Computer-Aided Civil and Infrastructure Engineering 50 (2026) 100156

Contents lists available at ScienceDirect

Computer-Aided Civil and Infrastructure Engineering

journal homepage: www.sciencedirect.com/journal/computer-aided-civil-and-infrastructure-engineering

Research Article

TopoCLR: Topology-guided contrastive learning of representations for unsupervised crack analysis

Barış Özcan *, Jörg Blankenbach

Geodetic Institute and Chair for Computing in Civil Engineering & Geo Information Systems, RWTH Aachen University, Mies-van-der-Rohe-Str. 1, Aachen, 52074, Germany

ARTICLE INFO

Keywords:
Concrete crack analysis
Topology-preserving representations
Unsupervised learning
Contrastive representation learning
Self-organizing maps
Image-based structural inspection
Fine-grained crack analysis

ABSTRACT

The reliable assessment of concrete surface damage is important for the maintenance and safety of civil infrastructure. In practice, visual inspection is often time consuming and subjective, while supervised learning methods require large amounts of labeled data that are often difficult to obtain for infrastructure inspection. Beyond this annotation challenge, crack analysis also benefits from representing gradual morphological variation, including orientation, width, and continuity. This motivates unsupervised methods that not only separate crack-related patterns from intact surfaces but also organize fine-grained crack variability in a structured way. In this paper, we introduce TopoCLR, an unsupervised framework for concrete crack analysis that combines contrastive learning with a Self-Organizing Map (SOM) in a unified training process. A convolutional encoder is trained on augmented image pairs using a contrastive objective and a topology-preserving constraint induced by the SOM. While the contrastive loss promotes discriminative and invariant features, the SOM guides the encoder toward a structured two-dimensional topology that preserves latent-space neighborhood relations. TopoCLR operates without manual annotations and provides a structured map-based organization, in which visually related samples are arranged in neighboring regions without relying on rigid predefined classes. Experiments on three public concrete crack datasets demonstrate purity values above 0.93 for the binary separation between crack and intact samples. To evaluate topology preservation under controlled gradual variation, we additionally introduce CCIC-O, an orientation benchmark derived from real cracks of one of these datasets. Under controlled rotations, TopoCLR achieves a purity of 0.835 and a label rank correlation of 0.972, the highest among the compared map-based methods. On the natural configuration, which consists entirely of real, un-rotated cracks with measured, continuous orientation labels, TopoCLR shows a similar trend at lower absolute values, as expected for the unbalanced, continuous labels, validating the method on fully real data. The learned map captures orientation-related crack variation while preserving the ordinal structure of related crack appearances.

1. Introduction

Buildings and civil infrastructure, such as bridges, tunnels, and roads, are subject to continuous ageing over their service life, which leads to progressive changes in material properties. Environmental influences and mechanical loads can cause a gradual decline in structural integrity over time (Biondini and Frangopol, 2016), which makes continuous assessment of structural condition increasingly important. Early indications of deterioration often appear as visible surface defects (Federal Highway Administration, 2012), with cracks, spalling, and other surface irregularities, that can indicate deeper structural issues (Gao & Spencer, 2008). Without timely detection and repair, these defects can compromise structural stability and create safety hazards, which makes early inspection essential and motivates scalable visual monitoring at large scale (Frangopol and Liu, 2007, Koch et al., 2015, Yeum and Dyke, 2015).

While structural damage has traditionally been assessed through periodic visual inspection by human experts, this practice is now increasingly supported by image-based methods (Chow et al., 2020a). Image-based documentation enables independent evaluation, direct comparison with previous conditions, and inspection of hard-to-access or large areas (Bartczak et al., 2023, Spencer et al., 2019), although

* Corresponding author at: Geodetic Institute and Chair for Computing in Civil Engineering & Geo Information Systems, RWTH Aachen University, Mies-van-der-Rohe-Str. 1, Aachen, 52074, Germany.
E-mail addresses: oezcan@gia.rwth-aachen.de (B. Özcan), blankenbach@gia.rwth-aachen.de (J. Blankenbach).

https://doi.org/10.1016/j.cacaie.2026.100156
Received 4 February 2026; Received in revised form 13 July 2026; Accepted 19 July 2026
Available online 21 July 2026
1093-9687/© 2026 The Author(s). Published by Elsevier Inc. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

the interpretation is still often manual and inherits the subjective uncertainties of direct visual inspection (Omer et al., 2021). To address these limitations, recent research increasingly focuses on automated evaluation of structural condition data, including images, using machine learning (Adeli, 2001, Mosalam and Gao, 2024). Classification, detection, and segmentation approaches are widely used to systematically identify and localize different types of damage (Gao and Mosalam, 2018, Yang et al., 2018), and recent work further indicates that machine-learning-based methods can contribute to highly automated inspection pipelines (Çelik et al., 2025).

The majority of machine learning approaches for damage analysis are based on supervised learning, in which models are trained using annotated image data (Özcan et al., 2024, Zhou et al., 2023). Damage areas must be marked manually by specialists with bounding regions or pixelwise masks, which requires considerable time and expertise (Benz and Rodehorst, 2024, Flotzinger et al., 2024), and inconsistent annotations can significantly weaken performance (Sylolypavan et al., 2023). In addition, damage appearances vary strongly in shape, texture, and context, which makes it difficult to compile datasets that cover real inspection variability (Benz and Rodehorst, 2025, Ro and Gong, 2024). This dependence on annotated data remains a key challenge for supervised approaches.

In response to these limitations, unsupervised and weakly supervised learning approaches are gaining in importance (Alam et al., 2025, Bhattacharya et al., 2025, Chun and Kikuta, 2024, Guo et al., 2021, Rafiei and Adeli, 2018, Zheng et al., 2022). These methods aim to discover recurring appearance patterns without explicit annotations, which reduces the effort required for data preparation and improves transferability to new application settings (Chen et al., 2024a). Recent crack-specific work in this direction includes weakly supervised localization based on class activation maps (Liu and Xu, 2022, Xu and Liu, 2022), image-level classification combined with superpixel segmentation (Liu and Xu, 2023), unsupervised semantic segmentation through superpixel clustering (Sun et al., 2024), and self-supervised contrastive learning for downstream segmentation tasks (Song et al., 2024). In parallel, general-purpose foundation models such as the Segment Anything Model (SAM) (Kirillov et al., 2023) can produce high-quality masks on unlabeled images. However, weakly supervised methods still rely on image-level class labels, and prompt-driven foundation models produce per-image masks rather than a structured representation across an unlabeled collection. This is particularly relevant for structural inspection, where surface conditions, materials, and imaging setups vary widely and labeled datasets rarely cover the full range of real-world variability.

Despite these advances, two limitations remain. First, methods that organize unsupervised representations on a topology-preserving map typically apply the map after representation learning, so the embedding is not shaped with the map in mind. Joint frameworks that integrate a SOM directly into the training process exist (Ferles et al., 2018, Forest et al., 2021, Fortuin et al., 2019, Manduchi et al., 2021, Pesteie et al., 2018), but their encoder objective is often governed by reconstruction fidelity or temporal prediction, which means that the criteria along which samples are organized cannot be directly controlled. Second, partition-oriented unsupervised methods focus on separating cracked from intact surfaces and do not preserve the continuous morphological variation of crack appearance. Cracks, however, vary along continuous dimensions such as width, orientation, and continuity, and these variations carry information about severity and possible underlying causes. A topology-preserving organization that reflects these gradual differences is therefore particularly relevant for crack analysis, but only if the criteria along which samples are organized can be aligned with the inspection-relevant variation that should be preserved.

In this paper, we present *TopoCLR*, a fully unsupervised framework for image-based analysis of concrete surface cracks. It addresses these limitations by combining contrastive representation learning with a Self-Organizing Map (SOM) in a unified, joint training process. A convolutional encoder is trained using a contrastive objective with augmentation-based positive pairs, while its feature space is simultaneously regularized by a topology-guided constraint induced by the SOM. The SOM is updated online via the classical competitive learning rule and adapts to the evolving representations during training. Through this coupling, the framework yields embeddings that are contrastively structured and organized on an interpretable map.

A central design principle of TopoCLR is the tight coupling between representation learning and topological organization: the encoder receives direct SOM-based feedback during training, so the embedding geometry is shaped to match the SOM neighborhood structure rather than being organized post-hoc. Through the choice of augmentations, the contrastive objective gives direct control over the criteria along which the encoder discriminates and the SOM organizes, which neither reconstruction-based nor temporal-prediction-based joint frameworks provide. The resulting organization supports systematic comparison of gradual and fine-grained variations in crack characteristics, including intermediate cases that rigid class definitions cannot capture.

The framework is primarily designed to analyze large collections of concrete surface images and to organize them according to their damage characteristics. As a complementary proof of concept, we additionally apply the framework in a patch-based mode to demonstrate label-free localization of crack regions in high-resolution images, presented in Appendix E. However, the core focus of this work is global image-level organization.

The main contributions of our research can be summarized as follows:

- Joint contrastive-topology framework:

  A fully unsupervised framework that integrates contrastive representation learning with online SOM updates in a single training procedure. Unlike existing joint formulations based on reconstruction error or temporal prediction, the contrastive encoder objective allows the criteria along which samples are organized to be directly steered through the choice of augmentations.

- Topology-preserving organization of crack appearance:

  An embedding organization that preserves SOM neighborhood relations and supports systematic comparison of gradual crack variability along width, orientation, and severity, including intermediate cases that rigid class definitions cannot capture.

- Comprehensive evaluation:

  Experiments on multiple crack datasets with quantitative metrics, qualitative diagnostics of representation structure and SOM organization, and ablation and sensitivity analyses on key design choices.

We position this work as a new methodological direction for fully unsupervised crack image analysis. To keep this focus, we investigate the framework in isolation, so that the reported results reflect the intrinsic properties of the method rather than auxiliary pipeline components. This methodological direction can be combined with complementary processing and downstream objectives in future work.

The remainder of the paper is organized as follows: Section 2 provides an overview of related work in image-based concrete crack analysis. Section 3 describes the proposed methodology. Section 4 outlines the experimental setup, including datasets, evaluation metrics, encoder architecture, training configuration, and the computational environment. Section 5 reports the experimental results, while Section 6 presents comparative studies. Section 7 discusses the main findings, methodological implications, and limitations. Section 8 concludes the paper with a summary and an outlook. Appendices A-D provide supplementary details, while Appendix E presents the complementary proof of concept for patch-based localization.

2. Related work

In this section, we review related work on image-based crack and surface damage analysis with a focus on unsupervised methods, covering classical image processing, hand-crafted feature clustering, representation learning, and approaches that explicitly preserve topological structure in the latent space. Based on this overview, the remaining research gap is outlined.

2.1. Classic image-processing pipelines

Early approaches to automated crack detection rely on predefined operators and rule-based pipelines that exploit characteristic visual properties of cracks, such as local contrast changes, elongated geometry, and abrupt intensity transitions relative to the surrounding surface (Hutchinson and Chen, 2006, Mohan and Poobal, 2018).

Typical pipelines highlight crack-like structures using edge- or line-sensitive processing (Abdel-Qader et al., 2003, Oliveira & Correia, 2014), convert the response into a crack mask using thresholding and morphological refinement (Oliveira and Correia, 2010, Yun et al., 2015), and apply multi-scale processing to handle varying crack widths (Nguyen et al., 2014, Subirats et al., 2006). Related approaches treat cracks explicitly as line-like objects and recover continuous traces using Hough-style or percolation-based formulations (Mathavan et al., 2017, Yamaguchi and Hashimoto, 2010).

Despite their efficiency and transparency, these methods are sensitive to variations in illumination, surface texture, and acquisition conditions, and depend on carefully tuned parameters specific to a given surface type or inspection setup (Mohan and Poobal, 2018, Oliveira and Correia, 2010, Yun et al., 2015). Their applicability in heterogeneous real-world scenarios is therefore limited.

2.2. Hand-crafted feature-based methods

Later unsupervised pipelines cluster hand-crafted feature descriptors instead of deriving crack candidates by fixed thresholds, separating crack-related patterns from intact surface appearance through similarity grouping, typically without labeled training data (Hsieh and Tsai, 2020, Yuan et al., 2024).

Typical descriptors include intensity- and contrast-based statistics (Xu et al., 2013), gradient-based and texture cues (Chen et al., 2022, Sun et al., 2023), and orientation-selective filters (Zalama et al., 2014). Clustering is then applied in feature space using k-means (Huyan et al., 2020), fuzzy clustering (Yao et al., 2025), or mixture models (Jutz and Özcan, 2022, Ogawa et al., 2019), often combined with multi-scale processing or spatial regularization to improve coherence (Bhardwaj et al., 2025, Jin et al., 2015, Li et al., 2019, Noh et al., 2017).

These methods introduce more flexible grouping than threshold-based pipelines, but performance still depends on manually designed features and similarity measures that remain fixed once selected, which limits adaptability across surface and acquisition conditions.

2.3. Learned representations for crack analysis

Representation learning approaches learn feature embeddings directly from image data, reducing reliance on manually designed descriptors. In unsupervised and self-supervised settings, neural networks are trained with surrogate objectives and used as feature extractors for downstream tasks such as clustering or anomaly detection.

Autoencoders and related generative models reconstruct the input and use reconstruction errors as anomaly indicators, often under the assumption that intact surface appearance dominates the training data (Chow et al., 2020b, Hou et al., 2020, Wu and Liu, 2025). Self-supervised approaches have also been applied to crack analysis, including cycle-consistent generative formulations (Zhang et al., 2020) and contrastive learning, in which encoders produce consistent embeddings for different views of the same region (Song et al., 2024). In the broader computer-vision literature, deep clustering frameworks couple representation learning with a clustering objective and optimize embeddings jointly with cluster assignments (Caron et al., 2018, Ghasedi Dizaji et al., 2017, Guo et al., 2017a,b, Van Gansbeke et al., 2020, Xie et al., 2016). However, in crack analysis, clustering is typically used as a downstream step on learned features, although first crack-specific clustering-inspired formulations have recently been explored (Chen et al., 2024b).

These methods produce more expressive representations than hand-crafted descriptors, but the global organization of the feature space is shaped only indirectly by the training objective and is therefore not explicitly controlled (Locatello et al., 2019). Neighborhood relations in the embedding space are not guaranteed to reflect gradual changes in crack morphology or severity, which limits the stability and usefulness of similarity groupings for inspection-oriented analysis.

2.4. Joint representation and topology learning

To address this lack of explicit structure, several approaches introduce explicit constraints that preserve neighborhood relations and impose an ordered organization on the latent space, shaping the geometry during learning rather than relying on structure that emerges implicitly.

The most widely used formulation in this context is the SOM (Kohonen, 2001), which produces a 2D grid in which similar samples occupy neighboring units. In crack and surface inspection, SOM-based pipelines have been applied with hand-crafted descriptors (Mathavan et al., 2012, Mathavan et al., 2015) and with learned representations from convolutional encoders or autoencoders (Sun and Chang, 2026, Yamaguchi and Mizutani, 2024), primarily to separate crack-related patterns from intact background.

Beyond SOM-centered pipelines for crack analysis, a broader line of work in the general representation learning literature integrates SOMs directly into deep neural network training. The first generation of these joint frameworks couples an autoencoder with a SOM, where the encoder is shaped primarily by a reconstruction objective and the SOM organizes the resulting latent space. Representative examples include Deep Neural Maps (Pesteie et al., 2018), DASOM (Ferles et al., 2018), SOM-VAE (Fortuin et al., 2019), DESOM (Forest et al., 2021), and (T-) DPSOM (Manduchi et al., 2021). Across these frameworks, the encoder optimization is not guided by an externally specifiable discrimination criterion, and discriminative structure between samples emerges only implicitly from the training objective. This limits how directly the criteria along which samples are organized can be controlled.

More recently, SOM-CPC (Huijben et al., 2023) has been introduced as the closest prior framework that combines a contrastive objective with a SOM. SOM-CPC is based on Contrastive Predictive Coding (van den Oord et al., 2018), where positive examples are defined through autoregressive prediction of future representations along a temporal axis. This makes the framework applicable to high-rate time series, but the underlying contrastive principle requires a meaningful temporal direction in which future signals can be predicted, which is not the case for static images. Beyond the data modality, the discrimination criterion is governed by the temporal-prediction task and cannot be specified externally to align with inspection-relevant similarity. Beyond SOM-centered formulations, more general topology-preserving objectives have also been proposed (Moor et al., 2020).

Table 1 summarizes how these frameworks differ along four axes that matter for the present work, namely the representation-learning signal that shapes the encoder, the mechanism by which the encoder is coupled to the SOM, the data modality the framework targets, and the extent to which the criterion of similarity used during organization can be specified externally rather than emerging implicitly from the training objective.

Across these approaches, the encoder objective shapes the latent

Table 1
Comparison of joint representation and SOM-based organization frameworks.

Method	Representation-Learning Signal	Encoder-SOM Coupling	Data Modality	Steerability of Discrimination Criterion
Deep Neural Maps	reconstruction and clustering	hard BMU; alternating SOM and clustering-guided encoder updates	static (images)	indirect, through reconstruction and clustering
DASOM	denoising reconstruction	hard BMU; energy-based topology loss	static	indirect, through reconstruction
SOM-VAE	continuous and quantized reconstruction	hard BMU; commitment and neighborhood losses	static and time series	indirect, through reconstruction
DESOM	reconstruction	hard BMU; Gaussian-neighborhood topology loss	static	indirect, through reconstruction
(T-) DPSOM	variational reconstruction and clustering (temporal losses in T-DPSOM)	soft assignments; clustering and neighborhood losses	static (DPSOM), time series (T-DPSOM)	indirect, through reconstruction and clustering
SOM-CPC	contrastive temporal prediction)	hard BMU; Gaussian-neighborhood topology loss	time series only	indirect, tied to the temporal-prediction task
TopoCLR (this work)	contrastive augmented views	soft encoder topology loss; separate BMU-based prototype updates; contrastive-topology curriculum	static (images)	direct, through the augmentation policy

space along criteria tied either to reconstruction, clustering, or to temporal-prediction objectives, while in many crack-specific pipelines the topological organization is imposed in a separate step after feature extraction. The criteria along which samples are organized are therefore determined indirectly, which can limit the usefulness of the learned topology for inspection tasks.

2.5. Limitations and research gap

Across existing approaches, a recurring limitation concerns how the criteria for organization in the latent space are determined. Classical and hand-crafted methods rely on fixed components that transfer poorly across heterogeneous surfaces. Learned representations are more expressive, but their global geometry is not an explicit design objective. Existing joint representation-topology frameworks impose structure on the latent space, but the criteria along which samples are organized remain governed by reconstruction, clustering, or temporal-prediction objectives rather than by an externally specifiable similarity definition.

This motivates a framework in which representation learning and topological organization are coupled in a single training process, and in which the criteria along which samples are organized can be aligned directly with the inspection-relevant variation that the SOM should preserve. The resulting organization should yield coherent groups of surface patterns while preserving meaningful neighborhood relations and smooth transitions between related cases, reflecting gradual changes in crack geometry and severity without relying on rigid predefined classes.

3. Methodology

3.1. Conceptual overview

This section outlines our methodology for unsupervised analysis of surface patterns on concrete surfaces, including both cracked and intact cases. The framework combines contrastive representation learning with a SOM in a single training process. Feature representations are learned from the data and shaped to align with the explicit 2D structure of the SOM, which is then used to organize samples on the grid and identify coherent neighborhoods of crack appearance.

The workflow is depicted in Fig. 1 and starts with the preprocessing of the input images, followed by the generation and subsequent standardization of two augmented views per image for contrastive learning. Targeted transformations, such as translations, zooming, and intensity variations generate two different views per image, which form a positive pair, while views from other images within the same mini-batch serve as in-batch negatives. All views are processed by a CNN-based encoder that is trained to produce feature vectors reflecting similarity and dissimilarity of crack appearance.

During training, the encoder and the SOM are optimized jointly. The encoder is updated using a combined contrastive and topology-guided objective, while the SOM adapts its prototype vectors to the evolving feature representations. The relative influence of the two objectives is not kept constant but changes over the course of training. In the early training stages, the contrastive objective dominates, encouraging the encoder to separate different crack-related appearance patterns. In later stages, the topology-guided term becomes more influential and drives the alignment of the learned embeddings with the two-dimensional grid structure of the SOM. At the same time, the SOM neighborhood size is gradually reduced, such that the map initially captures coarse variations in crack appearance and subsequently refines local and fine-grained differences. Through this co-evolution, the encoder and the SOM gradually form a consistent organization of crack patterns.

The framework operates at the image level and maps whole-image samples onto the SOM grid according to their dominant crack characteristics, enabling the identification of recurring crack patterns and gradual transitions between them. A complementary patch-based extension for the spatial localization of crack regions in high-resolution images is provided in Appendix E.

The individual steps of the method can be summarized as follows:

- Image normalization (Section 3.2.1):

Normalization of surface images with respect to spatial resolution and intensity statistics.

- Image pair formation (Section 3.2.2):

Generation of augmented views through targeted transformations and construction of a paired batch structure for the contrastive setup.

- Topology-guided contrastive representation learning (Section 3.2.3):

Learning crack-related feature representations with a contrastive objective and a topology-guided regularization induced by the SOM.

- Topological structuring with SOM adaptation (Section 3.2.4):

Alternating SOM updates that co-adapt the grid prototypes to the evolving encoder embeddings.

- Post-training SOM mapping and analysis (Section 3.2.5):

Analysis of the organized SOM to identify coherent groups of crack appearance and their mutual relationships.

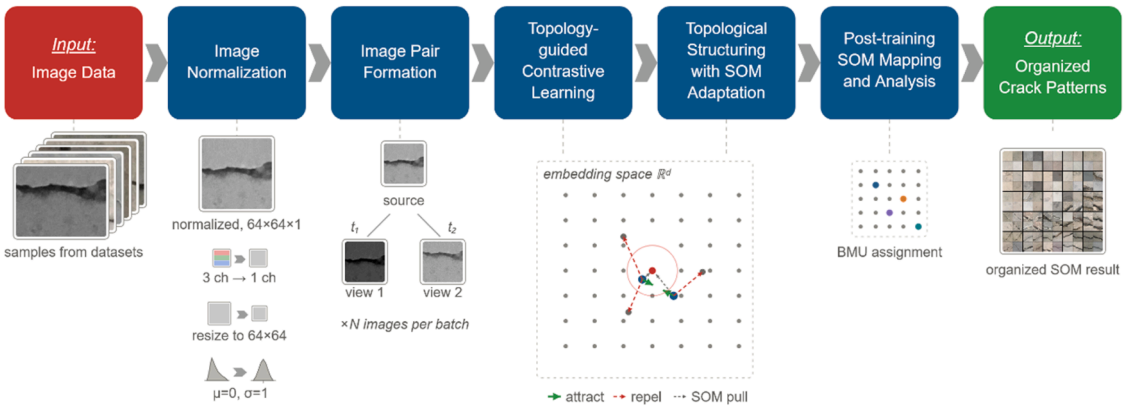


Fig. 1. Overview of our methodology. The per-image standardization is applied per augmented view at the encoder input.

The following subsections describe these methodological details. Section 3.2.6 then summarizes the training dynamics that result from the interaction of the schedules introduced in Sections 3.2.3 and 3.2.4.

3.2. Topology-guided representation learning and clustering

This section presents the core component of TopoCLR for topology-guided image-level clustering. It covers the data preparation for the contrastive setup and the joint optimization of the encoder and SOM that produce the SOM organization used for analysis.

3.2.1. Image normalization

The image normalization step comprises image preprocessing and intensity standardization to create consistent conditions for contrastive representation learning. During preprocessing, images are converted to grayscale, rescaled to the intensity range [0, 1], and resized to a fixed spatial resolution ($H \times W$), yielding uniformly preprocessed inputs for the subsequent image pair formation. The standardization is applied independently to each augmented view after the image pair formation described in Section 3.2.2 and immediately before it enters the encoder.

To standardize intensity values, we apply per-image Z-standardization to an image $x \in \mathbb{R}^{H\times W}$:

$$\widetilde{x} = \frac{x - \mu_x}{\sigma_x} \tag{1}$$

where μ_x and σ_x denote the mean and standard deviation of the pixel intensities of the image x.

This operation is used to reduce the influence of global brightness and contrast variations while preserving the relative intensity differences within the processed image. By centering the intensities around zero with unit variance, it also supports stable neural network training.

3.2.2. Image pair formation

Contrastive learning relies on comparing different views of the same image. Accordingly, generating paired views is a central step of our workflow. The objective is to encourage the network to bring visually similar crack structures closer together in feature space while separating dissimilar ones. For this purpose, two different views are generated from each preprocessed input image. These two views form a positive pair, whereas views originating from different images act as negatives within the same mini-batch.

For each preprocessed input image x_i, two independent transformations t_1 and t_2 are applied:

$$\left(x_i^{(1)}, x_i^{(2)}\right) = \left(t_1(x_i), t_2(x_i)\right) \tag{2}$$

where $x_i^{(v)}$ denotes the v-th augmented view, with $v \in \{1, 2\}$ of the preprocessed image x_i.

For a mini-batch of N_b images, two augmented views are generated per image, resulting in $2N_b$ views in total. The two views derived from the same input image constitute a positive pair, as they represent the same underlying content with altered appearance. For each view, its counterpart is the only positive sample, while the remaining $2N_b - 2$ views serve as negatives in the subsequent NT-Xent formulation (Section 3.2.3). This setup follows the principle of instance discrimination, where each image is treated as an individual pseudo-class.

To generate diverse yet semantically consistent views, we apply transformations that preserve crack-related structures while introducing sufficient variability for robust feature learning.

These comprise geometric transformations such as translations and isotropic zooming as well as photometric ones such as brightness, contrast, and noise variations. The concrete transformation set and its parameter ranges are task-dependent and are reported in Table 5 (Section 4.4.2).

In contrastive self-supervised learning, these transformations are not primarily used to increase the dataset size. Instead, they define the invariances of the learned representation by specifying which appearance changes should not alter the underlying semantic content. This enables two transformed views of the same image to be treated as a positive pair and allows the network to learn meaningful representations without manual labels. Consequently, the choice of transformations directly determines which variations in crack appearance are considered irrelevant and should be mapped to similar feature representations.

These paired views are used to optimize the encoder using the NT-Xent objective combined with the topology-guided term, as described in Section 3.2.3.

3.2.3. Topology-guided contrastive representation learning

The image pairs generated in the previous step form the basis for topology-guided contrastive representation learning. In this stage, a convolutional encoder is trained to map images of concrete surfaces into a feature space in which visually similar crack patterns are represented by nearby embeddings, while dissimilar surface appearances are separated. The resulting representation serves as the foundation for the subsequent organization and analysis on the SOM.

The encoder f_θ is implemented as a fully convolutional neural network (specific architecture described in Section 4.3) with global average pooling, followed by a lightweight projection head to map the feature vector onto a low-dimensional embedding space. For an augmented view $\widetilde{x}_i^{(v)}$, the encoder produces a feature vector:

$$\widetilde{z}_i^{(v)} = f_\theta(\widetilde{x}_i^{(v)}) \tag{3}$$

To obtain a representation suitable for similarity comparisons, this feature vector is subsequently L2-normalized:

$$z_i^{(v)} = \frac{\tilde{z}_i^{(v)}}{\| \tilde{z}_i^{(v)} \|_2} \tag{4}$$

This normalized embedding $z_i^{(v)}$ is used for all learning objectives. For notational simplicity, we collect the embeddings of all augmented views in the mini-batch as $\{z_k\}_{k=1}^{2N_b}$, where each z_k corresponds to one of the normalized view-specific embeddings $z_i^{(v)}$ (image i, view v) and $j(k)$ denotes the index of its paired view from the same source image.

The training optimizes two complementary objectives. The first objective enforces discriminative representation learning through a contrastive loss, while the second objective introduces a topology-guided regularization that aligns the learned representations with the structure of the SOM.

Contrastive Objective.

The contrastive component follows the SimCLR principle and is realized using the normalized temperature-scaled cross-entropy (NT-Xent) loss. For a mini-batch of N_b images, two augmented views are generated for each image, resulting in $2N_b$ embeddings. For a given embedding z_k, the corresponding positive example $z_{j(k)}$ originates from the same source image (the other augmented view), while all remaining embeddings in the mini-batch act as negatives.

The per-view loss is then defined as

$$l_{NT-Xent}(k) = -log \frac{\exp\left(sim\left(z_k, z_{j(k)}\right)/\tau_c\right)}{\sum_{a=1}^{2N_b} 1[a \neq k] \exp(sim(z_k, z_a)/\tau_c)}, \tag{5}$$

where $sim(.,.)$ denotes cosine similarity and τ_c is a temperature parameter that controls the softness of the contrastive distribution. The value of τ_c used in our experiments is reported in Table 4.

The contrastive loss for the mini-batch is obtained by averaging over all views,

$$L_{NT-Xent} = \frac{1}{2N_b} \sum_{k=1}^{2N_b} l_{NT-Xent}(k) \tag{6}$$

This objective pulls together embeddings of different augmented views of the same crack pattern and pushes apart embeddings corresponding to different surface appearances.

Topology-Guided Regularization.

In addition to contrastive discrimination, the encoder is encouraged to produce representations that can be organized in a structured manner on the SOM grid. To achieve this, a topology-guided regularization term is introduced, which promotes meaningful neighborhood relations between related surface patterns and explicitly aligns the learned embeddings with the topology of the self-organizing map. This regularization complements the contrastive objective by not only separating dissimilar appearances but also enforcing an ordered latent structure that is compatible with the subsequent SOM-based analysis.

Let $\{w_m\}_{m=1}^{M}$ denote the SOM prototype vectors, where M is the number of nodes in the two-dimensional grid and $w_m \in \mathbb{R}^d$. In a classical SOM, prototype updates rely on a discrete best-matching-unit assignment, which is not differentiable and therefore cannot be directly integrated into the encoder training. To enable a differentiable alignment between embeddings and map prototypes, we introduce a continuous soft assignment that is used exclusively during encoder optimization.

For each embedding z_k, a soft assignment over all SOM prototypes is computed based on Euclidean distances:

$$p_{k,m} = \frac{\exp\left(-\| z_k - w_m \|_2^2/\tau_t\right)}{\sum_{l=1}^{M} \exp\left(-\| z_k - w_l \|_2^2/\tau_t\right)}, \tag{7}$$

where τ_t is a temperature parameter controlling the sharpness of the assignment. The value of τ_t used in our experiments is reported in Table 4.

Based on these weights, an expected prototype representation is formed as:

$$\hat{w}_k = \sum_{m=1}^{M} p_{k,m} w_m \tag{8}$$

The topology-guided loss for a single embedding is then defined as the squared distance between the embedding and its expected prototype:

$$l_{topo}(k) = \frac{1}{2} \| z_k - \hat{w}_k \|_2^2 \tag{9}$$

The topology-guided loss for the mini-batch is obtained by averaging over all views:

$$L_{topo} = \frac{1}{2N_b} \sum_{k=1}^{2N_b} l_{topo}(k) \tag{10}$$

For encoder optimization, L_{topo} is fully differentiable with respect to z_k. The non-differentiable BMU does not appear on the encoder's gradient path: the soft assignment in Eq. (7) and the expected prototype in Eq. (8) are smooth functions of z_k, while the SOM prototypes $\{w_m\}$ are treated as constants during encoder backpropagation and updated separately via the classical competitive learning rule (Section 3.2.4).

The closed-form gradient of $l_{topo}(k)$ follows from the chain rule. With $d_{k,m} = \| z_k - w_m \|_2$, the soft assignment (Eq. 7) is a softmax over $-\frac{{d_{k,m}}^2}{\tau_t}$ and its derivative with respect to z_k is

$$\frac{\partial p_{k,m}}{\partial z_k} = \frac{2}{\tau_t} p_{k,m} \left(w_m - \hat{w}_k\right) \tag{11}$$

where the right-hand side uses the definition of $\hat{w}_k$ as the expectation of $\{w_m\}$ under p_k. Substituting into Eq. (8) yields

$$\frac{\partial \hat{w}_k}{\partial z_k} = \frac{2}{\tau_t} \left(\sum_{m=1}^{M} p_{k,m} w_m {w_m}^T - \hat{w}_k {\hat{w}_k}^T \right) = \frac{2}{\tau_t} Cov_p(w) \tag{12}$$

where $Cov_p(w)$ is the covariance matrix of the prototypes under the soft-assignment distribution. The gradient of $l_{topo}(k) = \frac{1}{2} \| z_k - \hat{w}_k \|_2^2$ is then

$$\frac{\partial l_{topo}(k)}{\partial z_k} = \left(I - \frac{2}{\tau_t} Cov_p(w) \right) (z_k - \hat{w}_k) \tag{13}$$

Two limit cases are instructive. When the soft assignment is sharply peaked on a single prototype w_b (the BMU), $Cov_p(w) \to 0$, and the gradient reduces to $(z_k - w_b)$, corresponding to a hard pull of z_k toward its closest prototype. Conversely, when the soft assignment is nearly uniform, $Cov_p(w)$ approaches the empirical prototype covariance, and the gradient is rescaled along directions of high prototype variability. The soft assignment therefore acts as a differentiable interpolation between these limits, and the encoder is consistently guided toward representations that can be organized on the SOM grid while the standard competitive learning rule of the SOM remains unchanged.

Joint Objective.

The total objective for a mini-batch in epoch e combines both terms:

$$L_{total}(e) = (1 - \lambda(e))\, L_{NT-Xent} + \lambda(e)\, L_{topo}, \tag{14}$$

where $\lambda(e)$ is an epoch-dependent weighting factor scheduled over training and controlling the balance between the contrastive and the topology-guided term. Details for this parameter and its effect are given in Section 3.2.6. At early training stages, the contrastive objective dominates and promotes discriminative feature learning. As training progresses, the influence of the topology term increases, encouraging the embeddings to align with the structure of the SOM grid.

Through this joint optimization, the encoder learns crack-sensitive representations that are not only discriminative but also organized in a way that is directly compatible with the subsequent SOM-based clustering and analysis.

3.2.4. Topological structuring with SOM adaptation

Within our framework, the SOM also adapts alongside the encoder and iteratively organizes the learned feature representations on a two-dimensional grid topology, yielding an explicit and interpretable neighborhood structure in the embedding space. While the update mechanism follows the classical competitive learning principle (BMU selection and neighborhood adaptation), we do not apply prototype updates sequentially per sample. Instead, we use a vectorized mini-batch formulation in which each prototype is updated toward a neighborhood-weighted mean of the current batch embeddings. This design supports stable prototype adaptation under stochastic mini-batch sampling, while retaining the neighborhood-preserving behavior of standard SOM learning.

The SOM consists of a two-dimensional grid of size $m \times n$, with $M = mn$ nodes. Each node $(u, v) \in \{1, \ldots, m\} \times \{1, \ldots, n\}$ is associated with a prototype vector in $\mathbb{R}^d$, where the dimensionality d matches that of the encoder embeddings. This allows distances between an embedding z_i and the prototypes to be computed directly in the same feature space using the Euclidean norm.

We initialize the SOM prototypes in a data-driven way using Principal Component Analysis (PCA) on the initial encoder embeddings. All prototypes are then L2-normalized to match the scale of the normalized embeddings.

To make neighborhood relations on the $m \times n$ lattice explicit, we switch from the single-index notation w_m used in Section 3.2.3 to the equivalent grid-coordinate notation $w_{u,v}$.

In the proposed workflow, the SOM is trained jointly with the encoder using an alternating update scheme. After each gradient update of the encoder, the current mini-batch is used to adapt the SOM prototypes. The SOM learning rate $\eta_{\text{som}}(e)$ and neighborhood width $\sigma(e)$ follow epoch-dependent schedules. Within a given epoch e, these parameters are kept fixed and are applied to all mini-batches of that epoch. We index the repeated SOM prototype adaptations within training by a mini-batch update step t.

For each original image, we aggregate the embeddings of its two augmented views into a single representation for the SOM update by averaging:

$$\bar{z}_i = \frac{1}{2}\left(z_i^{(1)} + z_i^{(2)}\right) \tag{15}$$

which yields a mini-batch $\{\bar{z}_i\}_{i=1}^{N_b}$.

For each $\bar{z}_i$, the BMU is the node whose prototype is closest in the embedding space:

$$b(i) = BMU(\bar{z}_i) = \underset{(u,v)}{\operatorname{argmin}} \|\bar{z}_i - w_{u,v}^{(t)}\|_2. \tag{16}$$

To propagate the influence of $\bar{z}_i$ to neighboring nodes, distances are measured on the 2D lattice. Let $b(i) = (u_i, v_i)$ denote the BMU coordinates of $\bar{z}_i$. The lattice distance between a node (u, v) and the BMU of $\bar{z}_i$ is defined as

$$d_{grid}((u,v), b(i)) = \|(u,v) - (u_i, v_i)\|_2. \tag{17}$$

The influence of $\bar{z}_i$ is propagated from its BMU to neighboring nodes using a Gaussian neighborhood function

$$h^{(e)}(d) = \exp\left(-\frac{d^2}{2\sigma(e)^2}\right) \tag{18}$$

where $\sigma(e)$ controls the neighborhood width at epoch e. Using this function, we define the neighborhood weight of sample i for node (u, v) as

$$h_i^{(e)}(u,v) = h^{(e)}\left(d_{grid}((u,v), b(i))\right). \tag{19}$$

Based on these weights, each node (u, v) receives a neighborhood-weighted target embedding given by the weighted mean over the mini-batch:

$$\bar{z}_{u,v}^{(e)} = \frac{\sum_{i=1}^{N_b} h_i^{(e)}(u,v)\bar{z}_i}{\sum_{i=1}^{N_b} h_i^{(e)}(u,v)}. \tag{20}$$

The prototypes are then updated toward $\bar{z}_{u,v}^{(e)}$ in a vectorized mini-batch form of the classical competitive learning rule:

$$w_{u,v}^{(t+1)} = (1 - \eta_{som}(e))\; w_{u,v}^{(t)} + \eta_{som}(e)\; \bar{z}_{u,v}^{(e)} \tag{21}$$

where $\eta_{\text{som}}(e)$ denotes the SOM learning rate.

As a last step, to maintain scale consistency with the normalized encoder embeddings, the prototypes are L2-normalized:

$$w_{u,v}^{(t+1)} \leftarrow \frac{w_{u,v}^{(t+1)}}{\| w_{u,v}^{(t+1)} \|_2}. \tag{22}$$

The SOM learning rate $\eta_{\text{som}}(e)$ and neighborhood width $\sigma(e)$ follow epoch-dependent schedules with details given in Section 3.2.6.

3.2.5. Post-training SOM mapping and analysis

After training, the encoder is applied to all original input samples to obtain final feature representations $z_i = f_\theta(\widetilde{x}_i) / \|f_\theta(\widetilde{x}_i)\|_2$ for $i = 1, \ldots, N_{data}$. Each representation is assigned to its BMU on the trained SOM grid, $b(i) = \underset{(u,v)}{\operatorname{argmin}} \|z_i - w_{u,v}\|_2$, yielding a discrete two-dimensional arrangement of the data. This map-based assignment constitutes the primary output of the organization stage and serves as the reference structure for subsequent analysis.

Based on this assignment, samples mapped to the same SOM node form a coherent group representing similar crack characteristics or surface conditions. The spatial arrangement of these groups on the grid enables qualitative analysis and interpretation at different levels of granularity, ranging from individual nodes to larger contiguous regions.

3.2.6. Training dynamics and encoder-SOM co-adaptation

During training, the interaction between the encoder and the SOM is governed by epoch-dependent schedules. These schedules control how strongly the topological structure influences the learned representations and how locally the SOM adapts to the encoder features at different stages of training.

Let e denote the current epoch and E the total number of training epochs. The weight of the topology-guided regularization follows a sigmoid schedule:

$$\lambda(e) = \lambda_{start} + (\lambda_{end} - \lambda_{start})\; \frac{1}{1 + \exp\left(-k\left(\frac{e}{E} - t_0\right)\right)}. \tag{23}$$

This keeps $\lambda(e)$ close to λ_{start} during early epochs and increases it smoothly towards λ_{end} after the inflection point t_0, controlled by slope parameter k.

The SOM neighborhood width $\sigma(e)$ and the SOM learning rate $\eta_{som}(e)$ are reduced linearly over time between an initial value σ_{start} and a final value σ_{end} as

$$\sigma(e) = \sigma_{start} + \frac{e}{E}(\sigma_{end} - \sigma_{start}) \tag{24}$$

$$\eta_{som}(e) = \eta_{start} + \frac{e}{E}(\eta_{end} - \eta_{start}). \tag{25}$$

This curriculum lets the contrastive objective dominate during early training, where it forms locally concentrated regions in the embedding space while the SOM adapts to coarse cluster structure under a large neighborhood radius (Fig. 2, panels 1-3). As training progresses, the increasing $\lambda(e)$ pulls the embeddings towards the SOM topology, while the shrinking $\sigma(e)$ shifts the SOM from global to local refinement. This produces a progressive alignment between encoder representations and

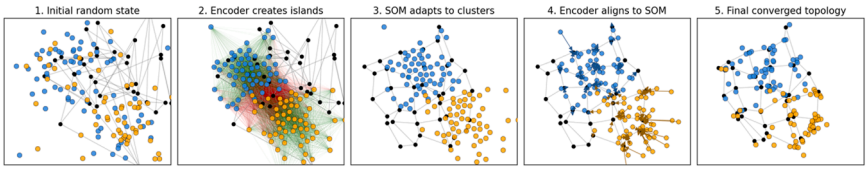


Fig. 2. Training dynamics of the topology-guided contrastive learning framework.

the SOM grid, with smooth transitions between neighboring nodes (Fig. 2, panel 4). At convergence, the encoder and the SOM are co-adapted, yielding a representation that is both discriminative and explicitly organized on the 2D map (Fig. 2, panel 5).

From left to right, the panels in Fig. 2 illustrate the evolution from an initially unstructured state, through contrastive separation of feature clusters and coarse SOM adaptation, to progressive alignment of encoder representations with the SOM grid, and finally a converged, topologically organized map reflecting coherent crack appearance patterns.

4. Experimental setup

In this section, we describe the experimental setup used to implement and evaluate the proposed methodology. We introduce the datasets and define the evaluation metrics, followed by presenting the encoder architecture, and specification of the training configuration and computational environment.

4.1. Datasets

Several public datasets of concrete surfaces are used to evaluate the proposed approach. Three datasets with image-level class labels are used to assess clustering quality (Section 4.1.1), and CCIC-O, an orientation benchmark derived from one of these datasets, is used to assess fine-grained ordinal organization (Section 4.1.2). All annotations are used exclusively for evaluation and are not used during training.

4.1.1. Binary crack datasets

We use three public datasets for the clustering experiments. Table 2 summarizes their main properties.

CCIC. The Concrete Crack Images for Classification (CCIC) dataset (Özgenel, 2019, Özgenel and Gönenç Sorguç, 2018) contains 40,000 RGB images with a resolution of 227 × 227 pixels and is evenly split into intact and cracked samples. The images are patch crops extracted from 458 original photographs with a resolution of 4032 × 3024 pixels. The original photographs show concrete surfaces of real buildings captured under natural lighting using a smartphone camera. Crack patterns range from fine hairline cracks to more pronounced damage, while intact samples include typical concrete textures and background structure. Overall, the surface appearance and acquisition conditions are relatively homogeneous, which results in clearly defined visual categories.

CPC. The Concrete and Pavement Crack (CPC) dataset (Omoebamije, 2023) consists of 30,000 RGB images of concrete and pavement surfaces with a resolution of 227 × 170 pixels and an equal number of intact and cracked samples. Data acquisition was performed at the Nigerian Army University Biu in Borno State, Nigeria, using a DJI Mavic 2 Enterprise drone for elevated structures and a smartphone for surfaces closer to ground level. This mixed recording setup introduces variation in viewpoint and scale. In addition, the dataset is visually heterogeneous across scenes, including different surface types and strong variation in texture and color, which can affect crack visibility.

BCD. The Bridge Crack dataset (BCD) (Xu et al., 2019) contains images of concrete bridge surfaces and originates from studies on automated bridge crack detection using convolutional neural networks. It is based on an earlier bridge crack dataset that was artificially enhanced. The dataset focuses on bridge-specific textures and crack patterns that differ from those found on building facades and pavements. The dataset contains 6,069 images with a resolution of 224 × 224 pixels. The surface appearance is very homogeneous, while cracks are often less pronounced than in CCIC. In this work, we use a balanced subset of 4,000 images.

Together, these datasets cover different surface types, acquisition conditions, and crack appearances, supporting an analysis of robustness and generalization for the proposed unsupervised clustering approach.

Table 2
Overview of the datasets used for clustering.

Dataset	No. of Images	Resolution	Damage Types	Label Type
CCIC	40,000	227 × 227	Intact, Crack	Class Label
CPC	30,000	227 × 170	Intact, Crack	Class Label
BCD*	4,000	224 × 224	Intact, Crack	Class Label

* Balanced subset of the full BCD dataset (6,069 images); see Section 4.1.1.

4.1.2. CCIC-O: an orientation crack benchmark for fine-grained evaluation

The topology-preserving aspect of our framework is most informative when evaluated on tasks in which the underlying data exhibit gradual, ordinal variation, allowing neighborhood relations on the SOM grid to be compared with meaningful similarity relations in data space. Binary or categorical labels without intrinsic ordering do not reveal topology preservation to the same extent. We therefore evaluate the framework in a setting where crack appearance varies along an ordinal axis, namely orientation. Since no publicly available crack dataset offers the combination of fine-grained ordinal labels and tightly controlled appearance variation required for this evaluation, we construct the benchmark ourselves.

Among the practically relevant morphological dimensions of cracks, both orientation and width are natural candidates, since they may reflect relevant structural conditions and damage-related processes. However, we choose orientation for a methodological reason related to the manual validation step. Orientation is visually unambiguous: a reviewer can readily determine whether a crack aligns with a chosen reference axis, for example the horizontal axis, and any subsequent rotation by a known angle preserves this property exactly. Width, by contrast, does not admit an equally clear visual definition. Mean width, maximum width, and width measured with or without accounting for the crack-edge gradient all yield different ground-truth values, none of which can be assessed reliably by visual inspection.

The dataset is constructed semi-automatically from CCIC. A subset of source images dominated by a single, edge-to-edge linear crack is first selected through an automated filtering pipeline and automatically aligned to a common horizontal reference orientation. The resulting images are then manually validated by the authors to exclude samples in which the crack does not form a single linear structure or in which the automatic reference alignment is visibly incorrect. From this validated base set of 1,008 images, rotated variants are synthesized in 10° increments over the full circle, yielding 36 physical orientations per source image. To avoid synthetic borders introduced by rotation, all images are reflectively padded prior to rotation and then center-cropped back to the original resolution of 227 × 227 pixels. Full construction details are reported in Appendix B.

The resulting benchmark, denoted CCIC-O (CCIC-Orientation), comprises 36,288 images at a resolution of 227 × 227 pixels and includes 36 physical rotation classes generated in 10° increments. In

addition, the orientation measured for each crack during alignment is retained as a continuous label for the original, un-rotated source image. Since this is the same angle that defines the alignment, the manual validation confirms the label together with the alignment. Notably, since linear cracks exhibit an inherent directional ambiguity, a crack at orientation θ is geometrically equivalent to one at $\theta + 180^\circ$, differing only in direction. We therefore treat orientation as defined modulo 180° in all experiments, collapsing the 36 physical classes into 18 effective ordinal classes.

All quantitative experiments evaluate orientation organization under two configurations, rotated and natural, each comprising N = 1,008 images. This number is anchored by the natural configuration, which comprises all 1,008 validated source cracks, and rotated is sized to match.

- Rotated: One rotated view of each crack, taken from CCIC-O, with the 36 orientations equally represented (28 cracks each, corresponding to 56 per effective orientation class under the 180° symmetry). This gives a balanced distribution and discrete labels in 10° steps. Since each crack appears only once, the result cannot come from repeated instances of the same crack.
- Natural: The original, un-rotated cracks from which CCIC-O was built, labeled with the orientations measured for them during that construction. This is the real data, with real appearance, continuous labels, and a naturally uneven orientation distribution.

Fig. 3 shows the resulting distribution of natural orientations, which is strongly concentrated near the horizontal (0° and 180°) and thins out toward the vertical (90°), in contrast to the uniform rotated configuration (56 cracks per class).

Both configurations are used and reported throughout, rotated as the balanced, discrete-label setting and natural as the real, continuous-label counterpart. Since the natural labels are continuous, some label-based metrics require a small adaptation, which is described in Section 4.2. The dataset is publicly available at https://doi.org/10.5281/zenodo.20042918 (Özcan, 2026). The rotated configuration is a direct subsample of it, and the natural configuration is based on the same cracks in their original, un-rotated form.

4.2. Evaluation metrics

We evaluate the proposed method along six complementary dimensions. External label agreement metrics quantify the consistency between the SOM-induced grouping and reference labels. Internal embedding geometry metrics characterize compactness and separation in the embedding space without using labels. Ordinal organization metrics assess how well the ordinal label factor is arranged across the map, from the local scale within a node, through neighboring nodes, up to the global scale of the entire map. Topology preservation metrics assess how faithfully the SOM map structure reflects the underlying data topology. Prototype quality metrics describe how well the SOM prototypes represent the embedding distribution. Training stability is assessed by measuring the consistency of sample assignments over the course of training. Two map-based diagnostics, namely the hit map and the U-matrix, complement the quantitative metrics with qualitative inspection of the trained map. Labels are used only for evaluation, not during training.

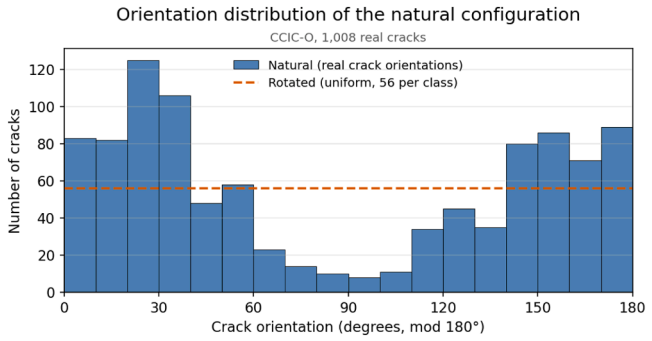


Fig. 3. Orientation distribution of the 1,008 real, un-rotated cracks in the natural configuration (10° bins, modulo 180°), in contrast to the uniform rotated configuration.

Table 3 summarizes the metrics used in this work, grouped by dimension, together with their value range and the direction of better performance. Formal definitions are provided in Appendix A.

The external label agreement metrics and LRC are defined on discrete classes and are therefore evaluated on the 18 effective orientation classes. In the rotated configuration these are the native classes. For the continuous labels of the natural configuration, the orientations are grouped into 18 equal 10° bins over the 180° range, which matches the resolution of the rotated classes and keeps the results of the two configurations as comparable as possible. The label variance metrics V_{intra} and V_{inter} operate on the continuous orientations directly and require no adaptation. TW is the discrete-class counterpart of V_{inter} and is reported only for the rotated configuration, since V_{inter} already measures the same local label transitions on the continuous orientations without any binning. The remaining metrics do not use labels and are unaffected. For the binary datasets, the class labels are discrete and no binning is involved.

4.3. Encoder architecture

The encoder is a custom CNN that is able to process grayscale images or patches of variable spatial size. It is designed to capture crack patterns across different scales, ranging from fine hairline structures to wider and more pronounced cracks, while also preserving broader surface context.

The network progressively aggregates spatial information through a stack of convolutional blocks at increasing channel widths, organized into stages separated by max-pooling. Each convolutional layer is followed by batch normalization and ReLU activation, and max pooling is used to progressively reduce spatial resolution. After the final convolutional block, global average pooling aggregates each feature map into a single value, yielding a fixed-length feature vector independent of

Table 3
Overview of evaluation metrics.

Dimension	Metric	Range	Direction
External Label Agreement	Purity (Pur)	[0, 1]	higher is better
External Label Agreement	Normalized Mutual Information (NMI)	[0, 1]	higher is better
External Label Agreement	Adjusted Rand Index (ARI)	[-1, 1]	higher is better
Internal Embedding Geometry	Davies-Bouldin Index (DBI)	[0, ∞)	lower is better
Internal Embedding Geometry	Silhouette Coefficient (SC)	[-1, 1]	higher is better
Ordinal Organization	Intra-Node Label Variance (V_{intra})	[0, 45]	lower is better
Ordinal Organization	Inter-Node Label Variance (V_{inter})	[0, 90]	lower is better
Ordinal Organization	Label Rank Correlation (LRC)	[-1, 1]	higher is better
Topology Preservation	Topographic Error (TE)	[0, 1]	lower is better
Topology Preservation	Transition Width (TW)	[0, 9] for 18 circular classes	lower is better
Prototype Quality	Quantization Error (QE)	[0, 2]	lower is better
Prototype Quality	SOM Usage Evenness (SUE)	[0, 1]	higher is better
Training Stability	Assignment Stability (AS)	[0, 1]	higher is better

the input resolution. Dropout is applied after global average pooling to reduce overfitting. The resulting features are then mapped into the contrastive embedding space using a two-layer projection head.

The specific architecture of our encoder f_Θ is defined by the following sequence of layers. Each Conv2D uses a 3 × 3 kernel, no bias, and is followed by BN + ReLU:

- (Conv2D (32 filters) → BN → ReLU) x 2 → MaxPooling2D
- (Conv2D (64 filters) → BN → ReLU) x 2 → MaxPooling2D
- (Conv2D (128 filters) → BN → ReLU) x 2 → MaxPooling2D
- Conv2D (128 filters) → BN → ReLU
- Global Average Pooling → Dropout (rate: 0.3)
- Dense (128 neurons) → BN → ReLU
- Dense (128 neurons)

This yields a backbone of seven convolutional blocks. The final 128-dimensional embedding vector is L2-normalized:

$$z = \frac{f_\Theta(x)}{\| f_\Theta(x) \|_2} \tag{26}$$

where $f_\Theta(x)$ denotes the encoder output for input x. This normalization enables similarity to be measured using cosine similarity and helps maintain a consistent embedding scale. The resulting 128-dimensional embeddings are used as input to the SOM.

The same encoder architecture is used across all experiments, both for the binary clustering task and for the fine-grained orientation task.

4.4. Training configuration

This section describes the training configuration used for all experiments, including hyperparameter setting and a brief overview of the training behavior.

4.4.1. Hyperparameters

The main hyperparameters are shared across all experiments and are summarized in Table 4. The augmentation set is the only component that differs between the binary and the orientation experiments and is reported separately in Table 5 (Section 4.4.2).

4.4.2. Augmentations

The augmentation parameters for the contrastive view generation are listed in Table 5. All images are grayscale with intensities in [0, 1], geometric transforms fill borders by reflection padding, and p denotes the probability of applying a transform to a view. Both tasks share the same geometric and appearance-related augmentations listed in Table 5, which makes the encoder invariant to position, scale, and appearance while preserving the orientation of the cracks.

For the binary experiments, rotation, flips, and slight shear are added, which makes the encoder slightly more orientation-invariant. These transforms are excluded for the orientation experiments, since they would alter the orientation, which is the label there.

Table 4
Hyperparameters used across all experiments.

Parameter	Value
Input size	64 × 64
Training epochs	50
Batch size	32
Embedding dimension	128
Optimizer (encoder)	Adam
Learning rate (encoder)	2×10^{-5}
Dropout (encoder)	0.3
SOM grid size	11 × 11
SOM loss weight λ_{start} and λ_{end}	0.1 and 0.9
SOM learning rate η_{start} and η_{end}	0.015 and 0.01
SOM neighborhood width σ_{start} and σ_{end}	grid side length (here 11) and 0.75
Contrastive temperature τ_c	0.5
Soft-assignment temperature τ_t	0.5
Sigmoid slope k	15
Sigmoid inflection point t_0	0.5

Table 5
Augmentation parameters used for contrastive view generation.

Transformation	Binary Experiments	Orientation Experiments
Brightness	$\delta \in$ [-0.10, 0.10], p=1.0	identical
Contrast	factor $\in$ [0.85, 1.15], p=1.0	identical
Gamma	$\gamma \in$ [0.85, 1.15], p=1.0	identical
Additive Gaussian noise	$\sigma \in$ [0.01, 0.025], p=0.5	identical
Gaussian blur	$\sigma \in$ [0.3, 0.9], 5 × 5 kernel, p=0.5	identical
Translation	$\in$ [-10%, 10%], p=0.5	identical
Isotropic zoom	scale $\in$ [0.90, 1.10], p=0.5	identical
Rotation	$\in$ [-3°, 3°], p=0.7	not used
Horizontal flip	p=0.5	not used
Vertical flip	p=0.5	not used
Shear	$\in$ [-0.05, 0.05], p=0.5	not used

4.4.3. Training behavior

Loss components and evaluation metrics are recorded throughout training to monitor learning dynamics. The contrastive and topology-guided objectives co-adapt as expected from the curriculum schedule, with the contrastive loss decreasing rapidly in early epochs and the topology-guided loss stabilizing later as the SOM prototypes align with the evolving embeddings. Purity, NMI, and the prototype quality metrics improve over training and stabilize toward the end. ARI and the topographic error instead reach their best values before the end of training and then degrade as the topology weight increases, which is the trade-off analyzed in Section 6.3.2. All reported values are those of the final epoch, and no metric-based model selection is applied. Detailed loss and metric curves for CCIC-O, as an illustrative example, are provided in Appendix C.

A typical training run completes in approximately 10 minutes over 50 epochs on the CCIC-O experimental subset with 1,008 image samples, and scales roughly with the number of samples. The implementation is research-grade and not optimized for runtime performance, so the reported duration is intended as a rough practical reference rather than a benchmark.

4.5. Computational environment

All training and evaluation steps were carried out on a laptop running Windows 11 Pro (64-bit). The main hardware components included:

- Processor: Intel Core i9-13950HX (13th generation, 24 cores, base clock speed 2.2 GHz)
- Memory: 64 GB DDR5 RAM
- Graphics card: NVIDIA RTX 3500 Ada Generation Laptop GPU with 12 GB dedicated VRAM

The implementation was developed in Python 3.10, using the following libraries for both training and evaluation:

- TensorFlow 2.15 – for deep learning models
- NumPy 1.26 – for numerical operations
- scikit-learn 1.4 – for evaluation routines
- OpenCV 4.9 – for image processing
- Matplotlib 3.8 – for visualizations

5. Results

In this section, we present the results of the proposed topology-

guided image-level organization. We first evaluate binary separation on three public crack datasets and then assess fine-grained topology preservation on the orientation-based CCIC-O dataset.

5.1. Binary crack clustering

We evaluate the proposed framework on the binary separation task between crack and intact samples using the three benchmark datasets CCIC, CPC, and BCD. In this setting, the aim is to assess whether the SOM-induced organization agrees with this coarse distinction, although no class labels are used during training. Since the SOM consists of 121 nodes for only two classes, purity is the most informative metric in this setting, whereas chance-corrected agreement metrics are dominated by the resulting spread of each class across many nodes (Section 6.3.1). Accordingly, Table 6 reports only purity. The fine-grained evaluation is therefore carried out on CCIC-O.

The results show high purity values across all three datasets, ranging from 0.9379 on CPC to 0.9921 on CCIC. This indicates that the learned SOM organization provides a clear separation between crack and intact samples under different dataset conditions. The highest value is obtained on CCIC, whose visual categories are comparatively well defined, while the slightly lower value on CPC is consistent with its stronger visual heterogeneity, including variation in surface type, texture, color, viewpoint, and scale.

To complement the quantitative evaluation, Appendix D provides the full qualitative results, including the SOM organization overviews, hit maps, U-matrices, and t-SNE projections for all three datasets. We summarize the main observations here.

Across the datasets, intact samples are generally mapped into coherent regions with comparatively homogeneous appearance, while crack-related samples occupy broader regions that reflect variation in orientation, width, and local texture. The U-matrices indicate clear transitions between predominantly crack and predominantly intact regions, while neighboring nodes within the same broad region remain visually related.

Overall, the binary experiments show that TopoCLR can organize the two categories into clearly distinguishable regions without supervision. At the same time, the SOM retains within-class variation rather than reducing the output to a single binary partition. This motivates the more fine-grained evaluation on CCIC-O in the following section.

5.2. Fine-grained orientation organization

We next evaluate the proposed framework on CCIC-O, an orientation-based dataset with 18 effective orientation classes. In contrast to the binary datasets, CCIC-O is designed to evaluate whether the learned SOM organization preserves gradual crack-related variation. Since the orientation labels define an ordered structure modulo 180°, neighboring classes represent gradual changes in crack direction. A topology-preserving organization should therefore arrange related orientations in neighboring map regions, rather than treating the classes as independent categories. This experiment therefore evaluates whether TopoCLR can organize crack images according to an ordered morphological property, while still maintaining class-level separation where orientations differ strongly.

5.2.1. Quantitative results

The quantitative evaluation uses all 13 metrics defined in Section 4.2, including the three ordinal organization metrics that are particularly relevant for this fine-grained setting. All values are reported as mean values with standard deviations across five independent runs with different seeds (Table 7).

Table 6
Quantitative results on the binary crack datasets CCIC, CPC, and BCD.

Dataset	CCIC	CPC	BCD
Purity	0.9921	0.9379	0.9407

The results show that TopoCLR organizes the orientation dataset in a structured and stable way, in both the rotated and the natural configuration. The external agreement metrics indicate a clear relation between the SOM-induced organization and the 18 effective orientation classes. Purity and NMI show that many map regions are dominated by specific orientations or by closely related orientations. At the same time, ARI remains more moderate, indicating that the classes are not represented as sharply separated compact clusters. This is consistent with the objective of preserving gradual orientation transitions rather than enforcing isolated class partitions.

The ordinal organization metrics support this interpretation. The high LRC in the rotated configuration indicates that the learned organization preserves the ordinal structure of the orientation labels. Thus, orientations that are close in label space are also arranged in related regions on the SOM, and the intra- and inter-node label variances confirm that individual nodes and their neighborhoods remain orientation-coherent. The topology preservation metrics show that the map captures gradual transitions. At the same time, not all neighborhood relations of the embedding are reproduced on the grid, which is consistent with the cyclic orientation axis being embedded in a square lattice. The prototype quality and training stability metrics further indicate that the map represents the embedding distribution well, is broadly utilized, and remains stable over training.

Both configurations show the same qualitative behavior, but the metrics that involve the orientation labels reach weaker values in the natural configuration. Two properties of that setting account for the difference. First, its orientation distribution is strongly uneven (Fig. 3), so the sparsely populated bins towards the vertical contain only a few cracks, which makes the corresponding class centroids on the grid noisy and lowers LRC in particular. Second, the continuous labels have to be grouped into the 18 orientation bins used for the discrete metrics (Section 4.2), which introduces a quantization of the label axis that the rotated configuration does not involve. The LRC of the natural configuration nevertheless remains clearly above that of every compared map-based method (Section 6.1), and the relative pattern across the metrics is unchanged, so the topology-preserving behavior of the learned organization is confirmed on fully real, un-rotated cracks.

5.2.2. Qualitative results

Fig. 4 shows the SOM organization for CCIC-O, with up to 4 image samples displayed for each BMU. The map shows a clear orientation-related structure, with neighboring nodes generally containing

Table 7
Quantitative results on CCIC-O, reported as mean values with standard deviations across five independent runs with different seeds. Arrows indicate the direction of better performance. For the natural configuration, Purity, NMI, ARI, and LRC use the 18 binned orientation classes (Section 4.2). Transition Width is reported only for the rotated configuration.

Category	Metric	rotated (mean ± std)	natural (mean ± std)
External Label Agreement	Purity ↑	0.8349 ± 0.0230	0.7143 ± 0.0291
	NMI ↑	0.6772 ± 0.0158	0.5931 ± 0.0287
	ARI ↑	0.2526 ± 0.0319	0.1692 ± 0.0336
Internal Embedding Geometry	DBI ↓	1.0756 ± 0.0206	1.1621 ± 0.0811
	SC ↑	0.1702 ± 0.0082	0.1666 ± 0.0208
Ordinal Organization	V_intra ↓	8.3395 ± 1.8506	16.4543 ± 4.0328
	V_inter ↓	24.6556 ± 5.7583	44.3316 ± 9.0723
	LRC ↑	0.9717 ± 0.0035	0.7121 ± 0.0204
Topology Preservation	TE ↓	0.1901 ± 0.0260	0.2552 ± 0.0639
	TW ↓	0.3689 ± 0.0465	n/a
Prototype Quality	QE ↓	0.1588 ± 0.0062	0.1486 ± 0.0238
	SUE ↑	0.9477 ± 0.0118	0.8981 ± 0.0268
Training Stability	AS ↑	0.9942 ± 0.0009	0.9910 ± 0.0010

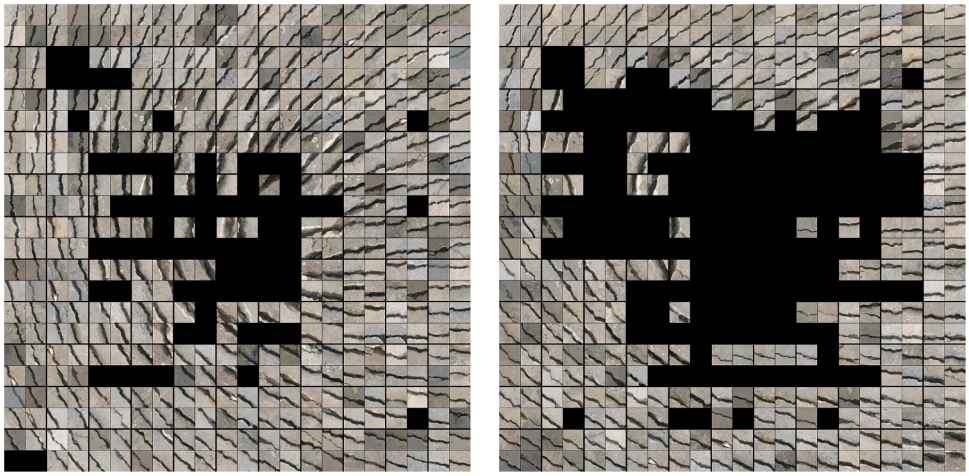


Fig. 4. Overview of the organized data of CCIC-O (left rotated, right natural), with up to 4 image samples mapped onto each SOM node.

visually similar crack orientations. Gradual changes in orientation are reflected by gradual movement across the grid, rather than by isolated and disconnected class regions. This is consistent with the label rank correlation reported in Table 7 and visually supports the topology-preserving behavior of the learned organization.

At the same time, the SOM is not uniformly occupied. The occupied nodes tend to form a ring-like structure, while several central nodes remain empty or sparsely populated. This pattern is consistent with the cyclic structure of the orientation labels. Since orientations are defined modulo 180°, neighboring classes form a closed sequence, while strongly different orientations appear in distant, approximately opposing map regions. As a result, central nodes may correspond to transitions between dissimilar orientation regions and receive few or no assignments. Visual inspection further suggests a secondary tendency along the radial direction of this ring-like structure, with crack appearance varying partly in thickness and visual prominence. This secondary trend is not evaluated quantitatively here and should therefore be interpreted only as a qualitative indication of additional visual ordering within the map.

In addition, Fig. 5 provides complementary views of the learned orientation structure. To summarize the orientation organization more compactly, we use an orientation proxy map in which each occupied SOM node is represented by the mean orientation of its assigned samples, while the corresponding standard deviation indicates local orientation variability. The proxy map indicates that neighboring nodes often have similar mean orientations and that orientation changes smoothly across large parts of the SOM. The t-SNE projection shows the orientation classes arranged along a closed curve in the learned feature space, reflecting the cyclic structure of the labels. The U-matrix highlights stronger transitions between map regions, while the hit map shows that the occupied nodes concentrate towards the periphery of the grid, while the central nodes remain largely empty.

Together, the qualitative results support the quantitative findings in Table 7. The learned map organizes related orientations in neighboring regions and shows a structured representation of gradual orientation-related crack variation.

6. Comparative studies

In this section, we analyze the proposed framework through three complementary studies. First, a benchmarking comparison positions TopoCLR against established unsupervised representation-learning and organization approaches. To assess practical generalization, this comparison is additionally evaluated across the two CCIC-O configurations, the rotated cracks and the original, un-rotated cracks. Second, ablation studies isolate the contributions of the contrastive and topology-guided objectives, comparing the full scheduled model against a topology-only ($\lambda = 1$) and a contrastive-only ($\lambda = 0$) variant. Third, sensitivity analyses assess the influence of the SOM grid size and the final topology-guided loss weight. All studies are conducted on CCIC-O, since this dataset provides a fine-grained orientation structure that makes differences in topology preservation and class-level organization more visible. Each experiment is repeated with five different random seeds, and results are reported as mean values with standard deviations. For the natural configuration, Purity, NMI, ARI, and LRC are computed on the 18 binned orientation classes and TW is omitted, following Section 4.2.

6.1. Benchmarking comparison

We compare TopoCLR against six baselines that represent different unsupervised strategies for representation learning, clustering, and topological organization:

- AE + k-means trains an autoencoder with a reconstruction loss and clusters the resulting bottleneck features post-hoc with k-means. This baseline represents reconstruction-based representation learning followed by conventional clustering.
- SimCLR + k-means trains a contrastive encoder with the NT-Xent loss and clusters the resulting embeddings post-hoc with k-means. This baseline represents contrastive representation learning without map-based organization.
- SimCLR + post-hoc SOM trains the same contrastive encoder and organizes the resulting embeddings on a SOM after training. This baseline tests whether a two-stage contrastive-learning and SOM pipeline can reproduce the behavior of joint training.
- DeepSOM, a baseline constructed for this comparison, trains a CNN encoder jointly with a SOM but without a contrastive loss, providing a topology-oriented baseline without explicit contrastive discrimination. Unlike the topology-only ablation of Section 6.2, it operates on single, unaugmented views and retains the gradual topology-weight schedule.
- DEC jointly learns representations and cluster assignments through a clustering-oriented KL-divergence objective without imposing a map-based organization.
- DESOM combines a reconstruction-based encoder with a SOM-style organization in the latent space and therefore replaces the contrastive signal of TopoCLR with an autoencoding signal.

To support comparability, all methods are trained under the same general conditions, including optimizer, batch size, and number of epochs. The contrastive baselines (SimCLR + k-means, SimCLR + post-hoc SOM) and DeepSOM share the encoder used by TopoCLR (Section

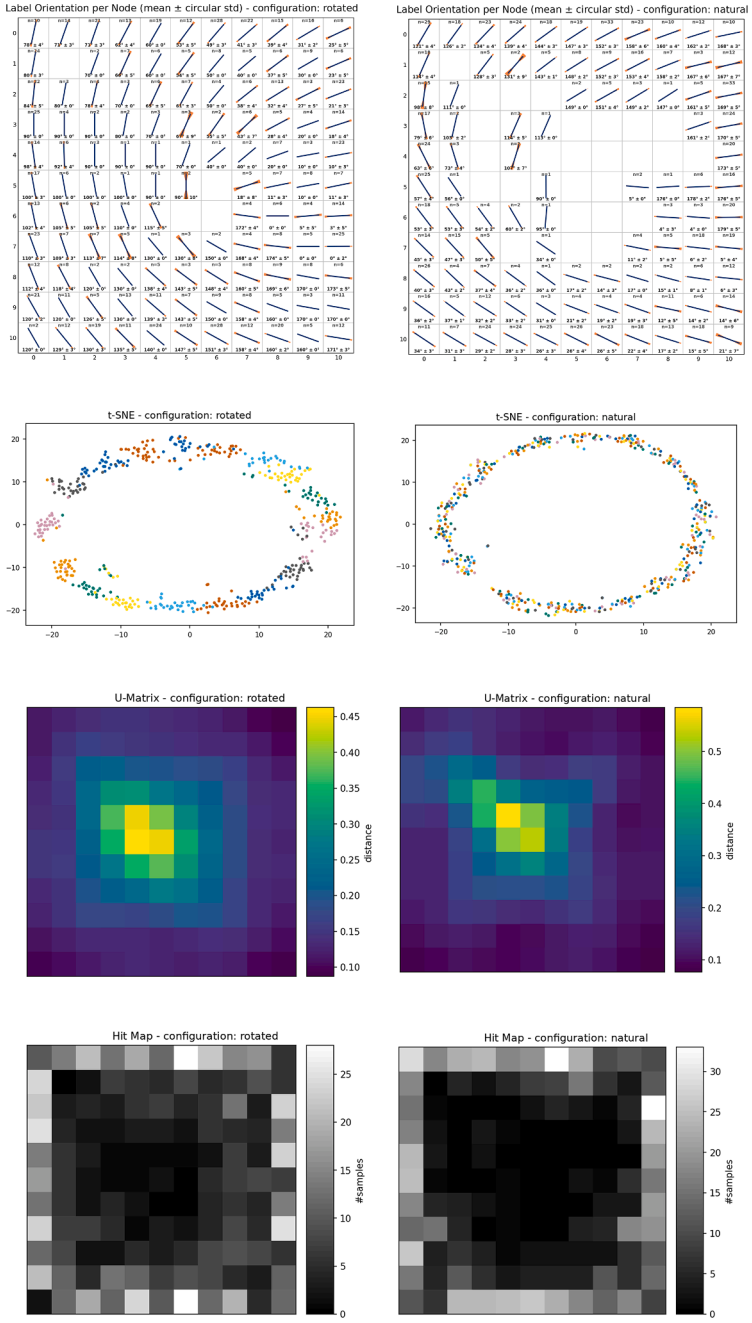


Fig. 5. CCIC-O orientation proxy map with per-node mean orientation and variability (top), together with t-SNE projection, U-matrix, and hit map (bottom) as complementary views of the learned structure. In each panel, the rotated configuration is shown on the left and the natural configuration on the right.

4.3), while AE + k-means, DEC and DESOM use their original autoencoder-based encoders, since the reconstruction objective is method-defining for these approaches. Augmentations are kept consistent where they are applicable to the respective method. SOM-specific hyperparameters are kept identical across all SOM-based methods, and DEC-specific hyperparameters follow the original formulation. All metrics are reported as mean values with standard deviations across five independent runs with different seeds. Arrows indicate the direction of better performance.

It should be noted that the compared methods differ conceptually in what their training objective can control. Several baselines are not designed to steer the criterion by which samples are discriminated and organized. Reconstruction-based methods are primarily shaped by input reconstruction, while post-hoc clustering or SOM variants organize representations that have already been learned independently of the final map structure. The benchmark therefore positions TopoCLR against representative unsupervised alternatives under comparable implementation conditions, but the differences in controllability of the organization criterion should be considered when interpreting the results.

Fig. 6 summarizes the benchmarking comparison across the two CCIC-O configurations. TopoCLR achieves the strongest overall performance on the external agreement metrics, leading on NMI and ARI in both configurations. Only for purity on the natural configuration does SimCLR + k-means reach a slightly higher mean, within one standard deviation. However, purity only measures how homogeneous the samples within each node are, and it is unchanged if the nodes are arbitrarily permuted across the grid, so it credits a fine partition while remaining

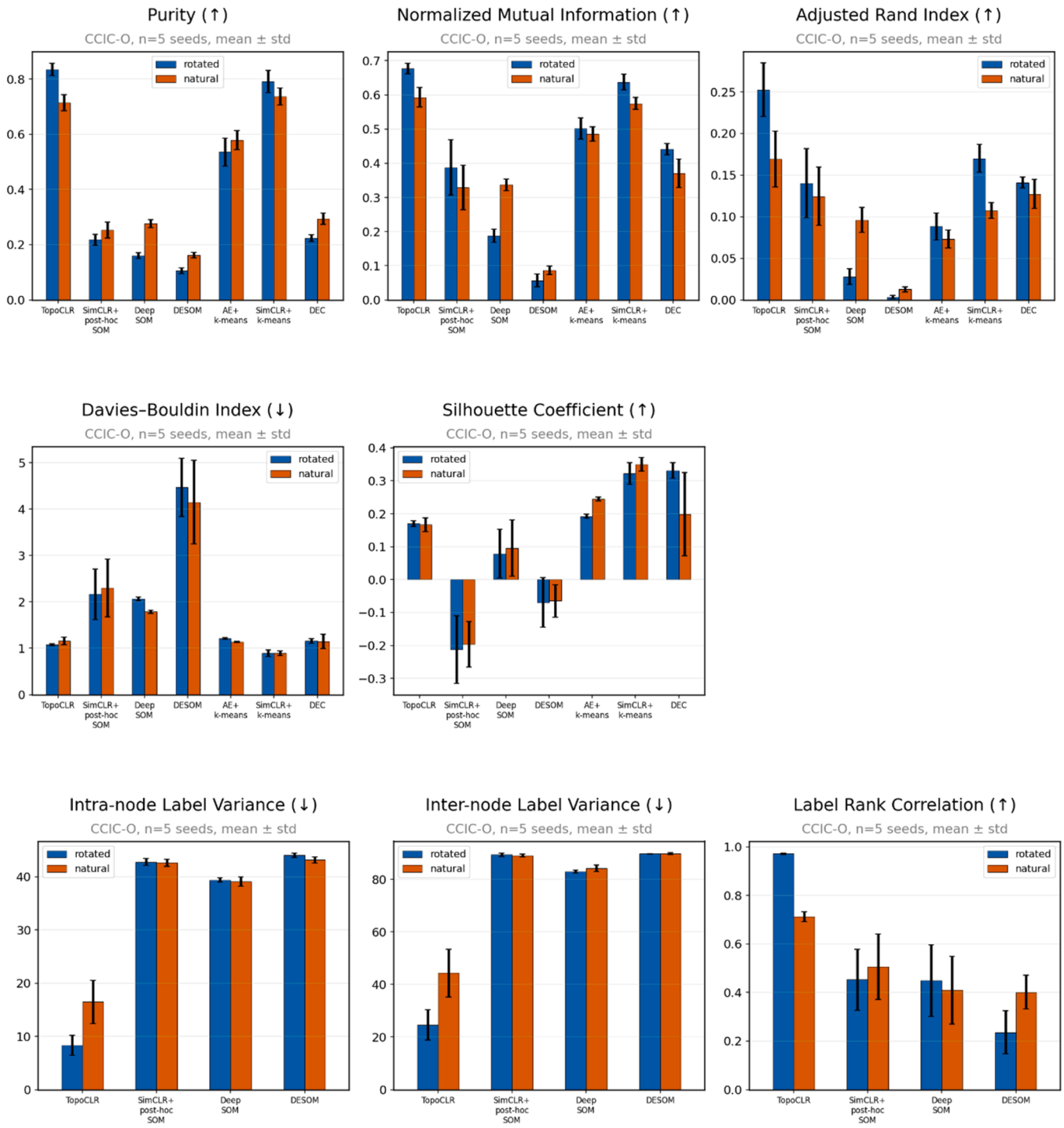


Fig. 6. Benchmarking comparison across the two CCIC-O configurations (rotated and natural) reported as mean values with standard deviations across five runs. External label agreement metrics and LRC for the natural configuration use the 18 binned orientation classes (Section 4.2). Metrics that require a map are shown only for the map-based methods, and transition width is reported only for the rotated configuration.

insensitive to whether related orientations end up next to each other. A flat clustering can therefore match a topology-preserving map on purity while providing no ordering at all, which is precisely what the map-based metrics below reveal. Beyond the external agreement metrics, some baselines are competitive on individual clustering metrics, for example a more favorable internal geometry on the Davies-Bouldin index and the Silhouette coefficient.

The clearest difference appears on the label-based topology metrics, which require a map and are therefore reported only for the map-based methods. TopoCLR leads by a wide margin, reaching the highest label rank correlation and the lowest label variation within and between nodes, so related orientations end up in neighboring map regions instead of being scattered. This advantage remains on the natural configuration, which shows that the improvement stems from the organization itself and not from the controlled rotation. The benchmark also isolates the role of the coupling. Organizing the same contrastive features with a post-hoc SOM instead of k-means adds a map that adapts smoothly to the embedding, reflected in its very low topographic error. Its weak external

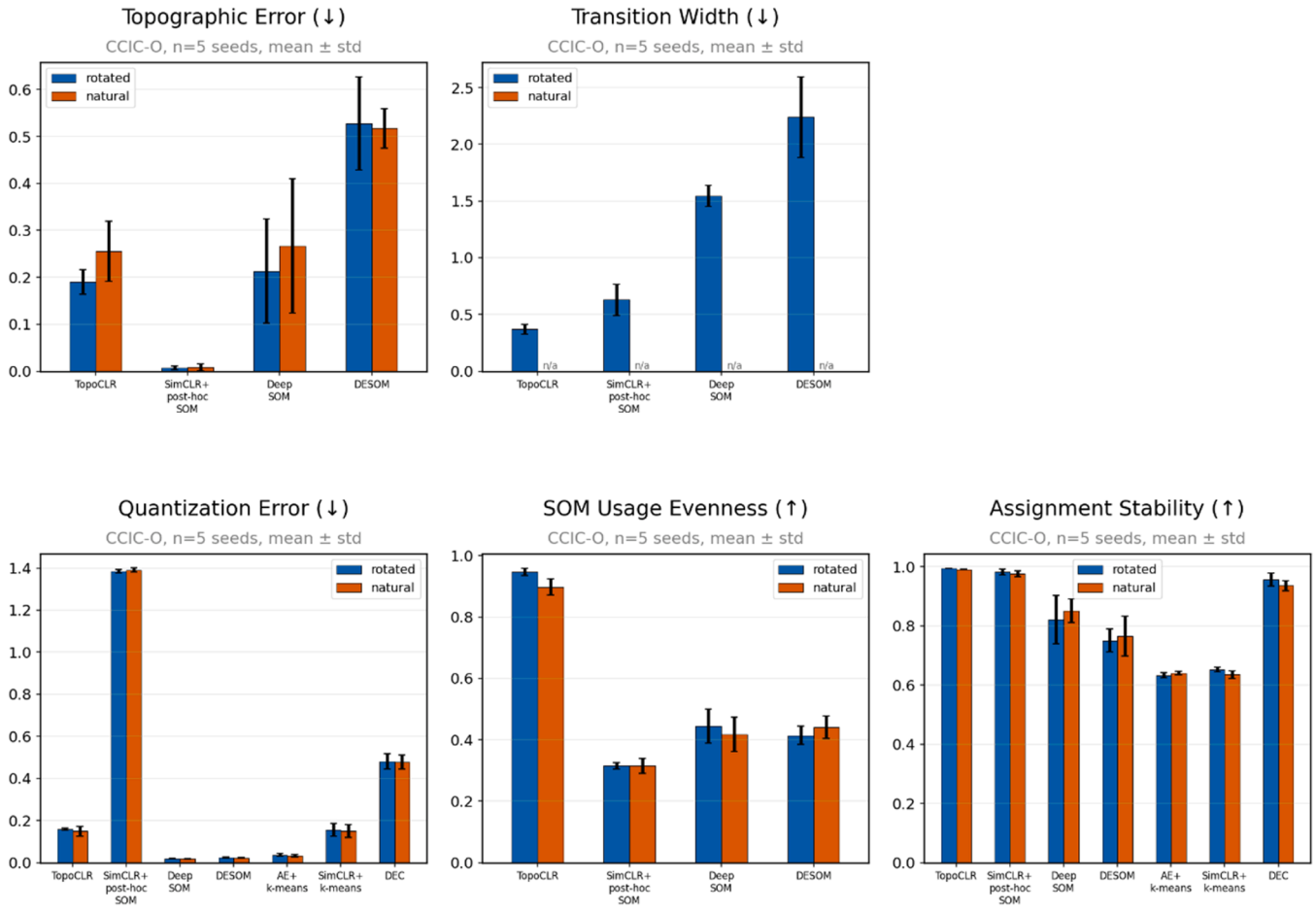


Fig. 6. (*continued*).

agreement and label metrics show, however, that this smoothness alone does not lead to an orientation-consistent organization. Coupling the map to the encoder during training, as in TopoCLR, raises the organization to the reported level. DESOM and DeepSOM, the two remaining map-based baselines, remain clearly behind on the orientation-related metrics.

Overall, no baseline matches TopoCLR across the full set of metrics. The methods that cluster well provide no ordered map, and the methods that provide a map organize the orientations clearly worse. Only the coupling of contrastive representation learning and SOM-based organization during training yields a representation that is both discriminative with respect to orientation and organized in a topology-preserving manner.

6.2. Ablation studies

To examine the contribution of the main training components, we perform two ablation studies. The first ablation replaces the scheduled topology weight with a constant value throughout training. In this variant, a constant weight of $\lambda = 1$ is used from the first epoch onward, so that the topology-guided term is active with full weight from the beginning and the contrastive term is switched off, which removes the gradual schedule and the contrastive objective together. The second ablation disables the topology-guided loss term entirely. In this case, the encoder is trained only with the contrastive objective, while the SOM is still updated through the competitive learning rule. This separates passive SOM adaptation to contrastive embeddings from the explicit encoder-level feedback provided by the topology-guided loss. All other hyperparameters follow the configuration in Section 4.4.1. All metrics are reported as mean values with standard deviations across five independent runs with different seeds. Arrows indicate the direction of better performance.

Fig. 7 reports the ablation results for all 13 metrics. Using the maximum topology weight from the first epoch strongly degrades the results, because, with the contrastive term switched off, the encoder is forced to align with the SOM without ever forming a discriminative feature space. Since this variant also removes the contrastive term, the schedule alone is examined in Section 6.3.2, where $\lambda_{end} = 1.0$ retains the gradual ramp at the same final weight and performs substantially better. Stability of the joint optimization therefore depends on the gradual increase of λ, not only on its final value. In contrast, disabling the topology-guided loss yields the strongest performance on several map-quality metrics. This is the expected behavior of a SOM that adapts passively to an already well-structured contrastive embedding without being pulled by the topology objective. However, the resulting external agreement (Purity, NMI, ARI) is substantially weaker than in the full configuration, indicating that the SOM organizes a less discriminative feature space. The full TopoCLR configuration represents a deliberate trade-off, sacrificing some topographic smoothness in exchange for a representation that is more strongly aligned with the orientation classes.

6.3. Sensitivity analyses

To characterize the robustness of the framework with respect to central design parameters, we perform two sensitivity analyses on CCIC-O with both configurations. The first analysis varies the SOM grid size, while the second varies the final topology-guided loss weight. All other settings follow the training configuration in Section 4.4. Each setting is

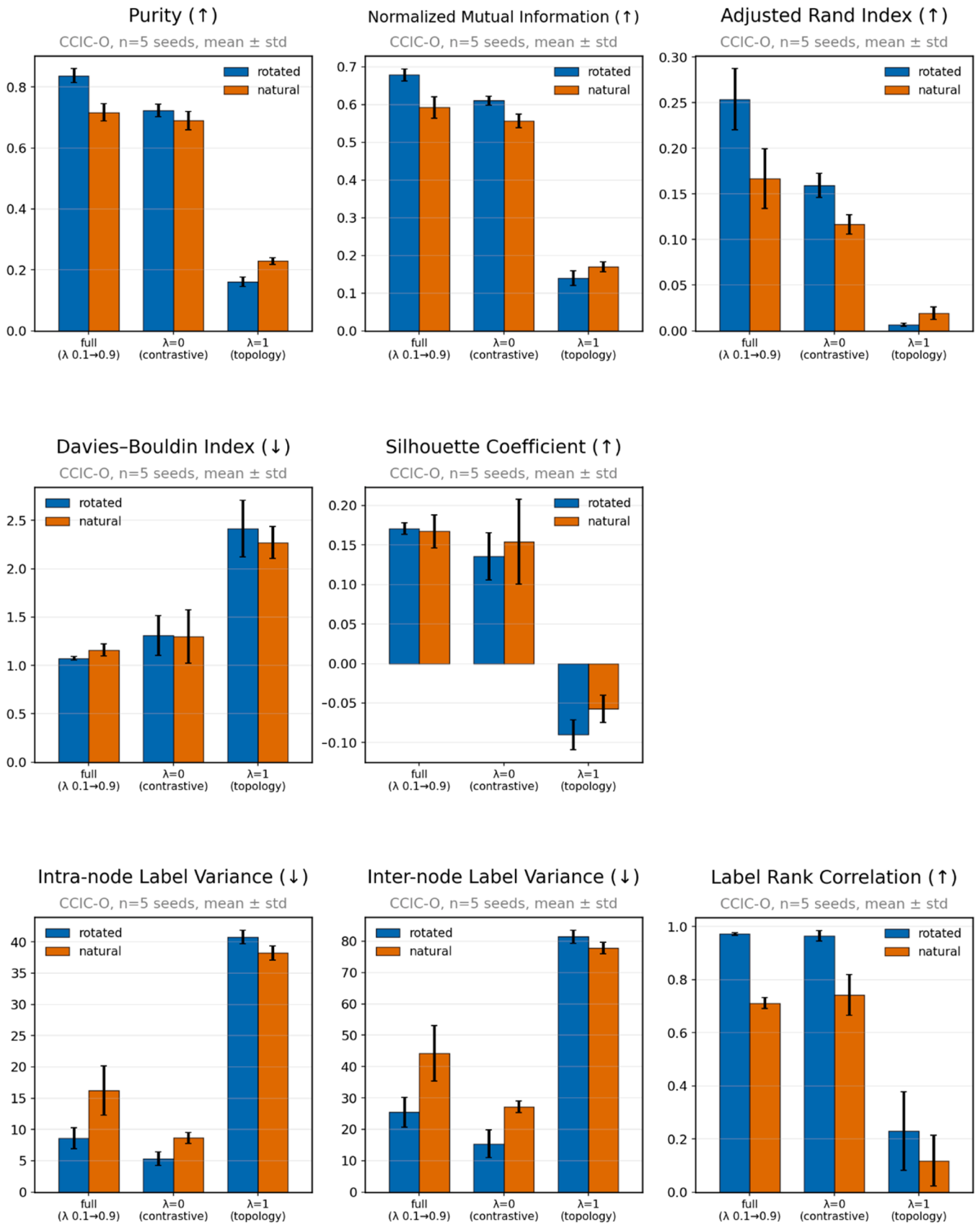


Fig. 7. Ablation study on CCIC-O (with both configurations) for all 13 metrics, comparing the full TopoCLR framework with two ablated variants.

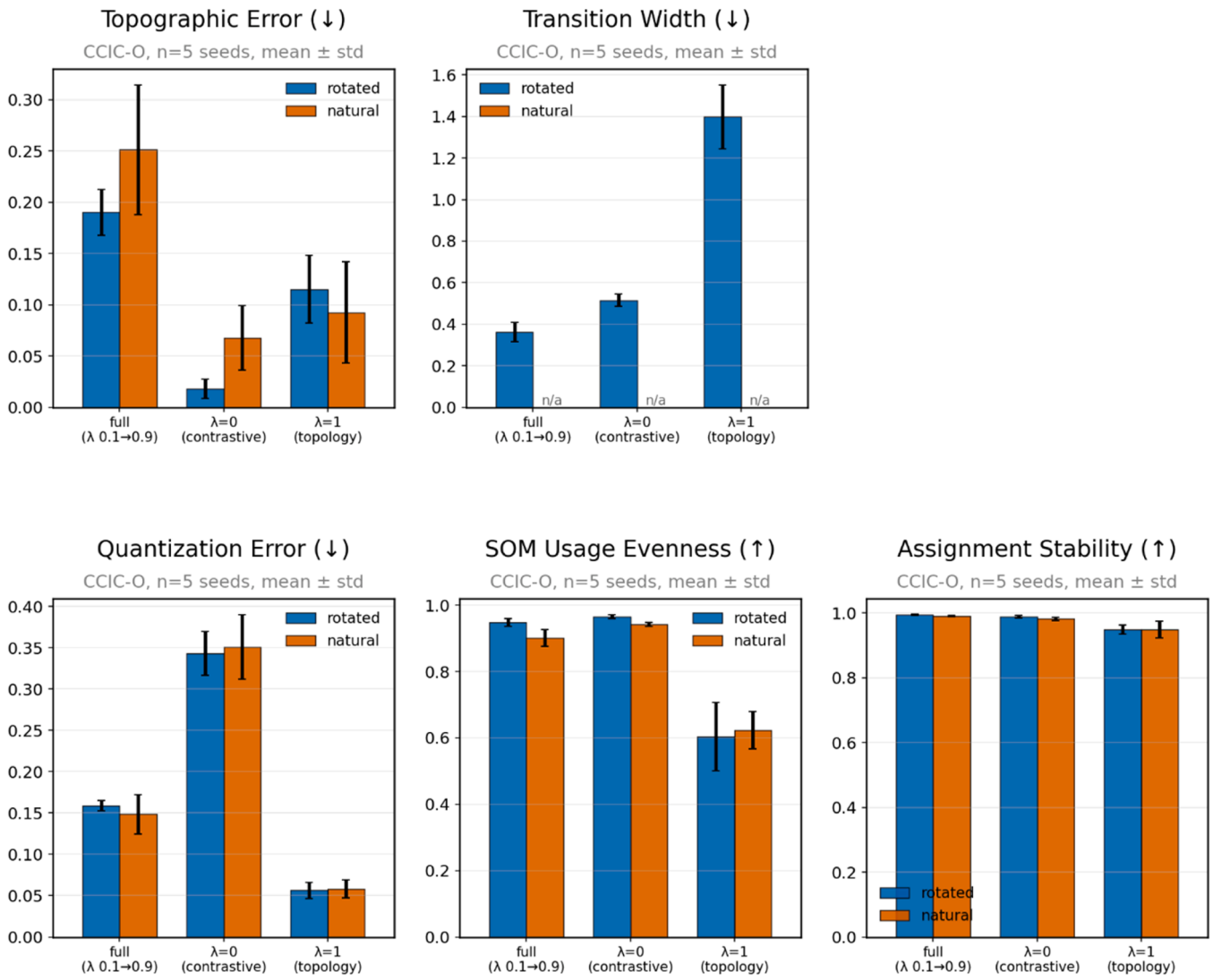


Fig. 7. (*continued*).

evaluated across five independent runs with different random seeds, and metrics are reported as mean values with standard deviations. Arrows indicate the direction of better performance.

6.3.1. SOM grid size

The grid size determines the number of available prototypes and the granularity of the topological organization. We evaluate seven grid sizes ranging from 3×3 to 15×15 in steps of two, while keeping all other hyperparameters fixed, except the initial neighborhood width σ_{start}, which is set to the grid side length and therefore scales with the grid size.

Fig. 8 shows the effect of the SOM grid size on the evaluation metrics. The grid-size analysis reveals two complementary trends. Purity increases with grid size, since more nodes provide more flexibility for majority-vote labeling. In contrast, NMI and ARI peak at smaller grid sizes and decrease for larger grids. This suggests that finer grids split coherent orientation regions into several sub-regions, which is penalized by chance- and entropy-corrected metrics but not by purity. The topology preservation metrics generally benefit from larger grids, although TE indicates that finer map resolutions can also introduce additional local discontinuities in the discrete neighborhood structure. Training stability and prototype quality remain comparatively stable across the tested grid sizes.

This pattern reflects the relation between node count and class count, as NMI and ARI peak when the number of nodes is close to the 18 effective orientation classes. As a practical guideline, the grid size should be chosen according to the intended granularity of the organization. Smaller grids yield a coarser grouping into few broad regions, whereas larger grids resolve gradual transitions and fine-grained variability in more detail. When discrete classes are expected, a node count on the order of the expected class count favors chance-corrected clustering agreement, while node purity benefits from a substantially larger grid. Overall, the 11×11 grid provides a balanced default rather than an optimum for every metric, combining high external agreement with sufficient map resolution for gradual orientation transitions.

6.3.2. Topology weight

While the ablation in Section 6.2 removes the gradual schedule entirely by applying the maximum topology weight ($\lambda = 1$) from the first epoch, this analysis examines how the final strength of the topology-guided loss affects clustering quality and topology preservation. We vary the final topology weight from 0.1 to 1.0 in steps of 0.1, while keeping the initial value fixed at 0.1 and leaving the sigmoid schedule otherwise unchanged.

Fig. 9 shows the effect of the final topology weight on the evaluation metrics. The topology-weight analysis shows a trade-off between discriminative organization, prototype fitting, and topological smoothness. Increasing the final topology weight generally improves external agreement and internal cluster geometry up to a certain range,

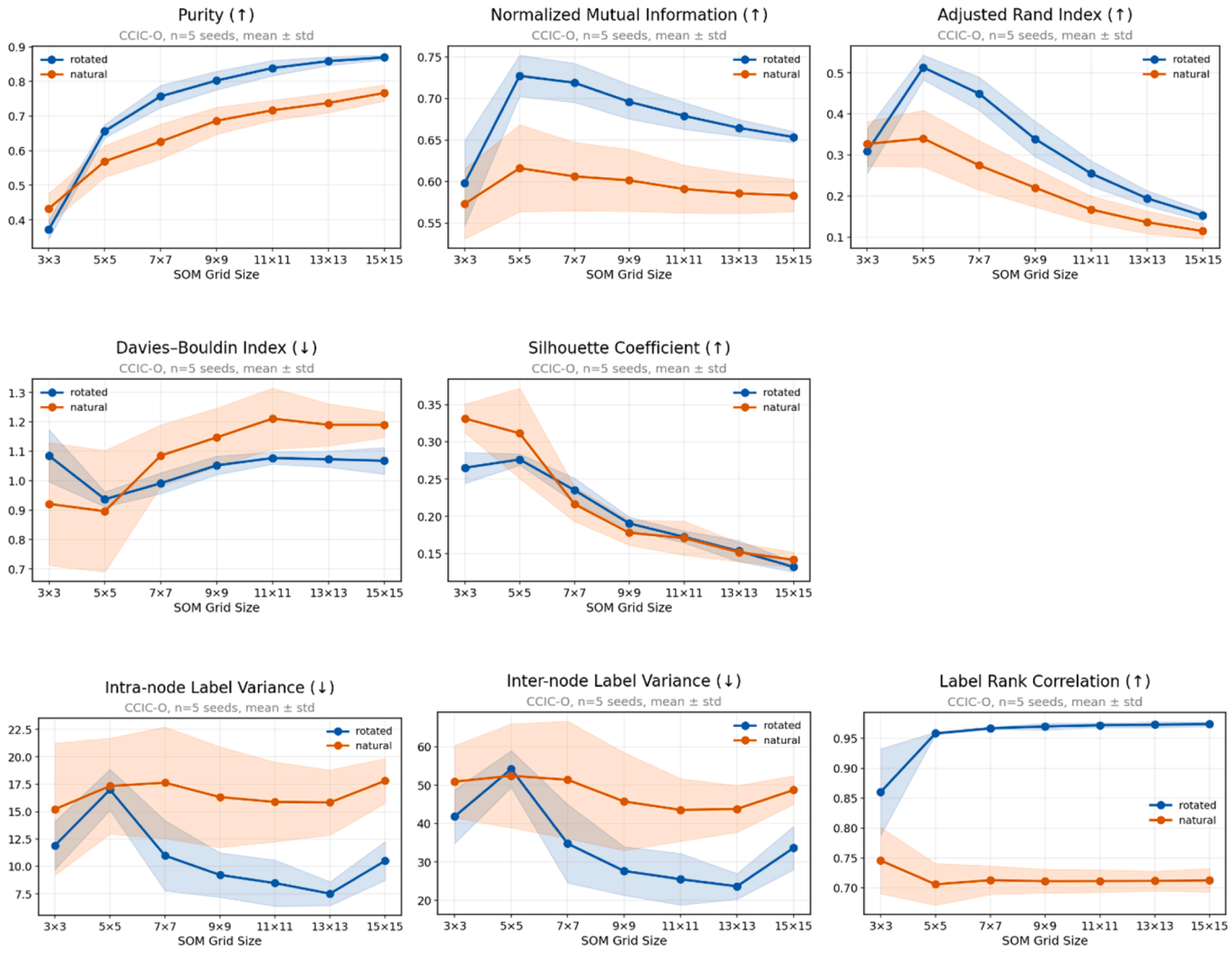


Fig. 8. Sensitivity analysis on CCIC-O (with both configurations), showing the effect of different SOM grid sizes on the evaluation metrics.

indicating that stronger topology-guided coupling helps align embeddings with the SOM prototypes. This is also reflected by the quantization error, which decreases as the coupling becomes stronger.

At the same time, very high topology weights can introduce local discontinuities in the map organization. This is visible in less favorable topology preservation and utilization metrics at the highest tested weights. These results indicate that moderate to high topology weights improve map alignment and clustering quality, whereas an excessively strong topology-guided term can reduce smoothness and balanced map utilization. Overall, an intermediate-to-high final topology weight provides the strongest empirical compromise in this experiment.

7. Discussion

In this section, we first interpret the results of the topology-guided organization of crack images. Subsequently, we reflect on the methodological design, including its strengths and limitations, and position the method with respect to related work and practical implications.

7.1. Analysis of results

Across the evaluated datasets, the SOM-based representation generally yields a clear separation between crack-dominated and intact surface patterns. The main remaining source of variability is often surface texture, which mainly affects local neighborhoods and transition areas, while the overall separation between crack and intact remains stable even when texture heterogeneity is pronounced. At the same time, these findings should be interpreted within the evaluated dataset conditions. The current datasets do not fully cover adverse inspection conditions such as strong shadows, water stains, severe surface contamination, highly variable crack width within the same image, and complex background clutter, which may introduce additional challenges and are not exhaustively represented here.

Within the crack-dominated region, the map shows structured regions and gradual transitions that align with typical variations in crack manifestation. Differences in dominant orientation and severity-related appearance cues tend to appear as neighboring areas rather than as isolated clusters, indicating that the organization retains fine-grained variability while maintaining a clear global separation.

Crucially, such fine-grained organization is important because variations in crack appearance carry information beyond a binary distinction between crack and intact. Differences in orientation, width, and continuity can relate to severity and may provide hints about possible underlying causes. Preserving these gradual changes therefore provides a basis for visual and systematic comparison of crack appearances.

In the complementary patch-based proof of concept, the back-projection provides coarse spatial crack localization in high-resolution images. When crack structures are distinct, the reconstructed masks

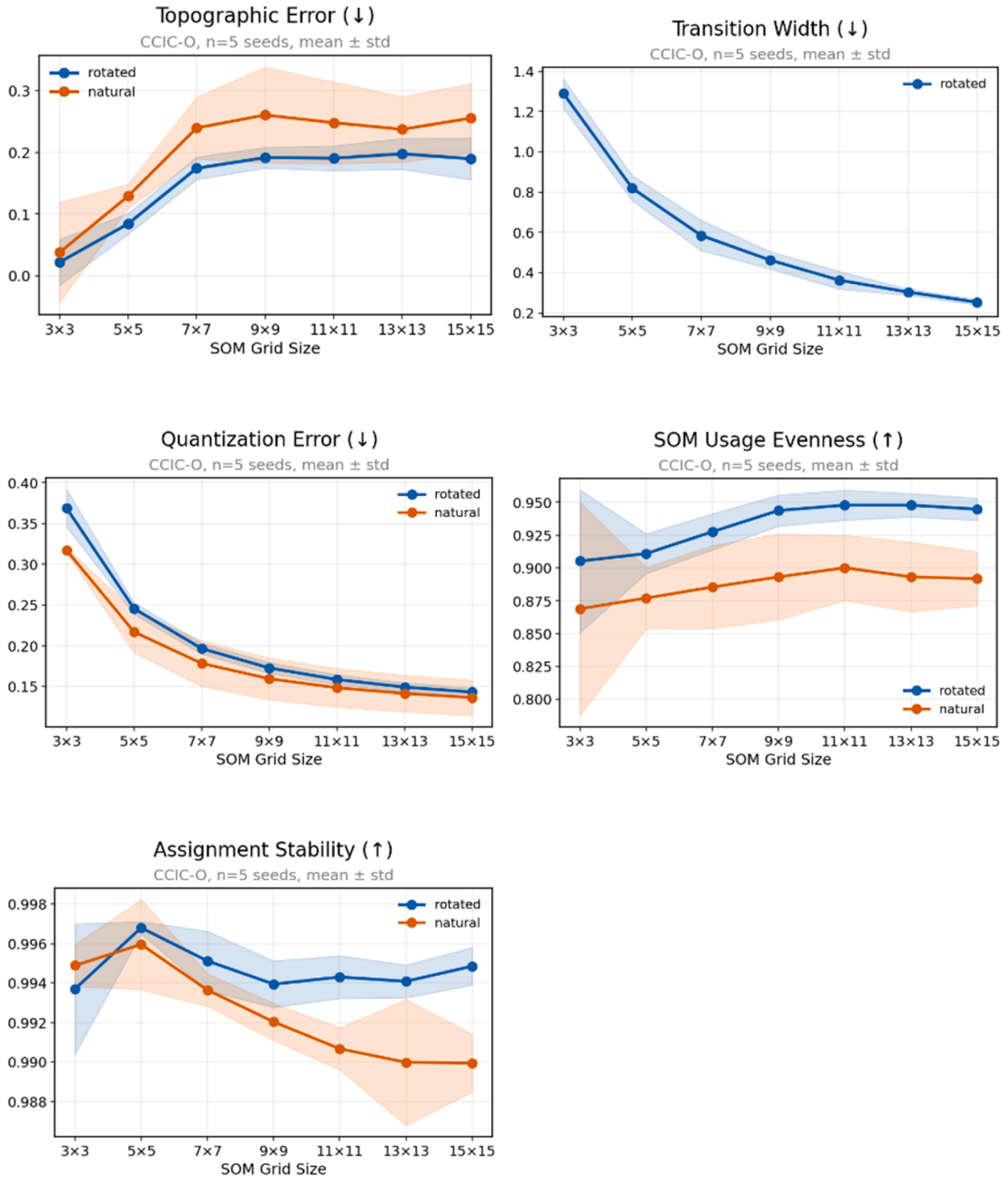


Fig. 8. (*continued*).

capture the overall crack course and form coherent regions. In strongly textured scenes, background structures increasingly influence the projection and lead to locally widened and blurred predictions. This behavior follows from fixed patch size and overlap together with hard patch assignment and majority voting. As a result, fine details and sharp boundaries are partially smoothed, so the output should be interpreted as a coarse localization cue rather than as boundary-accurate delineation or a basis for quantitative crack measurement.

In many images, this localization strategy can still identify the dominant crack region when cracks exhibit stronger, more characteristic structures than the surrounding intact texture. This often leads to more consistent BMU assignments for crack-containing patches, whereas

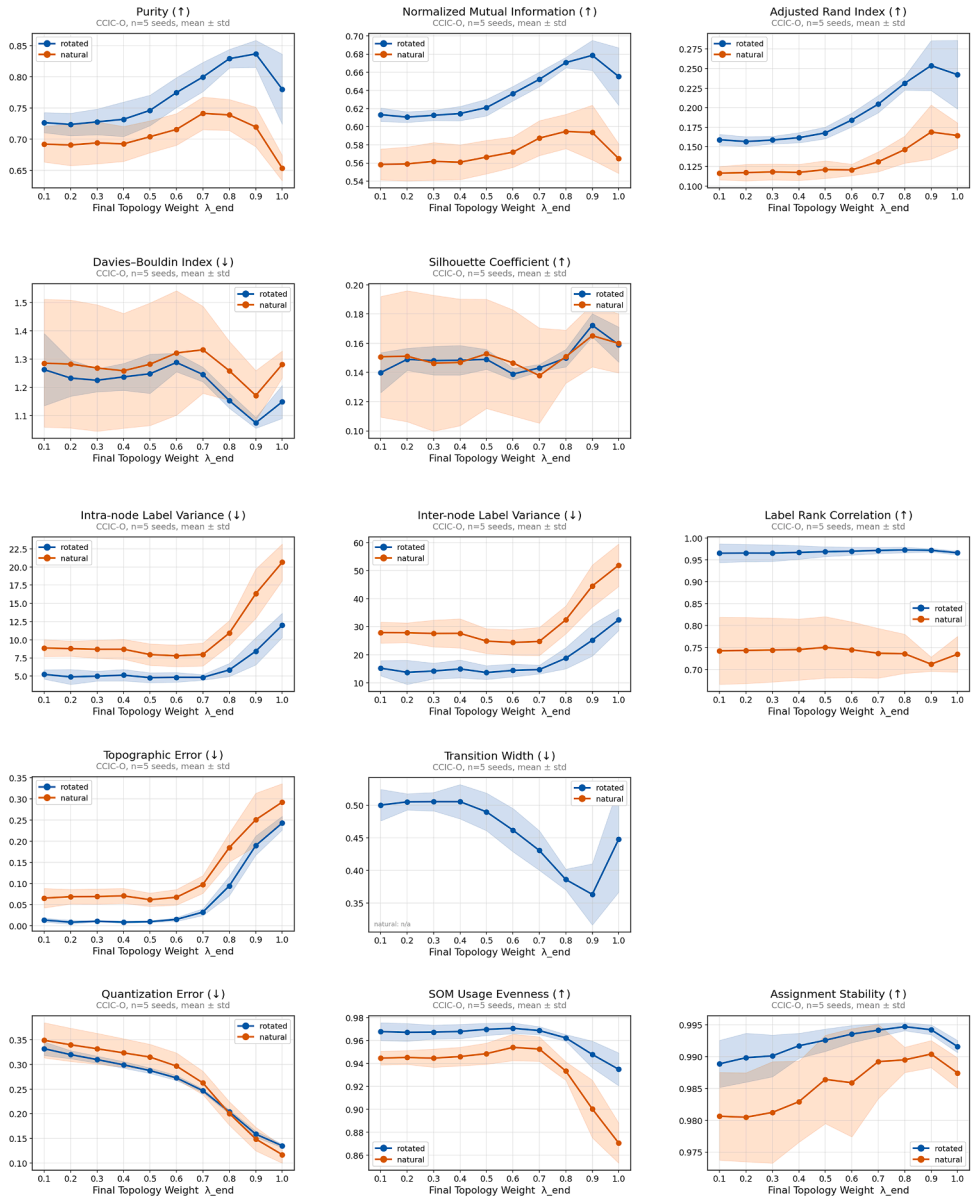


Fig. 9. Sensitivity analysis on CCIC-O (with both configurations) showing the effect of different final topology weights on the evaluation metrics.

background surface patches are frequently spread across several SOM nodes due to weaker and more heterogeneous cues. Consequently, after voting, consistency filtering, and connectivity-based selection, the crack region tends to remain as a comparatively continuous and stable component, while background regions appear more fragmented.

7.2. Methodological reflection

A central strength of the framework is its fully unsupervised design, which removes the need for annotated training data and predefined class labels. Instead of enforcing a small set of rigid categories, the embedding is organized along continuous, data-driven dimensions, which can enable fine-grained differentiation of surface conditions and retain transition regions between intact and damaged appearance.

The explicit SOM topology provides a structured visualization of the learned representation space. Neighborhood relations on the 2D grid make gradual changes in the learned organization observable at the level of prototypes and assigned samples. Complementary visualizations such as hit maps and U-matrices provide additional views of density and local separation. In this sense, interpretability is treated as an intended structural property of the organization rather than as empirically demonstrated decision support in practical inspection workflows.

Another methodological strength is the tight coupling between representation learning and organization. The SOM is not applied post-hoc to fixed features but co-evolves with the encoder through a topology-guided objective, so the embedding is shaped to be both contrastive-discriminative and map-consistent. This alignment makes organization an explicit part of representation learning rather than a purely downstream step, and it yields features that are better matched to stable organization on the map.

This coupling also implies that the organization on the SOM grid can be steered by the overall relative weighting between the contrastive and the SOM loss term. If the contrastive component is emphasized more strongly, the map tends to form more sharply separated areas on the grid with narrower transition zones, and larger gaps can appear between visually distinct areas. If the SOM term is emphasized more strongly, the grid typically represents gradual change more continuously, with broader zones of intermediate appearance and smoother transitions across neighboring nodes.

At the same time, the approach has inherent limitations. The

organization is driven by visual similarity and does not encode explicit semantic constraints, so visually similar but semantically different patterns may become neighbors. Strong textures can locally bias the structure, and rare patterns often occupy sparsely populated areas with less reliable neighborhood relations. Moreover, the topology depends on design choices such as grid resolution and neighborhood scheduling, which influence the balance between smoothness and fine-grained separation. The learned organization can also reflect biases introduced by acquisition conditions and augmentation choices.

The localization stage is intended as an initial proof-of-concept and introduces additional, method-specific limitations. Since reconstruction operates on patches and uses hard aggregation during back-projection, spatial context is reduced and boundaries are smoothed. Consequently, the output should be interpreted as coarse localization rather than contour-accurate delineation.

7.3. Positioning, differentiation, and implications

Many unsupervised approaches for concrete damage analysis primarily aim at coarse separation, including classical clustering, reconstruction-based models, and deep clustering variants (see Section 2). While these approaches can separate damaged from intact patterns, they often provide limited access to intermediate surface conditions and gradual transitions.

An alternative that avoids task-specific training is to combine a general-purpose foundation model such as SAM with a pretrained feature extractor and a post-hoc SOM. However, such a pipeline does not organize an unlabeled collection on its own. SAM is class-agnostic and returns masks for arbitrary segments rather than crack masks, so isolating the crack region requires either manual prompting, which reintroduces per-image manual input, or a hand-crafted selection heuristic, which has to be designed, tuned, and validated for cracks. Segmentation of thin, low-contrast structures is additionally a known failure mode of SAM (Guo et al., 2025, Zhang et al., 2025). On the binary datasets, a further difficulty is that half of the images are intact and contain no crack region to extract, so this stage operates without meaningful input on a large part of the data. A pipeline of this kind therefore depends on manual input or on crack-specific engineering.

The features of such a pipeline are frozen, so their invariances are fixed by pretraining on generic objects, in which thin, low-contrast crack morphology is underrepresented. They are therefore not guaranteed to encode the factor of interest, so a subsequent SOM cannot order the data along a property that the features do not represent. The benchmark in Section 6.1 isolates the consequence for the organization stage that both pipelines share, namely the post-hoc mapping of fixed features by a SOM. There, SimCLR + post-hoc SOM organizes features that were learned from the crack data itself and are therefore closer to the task than generic pretrained features, yet it still falls clearly behind TopoCLR on the orientation-related agreement metrics, as does the jointly optimized DESOM. Frozen features from a general-purpose encoder are therefore a weaker, not a stronger, starting point for the same stage. Moreover, such a pipeline processes every image with both a foundation model comprising hundreds of millions of parameters and the separate pretrained feature extractor. Although it requires no training, inference over the full collection therefore remains costly, and the overall system is considerably heavier than the lightweight encoder used in this work.

In contrast, our approach couples feature learning and organization during training, so the representation is explicitly shaped for a topology-preserving map rather than applying the SOM only as a post-processing step. This integrated design yields an output that is intended to provide a structured basis for exploring within-class variability and transition regions, instead of focusing solely on a final partition. This is particularly relevant in inspection settings, where differences in crack manifestation such as orientation, width, and continuity can relate to severity and may provide cues about possible underlying causes. Preserving these variations makes them more accessible for visual and systematic comparison beyond a coarse separation of damaged and intact cases.

From the research perspective, an implication is that multi-stage unsupervised inspection pipelines benefit from explicit compatibility between successive steps. Structural requirements of the intended organization can be incorporated already during representation learning, rather than connecting feature learning, grouping, and exploration only afterward. This frames alignment between representation and organization as a design principle for making unsupervised outputs more informative beyond a final partition, particularly when the goal is to analyze within-class variability, transition regions, and intermediate surface conditions.

From the practical perspective, the resulting organization may be useful for exploratory analysis and dataset pre-structuring when labels are scarce. The map provides a structured view of typical and atypical patterns and may serve as a basis for targeted selection for review or annotation. The localization cues may provide initial spatial indications for downstream labeling, while measurement-grade crack delineation and quantification remain downstream tasks when required.

8. Conclusion

8.1. Summary

In this paper, we presented TopoCLR, an unsupervised framework for concrete crack analysis that combines contrastive representation learning with a topology-preserving SOM organization. The method learns image representations without manual annotations and organizes them on a structured 2D map. This supports not only binary grouping of crack and intact samples, but also exploratory analysis of gradual variation in crack appearance.

The experiments show strong clustering agreement on three public concrete crack datasets, with purity values above 0.93. Beyond this binary setting, the orientation-based CCIC-O evaluation shows that the learned SOM organization captures gradual orientation-related crack variation. The high label rank correlation indicates that related orientations are arranged in coherent neighboring regions on the map, while the moderate ARI confirms that the task is not represented as a set of sharply separated compact clusters. This is consistent with the objective of preserving continuous transitions rather than enforcing rigid class partitions.

A complementary patch-based proof of concept further indicates that the framework can provide coarse crack localization cues in high-resolution images. The resulting masks highlight crack regions for visual inspection, but they remain coarse and tend to overestimate crack areas. This indicates that the localization is suitable for screening and flagging potential crack regions, but not for precise boundary delineation.

Overall, TopoCLR provides a structured basis for exploratory analysis and data pre-structuring in scenarios where labeled data are scarce or unavailable. By avoiding rigid class definitions and preserving fine-grained visual differences, the framework offers a topology-preserving organization of concrete surface images. This organization is intended to make gradual crack variations, such as orientation, width, and continuity, more accessible for visual and systematic exploration.

8.2. Outlook

Future work can extend the framework in several directions, including validation on additional damage types, improved scalability to large and heterogeneous datasets, extension of the fine-grained evaluation to further ordinal dimensions of crack appearance, such as width or continuity, and evaluation under realistic inspection conditions.

A first direction is to further develop analysis and partitioning methods for the trained SOM. Hit maps and U-matrices already indicate frequently occupied regions and strong transitions between neighboring nodes. Building on these signals, connected map regions could be

grouped into higher-level meta-clusters. While preserving the fine-grained organization of the SOM, this could make the map structure easier to summarize and compare across datasets.

A second direction is to explore alternative formulations of the encoder–SOM coupling. In the current framework, representation learning and SOM adaptation are coupled during training, but the current design preserves the classical competitive SOM update as a separate adaptation step. As a possible extension, the map component could be formulated as a more directly differentiable module, for example through differentiable prototype update or soft map adaptations. This may further simplify joint optimization and improve neighborhood consistency, especially for heterogeneous datasets.

For the complementary patch-based localization proof of concept, future work could extend the current coarse localization toward a more refined crack delineation. Possible directions include multi-scale patch extraction, adaptive strides, soft assignment maps, and confidence-weighted fusion based on assignment uncertainty. Boundary-aware post-processing, superpixel-based refinement, and weakly supervised localization strategies provide possible routes for converting coarse localization cues into more spatially refined crack masks. Together, these extensions could move the method closer to segmentation-level crack delineation.

A further direction concerns the empirical evaluation of the intended interpretability of the topology-preserving organization. While the present work analyzes structural properties of the learned SOM organization and shows quantitative metrics on representation organization, the practical benefit for domain experts remains an empirical question. A controlled user study or expert survey involving inspection professionals could assess how such visual organization is perceived and used in inspection-related analysis tasks. Such studies would require a dedicated experimental design, including expert recruitment, task definition, comparison methods, and evaluation protocols.

Finally, the proposed principles are not limited to concrete crack inspection. The framework may be applicable in settings where unlabeled data should be organized according to gradual visual or structural variation rather than rigid predefined classes. In such settings, the combination of representation learning and map-based organization can support exploratory grouping, similarity-based retrieval, and structured exploration of continuous variation. This includes inspection and monitoring tasks in civil engineering, mechanical engineering, manufacturing, and other domains where large collections of unlabeled data must be structured without relying on rigid predefined classes.

CRediT authorship contribution statement

Barış Özcan: Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Jörg Blankenbach:** Writing – review & editing, Supervision, Project administration.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A. – Evaluation metrics

This appendix provides the formal definitions of the evaluation metrics summarized in Section 4.2. Metrics are grouped according to the six complementary dimensions introduced in the main manuscript, namely external label agreement, internal embedding geometry, ordinal organization, topology preservation, prototype quality, and training stability. Additionally, two map-based diagnostics that support qualitative inspection are defined at the end of this appendix.

In the following, $b(i)$ denotes the SOM node assigned to sample i, y_i its ground-truth label, $w_{u,v}$ the prototype at grid node (u,v), and z_i the encoder embedding of sample i. The total number of samples is N, and the SOM grid contains M nodes in total. The set of directly adjacent node pairs on the grid is denoted by $\mathscr{E}$, and d_{grid} denotes the Euclidean distance on the 2D lattice.

A.1 External label agreement metrics

These metrics quantify the agreement between the SOM-induced grouping and the reference class labels. Since the method assigns each sample to a SOM node rather than to an explicit class, each node is first labeled by majority voting over its assigned samples. Let C denote the set of classes. For the natural configuration, whose labels are continuous, C consists of the 18 binned orientation classes described in Section 4.2.

Purity (Pur). Pur measures the fraction of samples whose ground-truth label matches the majority label of their assigned SOM node:

$$\hat{y}(k) = \underset{c \in \mathscr{C}}{\operatorname{argmax}} \sum_{i:b(i)=k} 1\{y_i = c\} \tag{A1}$$

$$\mathit{Purity} = \frac{1}{N} \sum_{i=1}^{N} 1\{y_i = \hat{y}(b(i))\} \tag{A2}$$

In our setup, purity is equivalent to clustering accuracy, since each SOM node is labeled via majority voting and no global label matching is performed.

Normalized Mutual Information (NMI). NMI measures the mutual information between the cluster assignment $B = \{b(i)\}$ and the ground truth labels $Y = \{y_i\}$, normalized to be invariant to cluster size:

$$\mathrm{NMI}(B,\ Y) = \frac{2 \cdot I(B;Y)}{H(B) + H(Y)} \tag{A3}$$

where $I(B;Y)$ is the mutual information and $H(\cdot)$ denotes entropy.

Adjusted Rand Index (ARI). ARI measures the pairwise agreement between B and Y, corrected for chance:

$$\mathrm{ARI}(B,Y) = \frac{\mathrm{RI}(B,Y) - \mathbb{E}[\mathrm{RI}]}{\max(\mathrm{RI}) - \mathbb{E}[\mathrm{RI}]} \tag{A4}$$

where RI is the unadjusted Rand Index.

A.2 Internal embedding geometry metrics

These metrics characterize the compactness and separation of clusters in the embedding space without reference labels. Clusters correspond to occupied SOM nodes, while empty nodes are ignored. Let K denote the number of occupied nodes, S_i the set of embeddings assigned to node i, and c_i the corresponding arithmetic mean.

Davies–Bouldin Index (DBI). DBI measures the ratio of within-cluster scatter to between-cluster separation. The within-cluster scatter at node i is given by

$$s_i = \frac{1}{|S_i|}\sum_{z\in S_i} \| z - c_i \|_2, \tag{A5}$$

and the resulting DBI is

$$DBI = \frac{1}{K}\sum_{i=1}^{K} \max_{j\neq i} \frac{s_i + s_j}{\| c_i - c_j \|_2} \tag{A6}$$

Silhouette Coefficient (SC). SC measures the relative closeness of an embedding to its own cluster compared to the nearest other cluster. For an embedding z assigned to node i, $a(z)$ denotes the mean distance from z to other samples in S_i and $b(z)$ the mean distance from z to samples in the nearest other cluster. The silhouette of z is then

$$s(z) = \frac{b(z) - a(z)}{\max(a(z), b(z))} \tag{A7}$$

and the overall silhouette coefficient is

$$SC = \frac{1}{N}\sum_{z} s(z) \tag{A8}$$

A.3 Ordinal organization metrics

These metrics assess how consistently the ordinal label factor (here the orientation) is arranged across the map, from the local scale within a node, through neighboring nodes, up to the global grid. They operate on the orientation labels y_i (which are continuous in the natural configuration) through the circular distance

$$\delta_P(a, b) = \min(|a - b| \bmod P,\ P - (|a - b| \bmod P)) \tag{A9}$$

with period $P = 180^\circ$ for axial orientation, and through the circular mean μ of a set of labels, obtained by averaging the labels as unit vectors on the orientation circle. To keep the measures robust on degenerate maps, nodes with fewer than two assigned samples are treated as undefined and contribute the worst-case circular distance ($\frac{P}{4}$ for V_{intra}, $\frac{P}{2}$ for V_{inter}) in place of the corresponding term in Eq. (A10) and Eq. (A11).

Intra-Node Label Variance (V_{intra}). V_{intra} is the mean circular spread of the labels within each node, averaged over the grid:

$$V_{intra} = \frac{1}{M}\sum_{(u,v)} \frac{1}{|S_{u,v}|}\sum_{i\in S_{u,v}} \delta_P\left(y_i, \mu_{u,v}\right) \tag{A10}$$

where $S_{(u,v)}$ denotes the samples assigned to node (u, v) and $\mu_{(u,v)}$ is the circular mean of their labels. Low values indicate that each node represents a narrow range of orientations.

Inter-Node Label Variance (V_{inter}). V_{inter} is the mean circular label difference between grid-adjacent nodes:

$$V_{inter} = \frac{1}{|\mathscr{E}|}\sum_{(u,v,\ u',v')\in\mathscr{E}} \delta_P\left(\mu_{u,v}, \mu_{u',v'}\right) \tag{A11}$$

where $\mathscr{E}$ is the set of directly adjacent node pairs on the grid. Low values indicate smooth orientation transitions between neighboring nodes.

Label Rank Correlation (LRC). LRC assesses whether the global arrangement of the orientation classes on the grid follows their orientation relations. It is computed on the 18 orientation classes $c \in C$ used for the external label agreement metrics (Section 4.2), which coincide with the native classes in the rotated configuration and are obtained through the 10° binning for the continuous natural labels. For each class c, let g_c denote its grid centroid, that is the mean node coordinate over the samples of class c, and let θ_c denote its representative orientation, given by the native rotation angle in the rotated configuration and by the bin center in the natural configuration. Over all class pairs,

$$LRC = \rho_{Spearman}\left(\delta_P(\theta_c, \theta_{c'}),\ d_{grid}(g_c, g_{c'})\right)_{c,\ c'\in C,\ c\ <\ c'} \tag{A12}$$

Unlike the local neighbor metrics, LRC is global and detects when classes with similar orientations are placed in distant regions of the grid even if the map is locally smooth.

A.4 Topology preservation metrics

These metrics assess whether the SOM grid topology faithfully reflects the data topology in the embedding space.

Topographic Error (TE). TE quantifies the fraction of samples whose best and second-best matching unit are not directly neighbors on the SOM grid. With $b_1(i)$ the BMU and $b_2(i)$ the second-best matching unit for sample i

$$TE = \frac{1}{N}\sum_{i=1}^{N} 1\{d_{grid}(b_1(i), b_2(i)) > 1\} \tag{A13}$$

Transition Width (TW). TW measures the local smoothness of the class arrangement on the grid by comparing the class labels of directly adjacent nodes. For ordinal class structures, such as the orientation classes in CCIC-O, TW is computed using a class-distance function d_{class} that reflects the underlying ordinal scale. For circular ordinal classes, such as orientations modulo 180°, d_{class} is the circular distance on the class index. Let $\hat{y}(u,v)$ denote the majority label of node (u,v). Then,

$$TW = \frac{1}{|\mathscr{E}|}\sum_{((u,v),(u',v'))\in\mathscr{E}} d_{class}(\hat{y}(u,v), \hat{y}(u',v')) \tag{A14}$$

To support comparison across different numbers of classes, TW can be normalized by the maximum class distance, so that values lie in [0,1]. In contrast to TE, which assesses whether the map is locally unfolded in the embedding space, TW evaluates how gradually class-level information varies across the grid. TW is the discrete-class counterpart of V_{inter} and is reported only for the rotated configuration, since V_{inter} covers the same local transitions on the continuous labels of the natural configuration.

A.5 Prototype quality metrics

These metrics characterize how well the SOM prototypes represent the embedding distribution as a whole, considering all prototypes and their utilization rather than only occupied clusters.

Quantization Error (QE). QE measures the mean distance between an embedding and its assigned prototype:

$$QE = \frac{1}{N}\sum_{i=1}^{N} \|z_i - w_{b(i)}\|_2 \tag{A15}$$

SOM Usage Evenness (SUE). SUE is calculated from the entropy of the hit map. Let $H(u,v)$ denote the hit count of node (u,v) as defined in A.7 below, and $p(u,v) = H(u,v)/N$ the corresponding normalized hit frequency. Then,

$$SUE = -\frac{1}{\log M}\sum_{(u,v)} p(u,v)\cdot\log p(u,v) \tag{A16}$$

where the convention $0\log 0 = 0$ is used for empty nodes. The metric reflects how evenly the samples are distributed across the available grid; lower entropy indicates that the data are concentrated on a small number of nodes.

A.6 Training stability

This dimension quantifies the stabilization of the SOM organization over the course of training within a single run.

Assignment Stability (AS). AS measures how consistent the BMU assignments of individual samples remain across consecutive training epochs. For each sample i and each pair of consecutive epochs $(e, e+1)$, we check whether the assigned nodes are within a tolerance distance d_{max} on the grid. Let $b^{(e)}(i)$ denote the BMU assignment at epoch e, and let E be the number of training epochs. Then,

$$AS = \frac{1}{N\cdot(E-1)}\sum_{i=1}^{N}\sum_{e=1}^{E-1} 1\{d_{grid}(b^{(e)}(i), b^{(e+1)}(i)) \le d_{max}\} \tag{A17}$$

In our experiments, we set $d_{max} = 1$, so that a sample is considered stably assigned if its BMU is within direct grid neighborhood across consecutive epochs. The metric quantifies the stability of sample assignments during training.

A.7 Map-based diagnostics

To support inspection of the SOM organization, two diagnostic quantities from standard SOM practice are used throughout the analysis. Both are defined per node on the trained map and can be inspected qualitatively as heatmaps over the grid or aggregated into quantitative summary measures.

Hit Map. The hit map quantifies the assignment frequency of samples to each node:

$$H(u,v) = \sum_{i=1}^{N} 1\{b(i) = (u,v)\} \tag{A18}$$

Nodes with high occupancy indicate frequently occurring surface patterns, while sparsely populated nodes often correspond to rare or transitional cases.

Unified Distance Matrix (U-matrix). The U-matrix characterizes the local dissimilarity structure of the SOM. For each node, it measures the average Euclidean distance between its prototype and the prototypes of its directly adjacent neighbors:

$$U(u,v) = \frac{1}{|\mathscr{N}(u,v)|}\sum_{(u',v')\in\mathscr{N}(u,v)} \|w_{u,v} - w_{u',v'}\|_2 \tag{A19}$$

where $\mathcal{N}(u,v)$ denotes the four-neighborhood of node (u,v) on the grid. Elevated values indicate strong transitions between neighboring regions, while low values indicate locally coherent areas.

Appendix B. – Construction of the CCIC-O dataset

This appendix describes the semi-automatic construction of the CCIC-O dataset summarized in Section 4.1.2. The pipeline operates in two stages.

B.1 Source selection and reference alignment

Stage 1 selects images from CCIC that contain a single edge-to-edge linear crack and aligns them to a horizontal reference orientation.

1. Convert the input image to grayscale.
2. Compute a coarse crack response by combining a smoothed dark-valley response (Gaussian $\sigma = 5.0$) with an inverted local-contrast response ($\sigma = 2.0$) via pixelwise maximum.
3. Threshold at the 80th percentile and refine morphologically through closing, opening, hole filling, and small-object removal.
4. Among the connected components, select the one most consistent with a single edge-to-edge crack using a score that rewards opposite-border touches and large normalized bounding-box spans.
5. Reduce the selected component to its skeleton and extract the longest path as the shortest path between its endpoints in an 8-connected weighted graph.
6. Smooth the path (Gaussian $\sigma = 2.0$) and compute chord length, arc length, straightness ratio, normalized mean deviation from the chord line, dominant orientation θ from PCA, PCA linearity, and number of branch points.
7. Accept the candidate only if it is sufficiently straight, spans the patch, and dominates the mask without competing edge-to-edge components. Exact thresholds are documented in the released code.
8. Reflectively pad the source image, rotate to align θ with the horizontal reference axis using bilinear interpolation, and center-crop to 227×227 pixels.
9. Manually inspect the aligned candidates and exclude residual cases in which the crack does not form a single linear structure or the alignment is visibly incorrect.

This procedure yields a base set of 1,008 horizontally aligned source patches.

B.2 Synthesis of rotated variants

Stage 2 expands the validated base set by generating rotated variants of each source patch. For each base image, 36 rotated variants are produced at orientations $\varphi \in \{0^\circ, 10^\circ, \cdots, 350^\circ\}$ through reflective padding, rotation by φ with bilinear interpolation, and a centered 227×227 crop. The padding-and-crop convention matches Stage 1, so that no synthetic borders enter the cropped region.

Class labels are derived deterministically from the rotation angle. The dataset retains all 36 physical orientations; the corresponding 18 effective ordinal classes used during evaluation, which arise from the 180° directional symmetry of linear cracks, are described in Section 4.1.2.

Appendix C. – Training dynamics

This appendix shows the training dynamics of the proposed framework on the orientation dataset CCIC-O, which serves as a representative example. Loss components and evaluation metrics defined in Appendix A are recorded once per epoch on the full evaluation set. For the natural configuration, Purity, NMI, ARI, and LRC are computed on the 18 binned orientation classes (Section 4.2). Transition Width is shown for the rotated configuration only. Figs. C.1-C.6 show the loss components, the external label agreement, the internal embedding geometry, the ordinal organization, the topology preservation, and the prototype quality and training stability, respectively, over the training epochs.

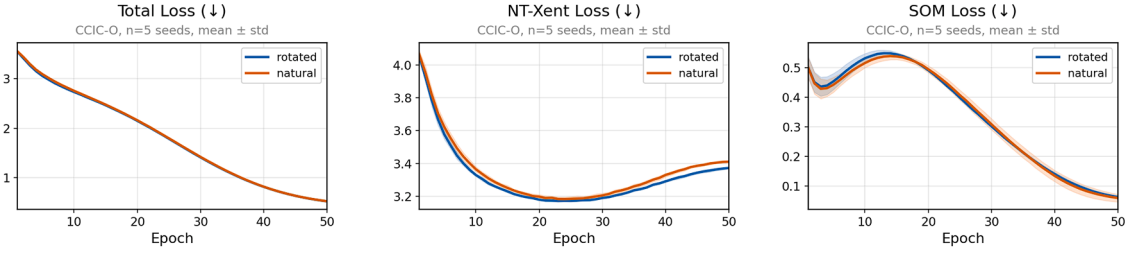


Fig. C.1. Loss components (total, NT-Xent, SOM) over training epochs for CCIC-O.

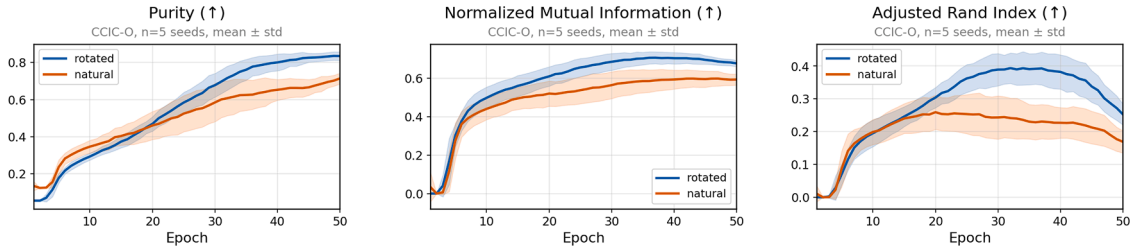


Fig. C.2. External label agreement metrics (Pur, NMI, ARI) over training epochs for CCIC-O.

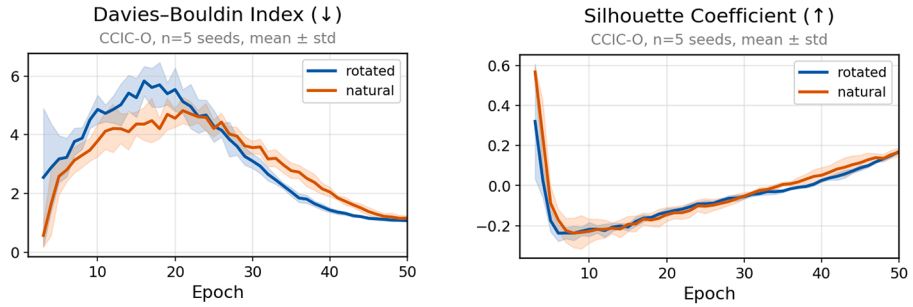


Fig. C.3. Internal embedding geometry metrics (DBI, SC) over training epochs for CCIC-O.

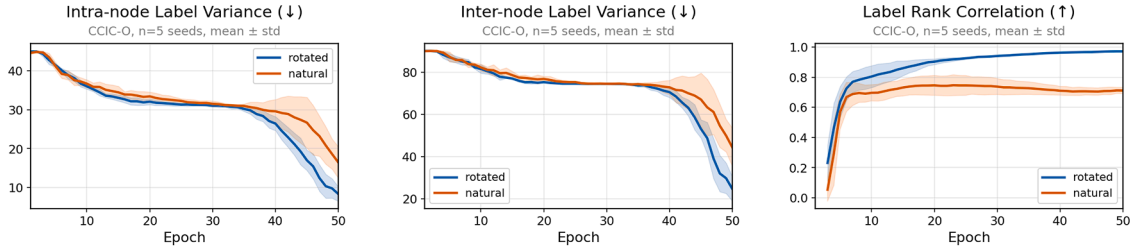


Fig. C.4. Ordinal organization metrics (intra- and inter-node variance, LRC) over training epochs for CCIC-O.

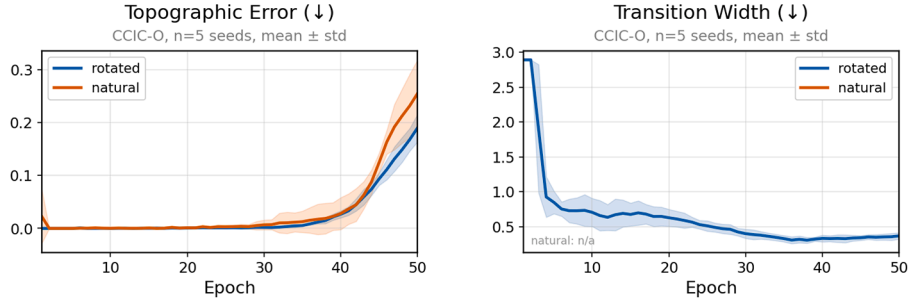


Fig. C.5. Topology preservation metrics (TE, TW) over training epochs for CCIC-O.

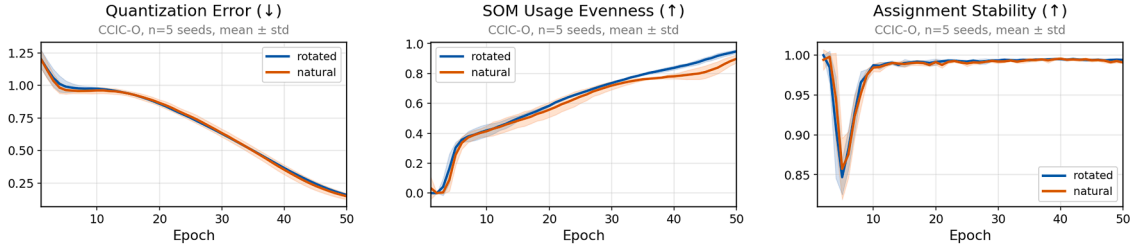


Fig. C.6. Prototype quality metrics (QE, SUE) and training stability (AS) over training epochs for CCIC-O.

Appendix D. – Qualitative results

This appendix shows qualitative visualizations of the learned SOM organization for the binary datasets, where crack and intact surfaces are organized within each map. Sections D.1, D.2, and D.3 cover CCIC, CPC, and BCD, respectively. The corresponding qualitative results for the

orientation dataset CCIC-O are presented in Section 5.2.2. For each dataset, the SOM overview shows sample images mapped to each BMU, the U-matrix indicates local dissimilarity between adjacent prototypes, and the hit map shows the assignment frequency per node.

D.1 CCIC

Fig. D.1 shows the SOM organization on CCIC with up to 9 sample images per BMU. Fig. D.2 shows the t-SNE projection, U-matrix, and hit map. Figs. D.2a and D.2b show randomly chosen samples assigned to node (10, 10) and node (0, 0), respectively, illustrating the corresponding crack and intact regions of the map.

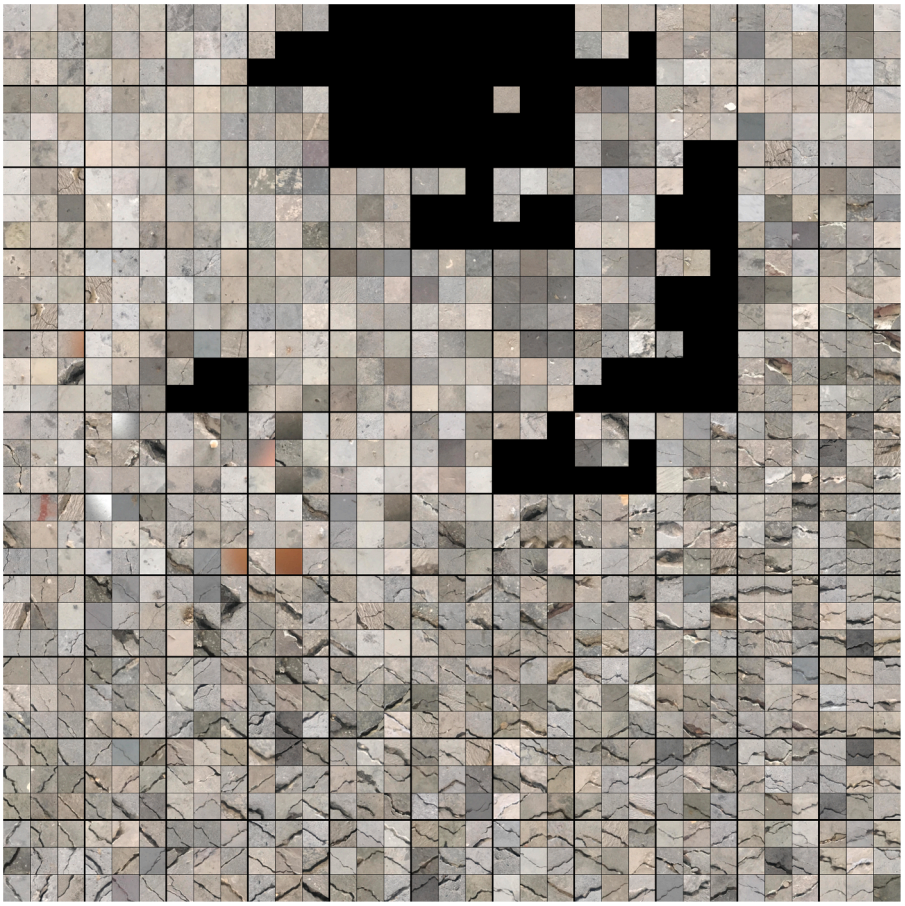


Fig. D.1. SOM organization on CCIC, with up to 9 sample images mapped onto each BMU.

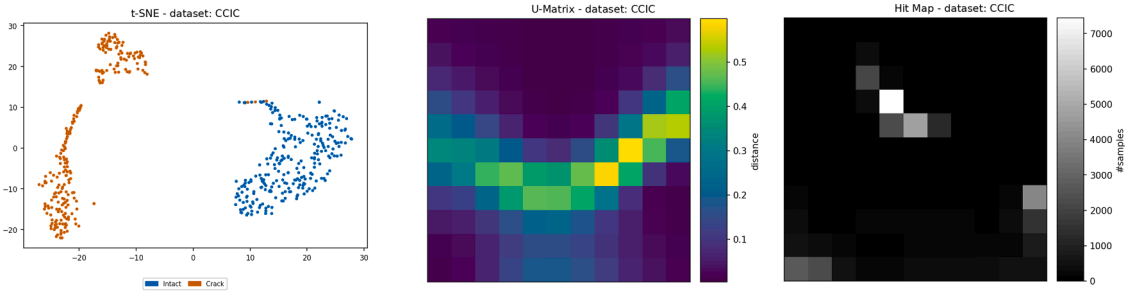


Fig. D.2. t-SNE projection of 500 data points (left), U-matrix (center) and hit map (right) for CCIC.



Fig. D.2a. Randomly chosen CCIC samples with BMU at SOM node (10, 10), showcasing cracks.

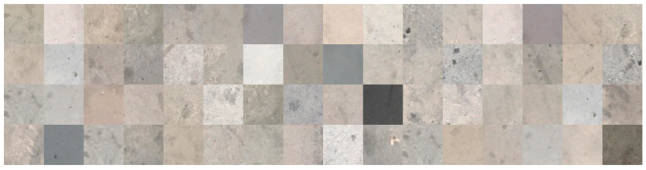


Fig. D.2b. Randomly chosen CCIC samples with BMU at SOM node (0, 0), showcasing intact surfaces.

D.2 CPC

Fig. D.3 shows the SOM organization on CPC with up to 9 sample images per BMU. Fig. D.4 shows the t-SNE projection, U-matrix, and hit map. Figs. D.4a and D.4b show randomly chosen samples assigned to node (10, 10) and node (0, 0), respectively, illustrating the corresponding crack and intact regions of the map.

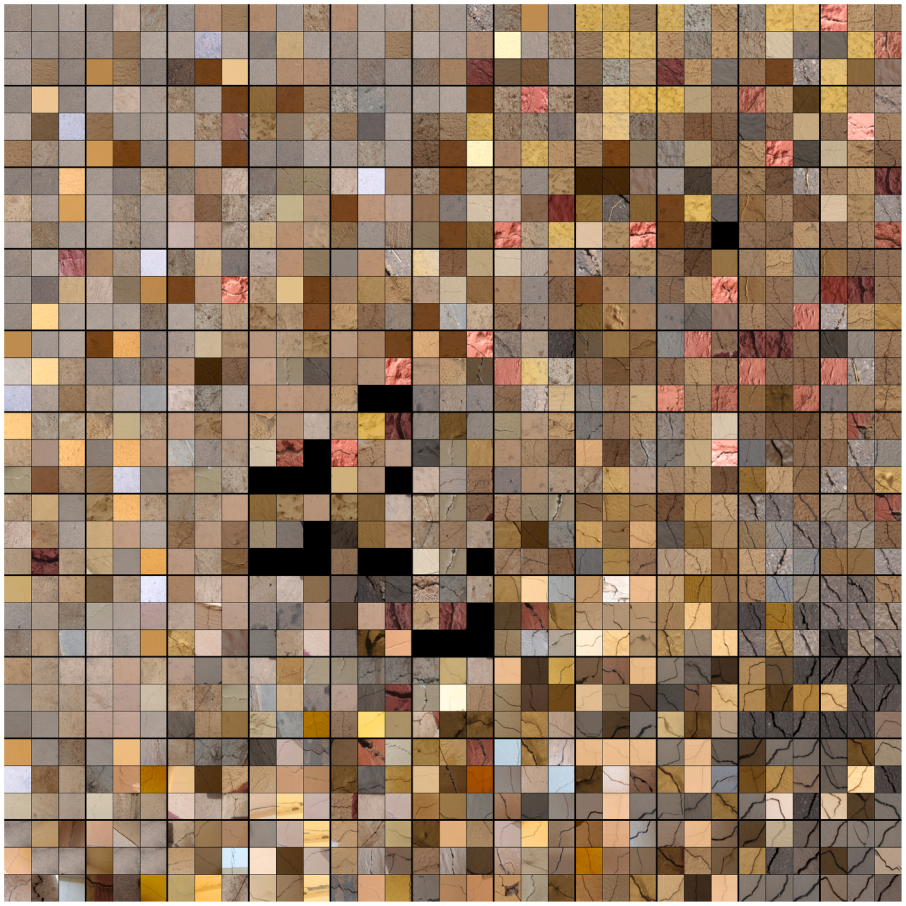


Fig. D.3. SOM organization on CPC, with up to 9 sample images mapped onto each BMU.

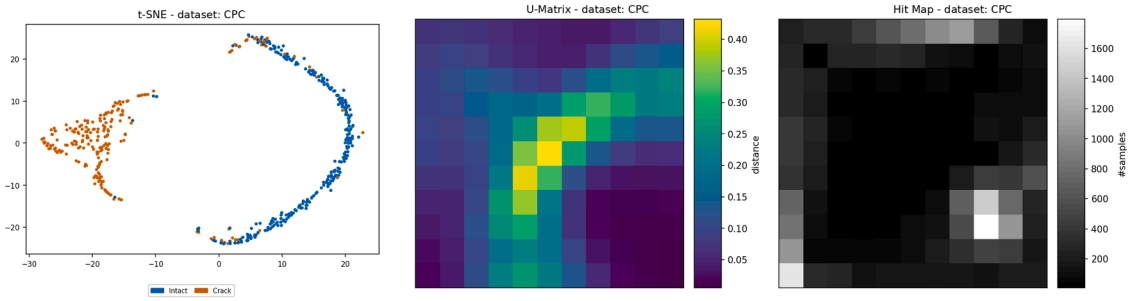


Fig. D.4. t-SNE projection of 500 data points (left), U-matrix (center) and hit map (right) for CPC.

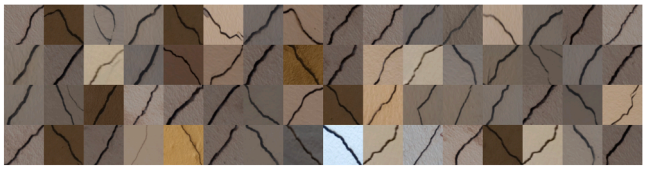


Fig. D.4a. Randomly chosen CPC samples with BMU at SOM node (10, 10), showcasing cracks.



Fig. D.4b. Randomly chosen CPC samples with BMU at SOM node (0, 0), showcasing intact surfaces.

D.3 BCD

Fig. D.5 shows the SOM organization on BCD with up to 9 sample images per BMU. Fig. D.6 shows the t-SNE projection, U-matrix, and hit map. Figs. D.6a and D.6b show randomly chosen samples assigned to node (10, 10) and node (0, 0), respectively, illustrating the corresponding crack and intact regions of the map.

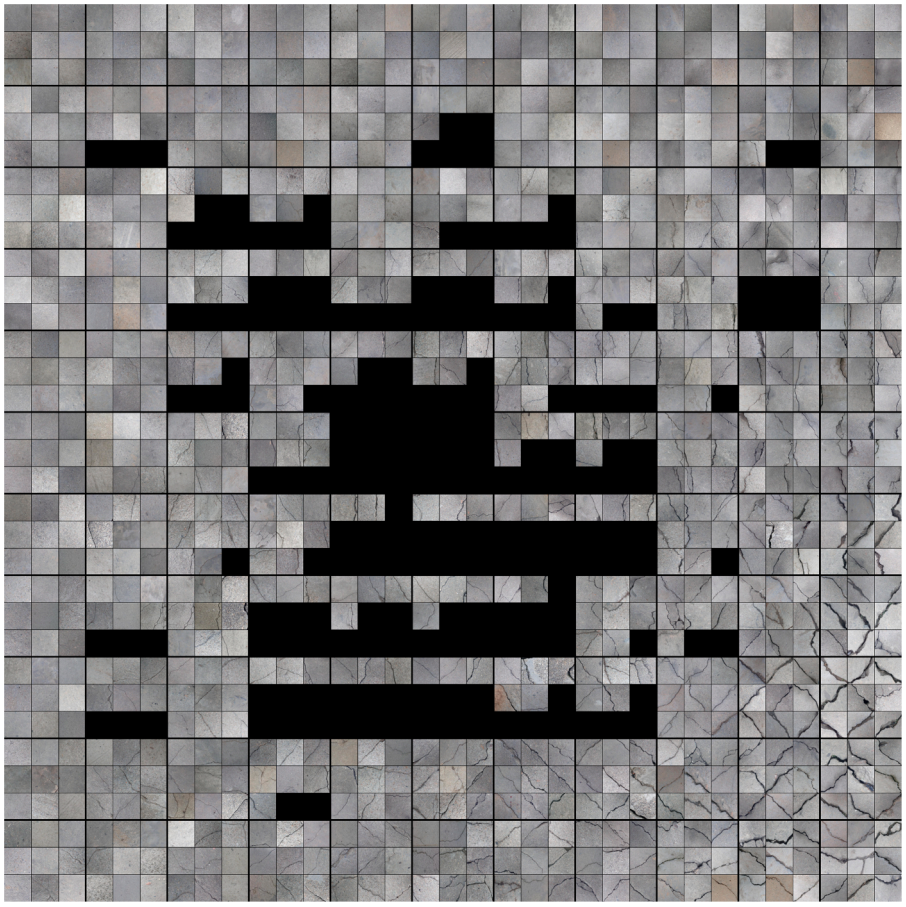


Fig. D.5. SOM organization on BCD, with up to 9 sample images mapped onto each BMU.

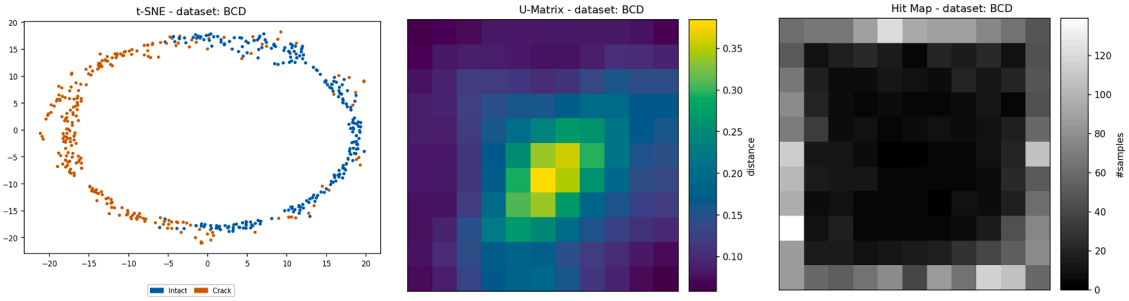


Fig. D.6. t-SNE projection of 500 data points (left), U-matrix (center) and hit map (right) for BCD.

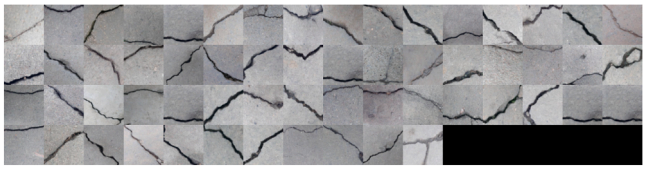


Fig. D.6a. Randomly chosen BCD samples with BMU at SOM node (10, 10), showcasing cracks.

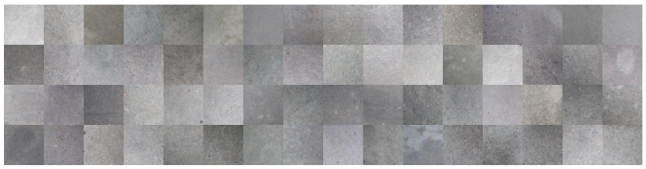


Fig. D.6b. Randomly chosen BCD samples with BMU at SOM node (0, 0), showcasing intact surfaces.

Appendix E. – Patch-based localization (proof of concept)

This appendix presents a patch-based extension of the framework as a complementary proof of concept for label-free crack localization in high-resolution images. The extension applies the framework introduced in Section 3 in a patch-based mode and provides coarse spatial localization instead of a fully developed segmentation pipeline.

E.1 Method

To localize cracks in single high-resolution images, the framework introduced in Section 3 is applied in a patch-based mode. The image is decomposed into overlapping patches, the patch-level organization induced by the framework is recovered, and the resulting BMU assignments are projected back to the image domain to form a coarse pixel-level localization mask.

E.1.1 Patch extraction

A sliding window of size $p \times p$ moves across the image with step size $s < p$, yielding a set of overlapping patches $P = \{p_1, p_2, \ldots, p_M\}$. The overlap ensures that each pixel is covered by multiple patches and avoids cutting fine structures at patch borders.

For an image of height H and width W, the total number of extracted patches is

$$M = \left\lfloor \frac{H-p}{s} + 1 \right\rfloor \cdot \left\lfloor \frac{W-p}{s} + 1 \right\rfloor. \tag{E1}$$

Patch size and step size jointly control the trade-off between spatial granularity, contextual information, and computational effort.

E.1.2 Patch-level topology-guided clustering

The patches form an image-specific dataset on which the full learning procedure of Section 3.2 is applied, including normalization and joint training of encoder and SOM. The SOM is therefore trained on the patch set itself rather than reused as a fixed model. After training, each patch embedding is assigned to its BMU on the trained SOM, yielding a structured patch-level organization that serves as the basis for mask reconstruction.

E.1.3 Mask reconstruction

To form a localization mask, the patch-level BMU assignments are projected back into the image space. Each pixel (x, y) is covered by multiple patches, which may be assigned to different SOM nodes. Its label is therefore obtained by majority voting:

$$A(x,y) = mode\{a_i | (x,y) \in p_i\}. \tag{E2}$$

where a_i denotes the SOM node assigned to patch p_i.

The robustness of this assignment is captured by a consistency map, defined as the relative frequency of the dominant SOM node:

$$C(x,y) = \frac{n_{max}(x,y)}{n_{total}(x,y)} \tag{E3}$$

with $n_{\max}(x,y)$ the vote count for the most frequent node and $n_{\text{total}}(x,y)$ the total number of covering patches. Higher values indicate more robust assignments, lower values typically arise near structural transitions or crack boundaries.

For binary crack localization, pixels with $C(x,y)$ below a threshold (set to 0.6 in our implementation) are first suppressed. Within the high-consistency region, connected components are extracted per SOM label, and the largest component across all labels is selected as the final crack mask. This selection is effective because patches containing cracks typically yield more consistent SOM assignments than background patches, so the crack tends to form the dominant coherent component after consistency filtering. In highly textured scenes, large non-crack regions can occasionally dominate, which is a limitation of this minimal selection strategy.

The main output of the patch-based localization is therefore the binary crack mask. It is accompanied by the intermediate multi-region assignment map $A(x,y)$, which can support more differentiated analyses if multiple SOM regions are of interest, and by the consistency map $C(x,\ y)$, which provides a per-pixel confidence estimate.

E.2 Dataset

We use the Bochum Concrete Crack (BCC) dataset (Çelik and König, 2022), which contains 370 RGB images of cracks on concrete building surfaces at a resolution of 512 × 512 pixels, each annotated with a binary segmentation mask. This dataset features diverse surface textures and challenging background elements such as vegetation, which make crack localization more difficult when cracks are less visually dominant.

For our experiments, we evaluate on a subset of ten images with their corresponding ground-truth masks. Although limited in size, this subset provides sufficient resolution and pixel-level annotation to assess the patch-based localization scheme on representative cases.

E.3 Metrics

The predicted localization mask P is compared at pixel level with the ground-truth annotation G, treating crack pixels as foreground. The main metric is intersection over union (IoU), defined as

$$IoU = \frac{|P \cap G|}{|P \cup G|}. \tag{E4}$$

Values range from 0 to 1, with higher values indicating better agreement in crack extent and spatial location.

For a more differentiated assessment, precision and recall additionally capture false positives and missed crack pixels:

$$Precision = \frac{|P \cap G|}{|P|} \tag{E5}$$

$$Recall = \frac{|P \cap G|}{|G|} \tag{E6}$$

Values range from 0 to 1, with higher precision indicating fewer false positives and higher recall indicating more complete crack localization.

E.4 Results

Table E.1 reports IoU, precision, and recall for the ten evaluated BCC images, together with their mean values.

Table E.1
Quantitative localization results in terms of IoU, Precision, and Recall for the patch-based localization of cracks on the BCC subset.

Image	IoU	Precision	Recall
1	0.2567	0.2581	0.9794
2	0.1456	0.1518	0.7830
3	0.4463	0.4832	0.8539
4	0.1301	0.1301	1.0000
5	0.2311	0.2311	1.0000
6	0.0752	0.0800	0.5562
7	0.4921	0.5416	0.8433
8	0.4011	0.4011	1.0000
9	0.0869	0.0899	0.7190
10	0.0552	0.0552	1.0000
Mean	0.2320	0.2422	0.8735

The values fall in the lower to middle range, consistent with the coarse-localization focus of our approach. However, the high recall confirms that potential crack regions are captured to a large extent, while the lower precision and IoU values reflect systematic over-coverage, where the reconstructed masks include adjacent or structurally similar areas. Despite this inaccuracy, the results provide a consistent spatial indication of crack locations, confirming that the SOM-based representations provide coarse localization cues.

For visual assessment, Fig. E.1 shows the predicted localization masks overlaid on the ten BCC images. Image IDs correspond to the rows in Table E.1.

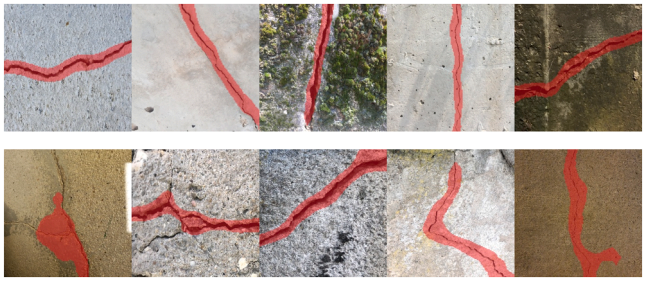


Fig. E.1. Crack localization results on the ten BCC images, illustrating typical behavior and limitations of the patch-based procedure.

The examples shown confirm the over-coverage tendency suggested by the metrics, with crack regions slightly overestimated and adjacent areas included in the mask. This behavior is inherent to the patch-based design, where the localization mask is derived from patch-level assignments and emphasizes coverage over precision. As a consequence, very fine or low-contrast cracks may be missed. Overall, this method should be interpreted as coarse localization rather than segmentation-level delineation.

Data availability

The CCIC-O dataset generated in this study is publicly available in the Zenodo repository at https://doi.org/10.5281/zenodo.20042918, while all other datasets analyzed in this study are publicly available from the sources cited in the article.

References

Abdel-Qader, I., Abudayyeh, O., & Kelly, M. E. (2003). Analysis of edge-detection techniques for crack identification in bridges. *Journal of Computing in Civil Engineering, 17*(4), 255–263. https://doi.org/10.1061/(ASCE)0887-3801(2003)17:4(255)

Adeli, H. (2001). Neural networks in civil engineering: 1989–2000. *Computer-Aided Civil and Infrastructure Engineering, 16*(2), 126–142. https://doi.org/10.1111/0885-9507.00219

Alam, M. N., Laxmi, V., Kumar, N., Kumari, R., & Sharma, S. (2025). Unsupervised machine learning for anomaly detection: A systematic review. *SSRG International Journal of Electrical and Electronics Engineering, 12*(7), 34–72. https://doi.org/10.14445/23488379/IJEEE-V12I7P104

Bartczak, E. T., Bassier, M., & Vergauwen, M. (2023). Case study for UAS-assisted bridge inspections. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, XLVIII-2/W3-2023*, 33–39. https://doi.org/10.5194/isprs-archives-XLVIII-2-W3-2023-33-2023

Benz, C., & Rodehorst, V. (2024). OmniCrack30k: A benchmark for crack segmentation and the reasonable effectiveness of transfer learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops* (pp. 3876–3886). https://doi.org/10.1109/CVPRW63382.2024.00392

Benz, C., & Rodehorst, V. (2025). CrackStructures and CrackEnsembles: The Power of Multi-View for 2.5D Crack Detection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)* (pp. 5990–5999). https://doi.org/10.1109/WACV61041.2025.00584

Bhardwaj, M., Khan, N. U., & Baghel, V. (2025). A novel fuzzy C-means clustering framework for accurate road crack detection: Incorporating pixel augmentation and intensity difference features. *Informatica, 49*(15), 27–40. https://doi.org/10.31449/inf.v49i15.7082

Bhattacharya, S., Zhang, C., Mishra, D. K., Yuen, M. M. F., & Zhang, J. (2025). Efficient unsupervised domain adaptation for crack segmentation with interpretable Fourier–Morphology blending and uncertainty-guided self-training. *Computer-Aided Civil and Infrastructure Engineering, 40*(29), 5790–5807. https://doi.org/10.1111/mice.70127

Biondini, F., & Frangopol, D. M. (2016). Life-cycle performance of deteriorating structural systems under uncertainty: Review. *Journal of Structural Engineering, 142*(9), Article F4016001. https://doi.org/10.1061/(ASCE)ST.1943-541X.0001544

Caron, M., Bojanowski, P., Joulin, A., & Douze, M. (2018). Deep clustering for unsupervised learning of visual features. In *Computer Vision – ECCV 2018, Lecture Notes in Computer Science (LNCS), 11218* pp. 139–156). Springer. https://doi.org/10.1007/978-3-030-01264-9_9

Çelik, F., & König, M. (2022). A sigmoid-optimized encoder–decoder network for crack segmentation with copy-edit-paste transfer learning. *Computer-Aided Civil and Infrastructure Engineering, 37*(14), 1875–1890. https://doi.org/10.1111/mice.12844

Çelik, R. F., Hoskere, V., & Kessler, S. (2025). From pixels to rating—A semiautomated system linking multi-damage segmentation and condition rating in concrete bridge inspections. *Computer-Aided Civil and Infrastructure Engineering, 40*(30), 5842–5866. https://doi.org/10.1111/mice.70145

Chen, C., Seo, H., Jun, C. H., & Zhao, Y. (2022). Pavement crack detection and classification based on fusion feature of LBP and PCA with SVM. *International Journal of Pavement Engineering, 23*(9), 3274–3283. https://doi.org/10.1080/10298436.2021.1888092

Chen, Y., Mancini, M., Zhu, X., & Akata, Z. (2024a). Semi-supervised and unsupervised deep visual learning: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 46*(3), 1327–1347. https://doi.org/10.1109/TPAMI.2022.3201576

Chen, Z., Lai, Z., Chen, J., & Li, J. (2024b). Mind marginal non-crack regions: Clustering-inspired representation learning for crack segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2024)* (pp. 12698–12708). https://doi.org/10.1109/CVPR52733.2024.01207

Chow, J. K., Su, Z., Wu, J., Li, Z., Tan, P. S., Liu, K.-F., … Wang, Y.-H. (2020a). Artificial intelligence-empowered pipeline for image-based inspection of concrete structures. *Automation in Construction, 120*, Article 103372. https://doi.org/10.1016/j.autcon.2020.103372

Chow, J. K., Su, Z., Wu, J., Tan, P. S., Mao, X., & Wang, Y.-H. (2020b). Anomaly detection of defects on concrete structures with the convolutional autoencoder. *Advanced Engineering Informatics, 45*, Article 101105. https://doi.org/10.1016/j.aei.2020.101105

Chun, P.-J., & Kikuta, T. (2024). Self-training with Bayesian neural networks and spatial priors for unsupervised domain adaptation in crack segmentation. *Computer-Aided Civil and Infrastructure Engineering, 39*(17), 2642–2661. https://doi.org/10.1111/mice.13315

Federal Highway Administration. (2012). *Bridge Inspector's Reference Manual (BIRM)*. Washington, DC: U.S. Department of Transportation. Publication No. FHWA-NHI-12-049 (Revised February 2012).

Ferles, C., Papanikolaou, Y., & Naidoo, K. J. (2018). Denoising autoencoder self-organizing map (DASOM). *Neural Networks, 105*, 112–131. https://doi.org/10.1016/j.neunet.2018.04.016

Flotzinger, J., Rösch, P. J., & Braml, T. (2024). dacl10k: Benchmark for semantic bridge damage segmentation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)* (pp. 8611–8620). https://doi.org/10.1109/WACV57701.2024.00843

Forest, F., Lebbah, M., Azzag, H., & Lacaille, J. (2021). Deep embedded self-organizing maps for joint representation learning and topology-preserving clustering. *Neural Computing and Applications, 33*(24), 17439–17469. https://doi.org/10.1007/s00521-021-06331-w

Fortuin, V., Hüser, M., Locatello, F., Strathmann, H., & Rätsch, G. (2019). SOM-VAE: interpretable discrete representation learning on time series. In *Proceedings of the 7th International Conference on Learning Representations (ICLR 2019)*. https://openreview.net/pdf?id=rygjcsR9Y7.

Frangopol, D. M., & Liu, M. (2007). Maintenance and management of civil infrastructure based on condition, safety, optimization, and life-cycle cost. *Structure and Infrastructure Engineering, 3*(1), 29–41. https://doi.org/10.1080/15732470500253164

Gao, Y., & Mosalam, K. M. (2018). Deep transfer learning for image-based structural damage recognition. *Computer-Aided Civil and Infrastructure Engineering, 33*(9), 748–768. https://doi.org/10.1111/mice.12363

Gao, Y., & Spencer, B. F., Jr. (2008). *Structural Health Monitoring Strategies for Smart Sensor Networks*. Urbana, IL, USA: Newmark Structural Engineering Laboratory Report Series, NSEL Report No. 011, University of Illinois at Urbana–Champaign.

Ghasedi Dizaji, K., Herandi, A., Deng, C., Cai, W., & Huang, H. (2017). Deep clustering via joint convolutional autoencoder embedding and relative entropy minimization. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)* (pp. 5747– 5756). https://doi.org/10.1109/ICCV.2017.612.

Guo, J., Wang, Q., & Li, Y. (2021). Semi-supervised learning based on convolutional neural network and uncertainty filter for façade defects classification. *Computer-Aided Civil and Infrastructure Engineering, 36*(3), 302–317. https://doi.org/10.1111/mice.12632

Guo, X., Gao, L., Liu, X., & Yin, J. (2017b). Improved deep embedded clustering with local structure preservation. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence (IJCAI-17)* (pp. 1753–1759). https://doi.org/10.24963/ijcai.2017/243

Guo, X., Liu, X., Zhu, E., & Yin, J. (2017a). Deep Clustering with Convolutional Autoencoders. In D. Liu, S. Xie, Y. Li, D. Zhao, & E. S. El-Alfy (Eds.), *Neural Information Processing (ICONIP 2017). Lecture Notes in Computer Science, 10635* pp. 373–382). Cham: Springer. https://doi.org/10.1007/978-3-319-70096-0_39.

Guo, Y., Xu, Y., Cui, H., Dang, M., & Li, S. (2025). Segment anything model-based crack segmentation using low-rank adaption fine-tuning. *Structural Health Monitoring, 24*(4), 2579–2591. https://doi.org/10.1177/14759217241261089

Hou, Y., Chen, Y.-H., Gu, X.-Y., Mao, Q., Cao, D.-D., Wang, L.-B., & Jing, P. (2020). Automatic Identification of Pavement Objects and Cracks Using the Convolutional Auto-encoder. *China Journal of Highway and Transport, 33*(10), 288–303. https://doi.org/10.19721/j.cnki.1001-7372.2020.10.022

Hsieh, Y.-A., & Tsai, Y. J. (2020). Machine learning for crack detection: Review and model performance comparison. *Journal of Computing in Civil Engineering, 34*(5), Article 04020038. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000918

Huijben, I. A. M., Nijdam, A. A., Overeem, S., van Gilst, M. M., & van Sloun, R. J. G. (2023). SOM-CPC: Unsupervised contrastive learning with self-organizing maps for structured representations of high-rate time series. In *, 202. Proceedings of the 40th International Conference on Machine Learning (ICML 2023)* (pp. 14132–14152). Proceedings of Machine Learning Research (PMLR).

Hutchinson, T. C., & Chen, Z. (2006). Improved image analysis for evaluating concrete damage. *Journal of Computing in Civil Engineering, 20*(3), 210–216. https://doi.org/10.1061/(ASCE)0887-3801(2006)20:3(210)

Huyan, J., Li, W., Tighe, S. L., Deng, R., & Yan, S. (2020). Illumination compensation model with k-means algorithm for detection of pavement surface cracks with shadow. *Journal of Computing in Civil Engineering, 34*(1), Article 04019049. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000869

Jin, H., Ye, Z., & Su, J. (2015). Pavement crack detection using spectral clustering method. *International Journal of Image, Graphics and Signal Processing, 7*(2), 56–62. https://doi.org/10.5815/ijigsp.2015.02.08

Jutz, L., & Özcan, B. (2022). Automated detection of cracks based on statistical analysis of image histograms. In *Proceedings of 33. Forum Bauinformatik* (pp. 140– 147). Munich, Germany.

Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A. C., Lo, W.-Y., Dollár, P., & Girshick, R. (2023). Segment Anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV 2023)* (pp. 4015–4026). https://doi.org/10.1109/ICCV51070.2023.00371

Koch, C., Georgieva, K., Kasireddy, V., Akinci, B., & Fieguth, P. (2015). A review on computer vision based defect detection and condition assessment of concrete and asphalt civil infrastructure. *Advanced Engineering Informatics, 29*(2), 196–210. https://doi.org/10.1016/j.aei.2015.01.008

Kohonen, T. (2001). Self-Organizing Maps (3rd ed.). In *Springer Series in Information Sciences: 30Self-Organizing Maps*. Berlin, Heidelberg: Springer https://doi.org/10.1007/978-3-642-56927-2.

Li, H., Song, D., Liu, Y., & Li, B. (2019). Automatic pavement crack detection by multi-scale image fusion. *IEEE Transactions on Intelligent Transportation Systems, 20*(6), 2025–2036. https://doi.org/10.1109/TITS.2018.2856928

Liu, C., & Xu, B. (2022). A night pavement crack detection method based on image-to-image translation. *Computer-Aided Civil and Infrastructure Engineering, 37*(13), 1737–1753. https://doi.org/10.1111/mice.12849

Liu, C., & Xu, B. (2023). Weakly-supervised structural surface crack detection algorithm based on class activation map and superpixel segmentation. *Advances in Bridge Engineering, 4*, 27. https://doi.org/10.1186/s43251-023-00106-0

Locatello, F., Bauer, S., Lucic, M., Rätsch, G., Gelly, S., Schölkopf, B., & Bachem, O. (2019). Challenging common assumptions in the unsupervised learning of disentangled representations. In *, 97. Proceedings of the 36th International Conference on Machine Learning (ICML 2019)* (pp. 4114–4124). PMLR. https://doi.org/10.48550/arXiv.1811.12359.

Manduchi, L., Hüser, M., Faltys, M., Vogt, J., Rätsch, G., & Fortuin, V. (2021). T-DPSOM: An interpretable clustering method for unsupervised learning of patient health

states. In *Proceedings of the Conference on Health, Inference, and Learning (ACM CHIL 2021)* (pp. 236–245). ACM. https://doi.org/10.1145/3450439.3451872.
Mathavan, S., Rahman, M., & Kamal, K. (2012). Application of texture analysis and Kohonen map for region segmentation of pavement images for crack detection. *Transportation Research Record, 2304*, 150–157. https://doi.org/10.3141/2304-17
Mathavan, S., Rahman, M., & Kamal, K. (2015). Use of a self-organizing map for crack detection in highly textured pavement images. *Journal of Infrastructure Systems, 21*(3), Article 04014052. https://doi.org/10.1061/(ASCE)IS.1943-555X.0000237
Mathavan, S., Vaheesan, K., Kumar, A., Chandrakumar, C., Kamal, K., Rahman, M., & Stonecliffe-Jones, M. (2017). Detection of pavement cracks using tiled fuzzy Hough transform. *Journal of Electronic Imaging, 26*(5), Article 053008. https://doi.org/10.1117/1.JEI.26.5.053008
Mohan, A., & Poobal, S. (2018). Crack detection using image processing: A critical review and analysis. *Alexandria Engineering Journal, 57*(2), 787–798. https://doi.org/10.1016/j.aej.2017.01.020
Moor, M., Horn, M., Rieck, B., & Borgwardt, K. (2020). Topological autoencoders. In *, 119. Proceedings of the 37th International Conference on Machine Learning (ICML 2020)* (pp. 7045–7054). PMLR. https://doi.org/10.48550/arXiv.1906.00722.
Mosalam, K. M., & Gao, Y. (2024). *Artificial Intelligence in Vision-Based Structural Health Monitoring*. Cham: Springer. https://doi.org/10.1007/978-3-031-52407-3
Nguyen, H. N., Kam, T. Y., & Cheng, P. Y. (2014). An automatic approach for accurate edge detection of concrete crack utilizing 2D geometric features of crack. *Journal of Signal Processing Systems, 77*(3), 221–240. https://doi.org/10.1007/s11265-013-0813-8
Noh, Y., Koo, D., Kang, Y. M., Park, D., & Lee, D. (2017). Automatic crack detection on concrete images using segmentation via fuzzy C-means clustering. In *2017 International Conference on Applied System Innovation (ICASI)* (pp. 877–880). https://doi.org/10.1109/ICASI.2017.7988574
Ogawa, S., Matsushima, K., & Takahashi, O. (2019). Crack detection based on Gaussian mixture model using image filtering. In *2019 International Symposium on Electrical and Electronics Engineering (ISEE)* (pp. 79–84). https://doi.org/10.1109/ISEE2.2019.8921060
Oliveira, H., & Correia, P. L. (2010). Automatic crack detection on road imagery using anisotropic diffusion and region linkage. In *Proceedings of the 18th European Signal Processing Conference (EUSIPCO)* (pp. 274–278).
Oliveira, H., & Correia, P. L. (2014). CrackIT—An image processing toolbox for crack detection and characterization. In *Proceedings of the 2014 IEEE International Conference on Image Processing (ICIP)* (pp. 798–802). https://doi.org/10.1109/ICIP.2014.7025160
Omer, M., Margetts, L., Mosleh, M. H., & Cunningham, L. S. (2021). Inspection of concrete bridge structures: Case study comparing conventional techniques with a virtual reality approach. *Journal of Bridge Engineering, 26*(10), Article 05021010. https://doi.org/10.1061/(ASCE)BE.1943-5592.0001759
Omoebamije, O. (2023). Concrete & Pavement Crack Dataset [Data set]. *Mendeley Data*. https://doi.org/10.17632/429vzbgmbx.1
Özcan, B. (2026). *CCIC-O (Concrete Crack Images for Classification - Orientation) [Data set]*. Zenodo. https://doi.org/10.5281/zenodo.20042918
Özcan, B., Crampen, D., Kratzer, Z., & Blankenbach, J. (2024). Holistic image-based analysis of damage on concrete surfaces—A multifaceted approach based on supervised machine learning. *Structural Concrete, 25*(5), 3121–3144. https://doi.org/10.1002/suco.202300626
Özgenel, Ç. F. (2019). Concrete Crack Images for Classification [Data set]. *Mendeley Data*. https://doi.org/10.17632/5y9wdsg2zt.2
Özgenel, Ç. F., & Gönenç Sorguç, A. (2018). Performance comparison of pretrained convolutional neural networks on crack detection in buildings. In *Proceedings of the 35th International Symposium on Automation and Robotics in Construction (ISARC 2018)* (pp. 693–700). https://doi.org/10.22260/ISARC2018/0094
Pesteie, M., Abolmaesumi, P., & Rohling, R. (2018). Deep Neural Maps. In *ICLR 2018 Workshop*. arXiv:1810.07291.
Rafiei, M. H., & Adeli, H. (2018). A novel unsupervised deep learning model for global and local health condition assessment of structures. *Engineering Structures, 156*, 598–607. https://doi.org/10.1016/j.engstruct.2017.10.070
Ro, S. H., & Gong, J. (2024). Scalable approach to create annotated disaster image database supporting AI-driven damage assessment. *Natural Hazards, 120*, 11693–11712. https://doi.org/10.1007/s11069-024-06641-x
Song, Q., Yao, W., Tian, H., Guo, Y., Muniyandi, R. C., & An, Y. (2024). Two-stage framework with improved U-Net based on self-supervised contrastive learning for pavement crack segmentation. *Expert Systems with Applications, 238*, Article 122406. https://doi.org/10.1016/j.eswa.2023.122406
Spencer, B. F., Jr., Hoskere, V., & Narazaki, Y. (2019). Advances in computer vision-based civil infrastructure inspection and monitoring. *Engineering, 5*(2), 199–222. https://doi.org/10.1016/j.eng.2018.11.030
Subirats, P., Dumoulin, J., Legeay, V., & Barba, D. (2026). Automation of pavement surface crack detection using the continuous wavelet transform. In *Proceedings of the 2006 International Conference on Image Processing (ICIP)* (pp. 3037–3040). https://doi.org/10.1109/ICIP.2006.313007
Sun, Q., Zheng, R., & Xu, B. (2024). Bayesian-optimized unsupervised semantic segmentation model for structural crack detection. *Advances in Structural Engineering, 27*(3), 432–449. https://doi.org/10.1177/13694332231222093
Sun, X., & Chang, P. (2026). Unsupervised and noise-resilient crack detection in concrete and masonry via self-organizing maps. *Engineering Structures, 352*, Article 122130. https://doi.org/10.1016/j.engstruct.2026.122130
Sun, Z., Caetano, E., Pereira, S., & Moutinho, C. (2023). Employing histogram of oriented gradient to enhance concrete crack detection performance with classification algorithm and Bayesian optimization. *Engineering Failure Analysis, 150*, Article 107351. https://doi.org/10.1016/j.engfailanal.2023.107351
Sylolypavan, A., Sleeman, D., Wu, H., & Sim, M. (2023). The impact of inconsistent human annotations on AI driven clinical decision making. *npj Digital Medicine, 6*, 26. https://doi.org/10.1038/s41746-023-00773-3
van den Oord, A., Li, Y., & Vinyals, O. (2018) Representation learning with contrastive predictive coding. arXiv preprint. doi:10.48550/arXiv.1807.03748.
Van Gansbeke, W., Vandenhende, S., Georgoulis, S., Proesmans, M., & Van Gool, L. (2020). SCAN: Learning to classify images without labels, *12355. Computer Vision – ECCV 2020, Lecture Notes in Computer Science (LNCS)* (pp. 268–285). Springer. https://doi.org/10.1007/978-3-030-58607-2_16
Wu, J., & Liu, C. (2025). VQ-VAE-2 based unsupervised algorithm for detecting concrete structural apparent cracks. *Materials Today Communications, 44*, Article 112075. https://doi.org/10.1016/j.mtcomm.2025.112075
Xie, J., Girshick, R., & Farhadi, A. (2016). Unsupervised deep embedding for clustering analysis. In *, 48. Proceedings of the 33rd International Conference on Machine Learning (ICML 2016)* (pp. 478–487). PMLR.
Xu, B., & Liu, C. (2022). Pavement crack detection algorithm based on generative adversarial network and convolutional neural network under small samples. *Measurement, 196*, Article 111219. https://doi.org/10.1016/j.measurement.2022.111219
Xu, H., Su, X., Wang, Y., Cai, H., Cui, K., & Chen, X. (2019). Automatic bridge crack detection using a convolutional neural network. *Applied Sciences, 9*(14), 2867. https://doi.org/10.3390/app9142867
Xu, W., Tang, Z., Zhou, J., & Ding, J. (2013). Pavement crack detection based on saliency and statistical features. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)* (pp. 4093–4097). https://doi.org/10.1109/ICIP.2013.6738843
Yamaguchi, T., & Hashimoto, S. (2010). Fast crack detection method for large-size concrete surface images using percolation-based image processing. *Machine Vision and Applications, 21*(5), 797–809. https://doi.org/10.1007/s00138-009-0189-8
Yamaguchi, T., & Mizutani, T. (2024). Road crack detection interpreting background images by convolutional neural networks and a self-organizing map. *Computer-Aided Civil and Infrastructure Engineering, 39*(11), 1616–1640. https://doi.org/10.1111/mice.13132
Yang, X., Li, H., Yu, Y., Luo, X., Huang, T., & Yang, X. (2018). Automatic pixel-level crack detection and measurement using fully convolutional network. *Computer-Aided Civil and Infrastructure Engineering, 33*(12), 1090–1109. https://doi.org/10.1111/mice.12412
Yao, Y., Zhang, Y., Liu, Z., & Yuan, H. (2025). A bridge crack segmentation algorithm based on fuzzy C-means clustering and feature fusion. *Sensors, 25*(14), 4399. https://doi.org/10.3390/s25144399
Yeum, C. M., & Dyke, S. J. (2015). Vision-based automated crack detection for bridge inspection. *Computer-Aided Civil and Infrastructure Engineering, 30*(10), 759–770. https://doi.org/10.1111/mice.12141
Yuan, Q., Shi, Y., & Li, M. (2024). A review of computer vision-based crack detection methods in civil infrastructure: Progress and challenges. *Remote Sensing, 16*(16), 2910. https://doi.org/10.3390/rs16162910
Yun, H. B., Mokhtari, S., & Wu, L. (2015). Crack recognition and segmentation using morphological image-processing techniques for flexible pavements. *Transportation Research Record: Journal of the Transportation Research Board, 2523*, 115–124. https://doi.org/10.3141/2523-13
Zalama, E., Gómez-García-Bermejo, J., Medina, R., & Llamas, J. (2014). Road crack detection using visual features extracted by Gabor filters. *Computer-Aided Civil and Infrastructure Engineering, 29*(5), 342–358. https://doi.org/10.1111/mice.12042
Zhang, K., Zhang, Y., & Cheng, H. D. (2020). Self-supervised structure learning for crack detection based on cycle-consistent generative adversarial networks. *Journal of Computing in Civil Engineering, 34*(3), Article 04020004. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000883
Zhang, Y., Konz, N., Kramer, K., & Mazurowski, M. A. (2026). Quantifying the limits of segmentation foundation models: Modeling challenges in segmenting tree-like and low-contrast objects. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)* (pp. 5205–5215). https://doi.org/10.1109/WACV61042.2026.00505
Zheng, Y., Gao, Y., Lu, S., & Mosalam, K. M. (2022). Multistage semisupervised active learning framework for crack identification, segmentation, and measurement of bridges. *Computer-Aided Civil and Infrastructure Engineering, 37*(9), 1089–1108. https://doi.org/10.1111/mice.12851
Zhou, S., Canchila, C., & Song, W. (2023). Deep learning-based crack segmentation for civil infrastructure: data types, architectures, and benchmarked performance. *Automation in Construction, 146*, Article 104678. https://doi.org/10.1016/j.autcon.2022.104678