-wise visual descriptors using the nine-dimensional feature representation defined in Section 2. Prior to clustering, PCA is applied to the standardized feature space to examine intrinsic dimensionality

and provide a compact embedding for clustering. The SOM class number is then selected via the elbow method on within-cluster sum-of-squares (Figure 11).

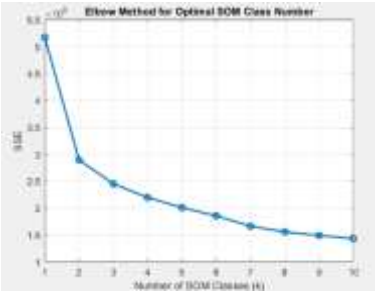
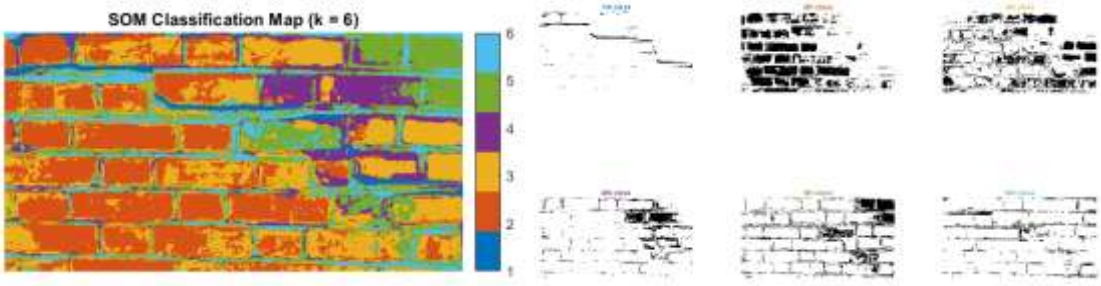


Figure 11. PCA dimensionality inspection and elbow-based selection of SOM class number for the brick wall surface

Using the elbow-selected class number, the SOM produces a dense pixel-wise semantic partition of the brick wall image. Figure 12(a) shows the SOM classification map, and Figure 12(b) displays the corresponding per-class binary maps. Despite the strong background texture, one SOM class forms a spatially coherent, line-like structure consistent with crack morphology, while other classes primarily capture brick textures, mortar joints, or illumination variations.

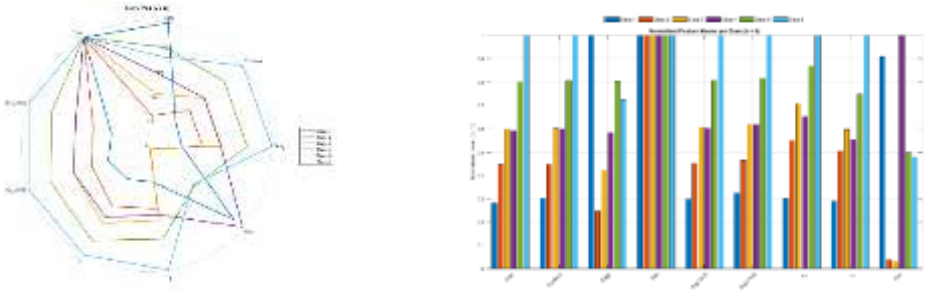


(a) SOM classification map (b) Binary maps for each SOM class

Figure 12. SOM clustering results on the brick wall surface (k = 6)

To support explainable crack-class identification, Figure 13 summarizes the cluster-level feature responses using the same radar plot and grouped bar chart visualization (cluster means normalized by the feature-wise maxima across clusters). The crack-related cluster exhibits a distinct structural signature, typically characterized by pronounced responses in edge/line-strength and thin-structure

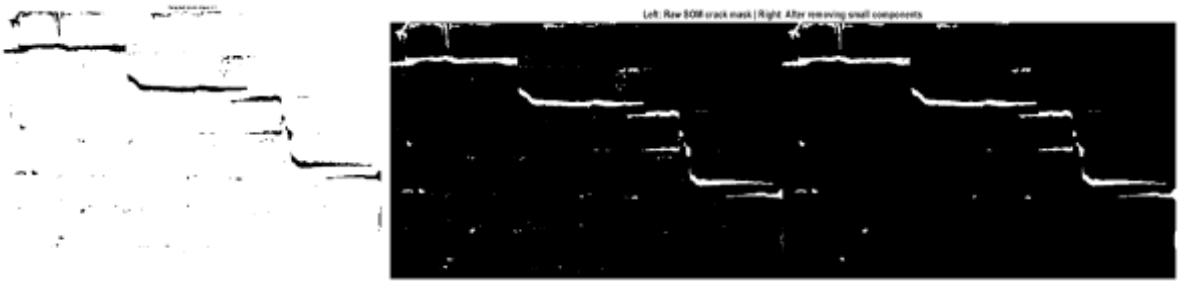
descriptors, together with comparatively lower grayscale intensity and consistent neighborhood statistics, distinguishing it from texture-driven clusters.



(a) Radar plot of normalized feature means for each SOM cluster (b) Grouped bar chart of normalized feature means (same normalization as the radar plot)

Figure 13. Interpretable cluster-level feature responses (k = 6)

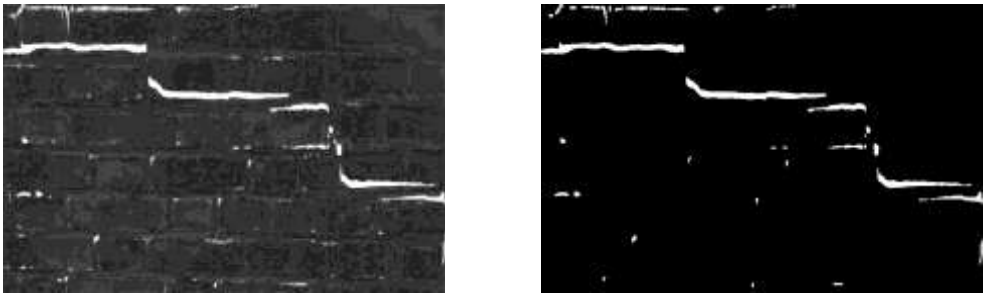
The crack class is automatically initialized using a minimum-intensity criterion (darkest cluster), and subsequently audited through multi-feature cluster signatures (radar and bar plots) to verify crack-consistent morphology (edge/line dominance). To suppress isolated noise while preserving spatially continuous crack segments, we apply the same connected-component filtering strategy as in Section 3.1: components smaller than 1% of the largest connected crack region are removed (Figure 14). This step reduces scattered texture-induced fragments while retaining the dominant crack structure.



(a) Automatically selected crack class (Class 1) (b) Raw SOM crack mask (left) and after removing small components (right), using a 1% threshold of the largest connected component

Figure 14. Automatic crack-class identification and SOM mask post-processing

To further improve spatial continuity and produce a smoother crack boundary, a lightweight U-Net is trained using the filtered SOM crack mask as pseudo-labels. Importantly, the CNN is not trained on any human-annotated ground truth; it only learns from SOM-generated pseudo-labels and serves as a smoothing/refinement module rather than a supervised detector. Figure 15(a) shows the CNN-predicted crack probability map, and Figure 15(b) shows the final CNN-refined crack mask after thresholding and connected-component area filtering (≥ 1% of the largest connected component).



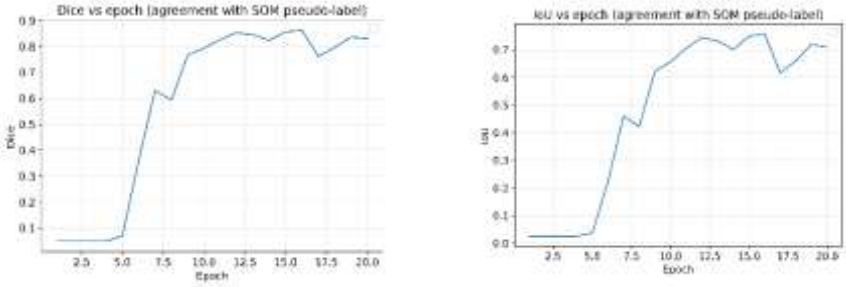
(a) CNN-predicted crack probability map learned from SOM pseudo-labels (b) Final CNN-refined crack mask (connected-component area filtering, retain ≥ 1% of the largest component)

Figure 15. SOM-guided CNN smoothing results (no manual labels)

For direct visual comparison, Figure 16 overlays the SOM boundary (green) and CNN boundary (red) on the original RGB image. The CNN output preserves the overall crack topology indicated by the SOM while reducing local jaggedness and suppressing isolated artifacts induced by brick texture.

Figure 16. Boundary overlay comparison: SOM (green) vs CNN (red)

Finally, Figure 17 reports the CNN training dynamics measured against the SOM pseudo-labels (Dice and IoU), demonstrating stable convergence under this pseudo-supervised setting.



(a) Dice vs epoch (b) IoU vs epoch

Figure 17. CNN training curves under SOM pseudo-supervision: Dice and IoU vs epoch.

In terms of computational cost, the SOM stage took 389.21 s and the CNN refinement stage took 444.70 s for this brick wall case (CPU), with an automatically selected PCA dimension of 6 and an elbow-selected SOM class number of k = 6.

3.3 Shadow-Affected Surface: Suppressing Illumination-Induced "False Crack" Patterns

This subsection evaluates the proposed SOM–CNN framework on a shadow-affected surface image (323×275 pixels), where illumination discontinuities and cast shadows create prominent "false crack" patterns. This case is intentionally challenging because shadow boundaries can exhibit strong contrast and elongated edges that resemble cracks, often leading to false positives in classical edge- or threshold-based methods. The objective here is to demonstrate that the proposed unsupervised clustering can separate illumination artifacts from true crack-like structures, and that the SOM-guided CNN can further enhance spatial continuity without any manual labels or parameter tuning.

Figure 18. Original shadow crack image

Figure 18 shows the original shadow-affected image together with the SOM-derived pseudo-label boundary overlay. As in the previous cases, the SOM operates on pixel-wise visual descriptors using the same nine-dimensional feature representation defined in Section 2. Prior to clustering, PCA is applied to the standardized feature space to inspect intrinsic dimensionality and provide a compact embedding, and the SOM class number is selected via the elbow method on within-cluster sum-of-squares (Figure 19).

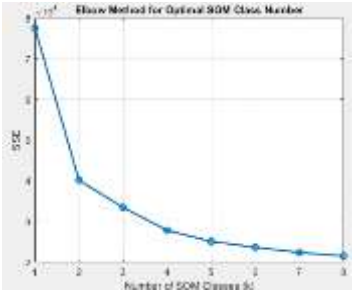


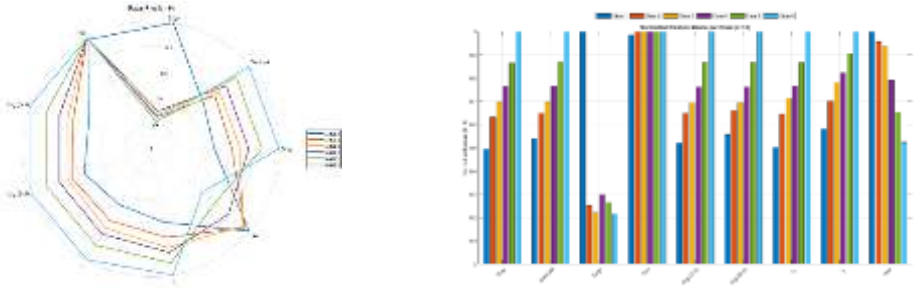
Figure 19. PCA dimensionality inspection and elbow-based selection of SOM class number for the shadow crack

Using the elbow-selected class number, the SOM produces a dense pixel-wise semantic partition of the image. Figure 20(a) shows the SOM classification map, and Figure 20(b) displays the corresponding per-class binary maps. In this shadow-affected scenario, multiple clusters may capture illumination transitions and shadow regions; however, the crack-like cluster remains distinguishable by its thin, line-structured geometry and localized connectivity, in contrast to the broader and more region-like shadow clusters.

(a) SOM classification map (b) Binary maps for each SOM class

Figure 20. SOM clustering results on the shadow crack (k = 6)

To support explainable crack-class identification, Figure 21 summarizes the cluster-level feature responses using the radar plot and grouped bar chart visualization (cluster means normalized by the feature-wise maxima across clusters). Compared with shadow-driven clusters, the crack-related cluster typically exhibits stronger responses in edge/line-strength and thin-structure descriptors, together with characteristic neighborhood statistics consistent with narrow, filamentary structures.

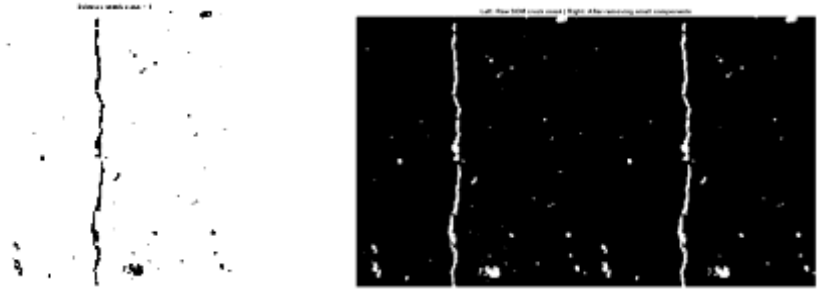


(a) Radar plot of normalized feature means for each SOM cluster (b) Grouped bar chart of normalized feature means (same normalization as the radar plot)

Figure 21. Interpretable cluster-level feature responses (k = 6)

The crack class is automatically initialized using a minimum-intensity criterion (darkest cluster), and subsequently audited through multi-feature cluster signatures (radar and bar plots) to verify crack-consistent morphology (edge/line dominance). To suppress isolated noise while preserving spatial continuity, we apply the same connected-component filtering strategy: components smaller

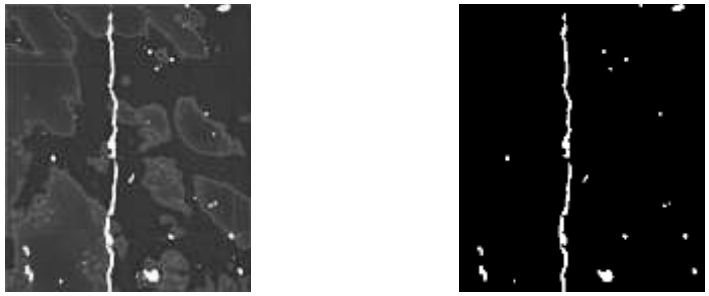
than 1% of the largest connected crack region are removed (Figure 22). This step helps reduce scattered fragments induced by shadow edges while retaining the dominant crack structure.



(a) Automatically selected crack class (Class 1) (b) Raw SOM crack mask (left) and after removing small components (right), using a 1% threshold of the largest connected component

Figure 22. Automatic crack-class identification and SOM mask post-processing

To further improve spatial continuity and produce a smoother crack boundary, a lightweight U-Net is trained using the filtered SOM crack mask as pseudo-labels. Importantly, the CNN is not trained on any human-annotated ground truth; it only learns from SOM-generated pseudo-labels and serves as a smoothing/refinement module rather than a supervised detector. Figure 23(a) shows the CNN-predicted crack probability map, and Figure 23(b) shows the final CNN-refined crack mask after thresholding and connected-component area filtering (≥ 1% of the largest connected component).



(a) CNN-predicted crack probability map learned from SOM pseudo-labels (b) Final CNN-refined crack mask (connected-component area filtering, retain ≥ 1% of the largest component)

Figure 23. SOM-guided CNN smoothing results (no manual labels)

For direct visual comparison, Figure 24 overlays the SOM boundary (green) and CNN boundary (red) on the original RGB image. The CNN output preserves the overall crack topology indicated by the SOM while reducing jaggedness and suppressing illumination-induced artifacts.

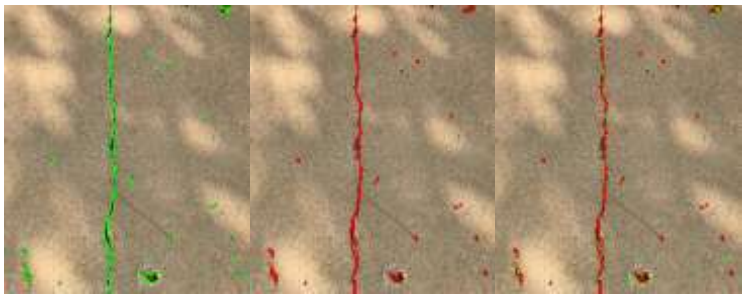
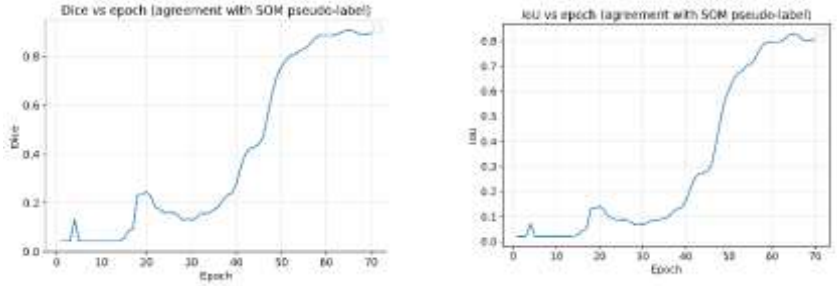


Figure 24. Boundary overlay comparison: SOM (green) vs CNN (red)

Finally, Figure 25 reports the CNN training dynamics measured against the SOM pseudo-labels (Dice and IoU), demonstrating stable convergence under this pseudo-supervised setting.



(a) Dice vs epoch (b) IoU vs epoch

Figure 25. CNN training curves under SOM pseudo-supervision: Dice and IoU vs epoch.

In terms of computational cost, the SOM stage took 82.89 s and the CNN refinement stage took 169.85 s for this shadow-affected case, with an automatically selected PCA dimension of 6 and an elbow-selected SOM class number of k = 6.

3.4 Discussion

Compared with our prior work [26], this study upgrades the SOM-based crack segmentation framework into a fully automatic pipeline by eliminating two manual bottlenecks for field

deployment: pre-setting the number of clusters and GUI-based crack-class selection. Automation is achieved through (1) PCA-based automatic dimension selection, (2) elbow-based automatic class-number selection, and (3) an intensity-based crack-class rule (darkest-cluster selection), followed by a fixed connected-component area filter to suppress scattered fragments. Radar/bar plots are used as post-hoc XAI verification to audit whether the automatically selected class exhibits crack-consistent signatures, rather than as an interactive decision step.

Because the pipeline operates in a strictly label-free setting without pixel-wise ground truth, we evaluate internal consistency between SOM pseudo-labels and CNN refinement outputs, and discuss what these agreement metrics imply for stability and field usability.

Unlike our prior study [26], which benchmarked segmentation accuracy using a grid-based reference protocol, the present work targets a strictly label-free deployment setting where such external references are unavailable. Therefore, Dice/IoU are reported as internal consistency metrics measuring CNN-to-SOM agreement, to evaluate spatial coherence and refinement stability. Future work with annotated field datasets will enable direct accuracy benchmarking between manual and fully automatic workflows.

Runtime note. The reported CPU runtime (4–14 minutes per image) includes a one-time, per-image self-refinement training stage for the optional U-Net regularizer (in addition to SOM inference). This design targets inspection-time and edge-oriented workflows (e.g., bridge/building surveys) where removing human-in-the-loop decisions and improving boundary regularity outweigh millisecond-level latency. When faster turnaround is preferred, the deterministic SOM-only output provides an immediate label-free baseline.

Table 1 summarizes internal consistency (CNN vs SOM) and computational cost across three representative field conditions. The CNN achieves high pseudo-label agreement at the best epoch

(Dice(F1) = 0.9046/0.8620/0.9063 for concrete/brick/shadow), indicating that SOM-generated masks are spatially coherent and learnable by a spatial model. Final-epoch agreement remains stable (Dice(F1) = 0.8804/0.7687/0.8949), supporting the interpretation that the CNN functions as a controlled boundary regularizer rather than a semantic re-classifier.

Table 1. Internal consistency metrics and computational cost: Dice/IoU quantify CNN-to-SOM pseudo-label agreement (not accuracy against ground truth), since no pixel-wise annotations are available in this label-free evaluation.

Case	PCA dim	SOM time (s)	CNN time (s)	Best epoch	Dice(F1)_best	IoU_best	Prec_best	Dice(F1)_final	IoU_final
Concrete	5	54.39 s	206.59 s	34	0.9046	0.8257	0.8397	0.8804	0.7864
Brick	6	389.21 s	444.70 s	16	0.8620	0.7575	0.8411	0.7687	0.6243
Shadow	6	82.89 s	169.85 s	65	0.9063	0.8287	0.9072	0.8949	0.8098

High Dice/IoU values in Table 1 have a clear interpretation in this label-free setting: (i) the SOM pseudo-labels are geometrically plausible and thus learnable by a spatial model, (ii) the CNN refinement remains anchored to SOM semantics rather than re-partitioning the scene, and (iii) the overall pipeline behaves reproducibly across materials without parameter tuning. Therefore, Dice/IoU are used here as internal consistency indicators that the unsupervised clustering yields spatially coherent crack masks suitable for field workflows where pixel-wise labels are impractical.

A key observation is that the CNN consistently learns a smooth spatial prior without drifting away from the unsupervised semantics provided by SOM. This is non-trivial under real-world conditions: surface texture, lighting changes, and background clutter can easily cause a deep model trained without labels to "invent" patterns. Here, the refinement remains anchored to SOM pseudo-labels, suggesting that the CNN primarily performs boundary regularization, gap filling, and suppression of isolated fragments, while preserving the crack topology indicated by the unsupervised clustering. The "best-epoch vs final-epoch" behavior further clarifies the refinement mechanism. For instance, in the shadow-affected case, Dice(F1) peaks at 0.9063 (IoU 0.8287) and remains high at the final epoch (Dice 0.8949, IoU 0.8098), indicating that the refinement is robust even under illumination-induced ambiguity. In contrast, the brick case shows a larger drop from peak (Dice 0.8620, IoU 0.7575) to final (Dice 0.7687, IoU 0.6243), which is consistent with the intuition that strong repetitive textures allow the CNN to gradually over-smooth and slightly deviate from the pixel-level pseudo-label boundary. Importantly, this does not indicate instability or divergence; rather, it reflects the refinement module's tendency to converge toward a more consolidated geometric representation than the original pseudo-label. Practically, this motivates a transparent operational choice: selecting the peak Dice(F1) epoch when maximal adherence to SOM pseudo-labels is desired, or using later epochs when smoother boundaries and stronger consolidation are preferred. Validation of automatic crack-class selection is provided through the XAI signatures: across the three cases, the automatically selected crack cluster consistently exhibits crack-consistent feature profiles in the radar/bar plots (low intensity-related features with elevated edge/thin responses), supporting that the intensity-based rule correctly identifies the crack class in the tested conditions. This verification step is critical because it provides an auditable check even when no pixel-wise labels exist.

Although the refinement stage includes one-time per-image self-training, the resulting 4–14 min per image runtime remains compatible with batch inspection and offline structural survey workflows, where eliminating human intervention and ensuring reproducible decision chains outweigh millisecond-level latency.

From a reviewer's perspective, two limitations must be stated explicitly.

First, pseudo-label agreement is not ground truth accuracy. A high Dice(F1)/IoU between CNN and SOM guarantees that pseudo-supervision is learnable and stable; it does not prove the correctness of the pseudo-label itself. This is precisely why the SOM stage is designed to be interpretable: cluster-level feature profiles (radar/bar plots) enable practitioners to audit whether the crack cluster is physically plausible (thin, line-structured, edge-consistent) rather than relying on a black-box deep model. In the intended setting—where dense labels are unavailable—the priority is an auditable pipeline whose intermediate decisions can be inspected and corrected, instead of a single supervised score on a curated dataset.

Second, the current implementation uses an automatic crack-class rule based on minimum mean grayscale intensity ("darkest cluster"), which is effective for many surface-crack scenarios but not universally valid. Cracks can appear light due to chalking/efflorescence, dust filling, overexposure, or reflective coatings; conversely, non-crack regions can be dark due to stains, wet patches, or shadow boundaries. In such cases, an intensity-only selector can misidentify the crack class even when the underlying SOM clustering remains meaningful. This limitation is not a weakness of the SOM–CNN framework itself, but rather of one specific selection heuristic.

A practical extension is to replace the single-feature "darkest cluster" selector with a multi-criterion crackness score that combines interpretable cues already available in the SOM feature set (e.g., thin/edge responses, local contrast statistics, and simple geometric descriptors). This would

improve robustness in cases where cracks are not the darkest regions (e.g., efflorescence-filled cracks) or where non-crack regions appear dark (e.g., stains or strong shadows), without reintroducing manual intervention.

Failure modes. The intensity-driven crack-class rule can fail when non-crack dark regions dominate the image (e.g., deep shadows, stains, black markings, or very dark aggregates). In such cases, the selected cluster may violate expected crack morphology. While the present work keeps the selection rule fully deterministic, we provide cluster signatures (radar/bar summaries) to audit the decision and to flag suspicious selections (e.g., low edge/line response and weak thinness cues). Future work will incorporate an explicit rejection/flagging criterion based on these signatures (e.g., minimum edge/thinness thresholds or a template-distance check) to trigger "no-decision" rather than returning a potentially incorrect mask.

Relatedly, the connected-component filtering step (removing components smaller than 1% of the largest region) is a pragmatic noise suppressor but can fail when cracks are genuinely fragmented (e.g., multiple small cracks or microcracking) or when the dominant connected component corresponds to a non-crack structure (e.g., a shadow boundary). This can be improved without labels by replacing the area-based threshold with an adaptive, geometry-aware filter that retains components with high elongation or long skeleton length even if their area is small, thereby better preserving multiple crack branches while still rejecting texture speckles.

In particular, when multiple thin cracks are spatially disconnected (e.g., parallel hairline cracking), a fixed area-based criterion may remove smaller but physically meaningful crack segments. Future work can replace the relative-area rule with a geometry-aware filter that retains elongated or high-eccentricity components even when their area is small.

Finally, elbow-based cluster-number selection can be ambiguous in some images. Nevertheless, the pipeline's stability is not solely dependent on a single k choice, because downstream crack selection is based on cluster semantics and morphology rather than manual tuning. Future work can strengthen this step via stability checks over multiple SOM initializations or a simple consensus criterion, still without requiring ground truth.

Taken together, the results support the main engineering value of the proposed method: it provides a label-free yet auditable chain of decisions (features → clustering → crack-class selection → refinement) where failure modes are identifiable and correctable without transitioning to full supervised training. This is particularly relevant for structural health monitoring in the field, where distribution shifts (material texture, illumination, staining, weathering) are unavoidable and interpretability of intermediate decisions is often as important as the final segmentation output.

More broadly, the proposed pipeline is physics-agnostic at the decision-chain level: it couples interpretable local descriptors, unsupervised partitioning, and deterministic post-processing to produce structure-preserving masks without pixel-wise supervision, and is readily transferable to other surface-driven inspection tasks where structure-preserving segmentation is needed.

4. Conclusion

This study moves SOM-based crack segmentation from an unsupervised yet manually steered workflow toward a fully automatic and auditable pipeline that is realistic for inspection settings. Using the same nine interpretable pixel descriptors, the proposed methodology removes two recurring barriers to practical adoption: subjective decisions on feature-space dimensionality and cluster number, and manual crack-class selection by visual trial-and-error. The resulting workflow is end-to-end and deterministic. It integrates PCA with automatic dimension retention, elbow-based selection of the number of clusters, automatic crack-class identification, and fixed post-

processing. An explainability layer (radar and bar signatures) is retained throughout, and an optional pseudo-label CNN refines boundaries as a spatial regularizer without requiring human annotations, pretrained models, or ground-truth supervision.

Key findings — The pipeline successfully transitioned crack detection from expert-dependent labeling to data-driven automated discovery without sacrificing interpretability. Across three representative conditions (aggregate-textured concrete under clean lighting, texture-dominated masonry, and shadow-affected imagery), the PCA stage adapted the retained dimension to scene complexity (5 PCs for concrete and 6 PCs for brick and shadow). The elbow rule similarly produced material-dependent cluster counts (e.g., $k = 5$ for concrete and $k = 6$ for brick and shadow), avoiding manual tuning while keeping the clustering compact and explainable. The CNN refinement behaved as intended. It did not introduce a new semantic classifier, but instead provided a controlled smoothing mechanism anchored to SOM pseudo-label semantics, which is reflected by high SOM–CNN internal consistency (best-epoch Dice $\approx 0.905/0.862/0.906$ for concrete/brick/shadow). From a computational perspective, the full workflow is practical on a standard CPU-only workstation. Even when including CNN training for refinement, the total processing time remains on the order of minutes per image (approximately $\sim$ 4–14 minutes in our experiments), supporting inspection-time use without GPUs or data labeling. This CPU-only characteristic is also well suited to edge deployment in field inspection platforms with constrained onboard compute and power budgets (e.g., drones or climbing robots). The proposed pipeline exhibits consistent behavior across heterogeneous textures and illumination, suggesting that the interpretable signatures primarily reflect crack-like morphology rather than scene-specific artifacts.

Limitations — Two constraints warrant explicit discussion. First, internal consistency between CNN and SOM is not the same as ground-truth accuracy. Strong Dice/IoU values indicate that the

pseudo-supervision is stable and learnable, but they cannot substitute for pixel-level ground-truth validation in terms of absolute accuracy. This is why the paper emphasizes auditable cluster signatures rather than relying on black-box performance claims. Second, the current automatic selector based on the darkest cluster is physically motivated for many surface cracks but not universal. Cracks can appear bright due to efflorescence, dust filling, or exposure effects, while non-crack regions can be dark due to stains, wet patches, or cast shadows. These cases can produce failure modes even when the underlying clustering remains meaningful. Finally, the evaluation is intentionally label-free and focused on representative conditions (aggregate-textured concrete under clean lighting, texture-dominated masonry, and shadow-affected surfaces) to demonstrate automation, explainability, and robustness under characteristic field nuisances. Large-scale validation across diverse materials, surface states, and aging conditions remains essential for comprehensive benchmarking.

Future work — The most direct improvement is to strengthen the label-free crack decision rule itself. A natural next step is to replace the single-criterion darkest-cluster heuristic with a multi-criterion crackness score that combines cues already present in the interpretable feature set, such as thin/edge responses, local contrast statistics, and lightweight geometry or connectivity measures. In addition, the connected-component filter can be extended beyond area-only retention to geometry-aware filtering (e.g., elongation or skeleton length), which would reduce the risk of discarding fragmented microcracks or multiple disconnected thin cracks. The elbow step can also be made more stable through lightweight consistency checks across adjacent k values or repeated SOM initializations, still without labels. Ultimately, large-scale field validation across broader materials and surface conditions, including coatings, stains, wetness, and diverse illumination

regimes, is the key step toward standardizing this approach as a practical and transparent inspection module under real-world distribution shifts.

References

[1] P. Kumar, A. Sharma, and S. R. Kota, "Automatic Multiclass Instance Segmentation of Concrete Damage Using Deep Learning Model," *IEEE Access*, vol. 9, pp. 90330–90345, 2021, doi: 10.1109/ACCESS.2021.3090961.

[2] M. Memari, P. Shakya, M. Shekaramiz, A. C. Seibi, and M. A. S. Masoum, "Review on the Advancements in Wind Turbine Blade Inspection: Integrating Drone and Deep Learning Technologies for Enhanced Defect Detection," *IEEE Access*, vol. 12, pp. 33236–33282, 2024, doi: 10.1109/ACCESS.2024.3371493.

[3] J. M. G. Payawal and D.-K. Kim, "A Review on the Latest Advancements and Innovation Trends in Vibration-Based Structural Health Monitoring (SHM) Techniques for Improved Maintenance of Steel Slit Damper (SSD)," *IEEE Access*, vol. 12, pp. 44383–44400, 2024, doi: 10.1109/ACCESS.2024.3381625.

[4] Y. Cha, W. Choi, and O. Büyüköztürk, "Deep learning-based crack damage detection using convolutional neural networks," *Comput. Civ. Infrastruct. Eng.*, vol. 32, no. 5, pp. 361–378, 2017.

[5] L. Zhang, F. Yang, Y. D. Zhang, and Y. J. Zhu, "Road crack detection using deep convolutional neural network," presented at the 2016 IEEE international conference on image processing (ICIP), IEEE, 2016, pp. 3708–3712.

[6] G. A. Pathrikar, J. Bharti, and B. Roy, "A Review of Deep Learning-Based Methods for Crack Detection and Segmentation in Civil Infrastructure," in *2025 International Conference on Engineering Innovations and Technologies (ICoEIT)*, 2025, pp. 95–101. doi: 10.1109/ICoEIT63558.2025.11211576.

[7] J. Deng, A. Singh, Y. Zhou, Y. Lu, and V. C.-S. Lee, "Review on computer vision-based crack detection and quantification methodologies for civil structures," *Constr. Build. Mater.*, vol. 356, p. 129238, Nov. 2022, doi: 10.1016/j.conbuildmat.2022.129238.

[8] H. Kaveh and R. Alhajj, "Recent advances in crack detection technologies for structures: a survey of 2022-2023 literature," *Front. Built Environ.*, vol. 10, p. 1321634, 2024.

[9] H. Liu *et al.*, "Deep Domain Adaptation for Pavement Crack Detection," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 2, pp. 1669–1681, 2023, doi: 10.1109/TITS.2022.3225212.

[10] J. L. Walker, Z. Zeng, C. L. Wu, J. S. Jaffe, K. E. Frasier, and S. S. Sandin, "Underwater Object Detection Under Domain Shift," *IEEE J. Ocean. Eng.*, vol. 49, no. 4, pp. 1209–1219, 2024, doi: 10.1109/JOE.2024.3425453.

[11] M. M. Azad and H. S. Kim, "An explainable artificial intelligence-based approach for reliable damage detection in polymer composite structures using deep learning," *Polym. Compos.*, vol. 46, no. 2, pp. 1536–1551, 2025.

[12] H. El Mir and S. Perinpanayagam, “Certification of machine learning algorithms for safe-life assessment of landing gear,” *Front. Astron. Space Sci.*, vol. 9, p. 896877, 2022.

[13] M. M. Shamszadeh, K. Kumar, A. Ferche, O. Bayrak, and S. Salamone, “Explainable Boosting Machine for Structural Health Assessment: An Interpretable Approach to Data-Driven Structural Assessment,” *Proc. 15th Int. Workshop Struct. Health Monit.*, 2025, [Online]. Available: https://api.semanticscholar.org/CorpusID:282231226

[14] M. Jambulkar, S. Ojha, A. Shelke, and A. Habib, “Autonomous defect estimation in aluminum plate and prognosis through stochastic process modeling,” *Sci. Rep.*, vol. 15, no. 1, p. 29549, 2025.

[15] A. S. Abosinnee, G. Bencsik, and F. Abedi, “Edges in image with illumination variations scenarios: a review,” *Vis. Comput.*, vol. 41, no. 14, pp. 12277–12305, Nov. 2025, doi: 10.1007/s00371-025-04157-4.

[16] J. König, M. D. Jenkins, M. Mannion, P. Barrie, and G. Morison, “Weakly-Supervised Surface Crack Segmentation by Generating Pseudo-Labels Using Localization With a Classifier and Thresholding,” *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 12, pp. 24083–24094, 2022, doi: 10.1109/TITS.2022.3204853.

[17] Z. Pan, S. L. H. Lau, X. Yang, N. Guo, and X. Wang, “Automatic pavement crack segmentation using a generative adversarial network (GAN)-based convolutional neural network,” *Results Eng.*, vol. 19, p. 101267, Sep. 2023, doi: 10.1016/j.rineng.2023.101267.

[18] A. Sepidbar and M. Sabouri, “Asphalt Pavement Crack Detection Using Image-to-Image Translation,” *Sci. Iran.*, 2024.

[19] Q. Sun, R. Zheng, and B. Xu, “Bayesian-optimized unsupervised semantic segmentation model for structural crack detection,” *Adv. Struct. Eng.*, vol. 27, no. 3, pp. 432–449, 2024, doi: 10.1177/13694332231222093.

[20] S. Bhattacharya, C. Zhang, D. K. Mishra, M. M. Yuen, and J. Zhang, “Efficient unsupervised domain adaptation for crack segmentation with interpretable Fourier–Morphology blending and Uncertainty-guided self-training,” *Comput. Civ. Infrastruct. Eng.*, vol. 40, no. 29, pp. 5790–5807, 2025.

[21] C. M. Hoang and B. Kang, “Pixel-level clustering network for unsupervised image segmentation,” *Eng. Appl. Artif. Intell.*, vol. 127, p. 107327, Jan. 2024, doi: 10.1016/j.engappai.2023.107327.

[22] A. L. Dai Pra *et al.*, “Dynamic speckle image segmentation using self-organizing maps,” *J. Opt.*, vol. 18, no. 8, p. 085606, 2016.

[23] T. Kohonen, *Self-organizing maps*, vol. 30. Springer Science & Business Media, 2012.

[24] T. Yamaguchi and T. Mizutani, “Road crack detection interpreting background images by convolutional neural networks and a self-organizing map,” *Comput. Civ. Infrastruct. Eng.*, vol. 39, no. 11, pp. 1616–1640, 2024.

[25] D. Sacha *et al.*, “SOMFlow: Guided Exploratory Cluster Analysis with Self-Organizing Maps and Analytic Provenance,” *IEEE Trans. Vis. Comput. Graph.*, vol. 24, no. 1, pp. 120–130, 2018, doi: 10.1109/TVCG.2017.2744805.

[26] X. Sun and P. Chang, “Unsupervised and noise-resilient crack detection in concrete and masonry via self-organizing maps,” *Eng. Struct.*, vol. 352, p. 122130, Apr. 2026, doi: 10.1016/j.engstruct.2026.122130.

[27] L. E. Campbell, R. J. Connor, J. M. Whitehead, and G. A. Washer, “Human factors affecting visual inspection of fatigue cracking in steel bridges,” *Struct. Infrastruct. Eng.*, vol. 17, no. 11, pp. 1447–1458, Oct. 2021, doi: 10.1080/15732479.2020.1813783.

[28] L. E. Ekemeyong Awong and T. Zielinska, “Comparative Analysis of the Clustering Quality in Self-Organizing Maps for Human Posture Classification,” *Sensors*, vol. 23, no. 18, 2023, doi: 10.3390/s23187925.

[29] Z. Hamida and J. Goulet, “Modeling infrastructure degradation from visual inspections using network-scale state-space models,” *Struct. Control Health Monit.*, vol. 27, no. 9, p. e2582, 2020.

[30] H. Abdi and L. J. Williams, “Principal component analysis,” *Wiley Interdiscip. Rev. Comput. Stat.*, vol. 2, no. 4, pp. 433–459, 2010.

[31] I. Jolliffe, “Principal component analysis,” in *International encyclopedia of statistical science*, Springer, 2011, pp. 1094–1096.

[32] J. Shlens, “A tutorial on principal component analysis,” *ArXiv Prepr. ArXiv14041100*, 2014.

[33] S. Wold, K. Esbensen, and P. Geladi, “Principal component analysis,” *Chemom. Intell. Lab. Syst.*, vol. 2, no. 1–3, pp. 37–52, 1987.

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

[35] T. Kohonen, “Essentials of the self-organizing map,” *Neural Netw.*, vol. 37, pp. 52–65, 2013.

[36] D. J. Ketchen and C. L. Shook, “The application of cluster analysis in strategic management research: an analysis and critique,” *Strateg. Manag. J.*, vol. 17, no. 6, pp. 441–458, 1996.

[37] T. M. Kodinariya and P. R. Makwana, “Review on determining number of Cluster in K-Means Clustering,” *Int. J.*, vol. 1, no. 6, pp. 90–95, 2013.

[38] S. Salvador and P. Chan, “Determining the number of clusters/segments in hierarchical clustering/segmentation algorithms,” presented at the 16th IEEE international conference on tools with artificial intelligence, IEEE, 2004, pp. 576–584.

[39] V. Satopaa, J. Albrecht, D. Irwin, and B. Raghavan, “Finding a" kneedle" in a haystack: Detecting knee points in system behavior,” presented at the 2011 31st international conference on distributed computing systems workshops, IEEE, 2011, pp. 166–171.

[40] R. L. Thorndike, “Who belongs in the family?,” *Psychometrika*, vol. 18, no. 4, pp. 267–276, 1953.