Material-Condition Fingerprints Enable Scalable, Zero-Label SOM Crack Segmentation via Weight-Vector Transfer

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 label-free and interpretable crack segmentation by clustering pixels in a physically meaningful descriptor space, but their practical use is constrained by a key scalability bottleneck: a SOM must be newly formed for every image. This work removes that bottleneck by introducing a within-material weight-vector transfer mechanism that calibrates once and deploys many times under consistent surface conditions. A single SOM is formed on a representative reference image in the nine-dimensional interpretable descriptor space established in our prior work, and its prototype vectors and internal normalization parameters are stored as a compact material-condition fingerprint. For subsequent images of the same material, segmentation is performed through deterministic nearest-prototype assignment, eliminating per-image SOM formation.

Experiments on aggregate-textured concrete (three classes) and texture-dominated masonry (five classes) show that the transferred SOM closely matches Direct-SOM behavior under strictly zero-label evaluation while yielding substantial runtime reductions that enable full-resolution batch processing. Beyond efficiency, transfer stabilizes decision geometry across repeated inspections

and enables rapid screening of crack-free controls without introducing systematic false positives. An auxiliary U-Net refinement trained solely on SOM-generated pseudo-labels further demonstrates that transferred pseudo-labels are sufficiently stable to support downstream spatial regularization without manual annotation.

The proposed mechanism strengthens three deployment-facing advantages—computational efficiency, operational consistency, and zero-label overhead—and highlights limitations under illumination shifts and cross-material use, where re-calibration or condition-specific prototype libraries may be required.

Keywords: Crack segmentation; Self-organizing map; Unsupervised learning; Within-material transfer; Infrastructure inspection; Zero-label workflow

1. Introduction

Crack segmentation is a fundamental step in visual inspection for civil infrastructure [1–3], yet many high-performing solutions remain difficult to deploy at scale [4–8]. Fully supervised deep models typically require large volumes of pixel-level annotations and compute-intensive training, which introduces substantial cost and limits practical adoption in routine inspection workflows [9–11]. Even when accuracy is high, field deployment often prioritizes predictable runtime [12–14] and consistent decision logic across large image collections [15–17]—such as UAV batch surveys or repeated monitoring of the same structural component—rather than re-optimizing a model separately for every image [18–21].

To reduce labeling dependence, our prior work established a training-free, pixel-level crack segmentation framework based on Self-Organizing Maps (SOMs) [22–24] operating in an interpretable descriptor space [25]. The SOM clusters pixels without human annotations, and a

deterministic interpretation rule extracts the crack-related class, yielding a fully label-free pipeline grounded in transparent feature-space evidence [25]. However, a key scalability bottleneck remains: Direct SOM must be formed per image. This is computationally expensive at high resolution and can also introduce **per-image fitting variability**, since each image's feature distribution can shift cluster boundaries, and random initialization may cause run-to-run differences even on the same image [15,26–28]. In practice, inspectors would prefer to "lock in" a reliable crack/non-crack separation once on a representative surface condition and then apply it consistently across subsequent images of the same material [29–33].

This paper addresses that practicality question with a within-material **weight-vector transfer** mechanism. The core idea is simple: calibrate the SOM once on a representative reference image, then deploy it repeatedly on an arbitrary number of target images of the same material condition. This "calibrate once, deploy N times (same material)" formulation transforms a per-image unsupervised analysis tool into a scalable, batch-ready workflow.

Technically, weight-vector transfer operates as follows. A SOM is formed once in the nine-dimensional interpretable descriptor space, and its prototype weight vectors together with the implementation's internal normalization are saved as a compact **material-condition fingerprint**. For any target image, segmentation becomes deterministic, training-free inference: each pixel descriptor is normalized using the saved parameters and assigned to the nearest prototype. Computationally, this replaces iterative per-image SOM formation with a lightweight nearest-prototype distance calculation, enabling batch deployment with stable decision geometry.

This formulation yields three deployment-facing benefits. First, it improves **Computational Efficiency**: after a one-time calibration, inference on each new image requires no per-image SOM formation and can run on CPU and edge hardware. Second, it promotes **Operational Consistency**:

fixed prototypes enforce a consistent crack decision logic across repeated inspections of the same material, supporting defensible reporting and reliable screening. Third, it enables **Zero-labeling overhead**: the entire pipeline remains unsupervised and label-free, avoiding the annotation burden that dominates many supervised crack segmentation workflows.

The main contributions of this study are:

1. **Within-material weight-vector transfer for unsupervised SOM segmentation**: we show that SOM prototypes learned on a reference image can be reused across images of the same material via nearest-prototype assignment, eliminating per-image SOM formation.
2. **Validation on concrete and masonry**: we demonstrate that transfer closely reproduces Direct-SOM behavior while substantially reducing runtime, turning SOM segmentation into a practical large-batch workflow.
3. **Stability analysis under CNN refinement**: we show that transferred pseudo-labels remain stable enough to support lightweight U-Net refinement without manual annotation, enabling label-free spatial regularization when desired.
4. **Deployment boundaries and implications**: we analyze when transfer is reliable (within-material, similar imaging regimes), when it breaks down (e.g., shadow-driven regime shifts or cross-material use), and how reference selection functions as a one-time calibration step.

We evaluate weight-vector transfer on two representative surface conditions validated in our prior work: aggregate-textured concrete with a three-class partition and texture-dominated masonry with a five-class partition [25]. Under this strictly zero-label setting, we quantify transfer consistency by comparing Direct-SOM outputs with transfer outputs using standard agreement metrics (precision, recall, F1-score, Dice, and IoU). In addition, we include an optional U-Net refinement

module trained purely on SOM-generated pseudo-labels as a lightweight spatial regularizer, used to probe whether transfer-based pseudo-labels remain stable enough to support downstream smoothing without manual annotation. The full pipeline is summarized in Figure 1.

2. Methodology

2.1 Overview: calibrate once, deploy N times (same material)

This study proposes a within-material weight-vector transfer mechanism for Self-Organizing Maps (SOMs) in fully unsupervised crack segmentation. The objective is to eliminate per-image SOM formation while preserving a label-free and interpretable decision chain in a fixed pixel-descriptor space. Given a representative reference image under a specific material condition, we (i) fit a SOM once in the interpretable descriptor space, (ii) save the resulting prototype weight vectors together with the implementation's internal feature normalization, and then (iii) apply the same prototypes to an arbitrary number of target images of the same material via deterministic nearest-prototype assignment. This "calibrate once, deploy N times" workflow yields predictable computational cost and consistent decision logic across inspection batches.

Two fully unsupervised settings are compared throughout:

Direct SOM (baseline): form a SOM independently on each image, then extract the crack class using the same deterministic rule.

Weight-vector transfer (proposed): form the SOM only once on a reference image, then reuse the learned prototypes to segment all other images without re-forming a SOM on each target.

An optional U-Net module is included only as a lightweight spatial regularizer trained on SOM-generated pseudo-labels; it introduces no human annotations and does not alter the core unsupervised evidence source.

Figure 1 summarizes the overall pipeline.

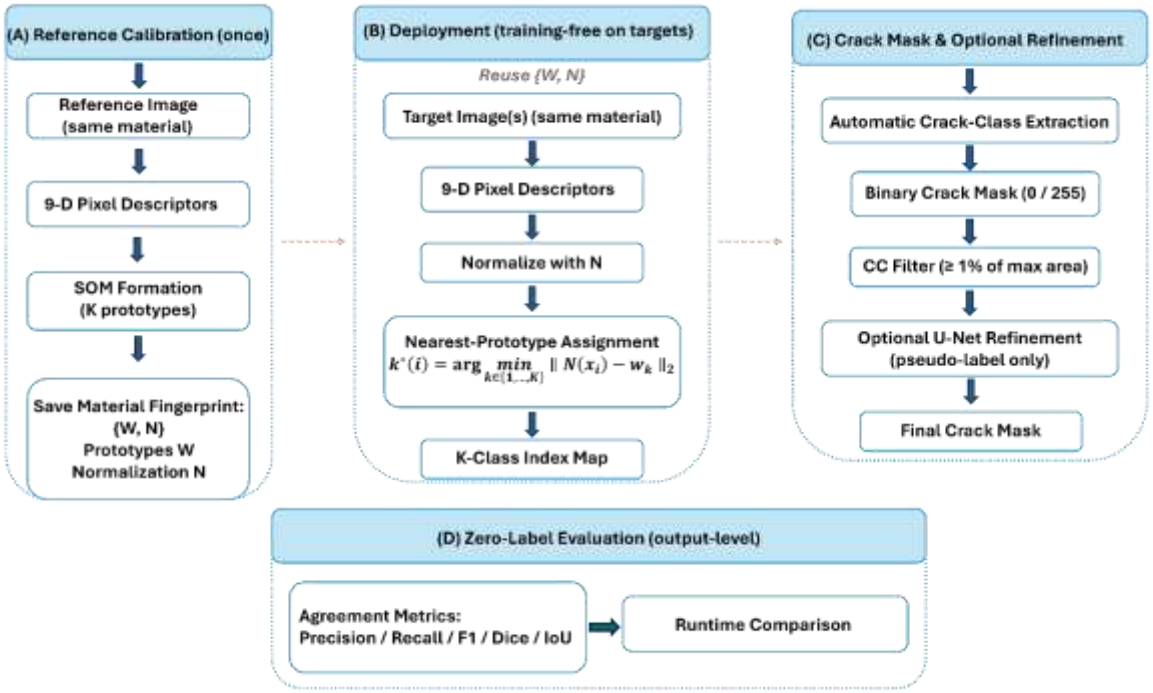


Figure 1. Flow chart of the proposed within-material weight-vector transfer pipeline

2.2 Interpretable pixel-wise descriptor space

For each RGB input image, each pixel is mapped to an interpretable nine-dimensional descriptor vector designed to capture crack-like morphology (thin/line/edge behavior), local contrast and intensity statistics, neighborhood texture structure, and color cue. The descriptor definition follows our prior Engineering Structures work to preserve interpretability and ensure that any performance differences arise from the transfer mechanism rather than feature redesign. Let $x_i \in \mathbb{R}^9$denote the descriptor for pixel i. Details of the exact feature definitions are provided in the prior work and are not repeated here to avoid redundancy.

2.3 SOM formation and prototype learning in descriptor space

Given a set of pixel descriptors from an image, SOM formation yields K prototype weight vectors $\{w_k\}_{k=1}^{K}$ that represent recurring pixel modes in the descriptor space. Conceptually, the SOM iteratively adjusts prototype locations via competitive learning: for each input descriptor, the best-matching unit (BMU)—the nearest prototype—and its topological neighbors are updated to better represent that region of feature space. After convergence, the prototypes partition the descriptor space into Voronoi regions, providing a prototype-based multi-class clustering of pixels.

Crucially, practical SOM implementations (including MATLAB's SOM toolbox) apply internal feature normalization or scaling before distance computation. To preserve the original decision geometry under transfer, we treat the reference SOM as a complete object consisting of: (i) prototype weight vectors $W = \{w_k\}$ and (ii) the normalization operator $N(\cdot)$ used internally prior to distance evaluation (e.g., per-dimension scaling). The pair (W, N) serves as a material-condition fingerprint: a compact representation of how the reference surface organizes pixels into semantically meaningful modes (crack, background texture, joints/residue, etc.) in the shared descriptor space.

2.4 Weight-vector transfer via nearest-prototype assignment

For a target image of the same material condition, no new SOM formation is performed. Instead, each target pixel descriptor x_i is normalized using the reference normalization $N(\cdot)$ and assigned to the nearest prototype in Euclidean distance:

$$k^*(i) = \arg \min_{k \in \{1,\ldots,K\}} \| N(x_i) - w_k \|_2 \quad (1)$$

This produces a full-resolution cluster-index map with the same K categories as the reference SOM. Computationally, this step is a deterministic distance evaluation performed once per pixel, in

contrast to the iterative competitive learning required by per-image SOM formation. As a result, within-material deployment becomes “training-free” on target images after a one-time reference calibration.

2.5 Material-specific class taxonomies (from the established SOM partitions)

Because different materials exhibit different structured confounders, we adopt the SOM class count K and the associated semantic partition following the established settings validated in the prior work:

Concrete (aggregate-textured): $K = 3$, corresponding to background concrete, residual noise/spot artifacts, and cracks.

Masonry (brick): $K = 5$, explicitly separating crack pixels from structured non-crack modes. The five classes are interpreted as (i) crack, (ii) brick texture, (iii) noise, (iv) residue/background variation (e.g., staining or illumination-induced intensity drift), and (v) mortar-joint class.

This material-aware taxonomy prevents repetitive brick patterns and mortar seams from being forced into a single “background” bucket, which is essential under texture-dominated masonry conditions.

2.6 Automatic crack-class extraction (label-free interpretation rule)

SOM outputs are multi-class by construction. For crack-focused evaluation and refinement, we extract a binary crack mask using the same deterministic, label-free rule as the prior framework. The crack class is identified as the cluster whose member pixels exhibit characteristic crack signatures—low intensity relative to intact surface regions, strong thin-structure responses, and spatial connectedness patterns consistent with linear defects. This interpretation rule is applied consistently across all SOM outputs in both Direct-SOM and transfer settings, ensuring that crack

extraction remains fully automatic and annotation-free. The resulting crack mask is exported as an 8-bit PNG with crack pixels encoded as 0 (black) and background encoded as 255 (white).

2.7 Post-processing: connected-component filtering

To suppress isolated speckle-like artifacts while preserving extended crack structures, a connected-component (CC) filter is applied to the predicted binary mask. Components whose areas are smaller than 1% of the largest connected component area are removed. This post-processing follows the same protocol as the prior SOM pipeline to maintain strict comparability between Direct-SOM and transfer outputs.

2.8 Optional CNN refinement as a spatial regularizer (pseudo-label only)

To assess whether transfer-based pseudo-labels remain stable enough for downstream learning, we include an optional U-Net refinement module trained purely on SOM-produced masks (no manual annotations). The CNN is evaluated at full resolution using tiled inference without resizing. Importantly, this module is not treated as a semantic re-classifier; instead, it functions as a spatial regularizer that can enforce continuity and suppress isolated noise, enabling a controlled test of whether SOM pseudo-labels (Direct-SOM vs transfer) support consistent learning-based smoothing.

2.9 Output-level evaluation metrics (zero-label validation)

Because the study is strictly zero-label, evaluation is performed at the model-output level. For each image, we compute standard segmentation agreement metrics (precision, recall, F1-score, Dice, and IoU) between Direct-SOM binary masks and transfer-based binary masks to quantify within-material transfer consistency. For the CNN refinement experiments, Dice and IoU are also tracked

across epochs by comparing CNN predictions to their corresponding SOM pseudo-label source, enabling stability comparisons through training dynamics.

2.10 Computational efficiency, operational deployment, and implementation details

Direct-SOM requires per-image SOM formation, whereas weight-vector transfer performs a one-time calibration on a reference image and then applies deterministic nearest-prototype assignment to subsequent images. This difference underpins three deployment-facing advantages emphasized in this work:

Computational Efficiency: target-image inference is training-free after a single reference calibration; suitable for CPU/edge execution.

Operational Consistency: fixed prototypes enforce consistent decision logic across repeated inspections on the same material condition.

Zero-labeling overhead: no manual annotations are required at any stage; SOM remains the primary evidence generator in an interpretable descriptor space.

Feature extraction and SOM formation were implemented in MATLAB R2023b, while the U-Net 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.

3. Experiments

Building on the training-free, pixel-level SOM crack segmentation framework established in our prior work, this study addresses a scalability question that is critical for practical inspection workflows: if a SOM produces a clean crack/non-crack partition on one representative image, can its weight vectors be reused to segment other images of the same material, instead of fitting a new

SOM for every image? In large-scale inspections (e.g., UAV-based bridge surveys or repeated visual checks on similar concrete or masonry surfaces), per-image SOM fitting is computationally expensive and may introduce small run-to-run variations, whereas field users would prefer to "lock in" a reliable crack/non-crack separation once on a clear reference and then apply it consistently across similar images.

The core pipeline remains strictly label-free and unsupervised: SOM clustering is applied only in the interpretable feature space, and no human annotations are used to adjust the map or its weights. For each material group, we therefore compare two fully unsupervised settings. In the Direct SOM baseline, SOM clustering is applied independently to each image using the same pixel-wise descriptors. In the proposed weight-vector transfer setting, SOM clustering is applied once to a selected reference image with relatively clear crack appearance; the fully trained SOM model (including MATLAB's internal feature normalization) and its prototype weight vectors are then saved and directly reused to assign cluster indices on subsequent target images of the same material, without running SOM clustering again on those images.

Consistent with the prior framework, all experiments use the same interpretable nine-dimensional pixel descriptor and the same unsupervised crack-class selection rule: the crack-related cluster is identified using an intensity-driven criterion that associates cracks with low mean grayscale intensity (and thin-structure response) under spatial connectedness. Post-processing also follows the same protocol as before: a connected-component filter removes small speckle-like artifacts by retaining only components whose areas are at least 1% of the largest component area in the predicted crack mask. The resulting masks are exported as 8-bit PNG images with crack pixels encoded as black (0) and background as white (255), and serve both for qualitative inspection and as pseudo-labels for refinement.

Validation is performed entirely at the model-output level under this zero-label setting. For each material group and each image, we quantify within-material transfer consistency via standard segmentation agreement metrics (precision, recall, F1-score, Dice, and IoU) computed between the Direct SOM and weight-vector transfer outputs. In addition, we include an optional U-Net refinement module as a lightweight spatial regularizer: the U-Net is trained purely on SOM-generated pseudo-labels and evaluated at full resolution using tiled inference without resizing. This refinement step demonstrates that transferred pseudo-labels are sufficiently stable to train a lightweight CNN without manual annotation, confirming SOM's utility as a cost-free pseudo-labeling engine even under weight-vector reuse.

In the following subsections, we instantiate this comparison on two representative surface conditions: aggregate-textured concrete (Section 3.1) and texture-dominated masonry (Section 3.2).

3.1 Concrete crack segmentation via weight-vector transfer

We first test within-material weight-vector transfer on a concrete set of 11 images (C1–C11). C1 is the concrete test image from the previous study, whose three-node SOM (including prototype weight vectors and MATLAB's internal normalization parameters) serves as the reference map for transfer. C2–C10 are cracked concrete images spanning diverse crack morphologies, including long narrow fissures, wider cracks with surface spalling, branching and intersecting crack networks, and ultra-thin hairline cracks near the texture noise floor. C11 is a crack-free concrete surface included as a negative control to check that transfer does not produce spurious crack detections on intact material. Image sources and author attributions (own photographs, Unsplash images, and Flickr textures, including cropped sub-patches) are listed in Appendix A.

In all experiments, a three-node SOM is used so that unsupervised clustering yields three prototype-based pixel classes. After clustering, these three classes are semantically interpreted as background concrete, residual noise/spot artifacts, and crack pixels using the same intensity-driven rule and nine-dimensional feature representation introduced in the experimental framework. We compare two fully unsupervised settings. In the Direct SOM baseline, a three-class SOM is fit independently on each image's pixel-wise descriptors, and the intensity-driven rule identifies the crack-related class for that image. In the weight-vector transfer setting, no new SOM formation is performed on the concrete set: the reference SOM from C1 is loaded once, and every pixel in C2–C11 is assigned to one of the three classes by nearest-prototype matching in the shared nine-dimensional feature space, so that the concrete/noise/crack partition is entirely dictated by the reference map.

Post-processing follows the prior SOM pipeline: connected components whose area is smaller than 1% of the largest connected component in the predicted crack mask are removed, suppressing isolated speckles while preserving extended crack structures. The resulting binary masks are exported as 8-bit PNG images with crack pixels encoded as 0 (black) and background (concrete plus noise) encoded as 255 (white), and are used both for qualitative inspection and as pseudo-labels for the optional CNN refinement stage.

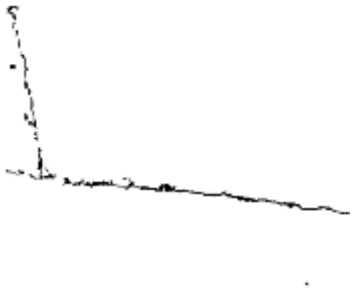
Evaluation is performed at the model-output level. For each concrete image, we compute standard agreement metrics—precision, recall, F1-score, Dice coefficient, and IoU—between the Direct SOM and weight-vector transfer segmentations, quantifying how closely the reference prototypes reproduce per-image SOM results under the same three-class partition. The crack-free image C11 plays a key role as a negative control: a robust transfer should keep the crack class essentially

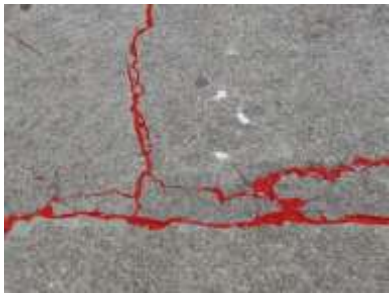
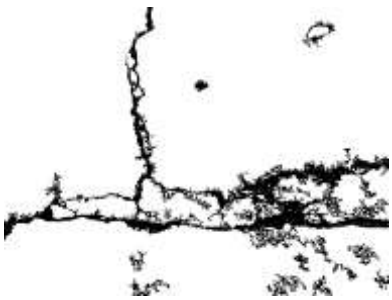
empty on this intact surface. Per-image metrics and runtime comparisons for the concrete set are reported in Table 1.

Table 1. Visual and runtime comparison between Direct SOM and weight-vector transfer pipelines on concrete images (C1–C11).

Original				
Direct SOM				
Direct SOM Time	5.0 s	6.6 s	8.4 s	1712.0 s
Direct SOM + CNN				

Transfer SOM				
Transfer SOM Time	5.0 s	2.5 s	2.7 s	59.9 s
Transfer SOM + CNN				
Original				
Direct SOM				

Direct SOM Time	1516.4 s	1706.6 s	23.2 s	45.6 s
Direct SOM + CNN				
Transfer SOM				
Time	46.7 s	41.3 s	2.6 s	4.7 s
Transfer SOM + CNN				

Original			
Direct SOM			
Direct SOM Time	605.2 s	19.9 s	13403.0 s
Direct SOM + CNN			
Transfer SOM			

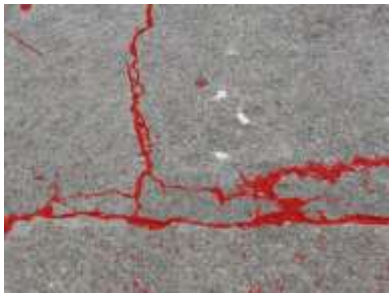
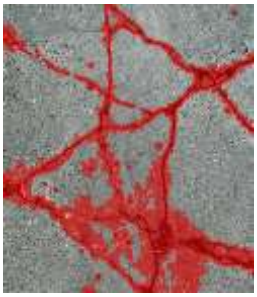
Transfer SOM Time	30.0 s	2.1 s	790.4 s
Transfer SOM + CNN			

Table 1 indicates that weight-vector transfer reproduces the Direct-SOM outputs on cracked concrete (C2–C10) with high consistency, while achieving markedly lower runtime because the SOM prototypes are obtained only once from the reference image (C1) and then reused for all subsequent images via nearest-prototype assignment in the shared 9-D feature space. Importantly, the crack-free negative control (C11) remains largely empty under both settings, indicating that transfer does not introduce systematic false positives on intact concrete.

To further assess whether transferred pseudo-labels are stable enough to support downstream learning, we train two identical U-Nets (same architecture, patch sampling, and inference tiling) using (i) Direct SOM masks and (ii) weight-vector transfer masks as pseudo-labels. Figures 2–4 report the full-resolution mean Dice and IoU between each U-Net prediction and its corresponding SOM-generated pseudo-labels across epochs. The two settings achieve comparable peak performance, while the transfer-based training shows higher last-epoch Dice/IoU and smoother trajectories, suggesting improved label stability and reduced sensitivity to late-epoch drift.

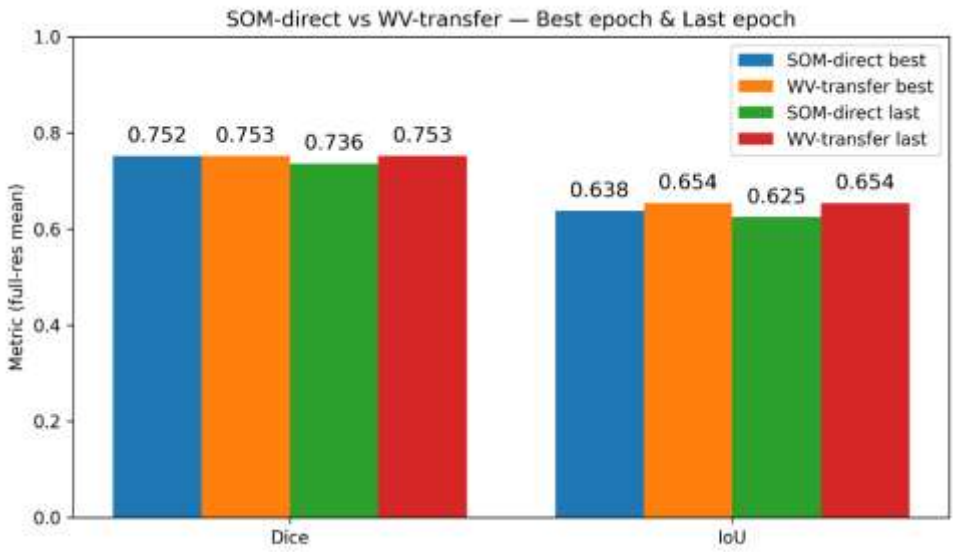


Figure 2. SOM-direct vs weight-vector transfer — best epoch and last epoch (full-resolution mean Dice and IoU).

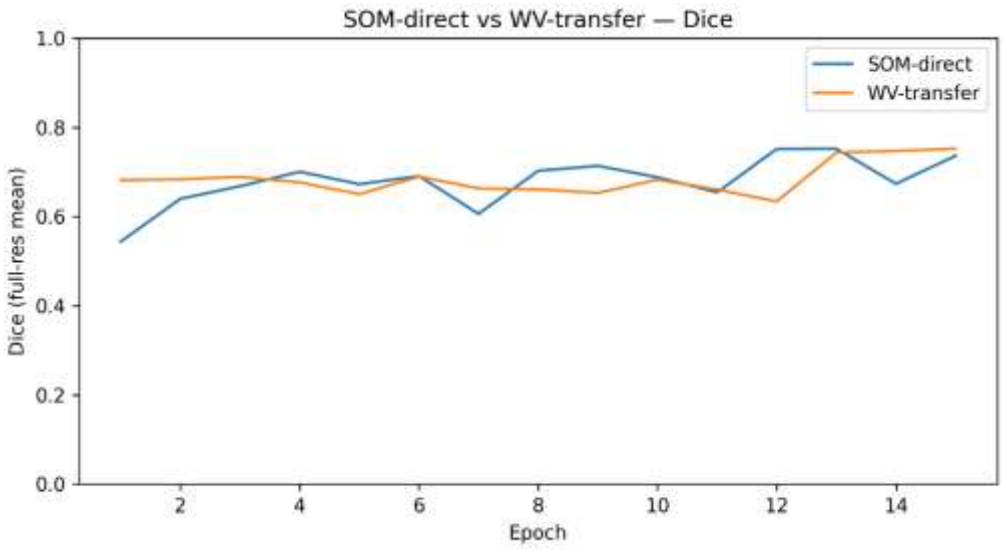


Figure 3. SOM-direct vs weight-vector transfer — epoch-wise full-resolution mean Dice.

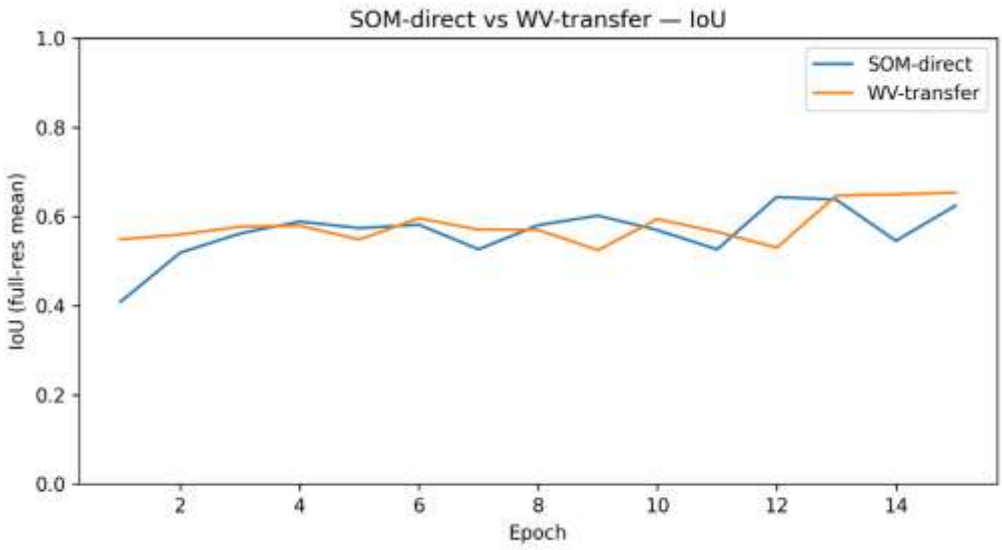


Figure 4. SOM-direct vs weight-vector transfer — epoch-wise full-resolution mean IoU.

WV-transfer achieves comparable best-epoch Dice to Direct-SOM on concrete (0.753 vs. 0.752), while yielding a marginally higher best-epoch IoU (0.654 vs. 0.638). Notably, the gap becomes clearer at the last epoch: WV-transfer maintains Dice/IoU of 0.753/0.654, whereas Direct-SOM drops to 0.736/0.625. The epoch-wise curves in Fig. 3–4 further show that WV-transfer tracks Direct-SOM closely throughout training, with a more stable trajectory in both Dice and IoU across epochs. Having established comparable (and slightly more stable) performance on concrete, we next evaluate whether the same weight-vector transfer remains effective under the texture-dominated masonry (brick) condition.

3.2 Masonry crack segmentation via weight-vector transfer

We next test within-material weight-vector transfer on a masonry (brick) set of 11 images (B1–B11). B1 is the brick test image from the prior study, whose five-class SOM (including prototype weight vectors and MATLAB's internal normalization parameters) serves as the reference map for transfer. B2–B10 are cracked brick images spanning diverse crack appearances and masonry

textures. B11 is a crack-free brick surface included as a negative control to confirm that transfer does not produce spurious detections on intact masonry. Image sources and author attributions for the masonry set (own photographs, Unsplash images, a Wikimedia Commons photo, and a Flickr photo, including screenshots and cropped sub-patches where applicable) are provided in Appendix A.

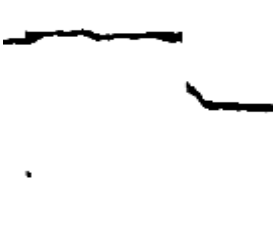
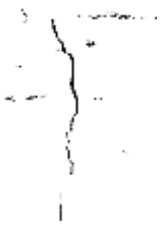
Compared with aggregate-textured concrete, brick surfaces exhibit strong periodic texture and structured discontinuities, where mortar joints, brick edges, and surface residues can produce crack-like responses in local contrast and thin-structure descriptors. To prevent such structured patterns from being absorbed into a single non-crack or noise class, the brick condition is formulated as a five-class segmentation problem.

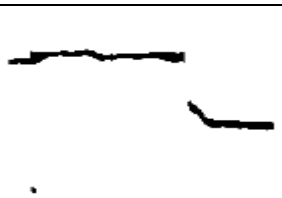
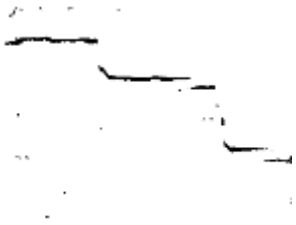
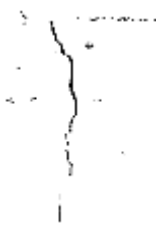
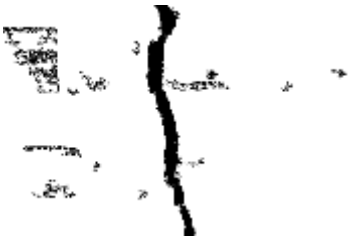
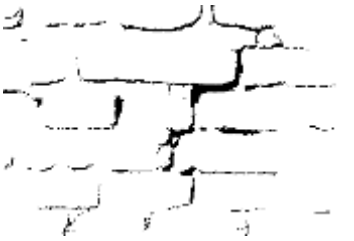
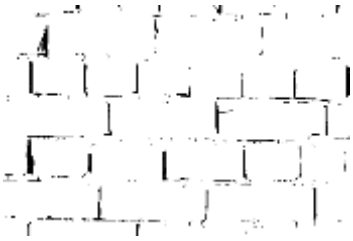
Specifically, unsupervised clustering yields five prototype-based pixel classes that are interpreted as: (i) crack, (ii) brick, (iii) noise, (iv) concrete residue / background-variation class (e.g., staining or illumination-induced intensity drift), and (v) mortar-joint class. This taxonomy explicitly separates true cracking from repetitive brick texture and mortar seams, distinguishing damage from background material patterns.

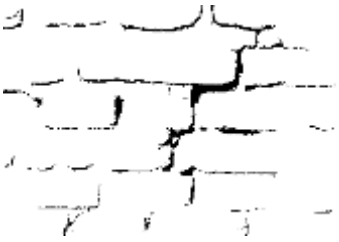
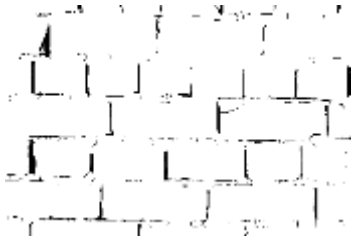
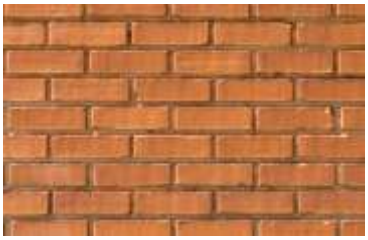
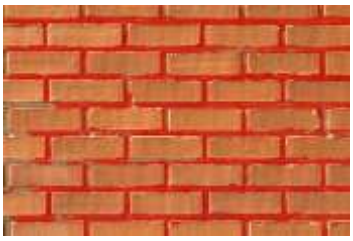
As in Section 3.1, we compare two fully unsupervised settings. In the Direct SOM baseline, a five-class SOM is fit independently on each image's pixel-wise descriptors, and the resulting classes are assigned to the above semantic categories using the same intensity- and structure-driven interpretation protocol. In the weight-vector transfer setting, SOM formation is performed only once on the reference image B1; the trained SOM model and its prototypes are then loaded and reused to assign class indices on B2–B11 via nearest-prototype matching in the shared 9-D feature space, without re-fitting a SOM for each target image.

Post-processing and mask export follow the same protocol as before. For evaluation and CNN refinement, the crack class is extracted as a binary mask (crack = 0, background = 255), with the same 1%-threshold connected-component filter applied. The crack-free image B11 serves as a negative control: a robust transfer should keep the crack class essentially empty on this intact masonry surface. Agreement metrics (precision, recall, F1-score, Dice, and IoU) and runtime are then reported to quantify how closely the transferred prototypes reproduce the per-image Direct SOM behavior under this more texture-dominated, multi-class setting.

Table 2. Visual and runtime comparison between Direct SOM and weight-vector transfer pipelines on brick images (B1-B11).

Original Image				
Direct SOM				
Time	8.0 s	82.2 s	1303.1 s	5441.7 s
CNN				

Weight Vector Transfer				
Time	8.0 s	3.0 s	24.4 s	78.9 s
CNN				
Original Image				
Direct SOM				
Time	3234.6 s	3693.8 s	6372.5	
CNN				

Weight Vector Transfer				
Time	73.85 s	56.03 s	129.9 s	
CNN				
Original Image				
Direct SOM				
Time	6220.1 s	1038.0 s	290.9 s	15.1 s
CNN				

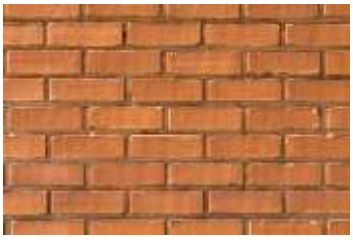
Weight Vector Transfer				
Time	161.3 s	16.57 s	8.0 s	4.5 s
CNN				

Table 2 summarizes the visual outputs and runtime comparison between Direct SOM and weight-vector transfer on the masonry set (B1–B11). Overall, weight-vector transfer achieves a pronounced runtime reduction by eliminating per-image SOM fitting: once the five-class prototypes are obtained on the reference image (B1), subsequent images are segmented via nearest-prototype assignment in the shared 9-D feature space, whereas Direct SOM can become computationally heavy on texture-dominated brick surfaces.

Beyond speed, Table 2 also reveals a practical robustness difference under challenging brick texture and residue patterns. In particular, Direct SOM occasionally absorbs structured brick texture or surface residue into the crack class (e.g., B2) and shows slightly noisier outputs on the crack-free control (B11), where mortar joints or background variation can be misinterpreted as cracking. In contrast, weight-vector transfer yields cleaner separation in these hard cases and keeps the crack class largely empty on B11, supporting its intended role as an intact-surface control.

To assess whether transfer-based pseudo-labels remain stable for downstream learning under this texture-dominated condition, we train two identical U-Nets using (i) Direct-SOM masks and (ii) transfer masks as

pseudo-labels. Figures 5–7 report full-resolution epoch-wise Dice and IoU between each U-Net prediction and its corresponding SOM-generated pseudo-labels.

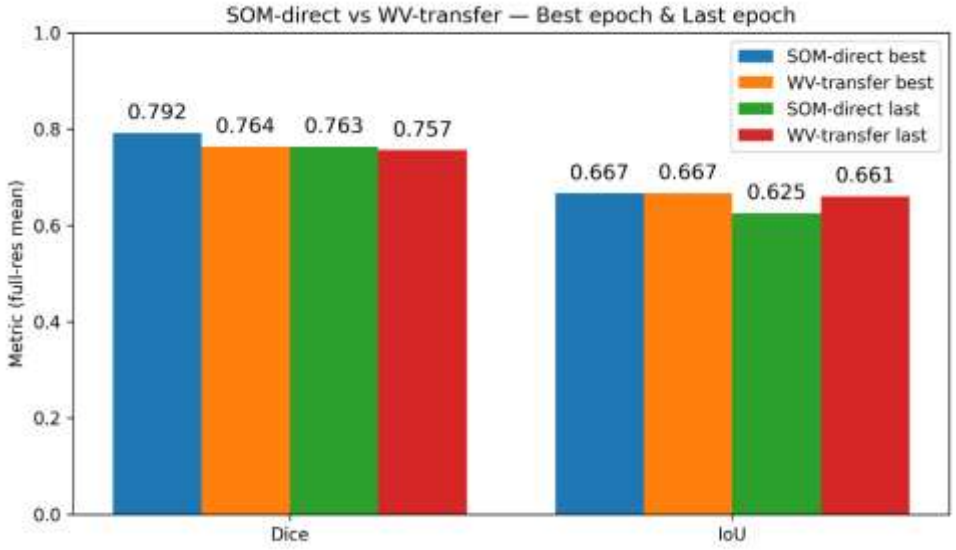


Figure 5. SOM-direct vs weight-vector transfer — best epoch and last epoch (full-resolution mean Dice and IoU).

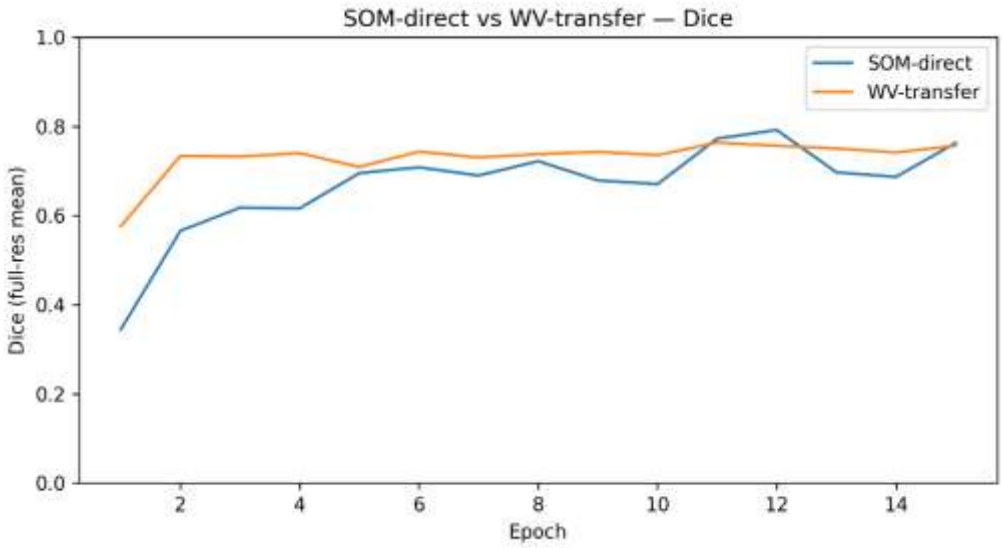


Figure 6. SOM-direct vs weight-vector transfer — epoch-wise full-resolution mean Dice.

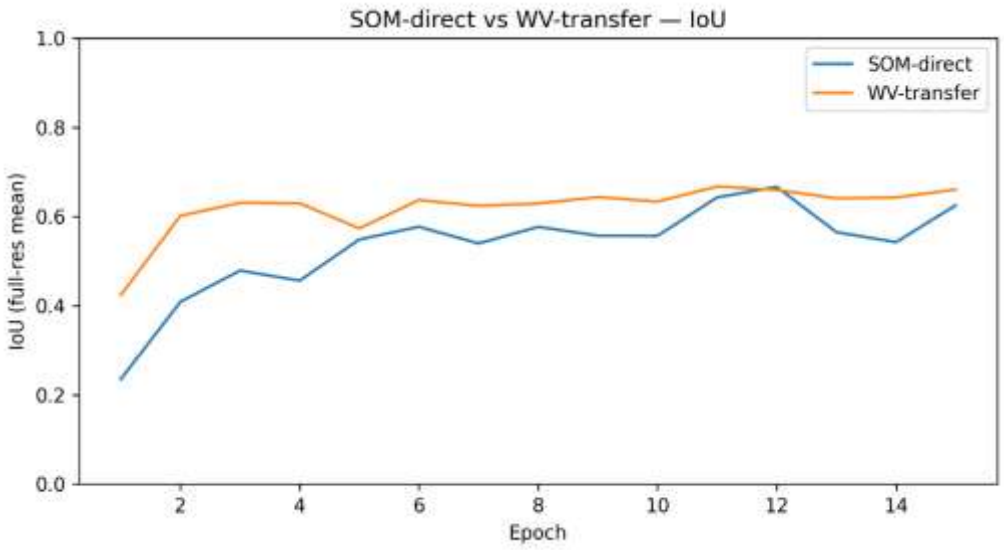


Figure 7. SOM-direct vs weight-vector transfer — epoch-wise full-resolution mean IoU.

Figures 5–7 further compare the two pseudo-label sources through full-resolution training dynamics. In terms of peak performance, Direct-SOM achieves a higher best-epoch Dice than WV-transfer on masonry (0.792 vs. 0.764), while the best-epoch IoU is essentially the same (0.667 vs. 0.667). Notably, the last-epoch behavior differs more clearly: WV-transfer maintains a higher IoU at the final epoch (0.661 vs. 0.625), while the final Dice values remain close (0.757 vs. 0.763). The epoch-wise curves (Fig. 6–7) show that WV-transfer reaches a strong Dice/IoU level earlier and remains comparatively stable across epochs, whereas Direct-SOM exhibits larger early-epoch variability and a more noticeable late-epoch fluctuation, especially in IoU.

With these concrete and masonry results established, the following section discusses the implications of weight-vector transfer for scalable, field-ready crack detection workflows.

4. Discussion

4.1 Concrete — ultra-thin cracks and the sensitivity–specificity trade-off under boundary locking

On aggregate-textured concrete, the most consequential discrepancies between Direct-SOM and WV-transfer arise near the texture noise floor, where ultra-thin cracking is weakly separable from background. WV-transfer tends to preserve hairline crack signatures because the crack/non-crack boundary is effectively *locked* by a reference SOM whose prototypes define a stable separation in the shared 9-D descriptor space. This anchoring reduces the chance that faint crack pixels are reassigned into non-crack clusters during per-image fitting when the image's dominant statistics are governed by aggregate texture and illumination variation.

The same boundary locking also explains transfer's predictable failure mode. If a target image contains low-contrast contamination patterns (dirt streaks, stains, mild intensity drift), their descriptor responses can partially overlap with the reference crack prototype and be pulled into the crack class. Direct-SOM can appear more conservative in such cases because re-fitting allows ambiguous pixels to be redistributed into non-crack clusters that better match that image's feature density. Practically, WV-transfer behaves as a sensitivity-biased but *consistent* decision rule defined by the reference boundary, whereas Direct-SOM behaves as an image-adaptive rule that may suppress certain artifact-induced detections but can under-represent the faintest cracks when they are not cleanly separable in that specific image.

4.2 Concrete — what CNN "refinement" is (and is not): learned regularization that can both complete and trim

The optional U-Net stage clarifies that a CNN trained on SOM pseudo-labels does not simply replicate the SOM, especially for borderline, ultra-thin structures. Because the network is optimized against pseudo-label masks, it learns a morphological prior—thinness, connectivity, and local continuity—that acts as a learned regularizer.

This produces a two-sided effect. First, the CNN can *complete* hairline cracks that Direct-SOM fragments or partially misses by favoring spatially coherent thin paths over isolated pixels. Second, the CNN can also *trim* extremely faint borderline pixels—particularly those produced by WV-transfer's higher sensitivity—if they are statistically inconsistent with the dominant connected-crack patterns present in the pseudo-label distribution (e.g., cases where transfer preserves very fine signals that the CNN later suppresses, such as C8). The CNN should therefore be interpreted as an optional spatial stabilizer rather than a semantic evidence source: it can improve coherence, but it can also erase the finest detections when they behave like outliers relative to learned crack geometry.

4.3 Concrete negative control — why predictability matters more than "winning" a single image

Crack-free controls reflect deployment reliability. In field inspection workflows, false crack alarms on intact surfaces drive unnecessary follow-up and degrade trust. The control case also exposes an inherent risk of per-image unsupervised fitting: even with no true cracking, a method that must partition pixels may still allocate one cluster to "the darkest/thinnest-looking pixels," which can correspond to texture extremes or noise rather than damage.

WV-transfer mitigates this by reusing a fixed prototype set. If an intact surface does not express the reference crack signature under the shared descriptor space, the crack class is naturally suppressed without per-image re-optimization. Table 1 shows that the crack-free control (C11) remains largely empty under both pipelines, confirming that WV-transfer provides a reliable "no-damage" screening mechanism without introducing systematic false positives. The deployment advantage is thus predictability: transfer reduces image-to-image fitting variability in the decision boundary that can otherwise convert harmless extremes into apparent cracks.

4.4 Masonry — why five classes are necessary, and why transfer benefits disproportionately

Masonry differs from concrete because its confounders are structured rather than random. Periodic brick texture, mortar joints, and residue/background variation can strongly activate the same contrast and thin-structure descriptors that cracks activate. Under a low-capacity partition, these structured patterns are forced to compete within a small number of clusters, increasing the chance that crack-like but non-damage signals are absorbed into the crack class.

The five-class formulation is therefore a capacity allocation strategy, not a cosmetic relabeling: it provides dedicated prototype attractors for brick texture, mortar-joint lines, and residue/background variation, reducing competition for the crack class and stabilizing semantic interpretation under strong texture. WV-transfer then benefits disproportionately because it anchors inference to a reference map that has already learned these distinctions, yielding more consistent routing of confounders into their intended non-crack categories across images. This is particularly evident on the crack-free masonry control (B11): while Direct-SOM can generate scattered crack-like noise from mortar-joint lines or texture extremes under per-image fitting, WV-transfer more consistently assigns these patterns to the mortar/brick/residue categories and keeps the crack class largely empty, reinforcing the intended "no-damage" interpretation.

4.5 Runtime — the scalability bottleneck that transfer directly resolves

Runtime is the central scalability bottleneck. Direct-SOM re-fits the SOM on every image, and cost grows rapidly with pixel count and texture complexity, making high-resolution batch inspection impractical. Table 1 provides an extreme but realistic example: for the high-resolution concrete image (7815×4737), Direct-SOM requires 13,403.0 s, whereas WV-transfer reduces this to 790.4 s by replacing per-image fitting with a single nearest-prototype assignment under a fixed reference map. This order-of-magnitude shift enables field-ready usage in large inspection batches (e.g., UAV surveys), where predictable cost and consistent decisions matter more than per-image re-optimization.

4.6 Reliability principle — reference selection as a one-time calibration step

WV-transfer is most reliable when the reference image is representative of the material condition and contains a clean, unambiguous crack signature. If the reference is atypical (unusual staining, lighting extremes, non-representative texture), transfer may inherit that bias and increase artifact sensitivity. Practically, selecting a clear reference functions as a one-time calibration step that trades per-image fitting variability for stable, reusable decision geometry—precisely the property required for scalable, repeatable inspection workflows.

5. Limitation

A primary limitation of weight-vector transfer is its reduced reliability under illumination-driven domain shifts, particularly for shadow-affected cracking. Shadow introduces a genuine change in the imaging physics: the same surface point can undergo different intensity distributions, local contrast behavior, and gradient responses depending on shadow geometry, penumbra softness, and scene lighting. These effects reshape the 9-D descriptor distribution, so the “crack prototype”

learned under one shadow configuration is not guaranteed to remain the nearest prototype under another.

This instability is amplified by a structural constraint: transfer assumes class-space compatibility between the reference map and the target task. For shadowed concrete, effective unsupervised separation requires a richer five-class partition—(i) crack, (ii) shadowed-concrete background, (iii) shadow-noise consisting of isolated speckle-like fragments induced by shadow variations, (iv) light-concrete noise corresponding to local residue under strong illumination, and (v) lit-concrete background representing uniformly illuminated intact texture—whereas the non-shadow concrete condition is well captured by a three-class partition. Because WV-transfer reuses a fixed prototype set, a 3-class reference cannot "expand" into a 5-class decomposition on a shadowed target: the additional shadow-specific modes have no dedicated prototypes to absorb them. The reverse direction is more plausible—5-class (shadow) → 3-class (lit) via semantic merging—so the transferability is inherently asymmetric for the same crack observed with and without shadow.

More broadly, WV-transfer inherits the strengths and weaknesses of the chosen reference image because it locks the decision geometry to that reference. When the reference SOM yields a clean and representative partition, subsequent transfer can be both fast and robust. Conversely, if the reference is atypical (unusual staining, lighting extremes, or non-representative texture), its prototypes encode that bias and transfer will propagate it consistently across the target set. In practice, reference selection becomes a one-time calibration step that governs the downstream operating point (sensitivity versus specificity) of the entire batch.

Finally, beyond illumination shifts, cross-material transfer is out of scope and not expected to hold. Different materials introduce distinct texture and confounder modes—such as mortar joints and periodic brick texture on masonry versus aggregate grain structure on concrete—that

fundamentally alter the feature-space organization the reference prototypes were designed to capture. For instance, a concrete reference trained to separate cracks from aggregate texture cannot meaningfully classify brick images where the dominant non-crack modes are mortar lines and brick periodicity rather than random grain patterns. When illumination regimes change sharply or the required taxonomy changes (e.g., 3-class versus 5-class), per-image SOM fitting—or an explicit re-calibration to learn condition-specific prototypes—is generally required to avoid forcing incompatible modes into incorrect prototype attractors.

6. Conclusion

This study investigated a practical scalability question for fully unsupervised, interpretable SOM-based crack segmentation: once a clean crack/non-crack separation is obtained on a representative reference image, can the learned SOM prototype weight vectors be reused to segment additional images of the same material without re-fitting a SOM per image? By formulating weight-vector transfer as nearest-prototype assignment in the shared nine-dimensional descriptor space (while preserving MATLAB's internal normalization), we demonstrated that a one-time SOM calibration can be "locked in" and then applied consistently across within-material image batches.

Across aggregate-textured concrete and texture-dominated masonry, weight-vector transfer closely reproduced the per-image Direct-SOM behavior while delivering substantial runtime reductions. Importantly, the value of transfer is not only speed: it also promotes decision consistency by removing per-image SOM re-optimization, which can otherwise introduce run-to-run variability and occasionally reallocate borderline pixels under heavy texture or surface residue.

On concrete, transfer tended to preserve ultra-thin crack signatures that may lie near the texture noise floor, enabling rapid recovery of hairline cracking that is operationally easy to miss in large inspection batches. Meanwhile, the crack-free controls illustrate a complementary deployment

benefit: when reference prototypes are representative, transfer provides a predictable “no-damage” screening mode that can keep the crack class largely empty on intact surfaces, supporting fast triage of large image collections where most images contain no actionable defects.

The optional U-Net refinement experiments further clarify how SOM-generated pseudo-labels can be leveraged downstream without manual annotation. Rather than acting as a semantic re-classifier, the CNN behaves as a spatial regularizer that learns continuity and thin-structure priors from the pseudo-labels. Depending on the pseudo-label characteristics, this regularization can either complete fragmented hairline traces or suppress extremely borderline detections. This highlights a useful interpretation for practice: SOM (direct or transferred) serves as the primary evidence generator in an interpretable feature space, while the CNN can be treated as an optional stability operator that improves spatial coherence when desired.

From an application perspective, the proposed within-material transfer mechanism enables a field-friendly workflow for rapid, large-scale screening. A user can calibrate the SOM once on a clear reference image of a given surface condition (e.g., a representative concrete deck or masonry façade), then apply the fixed prototypes to subsequent images to obtain consistent crack masks at predictable computational cost. This is particularly aligned with inspection scenarios such as UAV surveys, repeated visual monitoring under similar material conditions, and batch processing pipelines where the operational priority is stable and fast defect triage—detecting whether cracking is present, and capturing fine crack traces early—rather than re-fitting an unsupervised model separately on every image.

Future work will extend weight-vector transfer beyond its most reliable operating regime. The most important direction is regime-aware transfer under illumination shifts (especially shadow), where the descriptor distribution and the required semantic taxonomy can change. Promising

strategies include (i) automatic regime detection (lit vs. shadow) coupled with condition-specific prototype libraries, (ii) multi-reference transfer in which several calibrated maps cover typical surface variations, and (iii) taxonomy-aware transfer rules that allow controlled merging (e.g., 5-class shadow → 3-class lit) while triggering re-calibration when prototype expansion is required (e.g., 3-class lit → 5-class shadow). Extending the same framework to broader materials and acquisition conditions—while maintaining the interpretability and label-free nature of SOM decision making—will further strengthen the role of weight-vector transfer as a practical bridge between unsupervised, interpretable segmentation and scalable deployment, and may inform other repeated-use unsupervised screening settings where consistent, low-cost decisions are required across large batches.

Reference

[1] Cha Y, Choi W, Büyüköztürk O. Deep learning-based crack damage detection using convolutional neural networks. Comput Civ Infrastruct Eng 2017;32:361–78.

[2] Shi Y, Cui L, Qi Z, Meng F, Chen Z. Automatic Road Crack Detection Using Random Structured Forests. IEEE Trans Intell Transp Syst 2016;17:3434–45. https://doi.org/10.1109/TITS.2016.2552248.

[3] Zou Q, Cao Y, Li Q, Mao Q, Wang S. CrackTree: Automatic crack detection from pavement images. Pattern Recognit Lett 2012;33:227–38. https://doi.org/10.1016/j.patrec.2011.11.004.

[4] Chen J, Liu Y, Hou J. A lightweight deep learning network based on knowledge distillation for applications of efficient crack segmentation on embedded devices. Struct Health Monit 2023;22:3027–46. https://doi.org/10.1177/14759217221139730.

[5] Fang S, Lu L, Lin Z, Yang Z, Wang S. AFQSeg: An Adaptive Feature Quantization Network for Instance-Level Surface Crack Segmentation. Computers 2025;14. https://doi.org/10.3390/computers14050182.

[6] Haochen C, Weifan G, Baohua G, David B. Crack Detection in Civil Infrastructure: A Method-Scenario Review. vol. 631, 2025, p. 01001.

[7] Kyem BA, Asamoah JK, Aboah A. Context-cracknet: A context-aware framework for precise segmentation of tiny cracks in pavement images. Constr Build Mater 2025;484:141583.

[8] Steiner M, Baumann N, Lebovitz L, Magno M. A Robust and Real-Time Hyper-Spectral Sensor-Fusion Model for Concrete Crack Segmentation. 2023 IEEE Sens. Appl. Symp. SAS, 2023, p. 1–6. https://doi.org/10.1109/SAS58821.2023.10254080.

[9] L. Zhang, F. Yang, Y. Daniel Zhang, Y. J. Zhu. Road crack detection using deep convolutional neural network. 2016 IEEE Int. Conf. Image Process. ICIP, 2016, p. 3708–12. https://doi.org/10.1109/ICIP.2016.7533052.

[10] Liu Y, Yao J, Lu X, Xie R, Li L. DeepCrack: A deep hierarchical feature learning architecture for crack segmentation. Neurocomputing 2019;338:139–53.

[11] M. David Jenkins, T. A. Carr, M. I. Iglesias, T. Buggy, G. Morison. A Deep Convolutional Neural Network for Semantic Pixel-Wise Segmentation of Road and Pavement Surface Cracks. 2018 26th Eur. Signal Process. Conf. EUSIPCO, 2018, p. 2120–4. https://doi.org/10.23919/EUSIPCO.2018.8553280.

[12] Altaf A, Mehmood A, Filograno ML, Alharbi S, Iqbal J. Deployable Deep Learning Models for Crack Detection: Efficiency, Interpretability, and Severity Estimation. Buildings 2025;15. https://doi.org/10.3390/buildings15183362.

[13] Arabi S. Monitoring of civil infrastructure: A deep-learning-based computer vision approach. Iowa State University, 2022.

[14] Chu H, Chen W, Deng L. Refined segmentation of high-resolution bridge crack images via probability map-guided point rendering technique. Comput Civ Infrastruct Eng 2025;40:4946–69.

[15] Chu H, Yu D, Chen W, Ma J, Deng L. A rendering-based lightweight network for segmentation of high-resolution crack images. Comput Civ Infrastruct Eng 2025;40:323–47.

[16] Dai R, Wang R, Shu C, Li J, Wei Z. Crack Detection in Civil Infrastructure Using Autonomous Robotic Systems: A Synergistic Review of Platforms, Cognition, and Autonomous Action. Sensors 2025;25. https://doi.org/10.3390/s25154631.

[17] Spencer BF, Hoskere V, Narazaki Y. Advances in Computer Vision-Based Civil Infrastructure Inspection and Monitoring. Engineering 2019;5:199–222. https://doi.org/10.1016/j.eng.2018.11.030.

[18] Egodawela S, Khodadadian Gostar A, Buddika HADS, Dammika AJ, Harischandra N, Navaratnam S, et al. A Deep Learning Approach for Surface Crack Classification and Segmentation in Unmanned Aerial Vehicle Assisted Infrastructure Inspections. Sensors 2024;24:1936. https://doi.org/10.3390/s24061936.

[19] Harb S, Achanccaray P, Maboudi M, Gerke M. Multi-temporal crack segmentation in concrete structures using deep learning approaches. ArXiv Prepr ArXiv241104620 2024.

[20] K. -W. Tse, R. Pi, W. Yang, X. Yu, C. -Y. Wen. Advancing UAV-Based Inspection System: The USSA-Net Segmentation Approach to Crack Quantification. IEEE Trans Instrum Meas 2024;73:1–14. https://doi.org/10.1109/TIM.2024.3418073.

[21] Yuan Q, Shi Y, Li M. A Review of Computer Vision-Based Crack Detection Methods in Civil Infrastructure: Progress and Challenges. Remote Sens 2024;16. https://doi.org/10.3390/rs16162910.

[22] Kohonen T. The self-organizing map. Proc IEEE 1990;78:1464–80. https://doi.org/10.1109/5.58325.

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

[24] Kohonen T. Essentials of the self-organizing map. Neural Netw 2013;37:52–65.

[25] Sun X, Chang P. Unsupervised and noise-resilient crack detection in concrete and masonry via self-organizing maps. Eng Struct 2026;352:122130. https://doi.org/10.1016/j.engstruct.2026.122130.

[26] Chen J-H, Su M-C, Cao R, Hsu S-C, Lu J-C. A self organizing map optimization based image recognition and processing model for bridge crack inspection. Autom Constr 2017;73:58–66.

[27] Chu H, Chun P. Fine-grained crack segmentation for high-resolution images via a multiscale cascaded network. Comput Civ Infrastruct Eng 2024;39:575–94.

[28] Tse K-W, Pi R, Yang W, Yu X, Wen C-Y. Advancing UAV-Based Inspection System: The USSA-Net Segmentation Approach to Crack Quantification. IEEE Trans Instrum Meas 2024;73:1–14. https://doi.org/10.1109/TIM.2024.3418073.

[29] Choi Y, Park HW, Mi Y, Song S. Crack detection and analysis of concrete structures based on neural network and clustering. Sensors 2024;24:1725.

[30] Forest F, Porta H, Tuia D, Fink O. From classification to segmentation with explainable AI: A study on crack detection and growth monitoring. Autom Constr 2024;165:105497. https://doi.org/10.1016/j.autcon.2024.105497.

[31] Shim S. Self-supervised domain adaptive approach for extrapolated crack segmentation with fine-tuned inpainting generative model. Comput Civ Infrastruct Eng 2025;40:3333–58.

[32] Siriborvornratanakul T. Downstream semantic segmentation model for low-level surface crack detection. Adv Multimed 2022;2022:3712289.

[33] Tran TV, Nguyen-Xuan H, Zhuang X. Investigation of crack segmentation and fast evaluation of crack propagation, based on deep learning. Front Struct Civ Eng 2024;18:516–35.