Construction and Building Materials 344 (2022) 128264

Contents lists available at ScienceDirect

Construction and Building Materials

journal homepage: www.elsevier.com/locate/conbuildmat

TOPO-Loss for continuity-preserving crack detection using deep learning

B.G. Pantoja-Rosero [a], D. Oner [b], M. Kozinski [c], R. Achanta [d], P. Fua [b], F. Perez-Cruz [d], K. Beyer [a,*]

[a] Earthquake Engineering and Structural Dynamics Laboratory (EESD), EPFL, 1015 Lausanne, Switzerland
[b] Computer Vision Laboratory (CVLab), EPFL, 1015 Lausanne, Switzerland
[c] Institute of Computer Vision and Graphics, Technical University of Graz, 8010 Graz, Austria
[d] Swiss Data Science Center (SDSC), EPFL and ETH Zurich, 1015 Lausanne, Switzerland

ARTICLE INFO

Keywords:
Crack detection
Deep learning
Post-earthquake assessment
Masonry buildings

ABSTRACT

We present a method for segmenting cracks in images of masonry buildings damaged by earthquakes. Existing methods of crack detection fail to preserve the continuity of cracks, and their performance deteriorates with imprecise training labels. We address these problems by adapting an approach previously proposed for reconstructing roads in aerial images, in which a Convolutional Neural Network is trained with a loss function specifically designed to encourage the continuity of thin structures and to accommodate imprecise annotations. We evaluate combinations of three loss functions (the Mean Squared Error, the Dice loss and the new connectivity-oriented loss) on two datasets using TernausNet, a deep network shown to attain state-of-the-art accuracy in crack detection. We herein show that combining these three losses significantly improves the topology of the predictions quantitatively and qualitatively. We also propose a new continuity metric, named Cracks Per Patch (CPP), and share a new dataset of images of earthquake-affected urban scenes accompanied by crack annotations. The dataset and implementations are publicly available for future studies and benchmarking (https://github.com/eesd-epfl/topo_crack_detection and https://doi.org/10.5281/zenodo.6769028).

1. Introduction

Masonry buildings are among the most vulnerable structures under seismic loads [1], so it is important to evaluate their structural behavior after any such event. However, current methodologies for their post-earthquake damage assessment rely on visual inspection by engineers, which is time-consuming and arduous as well as subjective in nature [2]. This motivates the development of faster and objective approaches for damage assessment, which is possible thanks to the recent breakthroughs in deep learning and artificial intelligence.

In a post-earthquake assessment, the first step is invariably damage detection, particularly crack detection. Any damage features revealed by this assessment can be correlated with mechanical properties using constitutive models developed from experimental campaigns [3–5]. Crack detection can be automated through deep learning, such as by convolutional neural networks (CNNs) [6]. Numerous studies are dedicated to the use of CNNs to semantically segment cracks in different materials [7–21].

However, all such approaches suffer from a major shortcoming—cracks are detected at the pixel level, *i.e.*, each pixel is individually labeled as belonging to a crack or otherwise, with no regard for the continuity of the cracks. This problem arises due to the use of pixel-based loss functions, like the Mean Squared Error (MSE) or Dice loss, which often detect fragments of cracks without preserving their continuity. When used in numerical models, these automatically detected crack fragments can contribute considerable errors in the estimated stress distribution.

When damaged structures are numerically modeled, incorrectly detected cracks can affect analysis results. If a continuous crack is modeled as discontinuous, the model disguises the actual structural behavior, making it stiffer and redistributing the stress differently. Such differences in mechanical analyses may overturn crucial decisions for managing damaged structures. In extreme scenarios, overestimating the stiffness of the structure may lead decision makers to underestimate damage and attempt to repair a building that is close to structural collapse. Conversely, underestimating the stiffness could lead to overestimate the damage and demolish a building that could be repaired, generating unnecessary costs.

A further situation where continuity-preserving crack detection is important occurs in masonry structures. In these structures, a diagonal crack often follows the mortar joints and is thus not straight, resulting in changes in the opening mode along the crack (Mode I, Mode II and

* Corresponding author.
E-mail addresses: bryan.pantojarosero@epfl.ch (B.G. Pantoja-Rosero), doruk.oner@epfl.ch (D. Oner), mateusz.kozinski@icg.tugraz.at (M. Kozinski), radhakrishna.achanta@epfl.ch (R. Achanta), pascal.fua@epfl.ch (P. Fua), fernando.perezcruz@epfl.ch (F. Perez-Cruz), katrin.beyer@epfl.ch (K. Beyer).

https://doi.org/10.1016/j.conbuildmat.2022.128264
Received 24 February 2022; Received in revised form 20 June 2022; Accepted 23 June 2022
Available online 30 June 2022

mixed mode) [22]. After unloading, Mode I crack segments tend to close, whereas Mode II crack segments remain open. As a result, Mode I crack segments are frequently difficult to detect and a long diagonal crack may appear as many short segments. The current state-of-the-art cannot address such situations where it is critical to detect cracks while maintaining the continuity of the crack topology.

Another drawback of current crack segmentation techniques is that they require pixel-precise labels to train CNNs as well as to assess their performance. However, the preparation of such pixel-precise labels requires significant manual effort, making it difficult, if not impossible, to do so for real-world images due to the wide range of variation in the image sizes and quality. Moreover, it is often difficult to determine the exact trajectory of cracks in images of entire facades taken from a considerable distance.

In this paper, we overcome both of these shortcomings. We first make use of the topological loss (TOPO) [23], which specifically penalizes discontinuities. For the assessment of continuity, we introduce a new metric that counts the number of cracks in an input image. We next show that this loss can accommodate coarse labels, significantly reducing the labeling effort needed to deploy our system.

We present experiments on two datasets, in which we compare the performance of TernausNet [24], a standard deep network architecture previously used for crack detection [25], that has been trained with different loss functions. We specifically focus on the capacity of the network to correctly represent the topology of cracks. To compare the topology of the predicted and annotated cracks, we introduce a new dedicated metric that we call Cracks Per Patch (CPP). Both qualitative and quantitative results show that the use of TOPO significantly improves crack continuity while reducing false positives. We make both our code and the new dataset publicly available.

The contributions of this paper are as follows:

- A thorough evaluation of different loss functions in crack detection, emphasizing the correct representation of crack topology (preserving continuity of the detected cracks).
- A solution for continuity crack detection problems that does not require pixel-precise labels, attained through the adaptation of an existing method for road network segmentation.
- A new metric to assess continuity preservation in crack prediction.
- A new training dataset of real-world building images containing labeled cracks.

2. Related work

One way to automatically assess the state of a building affected by an earthquake is through image analysis, beginning with the detection of damage and deterioration. Recent approaches to this problem can be divided into three main categories [26,27]: heuristic feature extraction, change detection and deep learning. The first approach applies a threshold or a machine learning classifier to the output of a hand-crafted filter [28–30]. The second approach establishes a baseline representation of the structure that is compared against data from subsequent inspections [31–33]. The third approach employs deep learning and may be combined with heuristics [8,11,15].

One of the first examples of using deep learning to detect cracks in civil engineering structures was presented by Zhang et al. [7]. Inspired by similar works in computer vision and medical imaging, the authors proposed a CNN-based method for crack detection in pavements. Building on this approach, several other approaches were introduced for specific scenarios using variations in the network architecture to segment, detect or classify damage. For instance, Zhang et al. [8] proposed a CNN architecture that does not have pooling layers to downsize outputs of previous layers and used as input data feature maps generated by line filters. Cha et al. [11] used CNNs to detect concrete cracks without calculating the defect features, along with a sliding window technique to scan any image size. Chen et al. [34] integrated a CNN and a Naïve Bayes data fusion scheme to analyze individual video frames for crack detection. Hoskere et al. [13] proposed a framework for generating vision-based condition-aware models to aid inspection decisions by projecting CNN results to photogrammetry based 3D mesh models. Kim et al. [15] presented an automated detection technique using CNNs for crack morphology on concrete surface under an on-site environment. An appreciable improvement in crack predictions was presented by Liu et al. [16], where the widely used U-Net architecture [35] was trained to detect cracks in concrete.

Specifically related to buildings, Ghosh et al. [36] detected damage through a region-based CNN architecture that uses bounding boxes to locate four types of damage: cracks, spalling, spalling with exposed rebars and buckled reinforcement. Bai et al. [10] used different CNN models to instead perform pixel-wise semantic segmentation by training models to predict cracks in images at different levels—the pixel level, object level and structural level. For masonry material in particular, a complete review can be found in [37]. Chaiyasarn et al. [12] proposed a crack segmentation system that combined deep CNN and Support Vector Machines (SVM). Ali et al. [9] used R-CNN to detect damaged bricks in buildings, and Rezaie et al. [25] compared the performance of CNN and Digital Image Correlation (DIC) to detect cracks in plastered stone masonry walls from an experimental campaign. More recently, similar work was presented by Dai et al. [17], where the authors segmented cracks on masonry buildings using FCNN with pre-trained encoders trained with cross-entropy loss.

These approaches are highly accurate for detecting pixels that represent structural damage. However, as we will show, they still fail to reliably represent the continuity of cracks. This is not a contradiction, since it takes only a few mis-classified pixels to break crack continuity, though this incurs little penalty in terms of per-pixel accuracy. Addressing this shortcoming is the main contribution of this paper.

Li et al. [38] and Zhang et al. [39] present works in which continuity is helped to be preserved. In these works, continuity is maintained indirectly as a result of their proposed methodology, in which cracked patches are fused to provide a general context for the image. Unlike them, our approach focuses on directly solving the problem of conserving crack continuity on segmentation using CNN, as the loss function used is designed based on the crack pattern topology.

3. Methodology

In this section, we present our approach to crack detection. We first specify the network architecture, then define the loss functions, and finally detail the hyper-parameters used in our experiments.

3.1. Network architecture

All our experiments are based on the TernausNet architecture [24], which is a CNN consisting of an encoder and decoder, like the even more common U-Net [35]. The major difference between these architectures is that TernausNet uses the convolutional part of a pre-trained VGG network [40] as its encoder. This speeds-up convergence and produces more accurate results than a conventional U-Net, even when little training data is available. The architecture diagram is presented in Fig. 1.

Depending on the loss function that is used, we either train TernausNet to regress the truncated distance from each pixel to the nearest crack center or to classify pixels as cracks or background. For the classification experiments, we terminate the network as a Softmax layer. In the experiments aimed at distance regression, we use Rectified Linear Unite (ReLU) activation function.

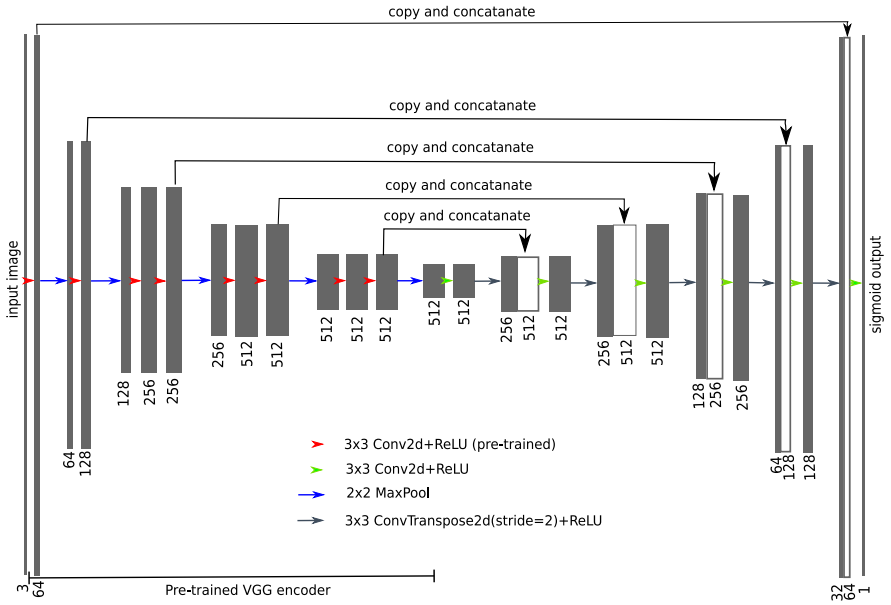

Fig. 1. The TernausNet CNN architecture [24]. The arrows represent the operations performed between the layers. The numbers on the layers' sides correspond to the number of kernels on the layer.

3.2. Loss functions—pixel classification

The standard approach for crack detection classifies each image pixel as crack or background. To formalize this approach, we denote the set of image pixels by I and individual pixels by $p \in I$. We denote the prediction by y_b and the corresponding binary annotation by $\hat{y}_b$. $y_b[p]$ and $\hat{y}_b[p]$ are the values of the prediction and the annotation for pixel p, where $\hat{y}_b[p] = 1$ if p represents a crack in a wall, and $\hat{y}_b[p] = 0$ otherwise. We evaluate the following two approaches to pixel classification.

*Mean Squared Error with binary ground truth (*MSE-BIN*).* The simplest approach for pixel classification is to enforce the per-pixel annotations on the network output by means of the MSE, defined as

$$L_{\text{MSE}}(y_b, \hat{y}_b) = \sum_{p \in I} (y_b[p] - \hat{y}_b[p])^2. \tag{1}$$

This basic loss here is typically outperformed by the more sophisticated Dice-loss alternative specified below.

*Dice loss (*DICE*).* The Dice loss [41], defined as

$$D(y_b, \hat{y}_b) = 1 - \frac{2 \sum_{p \in I} \hat{y}_b[p] y_b[p]}{\sum_{p \epsilon I} \hat{y}_b[p] + \sum_{p \epsilon I} y_b}, \tag{2}$$

has been specifically designed to maximize the Dice score (second term in Eq. (2) – also known as F1 score). In Section 4, we will show that it indeed outperforms the MSE-BIN on this metric.

3.3. Loss functions—distance regression

The alternative to pixel classification is to force the network to regress the distance from each pixel to the nearest crack center. The advantage of this approach is that, by design, it is robust to annotations that are not ideally precise, *i.e.*, where the annotated crack trajectories might be slightly off.

For the experiments based on regression to the nearest crack center, we denote the distance map produced by a deep net by y_d and the corresponding ground truth distance map by $\hat{y}_d$. $\hat{y}_d[p]$ is the distance from the pixel p to the nearest crack center, truncated at $d_{\max} = 20$ pixels ($d_{\max}$ is a hyper-parameter tuned during training through cross-correlation). Formally, $\hat{y}_d[p] = \min\{\min_{q \in I,\ \text{s.t. } y_b[q]=1} d_{pq}, d_{\max}\}$, where d_{pq} is the distance between the pixels p and q. We use the following two loss functions to train TernausNet in regression.

*Mean Squared Error (*MSE*).* The basic approach here enforces the correct distance map on the output of the network by means of the MSE, as

$$L_{\text{MSE}}(y_d, \hat{y}_d) = \sum_{p \in I} (y_d[p] - \hat{y}_d[p])^2. \tag{3}$$

While distance regression is more robust to annotation inaccuracy than pixel classification, it is prone to the same type of topological errors as MSE-BIN and DICE, particularly in terms of interruptions in crack continuity.

*The topological loss (*TOPO*).* To address the failure of the previously introduced losses to prevent crack interruptions, we resort to the connectivity-oriented loss function TOPO, proposed by Oner et al. [23], to encourage continuity in roads reconstructed from aerial images. The TOPO loss function is composed of two terms:

$$L_{\text{TOPO}}(y_d, \hat{y}_d) = L_{\text{conn}}(y_d, \hat{y}_d) + \beta L_{\text{disc}}(y_d, \hat{y}_d). \tag{4}$$

L_{conn} penalizes crack discontinuity, while L_{disc} penalizes false crack detection. We define these terms later in this section. The parameter β balances the influence of L_{conn} and L_{disc}. As we will demonstrate in Section 4, increasing β decreases the number of false positives.

The definition of both L_{conn} and L_{disc}, presented by Oner et al. [23], is based on the fact that a crack subdivides a small image patch into two disconnected background regions. If a crack is continuous in the annotation but interrupted in the prediction, the annotation contains two disconnected background regions, but in the prediction these regions connect (Fig. 2). This erroneous connection between the background regions and the misclassified pixels through which the regions connect can be detected by means of the *maximin* connectivity approach. This approach was first proposed by Turaga et al. [42] for modeling the connectivity of cells observed under an electron microscope, and it was reused by Oner et al. [23] for modeling the connectivity of background regions in aerial images.

The central idea of the maximin approach is the notion of the maximin path between two pixels p and p' in the distance map y_d. We denote the set of all paths that connect p and p' in the pixel lattice as $\Pi(p, p')$ and formalize the maximin path as

$$\pi_{pp'}(y_d) = \arg\max_{\pi \in \Pi(p,p')} \min_{q \in \pi} y_d[q]. \tag{5}$$

We call the smallest pixel on the maximin path the critical pixel and denote it as $q^*_{pp'}(y_d)$. A maximin path between two pixels on opposite sides of a discontinuous crack is presented in Fig. 2e. Note that the path passes through the disconnection, and that its smallest pixel, marked red, is the pixel over which the background regions connect. This is not a coincidence and is instead a property of maximin paths that follows from their definition. In a perfect distance map, the value of the smallest pixel between a pair of pixels on opposite sides of an uninterrupted crack on the maximin path is equal to zero simply because the path has to cross the crack. If a gap in the crack is present, the maximin path passes through this gap, and the value of its smallest pixel can be larger than zero. The formulation of L_{conn} leverages this property by minimizing the smallest pixel on the maximin paths between pairs of pixels on opposite sides of annotated cracks.

More formally, to compute L_{conn}, the predicted and ground truth distance maps are first divided into small square windows that we denote w. The window size s_w is a hyper-parameter of the method. As shown in Fig. 2b, the crack annotation is dilated to accommodate a possible lack of accuracy in crack annotations. The dilated crack region, denoted $\mathfrak{R}$, separates the window into background regions. In Fig. 2b, there are two such regions, denoted A and B. We denote the set of all background regions as $\mathfrak{B}$. The connectivity component of the loss is then defined as

$$L_{\text{conn}}(y_d, \hat{y}_d) = \sum_{w \in W} \sum_{\substack{A,B \in \mathfrak{B} \\ A \neq B}} \sum_{\substack{p \in A \\ p' \in B}} (y_d[q^*_{pp'}(y_d)])^2, \tag{6}$$

where W is the set of all windows. The loss tends to bring the smallest pixel between each pair of pixels on the opposite sides of a crack on the maximin path to zero, enforcing crack continuity.

The definition of L_{disc}, aimed at preventing false crack prediction and false connections between separate cracks, relies on preventing low values of the smallest pixels on maximin paths between pairs of pixels that belong to the same background region. Formally,

$$L_{\text{disc}}(y_d, \hat{y}_d) = \sum_{w \in W} \sum_{A \in \mathfrak{B}} \sum_{p,p' \in A} (y_d[q^*_{pp'}(y_d)] - \hat{y}_d[q^*_{pp'}(y_d)])^2. \tag{7}$$

The loss simply encourages the smallest pixel on the maximin path to take its ground truth value. This prevents this pixel from assuming values that are too low, which would falsely suggest the presence of a crack.

Even though Eqs. (6) and (7) are sums over pixel pairs, in practice they can be computed efficiently using a modified version of Kruskal's maximum spanning tree algorithm. Fig. 2d highlights the intuition between the relation of the maximum spanning tree and the smallest pixel on the maximin path. The subtrees on the two sides of the crack are connected over this pixel. We refer the reader to [42] for more details.

TOPO has been shown to work best in combination with MSE, wherein the resulting loss function is defined as

$$L_{\text{TOPO+MSE}}(y_d, \hat{y}_d) = \alpha L_{\text{TOPO}}(y_d, \hat{y}_d) + L_{\text{MSE}}(y_d, \hat{y}_d), \tag{8}$$

where α is a hyper-parameter. Increasing α encourages more connectivity in the predictions.

3.4. Parameters and hyper-parameters

For all the trained models, we used the Adam optimizer [43] with a batch size equal to sixteen. Data augmentation consisted of horizontal and vertical flips and brightness and contrast changes, each applied with the probability of 0.5. Since TernausNet relies on a pre-trained encoder [24], we normalized the images to match the statistics of those used for pre-training, consisting of a mean of $[0.485, 0.456, 0.406]$ and standard deviation of $[0.229, 0.224, 0.225]$.

For model selection, we performed cross-validation using training and validation data. Hyper-parameters such as number of epochs, learning rate (lr), threshold applied to the prediction (thr), truncation distance (d_{max}) and Topoloss parameters (α, β and $ws = 32$ px) were determined through grid search. Tables 1 and 2 present these hyper-parameters for the selected models.

Table 1
Hyper-parameters used on the Expe dataset.

	epoch	lr	α	β	thr
MSE-BIN	100	1.0e−4	n/a	n/a	0.53
DICE	100	4.0e−5	n/a	n/a	0.50
MSE	200	7.0e−5	n/a	n/a	2.00
TOPO	100	1.8e−4	1.0	0.1	2.00
DICE+TOPO	100	1.8e−4	1.0	0.1	2.00
MSE+TOPO	200	1.8e−4	0.01	0.001	2.00

Table 2
Hyper-parameters used on the Wild dataset.

	epoch	lr	α	β	thr
MSE-BIN	50	1.0e−5	n/a	n/a	0.53
DICE	50	1.0e−5	n/a	n/a	0.50
MSE	50	5.0e−6	n/a	n/a	6
TOPO	50	3.0e−5	100	10	2
DICE+TOPO	50	3.0e−5	100	10	2
MSE+TOPO	50	3.0e−5	100	10	4

4. Experiments

In this section, we present the compared loss functions, the sample datasets, the evaluation metrics used for the comparison, and finally, the quantitative as well as qualitative results.

4.1. Methods tested

We trained TernausNet [24] with six different loss functions:

- MSE-BIN: The mean squared error, used with binary ground truth.
- DICE: The Dice loss function [41], used with binary ground truth.
- MSE: The mean squared error, used with a distance map for the ground truth.
- TOPO: The topological loss function [23], used with a distance map for the ground truth.
- DICE+TOPO: A weighted sum of DICE and TOPO.
- MSE+TOPO: A weighted sum of MSE and TOPO.

4.2. Datasets

We performed on our experiments on two datasets:

- Expe: Experimental stone masonry walls [25] – publicly available dataset used to benchmark our methodology. This dataset contains patches of images depicting stone masonry walls damaged due to shear-compression loading in an experimental setting by the EESD laboratory at EPFL [44]. Since the initial aim of collecting this dataset was to apply DIC techniques, the walls were plastered and marked with speckles evenly distributed over their surface. As shown in the Fig. 3, the images were hand-labeled by carefully marking the pixels that represent cracks. In total, the published dataset is composed of 301 training patches, 129 validation patches and 100 test patches, all sized 256 × 256 pixels and containing crack information.
- Wild: Damaged buildings in the wild. This dataset comprises images of stone masonry buildings and urban scenes damaged in real earthquakes. These images have been collected over the years by the EESD laboratory from various locations around the world. In total, there are 162 images of different sizes ($[min, max, mean]_{size} = [3.25, 36.15, 13.31]$ Mpx), which were manually annotated with coarse labels (Fig. 4). This dataset is one of our contributions and is publicly available. For real life data it is very difficult to produce pixel-wise annotation because image sizes and quality vary largely. For this reason we put forward a method for which we claim that coarse labels (brushing over an area with a crack) are

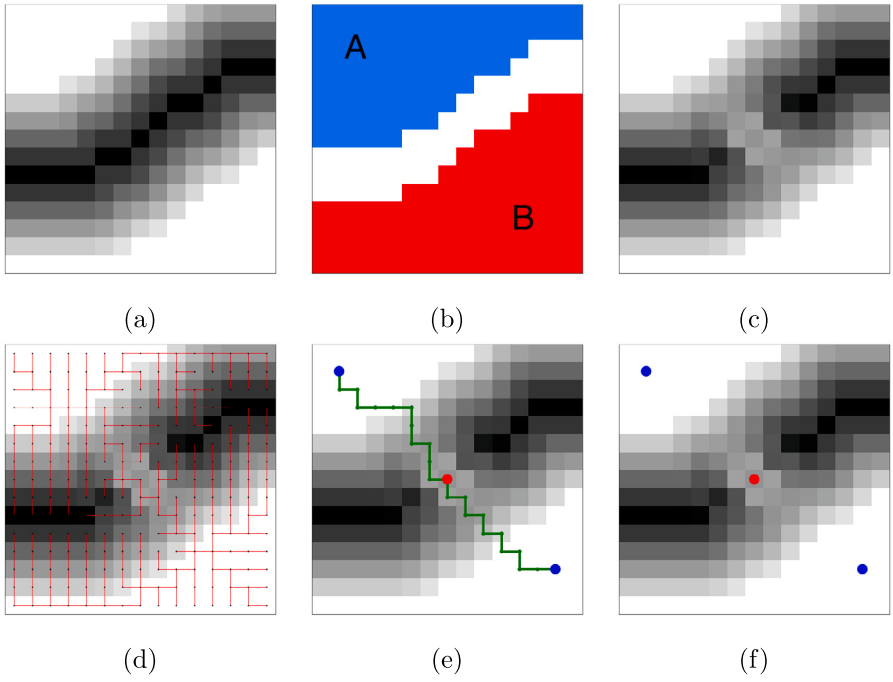

Fig. 2. *(a)* Ground truth distance map to crack center. *(b)* Background-connected components separated by the crack. *(c)* Prediction of the network. *(d)* Maximum spanning tree overlaid in red on top of the prediction. *(e)* Using the maximum spanning tree to find critical edges that create disconnections in the crack. The path connecting two points in the maximum spanning tree is called the maximin path, illustrated in green. *(f)* The edge with the minimum weight in the maximin path is the critical edge, illustrated in red, and the network is enforced to fix this edge by L_{TOPO}, hence fixing the disconnection [42]. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

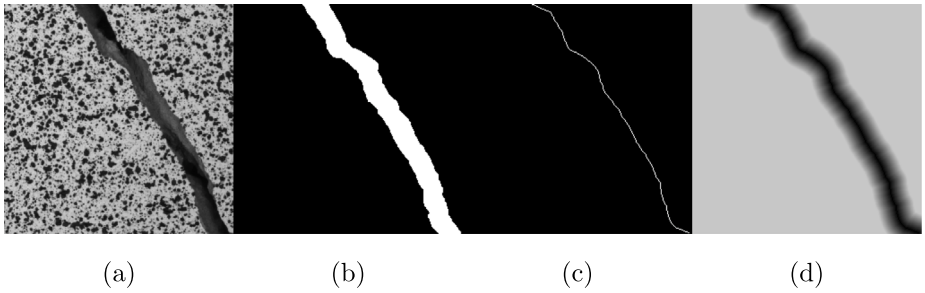

Fig. 3. Example image of the Expe dataset. *(a)* Original image, and *(b)* its manual binary annotation, used with MSE-BIN and DICE. *(c)* The skeleton of the binary and *(d)* its truncated distance map. We use such distance maps for training with MSE, TOPO and combinations thereof.

sufficient for training this deep learning model. We demonstrate this through the application of the method to this dataset.

To prepare the images for training, the original images were divided into non-overlapping patches of 256 × 256 pixels. Of these, only the patches depicting damaged parts (cracked images) of building walls were retained. This resulted in 5360 training, 1287 validation and 533 test patches. The training, validation and test patches all come from different full-resolution images. Along with these patches, we complete the dataset by adding 12 full-resolution images, which are the source of the 533 test patches, as well as their full-resolution manual annotations. Fig. 4 shows one of these images and the corresponding annotations.

Since MSE and TOPO rely on ground truths in the form of truncated distance maps to the crack center, we provide such annotations for both datasets along with binary masks. To generate the distance maps, we first skeletonized the binary annotations. Then, for each pixel, we computed its distance to the closest pixel of the skeleton and truncated the distance at the value of 20 pixels. Without this truncation, it would be difficult for a deep network to estimate the correct distance to the nearest crack at areas without any damage. Example distance maps are shown in Figs. 3d and 4f.

To classify each pixel as crack or background with MSE-BIN and DICE, a deep network was trained. This approach requires annotations in the form of binary masks, like the ones shown in Figs. 3b and 4d. However, as our labels are not pixel-precise especially for the Wild dataset, this could lead to noisy predictions when the network is trained with pixel-wise losses. Conversely, as we will demonstrate experimentally in this section, training the network to predict distance to the crack center, as opposed to classifying pixels as crack or background, makes the network more robust to this lack of precision in the annotations and that TOPO further amplifies this robustness.

4.3. Evaluation metrics

We used the following metrics to evaluate the results of our experiments:

- F_1, also known as the Dice score, is the most common metric for evaluating the results of binary segmentation algorithms. It is computed as the harmonic mean of the precision and recall that the algorithm attains in pixel classification. The expression

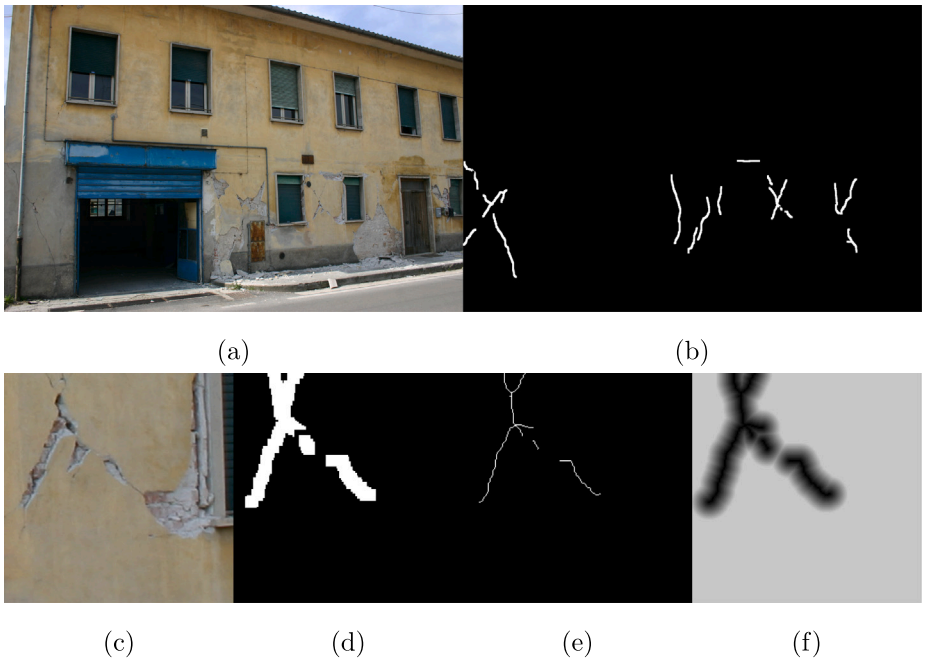

Fig. 4. Example image of the WILD dataset. *(a)* Original image, *(b)* Its coarse binary annotation. For training, we split the images into smaller non-overlapping patches. *(c)* A patch of the input image used for training, and *(d)* its annotation. *(e)* The binary skeleton of the annotation, and *(f)* the distance map computed from the skeleton.

for this metrics is:

$$\mathrm{F1} = \frac{2\sum_{p\in I} \hat{y}_{\mathrm{b}}[p] y_{\mathrm{b}}[p]}{\sum_{p\epsilon I} \hat{y}_{\mathrm{b}}[p] + \sum_{p\epsilon I} y_{\mathrm{b}}}, \tag{9}$$

where $y_{\mathrm{b}}[p]$ and $\hat{y}_{\mathrm{b}}[p]$ are the values of the prediction and the annotation for pixel $p \in I$.

It should be noted that this metric is intended to evaluate methods that use pixel-based loss functions for training and thus pixel precise-labels. We modify the metric to account for the fact that our approach is based on the crack skeleton. To accomplish this, we define a margin on the sides of the skeleton such that the skeleton prediction is considered correct if it falls within the margin. According to the truncated distance of the label distance maps, the margin is defined as 20 pixels on either side of the skeleton.

Please note that the F1 metric does not provide a way of evaluating the continuity preservation capability of our method because it is pixel-based. Therefore these values are presented as a reference since we are presenting a detection method in images.

- Cracks Per Patch (CPP) is a metric we propose for comparing crack topology. As argued in Section 3, correctly recovering the crack topology is crucial for producing useful crack segmentation results. The F_1, which is focused on evaluating the performance of the deep network in pixel classification, fails to capture the topological differences between the predicted and annotated cracks. To fill this gap, we compared the number of cracks in the annotation c_a to the number of cracks in the prediction c_p, as

$$\mathrm{CCP} = \left| c_a - c_p \right|. \tag{10}$$

4.4. Results

We present the results of the experiments on the EXPE dataset on the left side of Table 3. In terms of the F1 metric, the DICE outperformed other methods, followed in decreasing order of performance by MSE-BIN and MSE+TOPO. In contrast, in terms of our CPP measurement, MSE+TOPO attained the lowest error, outperforming other methods by a large margin. Upon inspection of the qualitative results in Fig. 5, this inconsistency of F1 with CPP may be attributed to the fact that MSE+TOPO produce confident and continuous crack predictions that are often thicker than or misaligned with the ground truth (GT), which produces a high penalty in terms of the pixel-level scores. This is expected, since TOPO focuses on preventing crack disconnections rather than enforcing pixel-level accuracy.

Table 3
Quantitative performance evaluation.

	EXPE		WILD	
	F1-skeleton	CPP	F1-skeleton	CPP
MSE-BIN	77.1	0.99	42.8	3.65
DICE	79.7	0.95	67.2	0.97
MSE	73.5	1.26	64.2	0.79
TOPO	72.0	0.91	64.4	1.82
DICE+TOPO	73.8	1.03	66.4	1.17
MSE+TOPO	76.5	0.24	68.5	0.62

The results attained for the WILD dataset are reported on the right side of Table 3. Here, MSE+TOPO outperformed other methods in terms of both performance measures. Inspection of the qualitative results in Figs. 6–8 confirms that this method yielded confident and uninterrupted cracks, albeit not perfectly overlapping the annotations. The difference in the results with respect to the EXPE dataset can be explained by the fact that the MSE and DICE are more sensitive to annotation inaccuracies. The relative lack of precision in the annotations we performed for WILD exposes this sensitivity and incurs a performance penalty. In contrast, the method based on distance regression, in particular MSE+TOPO, can accommodate such annotations without reducing the performance.

Superior scores of pixel-based metrics for the real world dataset (WILD dataset) can be obtained if we use more refined labels. However, as mentioned before, producing such detailed labeling in such images is a difficult task that is sensitive to their quality and size. As one of our method assets is the capability to deal with this type of datasets to train deep learning models and still produce good detections and preserve crack continuity, we do not focus in the refinement of the labels to improve such type of metrics (F1). Finally, it is important to highlight that there were no significant issues when the model was trained with

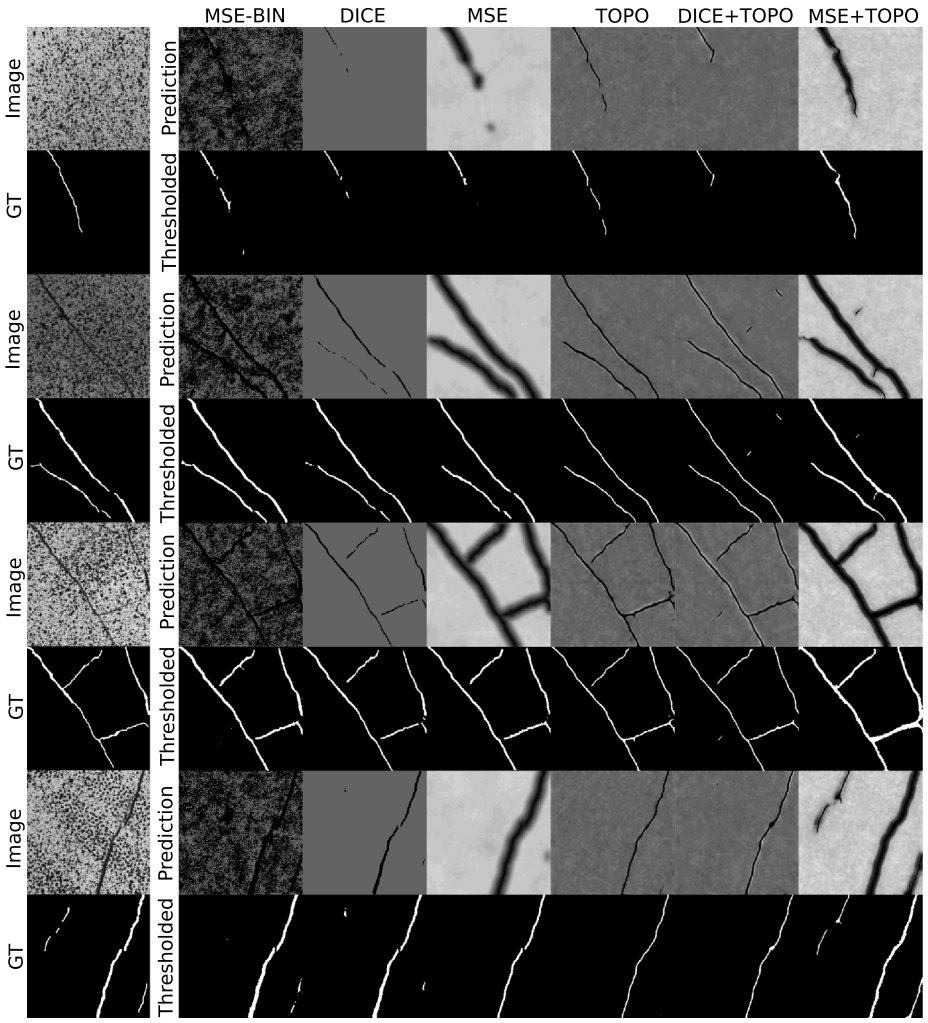

Fig. 5. The qualitative results of the Expe dataset. Images, ground truths, and predictions with their thresholded values for the different trained models.

the Wild dataset compared to the Expe dataset. This demonstrates the robustness of the CNN architecture as well as the loss function proposed in this work for images captured on-site.

Crack segmentation in the wild. Our model was tested on full-building images of varying dimensions much larger than the patch size 256×256 used for training the deep network. Figs. 9 and 10 show the capacity of the deep network trained with MSE+TOPO to detect cracks in images containing damaged buildings in the wild, that is, damaged buildings found in urban images also containing other elements of urban landscape. Visual features similar to cracks can cause false positive predictions, like the ones observed over the pavement in Fig. 10. These errors can be removed in post-processing, such as by detecting buildings in the images and constraining crack locations to lie within these detected areas. Another option for avoiding false positives is to add more training data containing similar information, from which the deep learning model can learn better how to make such distinctions. This can be seen, for example, in the case of window edges or handrails, where the model was able to distinguish cracks from them due to having enough image information during training. Fig. 9 presents the original images and the distance map outputs produced by the neural network. Figs. 10 presents the original image and its overlapped version with a thresholded binary mask obtained from the distance map.

5. Conclusions and future work

We demonstrated that correctly representing crack topology is crucial for the assessment of the mechanical properties of cracked structures. Unfortunately, the commonly used U-Net-like architectures are accurate in segmenting cracks, but fail to preserve their topology. Here, we show that the TOPO loss function can be used to improve the performance of U-Net in those aspects. We demonstrated this both in terms of qualitative and quantitative results, for which we proposed a novel metric, the CPP. Furthermore, TOPO allows to circumvent the use of precise labels reducing substantially the time for the annotation process which is known as one of the limitations of supervised deep learning techniques. Additionally, we showed that a deep network trained with a combination of TOPO and MSE yields convincing results when applied to images showing cracked buildings in the context of entire urban scenes. To facilitate future experimental comparisons, we released the Wild dataset and the code from our work.

The main drawback of using the topological loss function is a possible loss of precision in crack localization. Though loss enforces crack continuity, it relaxes the requirement of a perfect coincidence between the predicted and the annotated cracks. The estimated crack centers may therefore be slightly offset from the annotated ones, resulting in a lower accuracy in pixel classification, as highlighted in Section 4. Moreover, TOPO is designed to segment thin elongated structures, which may misrepresent the crack width. Additional post-processing may be applied if precise width estimates are desired.

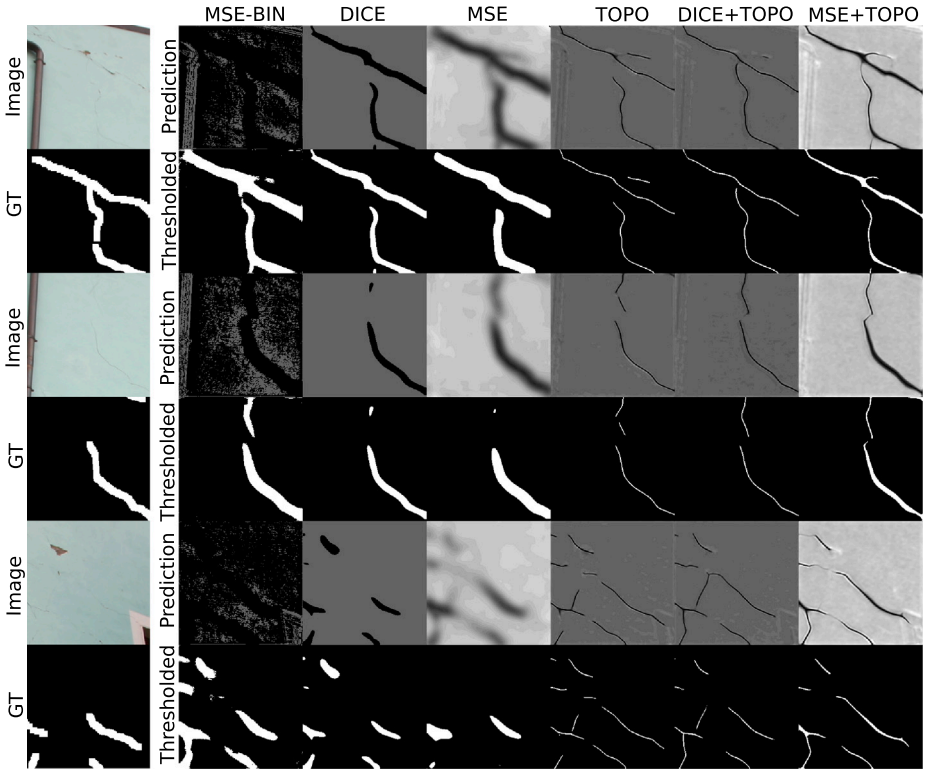

Fig. 6. Qualitative results for the WILD dataset. Patches from a single building A. Images, ground truths, and predictions with their thresholded values for the different trained models.

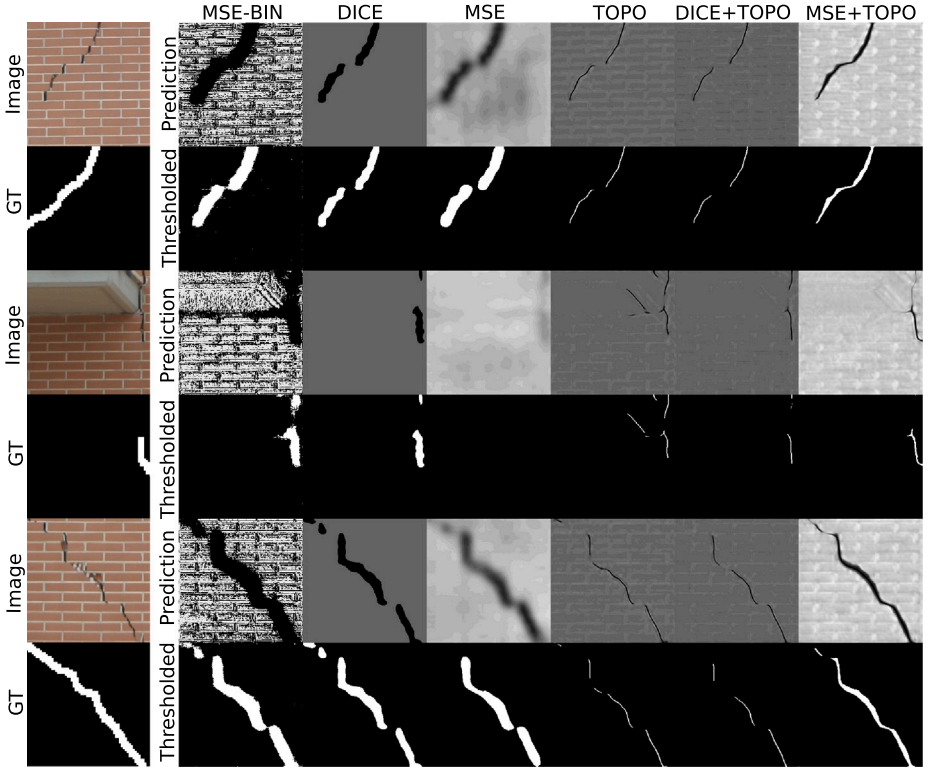

Fig. 7. Qualitative results for the WILD dataset. Patches from a single building B. Images, ground truths, and predictions with their thresholded values for the different trained models.

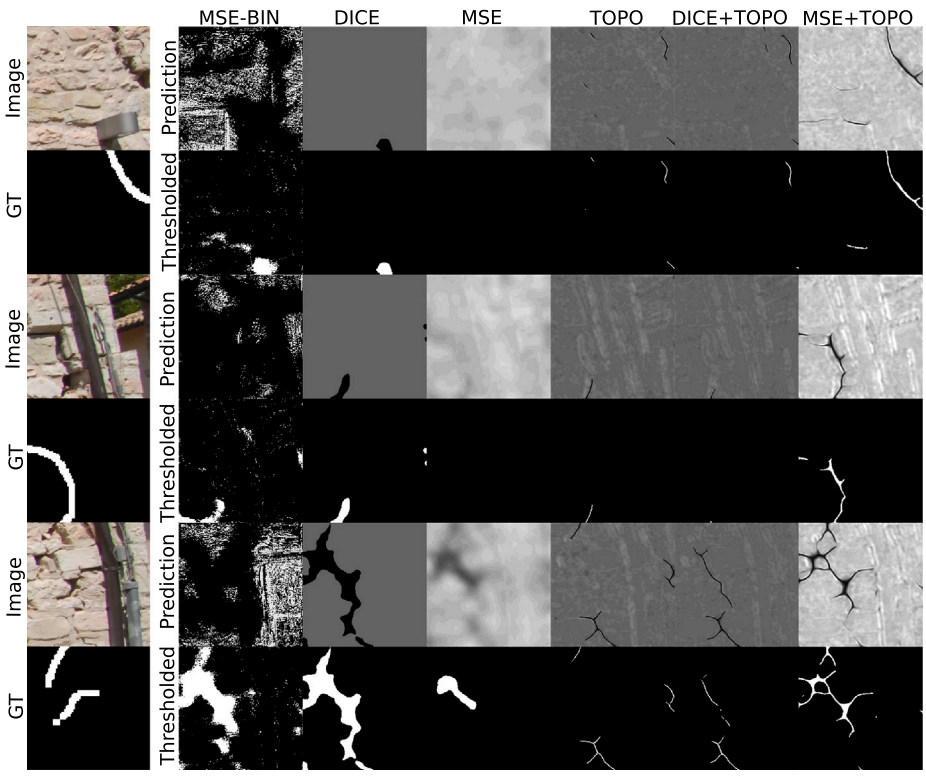

Fig. 8. Qualitative results for the WILD dataset. Patches from a single building C. Images, ground truths, and predictions with their thresholded values for the different trained models.

Fig. 9. Results of predictions on urban images. Original urban images and output distance map.

We plan to continue this work along two main directions. First, we will combine crack segmentation with semantic segmentation of urban scenes, which we expect to result in increased performance in detecting cracks in urban scenes. Second, to bridge the gap between crack detection in 2D images and the 3D assessment of the stability of damaged buildings, we plan to represent cracks in the geometric

Fig. 10. Results of predictions on urban images. Original urban images and overlapping thresholded cracks as binary mask.

context of the cracked structures. Our overarching aim with the work presented herein and our future experiments is to bring the automatic assessment of structural damage closer to real-life applications.

CRediT authorship contribution statement

B.G. Pantoja-Rosero: Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Writing – original draft, Writing – review & editing, Visualization. **D. Oner:** Methodology, Software, Conceptualization, Writing – original draft. **M. Kozinski:** Methodology, Conceptualization, Supervision, Writing – review & editing. **R. Achanta:** Conceptualization, Methodology, Resources, Supervision. **P. Fua:** Conceptualization, Supervision. **F. Perez-Cruz:** Resources, Supervision. **K. Beyer:** Conceptualization, Methodology, Investigation, Resources, Writing – original draft, Writing – review & editing, Supervision, Funding acquisition.

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.

Data availability

Dataset and implementations are publicly available.

Acknowledgments

This project is partially funded by the Swiss Data Science Center under grant C18-04 (“Towards an automated post-earthquake damage assessment”).

References

[1] F. Vanin, D. Zaganelli, A. Penna, K. Beyer, Estimates for the stiffness, strength and drift capacity of stone masonry walls based on 123 quasi-static cyclic tests reported in the literature, Bull. Earthq. Eng. 15 (12) (2017) 5435–5479, http://dx.doi.org/10.1007/s10518-017-0188-5, URL https://link.springer.com/article/10.1007%2Fs10518-017-0188-5.

[2] A. Athanasiou, A. Ebrahimkhanlou, J. Zaborac, T. Hrynyk, S. Salamone, A machine learning approach based on multifractal features for crack assessment of reinforced concrete shells, Comput.-Aided Civ. Infrastruct. Eng. 35 (6) (2020) 565–578, http://dx.doi.org/10.1111/mice.12509, URL https://onlinelibrary.wiley.com/doi/epdf/10.1111/mice.12509.

[3] K.M. Dolatshahi, K. Beyer, Stiffness and strength estimation of damaged unreinforced masonry walls using crack pattern, J. Earthq. Eng. 00 (00) (2019) 1–20, http://dx.doi.org/10.1080/13632469.2019.1693446.

[4] A. Rezaie, A.J.P. Mauron, K. Beyer, Sensitivity analysis of fractal dimensions of crack maps on concrete and masonry walls, Autom. Constr. 117 (May) (2020) 103258, http://dx.doi.org/10.1016/j.autcon.2020.103258.

[5] Y. Yao, E.T. Shue-Ting, G. Branko, Crack detection and characterization techniques - an overview, Struct. Control Health Monit. (May 2011) (2011) n/a–n/a, http://dx.doi.org/10.1002/stc.456, URL http://dx.doi.org/10.1002/stc.456.

[6] B.K. Oh, B. Glisic, H.S. Park, Convolutional neural network-based damage detection method for building structures, Smart Struct. Syst. 27 (6) (2021) 903–916, http://dx.doi.org/10.12989/sss.2021.27.6.903, URL https://yonsei.pure.elsevier.com/en/publications/convolutional-neural-network-based-damage-detection-method-for-bu.

[7] L. Zhang, F. Yang, Y. Daniel Zhang, Y.J. Zhu, Road crack detection using deep convolutional neural network, Proceedings - International Conference on Image Processing, ICIP, Vol. 2016-Augus (2016) 3708–3712, http://dx.doi.org/10.1109/ICIP.2016.7533052, URL https://ieeexplore.ieee.org/document/7533052.

[8] A. Zhang, K.C.P. Wang, B. Li, E. Yang, X. Dai, Y. Peng, Y. Fei, Y. Liu, J.Q. Li, C. Chen, Automated pixel-level pavement crack detection on 3D asphalt surfaces using a deep-learning network, Comput.-Aided Civ. Infrastruct. Eng. 32 (10) (2017) 805–819, http://dx.doi.org/10.1111/mice.12297, URL https://onlinelibrary.wiley.com/doi/abs/10.1111/mice.12297.

[9] L. Ali, W. Khan, K. Chaiyasarn, Damage detection and localization in masonry structure using faster region convolutional networks, Int. J. GEOMATE 17 (59) (2019) 98–105, http://dx.doi.org/10.21660/2019.59.8272, URL https://www.geomatejournal.com/sites/default/files/articles/98-105-8272-Luqman-July-2019-59g.pdf.

[10] Y. Bai, A. Yilmaz, H. Sezen, End-to-end deep learning methods for automated damage detection in extreme events at various, 2020, arXiv. URL https://arxiv.org/abs/2011.03098.

[11] Y.J. Cha, W. Choi, O. Büyüköztürk, Deep learning-based crack damage detection using convolutional neural networks, Comput.-Aided Civ. Infrastruct. Eng. 32 (5) (2017) 361–378, http://dx.doi.org/10.1111/mice.12263.

[12] K. Chaiyasarn, W. Khan, L. Ali, M. Sharma, D. Brackenbury, M. DeJong, Crack detection in masonry structures using convolutional neural networks and support vector machines, in: ISARC 2018 - 35th International Symposium on Automation and Robotics in Construction and International AEC/FM Hackathon: The Future of Building Things, 2018, http://dx.doi.org/10.22260/isarc2018/0016.

[13] V. Hoskere, Y. Narazaki, T.A. Hoang, B.F. Spencer, Towards automated post-earthquake inspections with deep learning-based condition-aware models, 2018, http://dx.doi.org/10.48550/arXiv.1809.09195, arXiv arXiv:1809.09195. URL https://arxiv.org/abs/1809.09195.

[14] V. Hoskere, Y. Narazaki, T.A. Hoang, B.F. Spencer, Vision-based structural inspection using multiscale deep convolutional neural networks, 2017, http://dx.doi.org/10.48550/arXiv.1805.01055, arXiv arXiv:1805.01055.

[15] B. Kim, S. Cho, Automated vision-based detection of cracks on concrete surfaces using a deep learning technique, Sensors 18 (10) (2018) http://dx.doi.org/10.3390/s18103452, URL https://www.mdpi.com/1424-8220/18/10/3452.

[16] Z. Liu, Y. Cao, Y. Wang, W. Wang, Computer vision-based concrete crack detection using U-net fully convolutional networks, Autom. Constr. 104 (January) (2019) 129–139, http://dx.doi.org/10.1016/j.autcon.2019.04.005.

[17] D. Dais, E. Bal, E. Smyrou, V. Sarhosis, Automatic crack classification and segmentation on masonry surfaces using convolutional neural networks and transfer learning, Autom. Constr. 125 (January) (2021) http://dx.doi.org/10.1016/j.autcon.2021.103606.

[18] C. Chen, S. Chandra, Y. Han, H. Seo, Deep learning-based thermal image analysis for pavement defect detection and classification considering complex pavement conditions, Remote Sens. 14 (1) (2022) http://dx.doi.org/10.3390/rs14010106, URL https://www.mdpi.com/2072-4292/14/1/106.

[19] Z. Dong, J. Wang, B. Cui, D. Wang, X. Wang, Patch-based weakly supervised semantic segmentation network for crack detection, Constr. Build. Mater. 258 (2020) 120291, http://dx.doi.org/10.1016/j.conbuildmat.2020.120291.

[20] F. Liu, L. Wang, Unet-based model for crack detection integrating visual explanations, Constr. Build. Mater. 322 (January) (2022) 126265, http://dx.doi.org/10.1016/j.conbuildmat.2021.126265.

[21] P. Miao, T. Srimahachota, Cost-effective system for detection and quantification of concrete surface cracks by combination of convolutional neural network and image processing techniques, Constr. Build. Mater. 293 (2021) 123549, http://dx.doi.org/10.1016/j.conbuildmat.2021.123549.

[22] B.G. Pantoja-Rosero, K.R. Maximiano dos Santos, R. Achanta, A. Rezaie, K. Beyer, Determining crack kinematics from imaged crack patterns, Constr. Build. Mater. 343 (128054) (2022) http://dx.doi.org/10.1016/j.conbuildmat.2022.128054, URL https://www.sciencedirect.com/science/article/pii/S0950061822017202?via%3Dihub.

[23] D. Oner, M. Kozinski, L. Citraro, N.C. Dadap, A.G. Konings, P. Fua, Promoting connectivity of network-like structures by enforcing region separation, 2020, pp. 1–11, http://dx.doi.org/10.48550/arXiv.2009.07011, arXiv arXiv:2009.07011. URL https://arxiv.org/abs/2009.07011.

[24] V. Iglovikov, A. Shvets, TernausNet: U-Net with VGG11 encoder pre-trained on ImageNet for image segmentation, 2018, http://dx.doi.org/10.48550/arXiv.1801.05746, arXiv:1801.05746. URL http://arxiv.org/abs/1801.05746.

[25] A. Rezaie, R. Achanta, M. Godio, K. Beyer, Comparison of crack segmentation using digital image correlation measurements and deep learning, Constr. Build. Mater. 261 (2020) 120474, http://dx.doi.org/10.1016/j.conbuildmat.2020.120474.

[26] B.F. Spencer, V. Hoskere, Y. Narazaki, Advances in computer vision-based civil infrastructure inspection and monitoring, Engineering 5 (2) (2019) 199–222, http://dx.doi.org/10.1016/j.eng.2018.11.030.

[27] S. Dorafshan, R.J. Thomas, M. Maguire, Comparison of deep convolutional neural networks and edge detectors for image-based crack detection in concrete, Constr. Build. Mater. 186 (2018) 1031–1045, http://dx.doi.org/10.1016/j.conbuildmat.2018.08.011.

[28] T. Nishikawa, J. Yoshida, T. Sugiyama, Y. Fujino, Concrete crack detection by multiple sequential image filtering, Comput.-Aided Civ. Infrastruct. Eng. 27 (1) (2012) 29–47, http://dx.doi.org/10.1111/j.1467-8667.2011.00716.x, URL https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-8667.2011.00716.x.

[29] T. Yamaguchi, S. Hashimoto, Fast crack detection method for large-size concrete surface images using percolation-based image processing, Mach. Vis. Appl. 21 (5) (2010) 797–809, http://dx.doi.org/10.1007/s00138-009-0189-8, URL https://link.springer.com/article/10.1007%2Fs00138-009-0189-8.

[30] W. Zhang, Z. Zhang, D. Qi, Y. Liu, Automatic crack detection and classification method for subway tunnel safety monitoring, Sensors (Switzerland) 14 (10) (2014) 19307–19328, http://dx.doi.org/10.3390/s141019307, URL https://www.mdpi.com/1424-8220/14/10/19307.

[31] A.P. Tewkesbury, A.J. Comber, N.J. Tate, A. Lamb, P.F. Fisher, A critical synthesis of remotely sensed optical image change detection techniques, Remote Sens. Environ. 160 (2015) 1–14, http://dx.doi.org/10.1016/j.rse.2015.01.006, URL https://www.sciencedirect.com/science/article/pii/S0034425715000152?via%3Dihub.

[32] M. Hussain, D. Chen, A. Cheng, H. Wei, D. Stanley, Change detection from remotely sensed images: From pixel-based to object-based approaches, ISPRS J. Photogramm. Remote Sens. 80 (2013) 91–106, http://dx.doi.org/10.1016/j.isprsjprs.2013.03.006.

[33] R.J. Radke, S. Andra, O. Al-Kofahi, B. Roysam, Image change detection algorithms: A systematic survey, IEEE Trans. Image Process. 14 (3) (2005) 294–307, http://dx.doi.org/10.1109/TIP.2004.838698, URL https://ieeexplore.ieee.org/document/1395984.

[34] F.C. Chen, M.R. Jahanshahi, NB-CNN: Deep learning-based crack detection using convolutional neural network and Naïve Bayes data fusion, IEEE Trans. Ind. Electron. 65 (5) (2018) 4392–4400, http://dx.doi.org/10.1109/TIE.2017.2764844.

[35] O. Ronneberger, P. Fischer, T. Brox, U-net: convolutional networks for biomedical image segmentation, in: N. Navab, J. Hornegger, W.M. Wells, A.F. Frangi (Eds.), Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015, Springer International Publishing, Cham, 2015, pp. 234–241, URL https://link.springer.com/chapter/10.1007/978-3-319-24574-4_28.

[36] T. Ghosh Mondal, M.R. Jahanshahi, R.T. Wu, Z.Y. Wu, Deep learning-based multi-class damage detection for autonomous post-disaster reconnaissance, Struct. Control Health Monit. 27 (4) (2020) 1–15, http://dx.doi.org/10.1002/stc.2507, URL https://onlinelibrary.wiley.com/doi/10.1002/stc.2507.

[37] M. Mishra, Machine learning techniques for structural health monitoring of heritage buildings: A state-of-the-art review and case studies, J. Cult. Herit. 47 (2021) 227–245, http://dx.doi.org/10.1016/j.culher.2020.09.005.

[38] H. Li, D. Song, Y. Liu, B. Li, Automatic pavement crack detection by multi-scale image fusion, IEEE Trans. Intell. Transp. Syst. 20 (6) (2019) 2025–2036, http://dx.doi.org/10.1109/TITS.2018.2856928, URL https://ieeexplore.ieee.org/document/8428669.

[39] X. Zhang, D. Rajan, B. Story, Concrete crack detection using context-aware deep semantic segmentation network, Comput.-Aided Civ. Infrastruct. Eng. 34 (11) (2019) 951–971, http://dx.doi.org/10.1111/mice.12477.

[40] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, 2014, http://dx.doi.org/10.48550/arXiv.1409.1556, arXiv e-prints, arXiv:1409.1556. URL https://arxiv.org/abs/1409.1556.

[41] F. Milletari, N. Navab, S.A. Ahmadi, V-Net: fully convolutional neural networks for volumetric medical image segmentation, in: Fourth International Conference on 3D Vision, IEEE, 2016, pp. 565–571, http://dx.doi.org/10.1109/3DV.2016.79, arXiv:1606.04797. URL https://ieeexplore.ieee.org/abstract/document/7785132.

[42] S.C. Turaga, K.L. Briggman, M. Helmstaedter, W. Denk, H.S. Seung, Maximin affinity learning of image segmentation, in: Advances in Neural Information Processing Systems 22 - Proceedings of the 2009 Conference, 2009, pp. 1865–1873, arXiv:0911.5372. URL https://arxiv.org/abs/0911.5372.

[43] D.P. Kingma, J.L. Ba, Adam: a method for stochastic optimization, in: International Conference on Learning Representations, 2015, pp. 1–15, http://dx.doi.org/10.48550/arXiv.1412.6980, arXiv:1412.6980. URL https://arxiv.org/abs/1412.6980.

[44] A. Rezaie, M. Godio, K. Beyer, Experimental investigation of strength, stiffness and drift capacity of rubble stone masonry walls, Constr. Build. Mater. 251 (2020) 118972, http://dx.doi.org/10.1016/j.conbuildmat.2020.118972.