Construction and Building Materials 261 (2020) 120474

Contents lists available at ScienceDirect

Construction and Building Materials

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

Comparison of crack segmentation using digital image correlation measurements and deep learning

Amir Rezaie [a], Radhakrishna Achanta [b], Michele Godio [c], Katrin Beyer [a,*]

[a] Laboratory of Earthquake Engineering and Structural Dynamics (EESD), School of Architecture, Civil and Environmental Engineering (ENAC), École Polytechnique Fédérale de Lausanne (EPFL), 1015 Lausanne, Switzerland
[b] Swiss Data Science Center, EPFL and ETH Zurich, Switzerland
[c] RISE Research Institutes of Sweden, Brinellgatan 4, 504 62 Borås, Sweden

HIGHLIGHTS

- Segmentation of cracks on speckled images taken for DIC.
- Deep learning results to a higher precision and dice score than threshold method.
- Geometry of cracks are better preserved by the deep learning method.

ARTICLE INFO

Article history:
Received 2 June 2020
Received in revised form 15 July 2020
Accepted 2 August 2020
Available online 3 September 2020

Keywords:
Crack segmentation
Digital image correlation
Deep learning
Threshold method
Masonry

ABSTRACT

Reliable methods for detecting pixels that represent cracks from laboratory images taken for digital image correlation (DIC) are required for two main reasons. Firstly, the segmented crack maps are used as an input for some DIC methods that are based on discontinuous fields. Secondly, detected crack patterns can serve as inputs for predictive empirical models to obtain the level of damage to a body. The aim of this paper is to compare the performance of two approaches for crack segmentation on grayscale images acquired from two experimental campaigns on stone masonry walls. In the first approach, a threshold is applied to the maximum principal strain map calculated using post-processed DIC results. In the second approach, a deep convolutional neural network is used. The two methods are compared in terms of standard segmentation criteria, namely precision, dice coefficient and sensitivity. It is shown that the precision and dice coefficient obtained from the deep learning approach are much higher than those obtained from the threshold method (by almost 47% and 34%, respectively). However, the sensitivity computed from the deep learning method is slightly (~4%) lower than the threshold method. These results show that the deep learning method can better preserve the geometry of detected crack patterns, and the prediction in terms of pixels belonging to a crack is finally more accurate than the threshold method.

1. Introduction

Digital image correlation (DIC) is an optical measurement technique widely used in experimental mechanics to compute the deformation of a body [1,2]. For the local subset DIC, images of a sample are compared at a regular time interval. In this process, a small group of pixels called subset is tracked [2] by comparing the reference image that is taken from the undeformed body with later images of the deformed body taken at intervals of time. To perform this procedure, called matching, a correlation error function that describes the difference between the subset in the reference image and the image taken at a later stage is minimized (refer to [2,3] for suggested error functions). However, the correlation process might fail if a crack passes through a subset; in this case, the user must exclude such areas in advance before the matching phase [4]. Additionally, multiple cracks crossing the sample can divide its surface into segregated areas, for which initial seed points—the initial pixels considered for the matching—must be defined separately. This can make it a cumbersome process to analyze the hundreds of sets of images usually taken during an experiment [4]. To tackle this problem, Helm [4] proposed using a cutoff on the correlation values obtained for each subset and discarding the subsets with error values higher than this cutoff. However, the threshold value must be defined by the user.

* Corresponding author.
E-mail address: katrin.beyer@epfl.ch (K. Beyer).

https://doi.org/10.1016/j.conbuildmat.2020.120474
0950-0618/

Detecting crack pixels on images could help automate the generation of the initial seed point in the local subset DIC method for segregated regions and to remove noisy results around the discontinuities [4]. Automated crack detection would also be useful for assessing the damage state of structural elements, because a pixel-wise segmentation of cracks, i.e., classifying all pixels of the image whether they belong to a crack or not, can be an important input for the damage assessment of a structural element [5–11]. This is particularly relevant for elements made of quasi-brittle materials, like concrete and masonry, for which cracks are the key damage feature.

In the present study, two approaches for crack segmentation are compared using images that were taken during two experimental campaigns conducted at École Polythecnique Fédérale de Lausanne (EPFL) [12,13]. The two approaches are: (i) crack segmentation by applying a threshold to the maximum principal strain map and (ii) crack segmentation using deep learning. To evaluate and compare the performance of each method, standard segmentation criteria in the computer vision domain, namely precision (PC), sensitivity (SE), and dice coefficient (DC) are used. The sensitivity of the threshold method to the subset size, which is an input parameter for performing the local DIC, is also explored. Furthermore, the effect of using a pre-trained encoder in the deep learning network is investigated. In what follows, previous studies on crack detection are reviewed in Section 2. A brief description of the conducted experimental campaigns is included in Section 3. Thereafter, the image dataset and the segmentation metrics are mentioned in Section 4. The segmentation methods are detailed in Section 5 and 6, and finally, the performance of the methods is compared in Section 7.

2. Related research studies

2.1. Crack segmentation by post-processing of DIC results

In some research, crack detection is integrated directly into the DIC method [4]. In the structural engineering community, however, it is addressed in post-processing results derived from DIC measurements, such as strain maps or displacement fields. Destrebecq et al. [14] tested a reinforced concrete beam under four-point bending. They verified the crack patterns obtained from the DIC method by comparing it to a visual inspection performed during the experiment. The criteria used for this verification was the location and width of cracks. Tung et al. [15] observed that the cracking of a 45 degrees brick wall under compression loading could be better depicted using the von Mises strain map than horizontal and vertical strain maps. However, this work did not include a quantitative justification for the accuracy of crack detection by applying a threshold on the von Mises strain maps. Ghorbani et al. [16] stated that visualizing the maximum principal strain (ε_1) can detect cracking of confined masonry walls tested under cyclic shear-compression loading. They compared a colored map of the maximum principal strain with the hand-drawn crack patterns at the ultimate state depicted during the experiments. They observed that masonry cracking can be captured by visualizing ε_1 maps, though again, a quantitative comparison was not mentioned. Similarly, Korswagen et al. [17,18] studied crack propagation in masonry walls using post-processed DIC results. Hoult et al. [19] used the horizontal strain map, defined as the horizontal displacement of adjacent subsets divided by the step size (i.e., the initial distance between subsets), to detect cracking of six reinforced concrete beams tested under four-point bending. Morgan [20] studied the initiation, propagation and coalescence of cracks using the DIC method by conducting several compression tests on Opalinus shale specimens. In this study, they applied a threshold to the horizontal strain ε_{xx} to obtain crack maps. By qualitatively comparing crack maps sketched during experiments and those obtained using the results of the DIC method, they concluded that the DIC analysis is capable of detecting more cracks than visual inspection. Cinar et al. [21] proposed a sequence of image processing algorithms to identify crack patterns, including phase congruency and active contour segmentation on the displacement field obtained from DIC measurements. They tested this method using both synthetic and laboratory data, but the efficiency was not verified for specimens with multiple cracks. Recently, Gehri et al. [22] proposed a pipeline to compute automatically the crack opening and slipping using the DIC measurements. They used the principal tensile strain to localize cracked area. They qualitatively evaluated the performance of crack detection by overlying the principal tensile strain map over the region of interest.

2.2. Crack segmentation using deep learning

Deep learning methods have been successfully applied to detect and segment cracks on natural images, such as asphalt, concrete, masonry and steel surfaces [23–32]. Generally, two approaches, namely classification and segmentation, have been used in the literature for crack detection. In the first method, small patches of an image are classified as crack or non-crack. In the second approach, every pixel is classified as a crack or non-crack.

Using the first methodology, Zhang et al. [33] proposed a shallow network to classify patches of road images as crack or non-crack. They observed that extracting hierarchical features using a convolutional network resulted in higher precision, recall, and F1-score (refer to Section 4.2 for the definition) compared to using hand-crafted features and designing an SVM (Support Vector Machine) or Boosting classifier. Cha et al. [34] classified patches by training a convolutional neural network using 40 k images of concrete surfaces (train/validation) with a resolution of 256×256 pixels. They achieved a classification accuracy of around 98%. Wand and Hu [35] trained a CNN (Convolutional Neural Network) to classify crack vs. non-crack patches using two patch sizes of either 32×32 pixels or 64×64 pixels. With the larger patch size, they observed an increase in the precision, recall and F1-score. To classify image blocks as background, crack or sealed crack on road images, Kaige et al. [36] trained a CNN utilizing the concept of transfer learning. In a second step, they trained a linear model to find an optimum threshold for each image block to segment the crack pixels. Finally, they used tensor voting [37] to connect broken crack fragments. Gopalakrishnan et al. [27] extracted features from pavement images using a pre-trained VGG16 (VGG: Visual Geometry Group at the University of Oxford) truncated deep network. They used five classifiers to classify image patches as crack or non-crack, including a single neural network layer, random forest, extremely random trees, support vector machines and logistic regression. It was observed that the single-layer neural network had the highest prediction accuracy, precision, recall, F1-score and Cohen's Kappa score. To detect cracks on metallic surfaces in video frames, Chen and Jahanshahi [38] proposed a CNN with naive Bayes data fusion.

Using the second methodology, i.e., pixel-level crack detection, Yang et al. [23] proposed a fully convolutional network to segment crack pixels on different surfaces. The proposed network consisted of a down-sampling part, wherein they used VGG19, and an up-sampling part. They achieved around 98% accuracy, 82% precision, 79% recall and an 80% F1-score. Zou et al. [39] proposed a hierarchical encoder-decoder architecture called "DeepCrack", in which a prediction map is created at multiple convolution stages by fusing feature maps from the encoder and decoder path. All fused maps are concatenated and fused to produce a multi-scale output. They stated that DeepCrack outperforms other crack segmentation

methods such as CrackForest [40] and CrackTree [41]. Liu et al. [42] proposed a variant of U-Net to predict crack pixels on concrete surfaces. They reported that U-Net requires only 57 images for training and validation, and it achieved an F1-score of 90%. Ji et al. [62] used DeepLabv3+ [43] to segment crack pixels on asphalt images followed by an algorithm to quantify crack maps in terms of length, mean width, maximum width, area and ratio.

The literature review presented above shows that the previous studies using the threshold method lack quantitative image-based criteria to verify the quality of crack detection. In this study, however, the threshold method is evaluated by quantitative metrics. With regard to the deep learning approach, all the conducted research has focused only on "natural images", i.e., images taken from structural elements without the presence of small black dots termed speckles. However, these approaches have never been tested for images taken in the laboratory for use in DIC, in short referred to as "laboratory images" in this article. One significant difference between natural images and laboratory images is the presence of speckles on the background (see Fig. 1 and Fig. 2), which complicates the detection process compared to natural images with a simpler background.

3. Description of the case studies

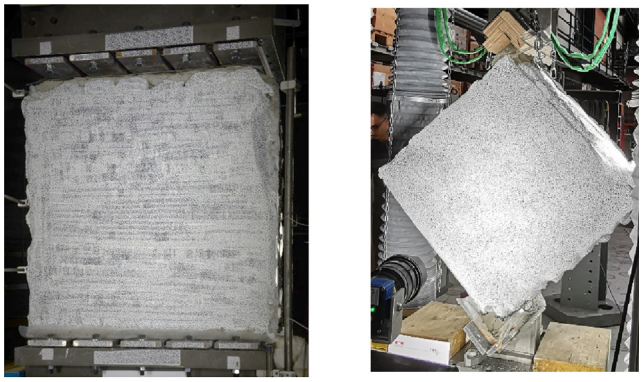
3.1. Experimental tests and procedures

As a case study, the data from two experimental campaigns conducted at EPFL, wherein DIC was applied to measure the displacement fields, were used in this work. These two experimental campaigns are briefly described below.

3.1.1. Tests on stone masonry walls of typology A [12]

This testing campaign included six shear-compression tests (specimen label: RS), three compression tests (specimen label: RSC) and three diagonal-compression tests (specimen label: RSD).

The specimens for shear-compression tests (RS) were 160 cm high, 160 cm long and 40 cm wide and were plastered on one side. Fig. 1(a) and (b) show views of the faces with and without plaster as well as views of the two stereo camera systems placed on either side of the wall to measure the 3D displacement fields via DIC. During the tests, the horizontal red actuator applied cyclic horizontal displacements while the vertical red actuators applied a constant vertical load. For more details about the experimental campaign, please refer to Rezaie et al. [12]. In addition to shear-compression tests, three diagonal-compression and three simple-compression tests were performed on wallettes that were 90 cm high, 90 cm long (one of the wallettes tested under simple compression load was 80 cm long) and 40 cm wide. Fig. 2(a) and (b) show images taken from simple-compression and diagonal-compression tests, respectively. As for the shear-compression tests, displacement fields on either side of the wallettes were measured by two sets of stereo cameras used in conjunction with DIC.



(a) (b)

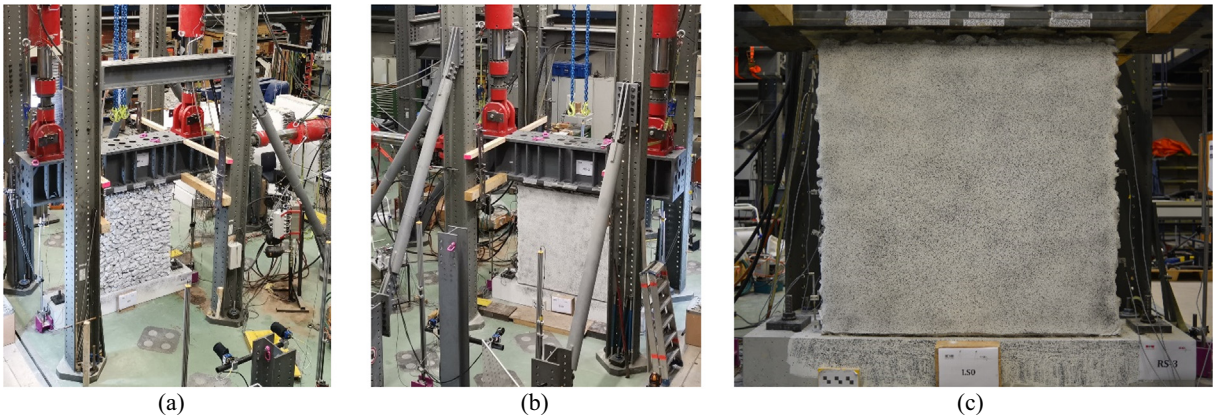
Fig. 2. Tests on wallettes: (a) simple-compression test and (b) diagonal compression test. The images show the plastered side of the wallettes.

3.1.2. Tests on stone masonry walls of typology E [13]

Godio et al. [13] conducted six cyclic shear-compression tests on stone masonry walls (specimen labels: SC) with regular typology. The walls were 90 cm high, 90 cm long and 20 cm wide. In addition to shear-compression tests, two simple-compression tests were carried out on specimens labelled RSM, measuring 90 cm high, 78 cm long and 20 cm wide and one diagonal-compression test on a wall measuring 78 cm high, 78 cm long and 20 cm wide. For a detailed explanation about the tests, please refer to Godio et al. [13]. Just as for the tests performed on walls of typology A by Rezaie et al. [12], the wall surfaces were speckled and two stereo camera systems were used to measure 3D displacement fields on either side of the walls.

3.2. Three-dimensional DIC setup

To use the DIC method, a number of preparatory steps were performed in both campaigns. On the specimens, the regions of interest (ROI) were first painted in white; then the wall surfaces were covered with random speckle patterns with an average dimension



(a) (b) (c)

Fig. 1. Test setup for shear-compression tests at EPFL. (a) view of the stone side without plaster, (b) view of the plastered side and (c) a closed view of the plastered side covered with random speckles.

of 2 mm. These black speckles were either sprayed using a paint pistol (see Fig. 1(c)) or a printer gun (see Fig. 2(a)). Additionally, stereo camera systems and lighting conditions were set up. Cameras were mounted on a rigid bar and placed far enough from the specimens to view the entire region of interest. They were set up such that the centres of the two cameras pointed towards the same location on the specimens. The average ratio of mm/pixel for all tested specimens ([12,13]) was almost 0.4. On each side of the walls, two LED lamps were placed to illuminate the regions of interest. The cameras were calibrated by taking around 80 pairs of images of a calibration board containing circular dots with known dimensions. For a more detailed explanation about the necessary preparation steps to use the DIC method, see previous work on the topic [2,44]. During the experiments, pairs of grayscale images were taken from both sides of the walls and wallettes at specified time intervals. For testing walls, images with dimensions of 4384 $\times$ 6576 pixels were acquired, while, for testing wallettes, images with dimensions of 2192 $\times$ 3288 pixels were acquired.

To perform the DIC method, the commercial software VIC-3D version 8.2.4 [45] was used, which uses a local subset DIC approach that divides the region of interest into subsets. The local subset DIC method is formulated as an optimization problem using a cross-correlation loss function [2,45]. As an illustration, Fig. 3 depicts a subset in a reference image (green box) and the corresponding deformed subset (red quadrilateral) in the deformed image. The result of the DIC method is the 3D deformation of the centres of each subset (the green circle shown in Fig. 3(a)), for which the correlation problem is solved. The correlation problem is usually solved for overlapping subsets, referred to as the step size. An important parameter affecting the accuracy and spatial resolution is the subset size [2,16,46,47]. One way to select a subset size is to take several images at the beginning of a test when a specimen is intact and investigate the standard deviation and bias of some measurements, such as horizontal or vertical displacements [16]. In this study, the subset size is considered as a variable parameter so that its influence on the segmentation metrics described in Section 4.2 could be investigated.

4. Crack segmentation dataset and metrics

In this paper, crack segmentation is performed on a pixel basis, i.e., by classifying each pixel as part of a crack or not. Two approaches of crack segmentation are applied and compared in this study: i) an approach based on a threshold applied on the principal strain map that is derived from the displacement field measured using DIC, and ii) an approach by deep learning making use of the images taken as input to DIC. Section 4.1 introduces the dataset of images used in the two approaches, while the criteria used in the evaluation and comparison of the approaches are defined in Section 4.2.

4.1. Training and test datasets

Small image patches were extracted from the full-size images by sliding a 256 $\times$ 256 pixels window with a stride of 256 pixels along the two directions (Fig. 4), which will be used as inputs to the network.

For the training and validation data (dataset A), 17 full-size laboratory images were selected from the specimens RSC2-3, RSD1, RSD2, RSM2, SC1-7, RS1-3. The images taken during the experimental campaigns illustrate the state of the plastered wall surfaces under different loading levels and support conditions [12,13]. The images that were selected as part of the training and validation dataset cover a wide combination of crack widths, speckle patterns and lighting conditions. In this way, the robustness of the crack segmentation approaches tested in the paper is challenged. From the 17 full-size images, 430 image patches were generated. The crack pixels contained in the patches were manually annotated by use of the open-source software "Pixel Annotation Tool" [48]. These manual annotations serve as ground truth.

For the test data (dataset B), three full-size images were selected from specimens RS4 and RS6 at different loading levels (see Fig. 5). To obtain a fair model, no images taken from specimens RS4 and RS6 were included in the training/validation data (dataset A). Using the sliding window with the same window size and stride as for the training/validation data, 100 image patches were generated, and the ground truth masks (binary images in which crack pixels are annotated, i.e., crack pixels are set to one and the background to zero) were annotated. These test image patches were used to compare the performance of the two segmentation approaches. All images and corresponding ground truth masks are made publicly available at https://www.epfl.ch/labs/eesd/data-sets/data_sets/.

Note that the manual labelling of the cracks was limited to what was visible, and the minimum crack width visible on the images from dataset B was about 0.1 mm. Moreover, perfect manual labeling was sometimes difficult, because speckles present on images and crack pixels are both black; therefore, for situations where the boundary of a crack is close to speckle pixels, the ground truth might not be without error. To overcome this issue, relaxed segmentation metrics [49] can be used; for instance, a buffer size of 3 pixels can be used to determine true crack pixels. In this study, however, this buffer was not conisdered. Furthermore, because the DIC results can be noisy around the border of the ROI, image patches corresponding to these areas were discarded even if they had cracks. Further explanation is provided in section 5. Fig. 6

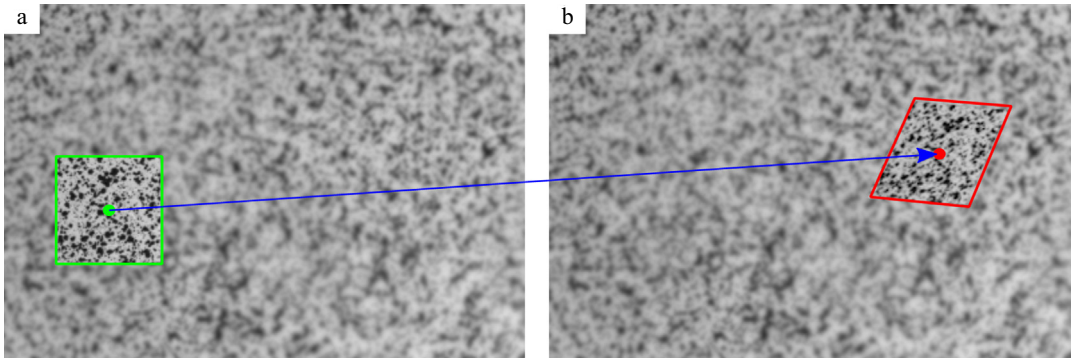


Fig. 3. Schematic illustration of DIC (a) reference and (b) deformed image. The selected green subset is deformed into the red quadrilateral.

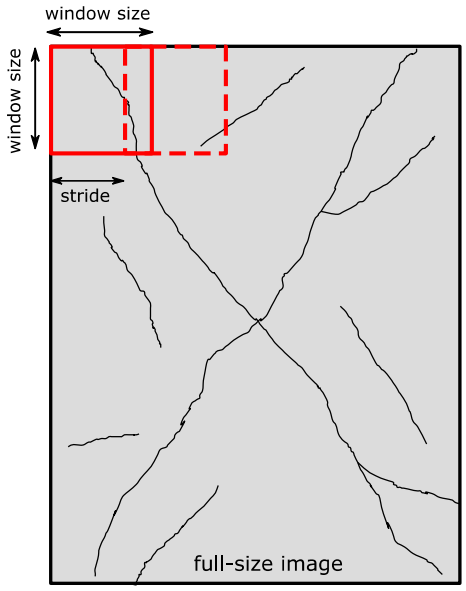


Fig. 4. Illustration of the sliding window used to obtain equally sized image patches from the full-size laboratory image.

depicts examples of image patches and the corresponding ground truth masks.

4.2. Segmentation metrics

Even though there is a broad range of options, one widely used group of semantic segmentation metrics is pixel-wise metrics [49], which can be divided into two categories: (i) evaluation of the segmentation by comparison of the segmentation results to the ground truth on a pixel-by-pixel basis, i.e., without skeletonization of the cracks, and (ii) evaluation of the segmentation by skeletonizing the crack pattern. In this study, the former approach was chosen. For this approach, three quantities were computed: the precision (PC), the sensitivity (SE), also known as recall, and the dice coefficient (DC), also known as F1-score, which is the harmonic mean of PC and SE. These are defined as:

$$\mathrm{PC} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{1}$$

$$\mathrm{SE} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{2}$$

$$\mathrm{DC} = \frac{2\,(\mathrm{SE} \times \mathrm{PC})}{\mathrm{SE} + \mathrm{PC}} \tag{3}$$

where TP, FP and FN are the number of true positive, false positive and false negative pixels, respectively. The dice coefficient measures the similarity between the predicted crack mask and the ground truth. If the predicted mask and the ground truth are identical, DC is equal to 1, while if the predicted crack mask and the ground truth do not intersect, DC is equal to 0. Otherwise, DC is between 0 and 1.

5. Crack segmentation by threshold method

5.1. Method implementation

The DIC method gives as output the 3D displacement field at the points for which the correlation problem is solved. This field was extracted from the software VIC-3D, and a python code was developed for post-processing the results. The post-processing includes generating a mesh of triangular elements (constant strain triangles) and calculating the maximum principal strain ε_1 at the center of each element. These strain values were then interpolated to obtain the ε_1-value for each pixel, and a threshold was applied to transform the obtained strain maps to binary crack masks.

To explore the effect of subset size on the crack segmentation results, four DIC analyses were performed by choosing different subset sizes, i.e., 15 × 15 pixels, 19 × 19 pixels, 23 × 23 pixels and 33 × 33 pixels. For all analyses, the step size was set to 2 × 2 pixels, and the ratio mm/pixel was 0.43 (for specimens RS4 and RS6). There is a lower bound on the ideal subset size, because when the subset size is too small, the subsets might contain too few speckles, which finally decreases the accuracy of the DIC method [2]. It is suggested that each subset should contain at least 3 × 3 speckle pixels [2]. Therefore, in this study, the subset size of 15 × 15 pixels (6.45 × 6.45 mm^2) was selected as the minimum subset size. Fig. 7 shows an example of the obtained ε_1-map for a subset size of 19 × 19 pixels for a selected loading level of specimens RS4 and RS6 (test dataset B).

To extract crack pixels, 25 threshold values were applied to the ε_1 maps: 0.5%, 1.5%, 2.5%, 3.5%,. . . , 24.5%. For each of these thresholds, the pixels with an ε_1 value larger than the threshold were labeled as crack, and not if otherwise.

(a) (b) (c)

Fig. 5. Full-size images used to create test data (dataset B): (a and b) two images of specimen RS4 taken at two load steps, and (c) an image of specimen RS6.

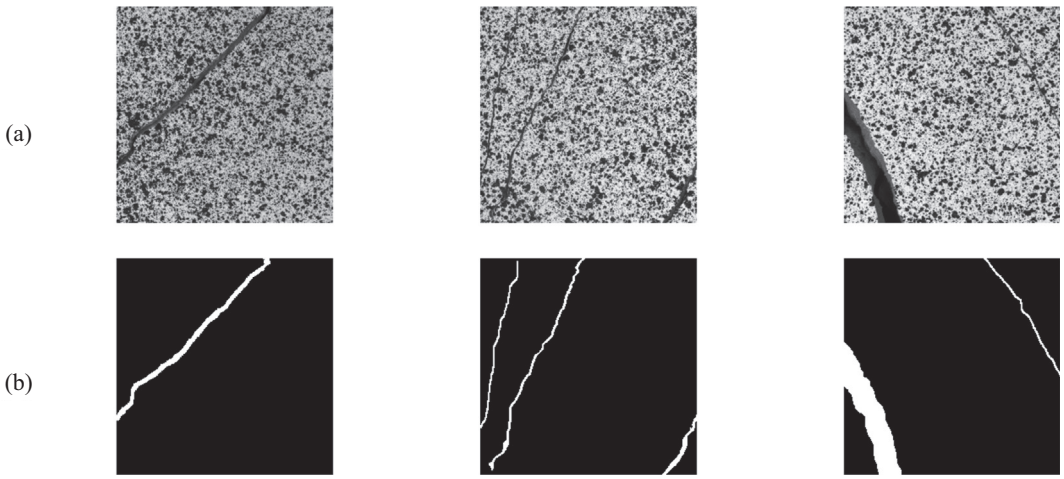


Fig. 6. Examples of the crack image database: (a) images and (b) annotated ground truth masks.

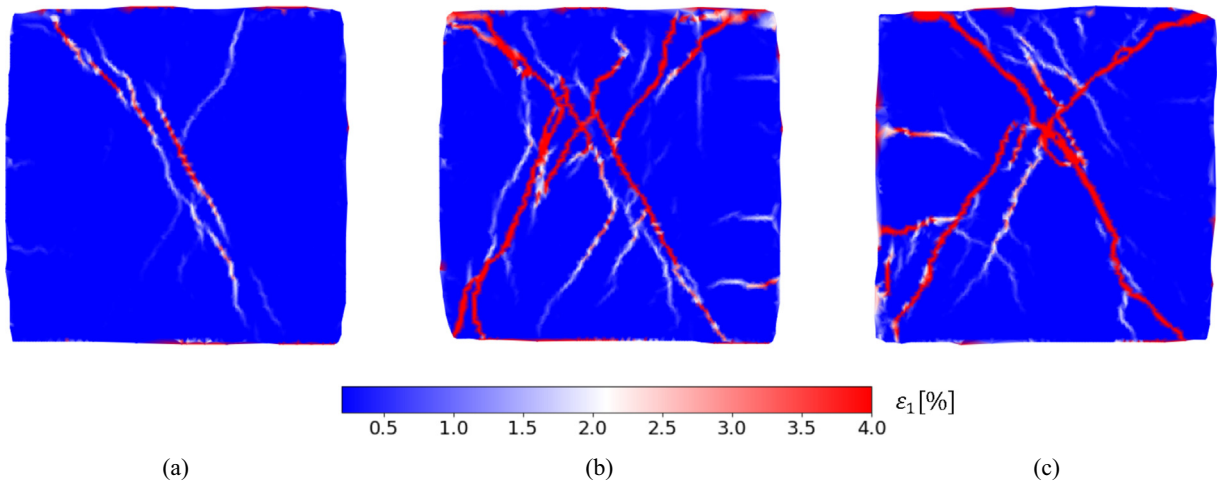


Fig. 7. Maximum principal strain maps for specimen RS4 (a and b) and specimen RS6 (c) shown in Fig. 5; subset size = 19 × 19 pixels.

5.2. Results

Fig. 8 shows the segmented cracks that were obtained by applying the smallest and largest threshold values to the strain map of Fig. 7(c). In this figure, the boundary of the wall is plotted with a dotted green line. It is evident from the figure that the extracted crack pattern is very sensitive to the applied threshold value. The threshold value of 0.5% (Fig. 8(a)) resulted in a highly noisy crack segmentation (see the ground truth mask Fig. 8(c)), which could be reduced in a post-processing step using mathematical morphology [50]. On the other hand, the crack map obtained by applying the threshold value of 24.5% is far less noisy, but there are still many pixels around the boundary of the ROI that are misclassified as a crack (see the ground truth mask Fig. 8(c)). This is because

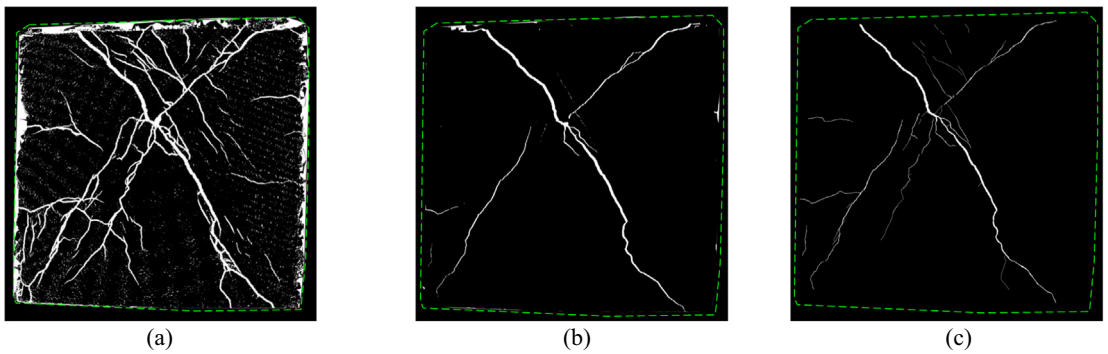


Fig. 8. Segmented crack patterns obtained by applying threshold values of (a) 0.5% and (b) 24.5% on the ε_1 map shown in Fig. 7(c); (c) manually annotated crack mask. The dashed box shows the boundary of the wall.

subset pixels are squares, and when the border of the ROI is curved, part of the background falls into the subset, which causes errors in the correlation step of the local subset DIC method [51]. Indeed, this problem can be avoided by choosing a smaller ROI; however, this comes at the cost of losing information, which is not desirable. Moreover, by applying such a high threshold, some crack pixels are wrongly misclassified as background, and the continuity of several thin cracks is not preserved. Furthermore, in addition to the above-mentioned qualitative reasons, the following analysis will show that even with the optimum threshold value and subset size, the results of the crack segmentation are poor.

The performance of the crack segmentation by application of a threshold to the ε_1 map was evaluated quantitatively by analysing 100 images of the test data (dataset B) and computing and plotting the segmentation metrics defined in Section 4.2 as a function of the threshold value. The results are presented in Fig. 9. It can be seen that for all subset sizes, as the threshold decreases, more pixels are classified as cracks, which corresponds to a high SE in Fig. 9a. However, the PC of segmentation is quite poor in Fig. 9b, which also reduces the DC because it is a harmonic mean of the SE and PC. By increasing the threshold value, though, the PC and DC are slightly improved while the SE is reduced.

The subset size affects the resolution, and therefore the accuracy of the crack segmentation. By increasing the subset size, the spatial resolution of the result decreases along with the SE, PC and DC values. This is most pronounced for the subset size of 33×33 pixels. In general, as the subset size decreases, crack pixels are more accurately classified.

The combination of subset size and threshold value that resulted in the maximum DC is the subset size of 15×15 pixels and the threshold value of 10.5%. Examples of the segmented crack patterns obtained using these values for nine crack images and the corresponding ground truth images are shown in Fig. 10(c). It is clear that the segmented cracks are quite coarse, i.e., wider than the width of the actual cracks (see for instance, ex. 1, 2, 5 of Fig. 10), which increases FP and decreases PC and DC. Furthermore, for very thin cracks of 1–2 pixels (such as ex. 3 in Fig. 10), the segmented cracks are not continuous.

6. Crack segmentation by deep learning

6.1. Network architecture

A UNet-like architecture [52] called "TernausNet" [53,54] was chosen for segmenting crack pixels. UNet-like architectures usually consist of two parts: encoder and decoder paths. In the encoder path, multiple blocks of operations, including stacks of convolutional layers, activation functions and pooling layers, are applied successively to the input of each layer. The in-plane size of the input to each layer reduces due to the convolutional and pooling operations deeper into the network. In the decoder path, the original size of the input is recovered by up-sampling the feature maps obtained in the encoder path. Moreover, to increase the localization accuracy of the prediction, the feature maps in the encoder path are copied into the decoder path via skip connections [52,54]. For this study, the TernausNet architecture [53,54] was chosen, which uses VGG16 [55] blocks as the encoder (see Fig. 11). This network was chosen because (i) previous studies have found that U-Net type architectures require few training examples for a satisfactory performance [52], which is required for our problem, and (ii) the encoder block is the VGG16, which can be initialized using the pre-trained weights on ImageNet. The detailed network architecture is summarized in Table 1, as implemented in PyTorch version 0.4.1. The table includes the number of filters at each layer, stride, padding, activation function and the up-sampling method.

6.2. Training setup and optimization scheme

As the goal was to obtain the highest performance in terms of both SE and PC, the dice loss function, which was first proposed by Milletari et. al. [56], was used to train models:

$$\text{DiceLoss} = 1 - \frac{\sum_{i=1}^{n} 2\hat{y}_i y_i + \epsilon}{\sum_{i=1}^{n} \hat{y}_i + \sum_{i=1}^{n} y_i + \epsilon} \tag{4}$$

where,y_i and $\hat{y}_i$ are the target (0 or 1) and predicted probability for pixel i, respectively, n is the number of pixels in the image, and $\epsilon = 1$.

Out of 430 image patches, 301 (70%) and 129 (30%) images were selected as training and validation data, respectively. The validation set guided the choice of learning rate and the best model (lowest validation loss). The initial learning rate was halved every 20 epochs. To increase the size of the database, the data was randomly augmented (with the probablity of 0.5). This was performed using the transforms module of Torchvision version 0.2.1 and included: a) random rotations with an angle equal to 90, 180 or 270 degrees; b) horizontal and vertical flips and c) brightness and contrast changes at a ratio of 0.2. The models were trained for 100 epochs using the Adam optimizer [57] with a batch size equal to 1 on a single NVIDIA Tesla P100 (12 GB) GPU. Apart from the learning rate, other parameters of the optimizer were set to the default values defined in the PyTorch optimization module. Weights of the encoder path were initialized using either model A or B. In model A, the weights of the encoder path were initialized randomly using the default method in PyTorch. For this case, a grid search led to the initial learning rate of 9e-5. In model B, the encoder path was

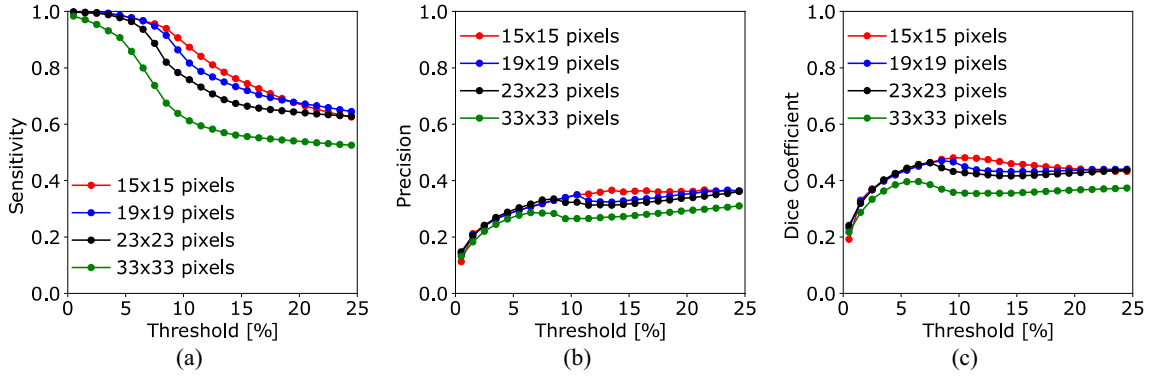


Fig. 9. Sensitivity (a), precision (b) and dice coefficient (c) of the threshold method as a function of the threshold value on the maximum principal strain and the subset size for the test data (dataset B).

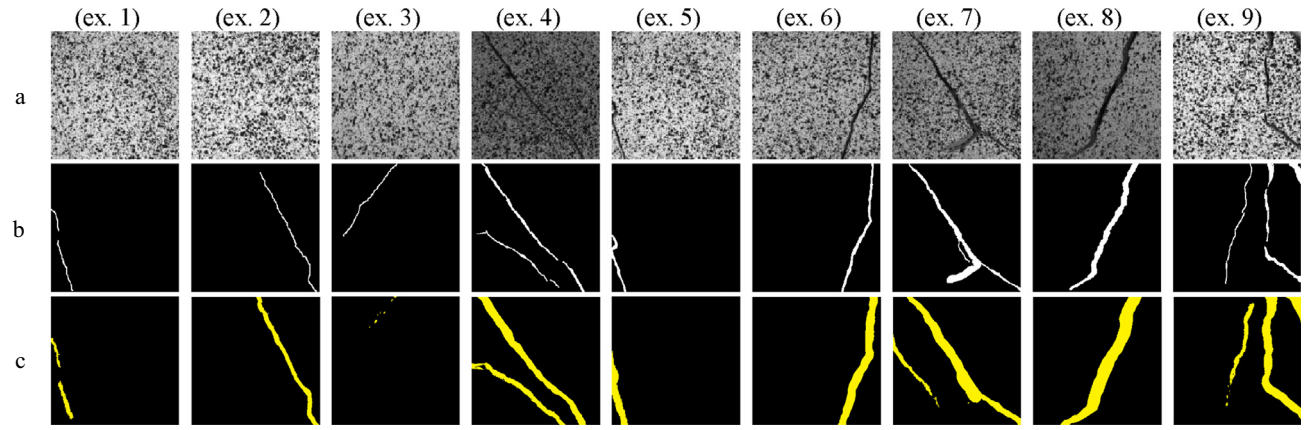


Fig. 10. Results of crack segmentation by the threshold method. (a) images, (b) ground truth masks, and (c) segmented crack mask made by applying the threshold value of 10.5% to the maximum principal strain map (post-processing of DIC results).

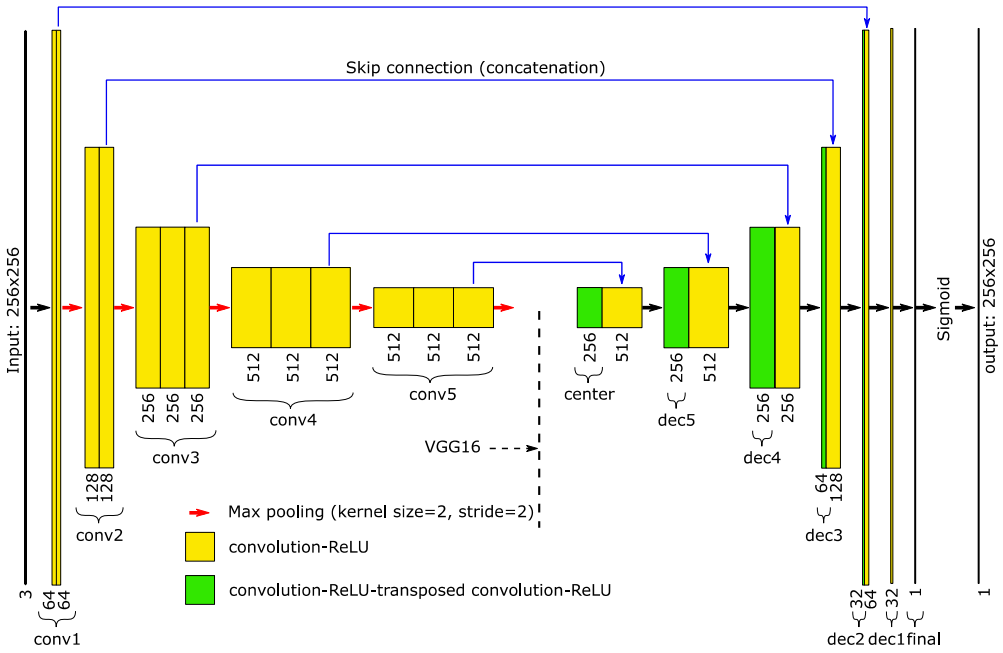


Fig. 11. TernausNet architecture [53,54].

initialized with parameters pre-trained on the ImageNet database [58]. For this case, a grid search led to the initial learning rate of 2e-4. Weights of the decoder path, however, were initialized randomly using the default method in PyTorch. All layers of the network were trained with a single learning rate.

The dice loss was evaluated on the training and validation data, and the trends are shown in Fig. 12 for both considered models (A and B). The plot of loss function vs. epoch shows that both training and validation losses decreased, while in higher epochs, the training loss became slightly lower than the validation loss due to overfitting. Therefore, the best model for each case was selected based on the validation loss. By comparing plots of loss function vs. epoch of both models, it can be observed that the dice loss over validation data converged faster (epoch = 67) and to a lower value (loss = 0.21) in model B compared to model A (epoch = 91, loss = 0.23).

6.3. Results

TernausNet (model B) was then used to predict the crack patterns in the images shown in Fig. 10(a). Fig. 13 depicts the predicted crack patterns. The figure shows that the trained TernausNet could detect cracks with a large width range, from very thin cracks to cracks of several pixels in width. In this set of examples, the continuity of the cracks is also well preserved. In Section 7, the TernausNet performance is quantitatively compared with the threshold method. In spite of the satisfactory prediction by TernausNet, there are some images in which the model fails to correctly predict crack pixels. Some examples are shown in Fig. 14. The yellow boxes in Fig. 14(b) and (c) highlight crack pixels that are not detected by the model (FN) and pixels that are incorrectly classified as a crack (FP).

Table 1
Details of operations in TernausNet.

Layer name	Operation	Kernel size	Number of kernels	Stride	Padding
conv1	Conv.*-ReLU**	3 × 3	64	1	1
	Conv.-ReLU	3 × 3	64	1	1
conv2	Conv.-ReLU	3 × 3	128	1	1
	Conv.-ReLU	3 × 3	128	1	1
conv3	Conv.-ReLU	3 × 3	256	1	1
	Conv.-ReLU	3 × 3	256	1	1
	Conv.-ReLU	3 × 3	256	1	1
conv4	Conv.-ReLU	3 × 3	512	1	1
	Conv.-ReLU	3 × 3	512	1	1
	Conv.-ReLU	3 × 3	512	1	1
conv5	Conv.-ReLU	3 × 3	512	1	1
	Conv.-ReLU	3 × 3	512	1	1
	Conv.-ReLU	3 × 3	512	1	1
center	Conv.-ReLU	3 × 3	512	1	1
	TransConv.***-ReLU	4 × 4	256	2	1
dec5	Conv.-ReLU	3 × 3	512	1	1
	TransConv.-ReLU	4 × 4	256	2	1
dec4	Conv.-ReLU	3 × 3	512	1	1
	TransConv.-ReLU	4 × 4	256	2	1
dec3	Conv.-ReLU	3 × 3	256	1	1
	TransConv.-ReLU	4 × 4	64	2	1
dec2	Conv.-ReLU	3 × 3	128	1	1
	TransConv.-ReLU	4 × 4	32	2	1
dec1	Conv.-ReLU	3 × 3	32	1	1
final	Conv.	1 × 1	1	1	1

* Conv. = convolution/ ** ReLU stands for the "Rectified Linear Unit" and is defined as max(0, the input of the function). / *** TransConv. = Transposed Convolution

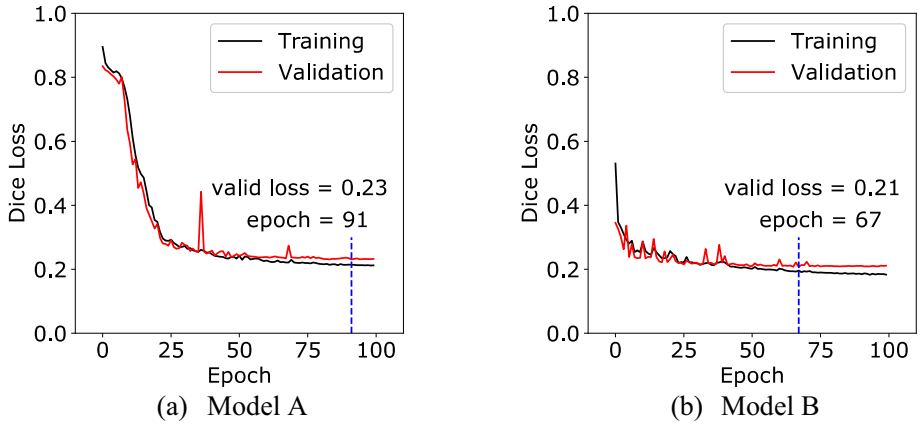


Fig. 12. Dice loss vs. epoch over training and validation data: (a) encoder weights initialized using random initialization, and (b) encoder weights initialized with pre-trained values.

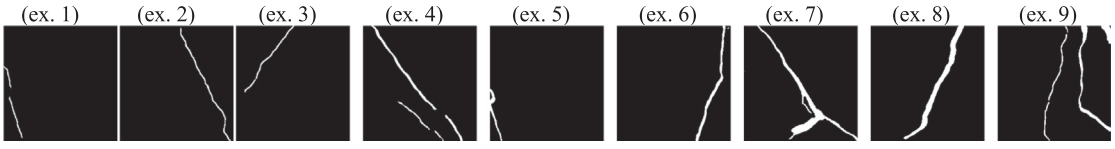


Fig. 13. Segmented cracks on examples images in Fig. 10(a) by TernausNet (model B).

7. Comparison of the methods

The performance of the two crack segmentation methods considered in this article, i.e., (i) the segmentation by applying a threshold value on the ε_1 map and (ii) the deep learning approach, on the test data is summarized in Table 2. For both methods, the best performing model was used for the comparison. This means that for the segmentation by applying a threshold value on the ε_1 map, the model with a subset size of 15 × 15 pixels and a threshold value of 10.5% was used. For the deep learning method, model B was used, wherein the parameters of the encoder path were initialized using pre-trained values on the ImageNet database. Regarding the SE value, the application of a threshold on the ε_1 maps slightly outperformed the deep learning approach. However, the PC value of the detection was quite poor (0.350) compared to that obtained by the deep learning approach (0.819). Similar performance indices were obtained for the DC metric, which was considered the most relevant metric and which served as the criterion for choosing the best parameter combination for each class of model—either the best combination of subset size and threshold value or the loss

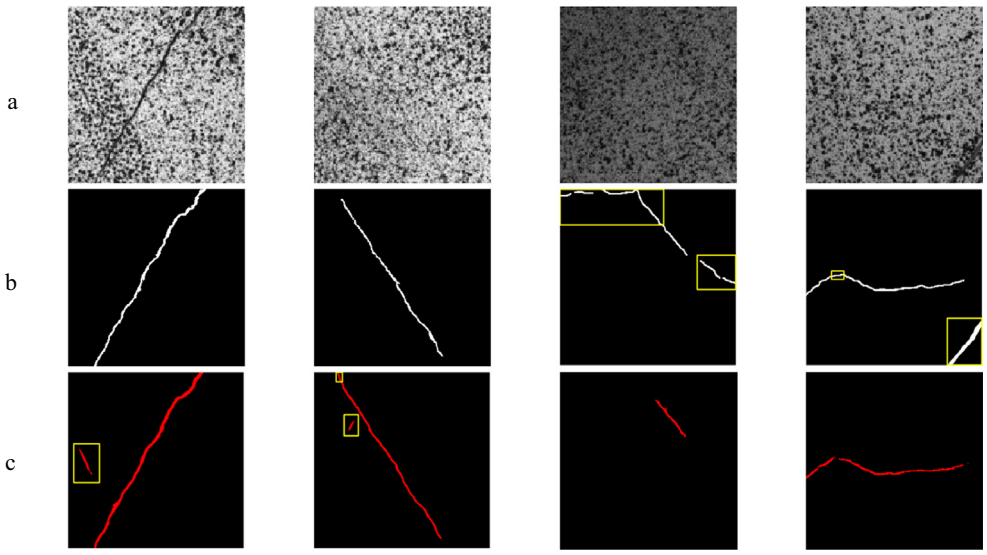


Fig. 14. Illustration of erroneous segmented crack pixels using TernausNet (model B). (a) Images, (b) ground truth masks, and (c) prediction.

Table 2
Comparison of the performance of the two segmentation methods on test data.

Segmentation method			SE	PC	DC
Apply a threshold to maximum principal strain map (Subset size = 15 × 15 pixels, threshold value = 10.5%)			**0.873**	0.350	0.481
Deep Learning (TernausNet)	Encoder weights	Random initialization	0.827	0.801	0.810
		Pre-trained weights on the ImageNet database	0.834	**0.819**	**0.821**

function to optimize. Note that the SE of the deep learning approach could be increased by choosing another loss function, such as Tversky loss [59] wherein the weights associated with FP and FN pixels can be tuned (in the dice loss, they are weighted equally) [59].

Although applying a threshold on the ε_1 map could detect the majority of the crack pixels, many more pixels were also incorrectly classified as cracks (false positive values). Therefore, the geometry of crack patterns was not well preserved by this method. Conversely, the deep learning approach proved a promising alternative method to the threshold method, as most of the crack pixels were segmented correctly and the precision of detection for crack/non-crack pixels was much higher than for the threshold method. For this reason, the deep learning method outperformed the crack segmentation obtained by applying a threshold value on the ε_1 map.

8. Conclusions

This study compares the performance of two methods—(i) the threshold and (ii) the deep learning method—for detecting crack pixels on laboratory images, i.e., grayscale images used as inputs for the local DIC method. The image database was selected from images taken from two experimental campaigns on stone masonry walls that were plastered on one side [12,13]. The DIC measurements discussed in this paper were obtained from this plastered side.

In the first approach, DIC analyses were performed by setting the subset size to 15 × 15 pixels, 19 × 19 pixels, 23 × 23 pixels, and 33 × 33 pixels (step size = 2 × 2 pixels, mm/pixel = 0.43) using VIC-3D software [45]. Then, from the resulting 3D displacement fields, the maximum principal strains were computed. Different threshold values were chosen (0.5%, 1.5%, 2.5%, 3.5%, … , 24.5%) to produce a binary crack mask from the maximum principal strain maps. In the second approach, a deep convolutional neural network called TernausNet was adopted to perform pixel-wise crack segmentation. TernausNet is an encoder-decoder architecture that uses VGG16 convolution blocks as the encoder and is initialized either using random initialization (model A) or pre-trained values on the ImageNet (model B).

The performance of the two approaches was examined in terms of sensitivity, precision and dice coefficient. It was shown that the deep learning model B, in which the encoder parameters were initialized using the pre-trained values on the ImageNet database, converged faster and to a lower loss compared to model A, where the encoder weights were initialized using the default PyTorch random initialization method. Additionally, it was found that the threshold method resulted in higher sensitivity (0.873) than the deep learning model B (0.834). However, the precision of the predictions made by the threshold method was quite poor (0.350) compared to the deep learning model B (0.819). Consequently, this led to a low dice coefficient of 0.481 for the threshold method, while the corresponding value for deep learning model B was 0.821. The high dice coefficient showed that the deep learning approach could detect the majority of the visible cracks and that the detection was precise. The higher sensitivity for the threshold method revealed that more cracks could be detected compared to the deep learning method, but as the precision of the detection was quite low, there were many pixels incorrectly labeled as cracks. It was observed that in the threshold method, detected pixels were fragmented for very thin cracks. In other words, the continuity of the crack was not preserved. Moreover, the detected cracks were wider than the actual cracks in this method. Conversely, the deep learning method better preserved the geometry of cracks.

Possible directions for future studies to improve the performance of the deep learning method include implementing an iterative refinement of predictions [49], increasing the size of the

database, and adding loss functions to consider the topology of the crack maps [49]. Post-processing of the predicted crack maps could also be performed to both remove the noisy outputs and connect the gaps between crack segments, which could be done by implementing methods such as tensor voting [36,37]. Furthermore, the deep learning approach can be used as a pre-processing step to detect crack pixels before the DIC analysis. This can be useful for automating the initial seed generation in the segregated areas. The crack geometry can also be used as an input for DIC methods that are based on discontinuous rather than continuous displacement fields [60,61].

CRediT authorship contribution statement

Amir Rezaie: Methodology, Validation, Formal analysis, Investigation, Data curation, Writing - original draft, Writing - review & editing, Visualization. **Radhakrishna Achanta:** Conceptualization, Methodology, Resources. **Michele Godio:** Investigation, Writing - original draft, Writing - review & editing, Visualization. **Katrin 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.

Acknowledgments

The project is supported by the Swiss National Science Foundation (grant 200021_175903/1 "Equivalent frame models for the in-plane and out-of-plane response of unreinforced masonry buildings").

References

[1] W.H. Peters, W.F. Ranson, Digital imaging techniques in experimental stress analysis, Opt. Eng. 21 (1982) 427–431, https://doi.org/10.1117/12.7972925.
[2] M.A. Sutton, J.J. Orteu, H. Schreier, Image correlation for shape, motion and deformation measurements: basic concepts, theory and applications, Springer Science & Business Media, 2009.
[3] J. Zhao, Y. Sang, F. Duan, The state of the art of two-dimensional digital image correlation computational method, Eng. Rep. 1 (2019), https://doi.org/10.1002/eng2.12038.
[4] J.D. Helm, Digital image correlation for specimens with multiple growing cracks, Exp. Mech. 48 (2008) 753–762, https://doi.org/10.1007/s11340-007-9120-2.
[5] K.M. Dolatshahi, K. Beyer, Stiffness and strength estimation of damaged unreinforced masonry walls using crack pattern, J. Earthquake Eng. (2019) 1–20, https://doi.org/10.1080/13632469.2019.1693446.
[6] H. Momeni, K.M. Dolatshahi, Predictive equations for drift ratio and damage assessment of RC shear walls using surface crack patterns, Eng. Struct. 190 (2019) 410–421, https://doi.org/10.1016/j.engstruct.2019.04.018.
[7] A. Ebrahimkhanlou, A. Athanasiou, H. Trevor D., O. Bayrak, S. Salamone, Fractal and multifractal analysis of crack patterns in prestressed concrete girders, J. Bridge Eng. 24 (2019) 4019059. doi:10.1061/(ASCE)BE.1943-5592.0001427.
[8] A. Farhidzadeh, E. Dehghan-Niri, A. Moustafa, S. Salamone, A. Whittaker, Damage assessment of reinforced concrete structures using fractal analysis of residual crack patterns, Exp. Mech. 53 (2013) 1607–1619, https://doi.org/10.1007/s11340-013-9769-7.
[9] 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. (2019), https://doi.org/10.1111/mice.12509.
[10] A. Rezaie, A.J.P. Mauron, K. Beyer, Sensitivity analysis of fractal dimensions of crack maps on concrete and masonry walls, Autom. Constr. 117 (2020), https://doi.org/10.1016/j.autcon.2020.103258.
[11] H.M. Madani, K.M. Dolatshahi, Strength and stiffness estimation of damaged reinforced concrete shear walls using crack patterns, Struct. Control Health Monitor. 27 (2020), https://doi.org/10.1002/stc.2494.
[12] 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, https://doi.org/10.1016/j.conbuildmat.2020.118972.
[13] M. Godio, F. Vanin, S. Zhang, K. Beyer, Quasi-static shear-compression tests on stone masonry walls with plaster: Influence of load history and axial load ratio, Eng. Struct. 192 (2019) 264–278, https://doi.org/10.1016/j.engstruct.2019.04.041.
[14] J.-F. Destrebecq, E. Toussaint, E. Ferrier, Analysis of cracks and deformations in a full scale reinforced concrete beam using a digital image correlation technique, Exp. Mech. 51 (2011) 879–890, https://doi.org/10.1007/s11340-010-9384-9.
[15] S.-H. Tung, M.-H. Shih, W.-P. Sung, Development of digital image correlation method to analyse crack variations of masonry wall, Sadhana 33 (2008) 767–779, https://doi.org/10.1007/s12046-008-0033-2.
[16] R. Ghorbani, F. Matta, M.A. Sutton, Full-field deformation measurement and crack mapping on confined masonry walls using digital image correlation, Exp. Mech. 55 (2015) 227–243, https://doi.org/10.1007/s11340-014-9906-y.
[17] P.A. Korswagen, M. Longo, J.G. Rots, High-resolution monitoring of the initial development of cracks in experimental masonry shear walls and their reproduction in finite element models, Eng. Struct. 211 (2020), https://doi.org/10.1016/j.engstruct.2020.110365.
[18] P.A. Korswagen, M. Longo, E. Meulman, J.G. Rots, Crack initiation and propagation in unreinforced masonry specimens subjected to repeated in-plane loading during light damage, Bull. Earthq. Eng. 17 (2019) 4651–4687, https://doi.org/10.1007/s10518-018-00553-5.
[19] N.A. Hoult, M. Dutton, A. Hoag, W.A. Take, Measuring crack movement in reinforced concrete using digital image correlation: overview and application to shear slip measurements, Proc. IEEE 104 (2016) 1561–1574, https://doi.org/10.1109/JPROC.2016.2535157.
[20] S.P. Morgan, Using Digital Image Correlation to Detect Cracking in Opalinus Shale, in: Fracture, Fatigue, Failure and Damage Evolution, Volume 8, Springer, 2017: pp. 125–131.
[21] A.F. Cinar, S.M. Barhli, D. Hollis, M. Flansbjer, R.A. Tomlinson, T.J. Marrow, M. Mostafavi, An autonomous surface discontinuity detection and quantification method by digital image correlation and phase congruency, Opt. Lasers Eng. 96 (2017) 94–106, https://doi.org/10.1016/j.optlaseng.2017.04.010.
[22] N. Gehri, J. Mata-Falcón, W. Kaufmann, Automated crack detection and measurement based on digital image correlation, Constr. Build. Mater. 256 (2020) 119383, https://doi.org/10.1016/j.conbuildmat.2020.119383.
[23] X. Yang, H. Li, Y. Yu, X. Luo, T. Huang, X. Yang, Automatic pixel-level crack detection and measurement using fully convolutional network, Comput.-Aided Civ. Infrastruct. Eng. 33 (2018) 1090–1109, https://doi.org/10.1111/mice.12412.
[24] Q. Mei, M. Gül, M.R. Azim, Densely connected deep neural network considering connectivity of pixels for automatic crack detection, Autom. Constr. 110 (2020), https://doi.org/10.1016/j.autcon.2019.103018.
[25] B.F. Spencer, V. Hoskere, Y. Narazaki, Advances in computer vision-based civil infrastructure inspection and monitoring, Engineering 5 (2019) 199–222, https://doi.org/10.1016/j.eng.2018.11.030.
[26] X. Zhang, D. Rajan, B. Story, Concrete crack detection using context-aware deep semantic segmentation network, Comput.-Aided Civ. Infrastruct. Eng. 34 (2019) 951–971, https://doi.org/10.1111/mice.12477.
[27] K. Gopalakrishnan, S.K. Khaitan, A. Choudhary, A. Agrawal, Deep Convolutional Neural Networks with transfer learning for computer vision-based data-driven pavement distress detection, Constr. Build. Mater. 157 (2017) 322–330, https://doi.org/10.1016/j.conbuildmat.2017.09.110.
[28] Y. Ren, J. Huang, Z. Hong, W. Lu, J. Yin, L. Zou, X. Shen, Image-based concrete crack detection in tunnels using deep fully convolutional networks, Constr. Build. Mater. 234 (2020) 117367, https://doi.org/10.1016/j.conbuildmat.2019.117367.
[29] S.E. Park, S.-H. Eem, H. Jeon, Concrete crack detection and quantification using deep learning and structured light, Constr. Build. Mater. 252 (2020) 119096, https://doi.org/10.1016/j.conbuildmat.2020.119096.
[30] 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, https://doi.org/10.1016/j.conbuildmat.2018.08.011.
[31] D. Brackenbury, I. Brilakisl, M. DeJong, Automated Defect Detection For Masonry Arch Bridges, in: International Conference on Smart Infrastructure and Construction 2019 (ICSIC): pp. 3–9. http://doi.org/10.1680/icsic.64669.003.
[32] 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: J. Teizer (Ed.), Proceedings of the 35th International Symposium on Automation and Robotics in Construction (ISARC), International Association for Automation and Robotics in Construction (IAARC), Taipei, Taiwan, 2018: pp. 118–125. doi:10.22260/ISARC2018/0016.
[33] L. Zhang, F. Yang, Y.D. Zhang, Y.J. Zhu, Road crack detection using deep convolutional neural network, in: 2016 IEEE International Conference on Image Processing (ICIP), 2016: pp. 3708–3712. doi:10.1109/ICIP.2016.7533052.
[34] Y.-J. Cha, W. Choi, O. Büyüköztürk, Deep learning-based crack damage detection using convolutional neural networks, Comput.-Aided Civ. Infrastruct. Eng. 32 (2017) 361–378, https://doi.org/10.1111/mice.12263.

[35] X. Wang, Z. Hu, Grid-based pavement crack analysis using deep learning, in: 2017 4th International Conference on Transportation Information and Safety (ICTIS), 2017: pp. 917–924. doi:10.1109/ICTIS.2017.8047878.
[36] Z. Kaige, C.H. D., Z. Boyu, Unified approach to pavement crack and sealed crack detection using preclassification based on transfer learning, J. Comput. Civ. Eng. 32 (2018) 4018001. doi:10.1061/(ASCE)CP.1943-5487.0000736.
[37] G. Medioni, C.-K. Tang, M.-S. Lee, Tensor voting: Theory and applications, Proc. RFIA (2000).
[38] F. 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 (2018) 4392–4400, https://doi.org/10.1109/TIE.2017.2764844.
[39] Q. Zou, Z. Zhang, Q. Li, X. Qi, Q. Wang, S. Wang, S. Member, DeepCrack : Learning hierarchical convolutional features for crack detection, IEEE Trans. Image Process. 28 (2019) 1498–1512, https://doi.org/10.1109/TIP.2018.2878966.
[40] Y. Shi, L. Cui, Z. Qi, F. Meng, Z. Chen, Automatic road crack detection using random structured forests, IEEE Trans. Intell. Transp. Syst. 17 (2016) 3434–3445, https://doi.org/10.1109/TITS.2016.2552248.
[41] Q. Zou, Y. Cao, Q. Li, Q. Mao, S. Wang, CrackTree: Automatic crack detection from pavement images, Pattern Recogn. Lett. 33 (2012) 227–238, https://doi.org/10.1016/j.patrec.2011.11.004.
[42] Z. Liu, Y. Cao, Y. Wang, W. Wang, Computer vision-based concrete crack detection using U-net fully convolutional networks, Autom. Constr. 104 (2019) 129–139, https://doi.org/10.1016/j.autcon.2019.04.005.
[43] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, H. Adam, Encoder-decoder with atrous separable convolution for semantic image segmentation, in: Proceedings of the European Conference on Computer Vision (ECCV), Springer International Publishing, 2018: pp. 801–818. http://doi.org/10.1007/978-3-030-01234-2_49.
[44] International Digital Image Correlation Society, Jones, E.M.C. and Iadicola, M.A. (Eds.) (2018). A Good Practices Guide for Digital Image Correlation. http://doi.org/10.32720/idics/gpg.ed1.
[45] Correlated Solutions, VIC-3D Software Manual. http://www.correlatedsolutions.com/supportcontent/VIC-3D-8-Manual.pdf.
[46] P. Zhou, K.E. Goodson, Subpixel displacement and deformation gradient measurement using digital image/speckle correlation, Opt. Eng. 40 (2001) 1613–1620, https://doi.org/10.1117/1.1387992.
[47] D. Lecompte, A. Smits, S. Bossuyt, H. Sol, J. Vantomme, D. Van Hemelrijck, A.M. Habraken, Quality assessment of speckle patterns for digital image correlation, Opt. Lasers Eng. 44 (2006) 1132–1145, https://doi.org/10.1016/j.optlaseng.2005.10.004.
[48] A. Breheret, Pixel Annotation Tool, (2017). https://github.com/abreheret/PixelAnnotationTool.
[49] A. Mosinska, P. Marquez-Neila, M. Koziński, P. Fua, Beyond the pixel-wise loss for topology-aware delineation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018: pp. 3136–3145. http://openaccess.thecvf.com/content_cvpr_2018/html/Mosinska_Beyond_the_Pixel-Wise_CVPR_2018_paper.html.
[50] R.M. Haralick, S.R. Sternberg, X. Zhuang, Image Analysis Using Mathematical Morphology, IEEE Transactions on Pattern Analysis and Machine Intelligence. PAMI-9 (1987) 532–550. doi:10.1109/TPAMI.1987.4767941.
[51] C. Sun, Y. Zhou, J. Chen, H. Miao, Measurement of deformation close to contact interface using digital image correlation and image segmentation, Exp. Mech. 55 (2015) 1525–1536, https://doi.org/10.1007/s11340-015-0055-8.
[52] O. Ronneberger, P. Fischer, T. Brox, U-net: Convolutional networks for biomedical image segmentation, in: International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer, 2015: pp. 234–241.
[53] A.A. Shvets, A. Rakhlin, A.A. Kalinin, V.I. Iglovikov, Automatic Instrument Segmentation in Robot-Assisted Surgery using Deep Learning, 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA). (2018). doi:10.1109/icmla.2018.00100.
[54] V. Iglovikov, A. Shvets, TernausNet: U-Net with VGG11 Encoder Pre-Trained on ImageNet for Image Segmentation, (2018). https://arxiv.org/abs/1801.05746.
[55] K. Simonyan, A. Zisserman, Very Deep Convolutional Networks for Large-Scale Image Recognition, in: International Conference on Learning Representations, 2015. https://arxiv.org/abs/1409.1556.
[56] F. Milletari, N. Navab, S.-A. Ahmadi, V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation, 2016 Fourth International Conference on 3D Vision (3DV). (2016) 565–571.
[57] D.P. Kingma, J. Ba, Adam: A Method for Stochastic Optimization, in: 3rd International Conference on Learning Representations, {ICLR} 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings, 2015. http://arxiv.org/abs/1412.6980.
[58] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A.C. Berg, L. Fei-Fei, ImageNet Large Scale Visual Recognition Challenge, (2014). https://arxiv.org/abs/1409.0575.
[59] S.S.M. Salehi, D. Erdogmus, A. Gholipour, Tversky loss function for image segmentation using 3D fully convolutional deep networks, in: International Workshop on Machine Learning in Medical Imaging, Springer, 2017: pp. 379–387.
[60] J. Réthoré, F. Hild, S. Roux, Extended digital image correlation with crack shape optimization, Int. J. Numer. Meth. Eng. 73 (2008) 248–272, https://doi.org/10.1002/nme.2070.
[61] S. Roux, J. Réthoré, F. Hild, Digital image correlation and fracture: an advanced technique for estimating stress intensity factors of 2D and 3D cracks, J. Phys. D Appl. Phys. 42 (2009), https://doi.org/10.1088/0022-3727/42/21/214004.
[62] Ankang Ji, Xiaolong Xue, Yuna Wang, Xiaowei Luo, Weirui Xue, An integrated approach to automatic pixel-level crack detection andquantification of asphalt pavement, Automation in Construction 114 (2020), https://doi.org/10.1016/j.autcon.2020.103176.