Smart Structures and Systems, *Vol. 33, No. 4 (2024) 301-311*
https://doi.org/10.12989/sss.2024.33.4.301

Crack growth prediction on a concrete structure using deep ConvLSTM

Man-Sung Kang and Yun-Kyu An*

Department of Architectural Engineering, Sejong University, 209, Neungdong-ro, Gwangjin-gu, Seoul, 05006, Republic of Korea

(*Received October 27, 2023, Revised April 15, 2024, Accepted April 25, 2024*)

Abstract. This paper proposes a deep convolutional long short-term memory (ConvLSTM)-based crack growth prediction technique for predictive maintenance of structures. Since cracks are one of the critical damage types in a structure, their regular inspection has been mandatory for structural safety and serviceability. To effectively establish the structural maintenance plan using the inspection results, crack propagation or growth prediction is essential. However, conventional crack prediction techniques based on mathematical models are not typically suitable for tracking complex nonlinear crack propagation mechanism on civil structures under harsh environmental conditions. To address the technical issue, a field data-driven crack growth prediction technique using ConvLSTM is newly proposed in this study. The proposed technique consists of the four steps: (1) time-series crack image acquisition, (2) target image stabilization, (3) deep learning-based crack detection and quantification and (4) crack growth prediction. The performance of the proposed technique is experimentally validated using a concrete mock-up specimen by applying step-wise bending loads to generate crack growth. The validation test results reveal the prediction accuracy of 94% on average compared with the ground truth obtained by field measurement.

Keywords: crack growth prediction; data-driven; deep learning; image stabilization; predictive maintenance

1. Introduction

Although civil infrastructures are designed to meet safety management standards, the degradation of structures is inevitable due to repeated external loading and harsh environmental variations under in-service conditions (Holt and Leivo 2004, Safiuddin *et al*. 2018, Wang 2018, Yu *et al*. 2023). The structural degradation may reduce the remaining service life of structures and even lead to catastrophic structural failure. Thus, structural regular safety inspection has been mandatory in many countries. In South Korea, for example, the regular safety inspection should be conducted at least once a half-year according to Special Act on the Safety Control and Maintenance of Establishments law since 1995.

Beside structural condition diagnosis, the main goal of the regular inspection is structural health prognosis which is to establish structural maintenance plan by analyzing time-history data. In particular, crack growth prediction on concrete structures has become one of the hot issues, because it would be crucial for structural integrity and safety in terms of strength loss (Ismail *et al*. 2010, Ni *et al*. 2019) as well as corrosion of internal rebars (Lai *et al*. 2020). However, expert-dependent crack inspection results could not be used as the time-history data for maintenance planning, because they are often subjective or unreliable. Recently, computer vision techniques have been widely adopted for structural crack inspection (Jang *et al*. 2022, Koh *et al*. 2022, Xu *et al*. 2023, Bae and An 2024), making it possible to cumulate reliable crack inspection data. Digital cameras have been combined with the various types of unmanned robot or vehicle for inspecting the cracks in extensive areas of target structures. For example, unmanned aerial vehicles have been widely utilized for crack inspection of dam and bridges (Reagan *et al*. 2018, Bae *et al*. 2021, Qi *et al*. 2021, Wu *et al*. 2022, Ding *et al*. 2023), because they can easily inspect areas where are difficult and dangerous for human beings to access. Also, unmanned climbing robots combined with multiple digital cameras (Jang *et al*. 2021) and fusion data-scanning system (Jang *et al*. 2024) were adopted for precise damage evaluation.

Even if the reliable crack inspection data can be obtained by various methodologies as time passed, crack growth prediction is still challenging. Since 1920's, the conventional fracture mechanics have been widely studied to calculate the driving force on a crack and characterize the material's resistance to fracture. Griffith firstly suggested that the brittle material contains population of fine small cracks and defects with varying size, geometries and orientation (Griffith 1921). In an initial stage of the fracture mechanics, linear elastic phenomenon was mainly used for material conditions which are predominantly linear elastic during the fatigue process. In 1960's, Paris and Erdogan (1963) proposed a fracture mechanics-based fatigue life prediction method, called Paris' law, which computed the fatigue crack growth rate with respect to the applied stress intensity factor, material coefficient and environmental conditions. Chang *et al*. (2020) utilized the Paris' law to characterize the crack tip and fatigue crack growth rate based on the relationship between the stress intensity and load cycle by considering the uncertainty of material coefficients. They obtained the material coefficients through

*Corresponding author, Ph.D., Professor,
E-mail: yunkyuan@sejong.ac.kr

ISSN: 1738-1584 (Print), 1738-1991 (Online)

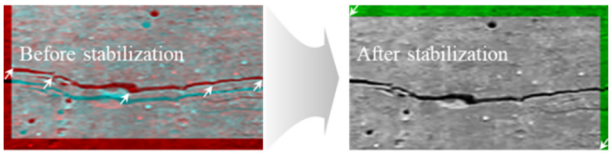
crack growth test data and regression analysis. However, the limitation of the Paris' law is that it cannot properly consider field uncertainties associated with test setups, measurement, geometric complexity, material property and so on. Although these conventional models have been well mathematically established, they still have difficulty in tuning both the linear and nonlinear parameters to solve complex nonlinear crack propagation mechanism in a field.

To overcome the limitations of parameter tuning, nowadays, machine learning algorithms have adopted which have the advantages of handling both linear and nonlinear features of different data types (Raja *et al*. 2020). Rovinelli *et al*. (2018) proposed machine learning and Bayesian network techniques to identify microstructural variables which are related to the direction and rate of the fatigue crack propagation. In addition, total seven parameters such as tensile, yield, fatigue strength, theoretical stress concentration factor, notch root radius, samples size and notch fatigue limit are adopted as inputs of machine learning framework for predicting the fatigue stress concentration factor (Wang *et al*. 2016). Pierson *et al*. (2019) used the convolutional neural network-based prediction for microstructure-sensitive fatigue crack path in a polycrystalline alloy using the Paris' law. More recently, Raja *et al*. (2020) applied machine learning techniques to predict nonlinearities of fatigue crack growth by learning the simple relationship between stress intensity factor and load cycles. However, the limitation of machine learning-based parameter tunning is trained and built on intentionally decided features such as stress, crack length and orientation instead of features learned from the raw data, which can result in information loss and model performance degradation.

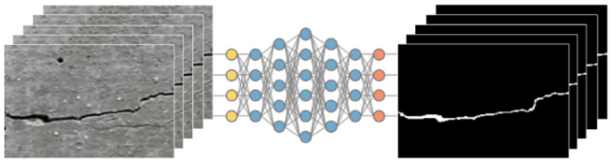
In this study, a deep convolutional long short-term memory (ConvLSTM)-based crack growth prediction technique is newly proposed to resolve the above technical limitations. The proposed technique has the following superiorities over the existing techniques. First, it utilizes only data-driven time-series crack propagation digital image data for testing the predictive network without relying on mathematical models. Second, precise crack detection and quantification can be possible through region of interest (ROI) matching, called stabilization, even if the time-series data are obtained with different camera pose and long-time interval. In addition, the efficient maintenance plan can be established through the proposed technique by considering structural maintenance criteria. The proposed technique consists of the following four steps: (1) time-series crack propagation image acquisition, (2) stabilization of the time-series digital image to minimize the unwanted camera pose change, (3) deep learning-based automated crack detection and quantification and (4) ConvLSTM-based crack growth prediction.

This paper is organized as follows. Section 2 develops the ConvLSTM-based crack growth prediction technique. The performance of the proposed technique is experimentally validated using a concrete mock-up specimen in Section 3. The extending potential of the proposed technique is then discussed in Section 4. Finally, the paper is concluded in Section 5.

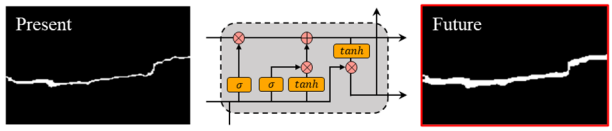
(a) Time-series digital image acquisition



(b) Stabilization of time-series digital images



(c) Deep learning-based crack quantification



(d) ConvLSTM-based crack growth prediction

Fig. 1 Overall procedure of ConvLSTM-based crack growth prediction

2. ConvLSTM-based crack growth prediction

Fig. 1 shows the overall procedure of ConvLSTM-based crack growth prediction. First, the data acquisition system is basically composed of digital camera combined with various types of inspection machine such as unmanned aerial vehicle and climbing robot, to name a few. Once the time-series digital images are acquired from a target structure, the stabilization procedure should be performed to minimize unnecessary mismatch among the time-different digital images. For precise crack growth prediction, afterward, only crack information is extracted and quantified using a deep learning-based crack detection network. Finally, the ConvLSTM-based crack growth prediction is carried out. The details of each procedure are described in the following sub-sections.

2.1 Time-series digital image acquisition

The time-series digital images can be regularly obtained from a target structure with a certain time interval by using a digital camera. Here, the digital camera can be mounted on drone or robot depending on a target structure and surrounding environments. The most important thing of this step for crack growth prediction is that the time-series digital images should contain cracks with same ROI. However, the technical challenge is how to retain the identical data acquisition conditions such as ROI, pixel resolution, angle, lens type and etc. in every image acquisition process with near-yearly inspection period. In

reality, it is almost impossible to achieve identical data acquisition conditions by considering civil infrastructures' lifetime of several decades. In particular, the rapid device development of unmanned robot system as well as digital camera makes it more difficult to retain the digital image quality as time goes on.

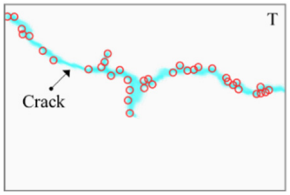
2.2 Stabilization of the time-series digital images

To resolve the aforementioned technical problem, the time-series digital images need to be stabilized prior to the subsequent deep learning-based crack analysis. The main assumption of this step is that the target surface condition within ROI will not be much changed with respect to existing cracks. Nevertheless, the camera pose cannot be kept in every image capture process when it comes to micro-scale cracks. To resolve the technical problem, the stabilization process of the time-series digital images is newly proposed. Initially, the feature points within the ROI of each time-series digital image are detected, and subsequently, their corresponding vector called descriptors are extracted from the timely-adjacent images. Since the corner features are discrete and distinguishable on each digital image, the corresponding feature points are detected using the Harris corner detector, which is one of the fastest corner detection algorithms (Harris and Stephens 1988).

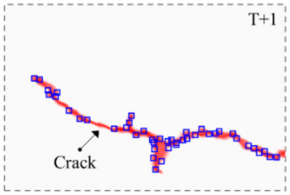
$$E(u,v) = \sum_{(x_k,y_k)\in W} [I(x_k+u, y_k+v) - I(x_k,y_k)]^2 \quad (1)$$

where $I(x_k, y_k)$ denotes the intensity of pixel location (x_k, y_k) of the image. W denotes the windowed patch. The corner could be found by calculating sum of squared difference (E) between initial W and shifted W which are produced by a shift (u,v) along the x- and y-directions. If W is shifted to the corner, then E will result in a large change.

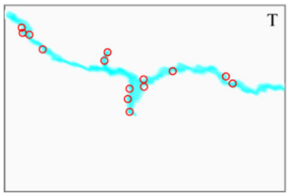
Figs. 2(a) and (b) show the timely-adjacent schematic crack images at T and T+1. These two images have different ROI marked as solid and dashed outer rectangles, respectively. The detected E of these crack images using Eq. (1) are indicated as the red circles and blue rectangles, respectively. Once the feature points are detected, the corresponding descriptors are extracted by adjusting brightness and gradient orientation around the detected feature points. To extract the descriptors, fast retina keypoint (FREAK) algorithm is employed in this study (Alahi *et al*. 2012). The FREAK descriptor is constructed based on intensity comparisons $s(P_a)$ between different pairs of sampling pattern around the detected feature point. Total 512 pairs of sampling pattern are used by comparing the pair-wise binary intensity.



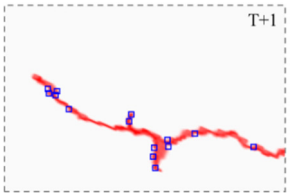
(a) Crack image at T with detected feature points



(b) Crack image at T+1 with detected feature points



(c) Matched descriptors at T



(d) Matched descriptors at T+1

Fig. 2 Schematics of timely-adjacent crack images with detected feature points and matched descriptors

$$s(P_a) = \begin{cases} 1 & if\ P_i > P_j \\ 0 & otherwise \end{cases} \quad (2)$$

where $P_a = (P_i, P_j)$ is the pair of sampling pattern, $i, j \in \{1,2,\dots,N\}$ and $i \neq j$. The presented comparison forms the basis for building the FREAK descriptor F as a N-dimensional bit string

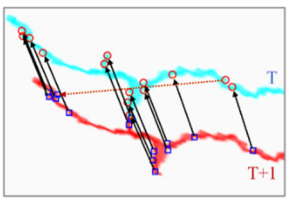
$$F = \sum_{0\le a<N} 2^a\, s(P_a) \quad (3)$$

Since F is extracted based on the detected features, the number of descriptors should be same with the detected feature ones. The descriptor matching between two images with different ROI can be easily identified by comparing the extracted descriptors from crack images at T and T+1. Figs. 2(c) and (d) show that the number of matched descriptors in the T and T+1 crack images.

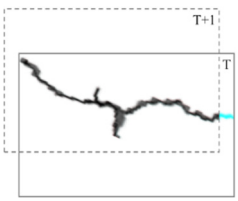
Although many descriptors are well matched as the black solid lines, the outlier noted as the red dotted line also often appear as shown in Fig. 3(a). To remove the outliers, a M-estimator sample consensus (MSAC) algorithm (Torr and Zisserman 2000) is applied. The MSAC algorithm will search for the valid inlier descriptors with respect to the given set of matched descriptors. Then, it derives an affine transformation, that makes the inliers from the T crack image, transform most closely with the inliers from the T+1 image. As a consequence, Fig. 3(b) shows the stabilized result between timely-adjacent crack images, achieved by mapping the inliers of matched descriptors in crack image T+1 to the inlier of matched descriptors in T, using different ROI.

2.3 Deep learning-based crack detection and quantification

Once the sequential time-series digital images are stabilized, only crack information are automatically



(a) The inlier and outlier descriptors matching



(b) Final stabilization result

Fig. 3 Representative stabilization results

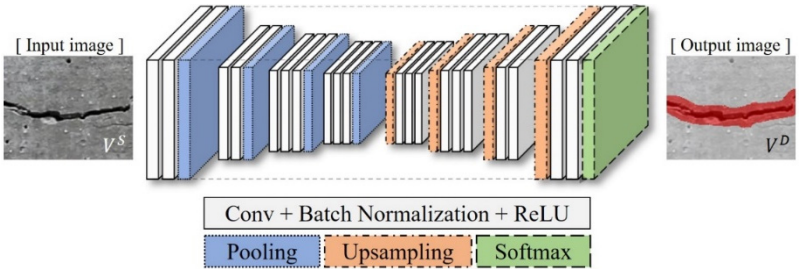


Fig. 4 Deep learning-based crack detection network: the V^S and V^D denote stabilized and crack detection images

detected from each stabilized image (V^S). In this study, a semantic segmentation model was constructed by modifying SegNet (Jang *et al.* 2021) and used for automated crack detection. The modified semantic segmentation model depicted in Fig. 4 comprises 10 convolutional layers, 4 pooling layers and 4 upsampling layers, while the SegNet utilizes 13 convolutional layers and 5 pooling layers in the encoder and decoder. In particular, the final fully connected softmax layer is modified to binary classification, i.e., crack and intact classes. The filter size for each convolutional layer is set to 3×3 across the entire model. The depth of the feature map is also fixed to 64 with a stride of 1, which is followed by batch normalization (Ioffe and Szegedy 2015) and rectified linear unit (ReLU) (Nair and Hinton 2010). The blue-colored layers in Fig. 4 represent the pooling layers with 2×2 window and a stride of 2. The upsampling layers upsample the preceding convolutional layers by receiving pooling indices from the corresponding pooling layers. The pixel-to-pixel binary class softmax classifier is placed at the end of the model. Total 1,021 crack images, each with a resolution of 360×480 pixels, were used to train the network. The training crack areas in each image were annotated in a pixel-unit using a polygon tool. During the training process, 70 epochs with 10 mini-batch size and Adam optimizer with an initial learning rate of 0.01 were used.

Next, the detected cracks (V^D) need to be precisely quantified for precise crack growth prediction. Although the crack areas are classified into the RGB image with respect to only the crack regions (V^C) as depicted Fig. 5, a lot of noise components are still included in the classified crack regions. Thus, the statistical denoising process for obtaining the binary crack image (I^B) is sequentially conducted on V^C. Firstly, V^C is converted from RGB to grayscale. Then, I^B is obtained by subtracting the resultant image of the median filter from the grayscale crack image, using a specific threshold based on Otsu's method (Otsu 1979).

Subsequently, the Euclidean distance transform and skeletonization processes are carried out using I^B image to extract crack shape and to compute the corresponding crack width. The skeletonization makes it possible to extract only crack shape that physically represent the linear connectivity between adjacent pixels. As shown in Figure 5, the skeletonization crack image (I^S) can be obtained based on morphological thinning algorithm (Lam *et al.* 1992), which is given by

$$I^S = [I^B \otimes \{B_K\}]^N = \left[I^B \setminus \left((I^B \odot B_1) \cdots \odot B_K\right)\right]^N, \quad (K = 1, \ldots, 8; N = 1, 2, 3, \ldots) \tag{4}$$

where $\setminus$ and $\odot$ denote the set difference and hit-or-miss transformation. Each N represent the numbers of cyclical iteration until convergence, and B is the structuring elements.

The Euclidean distance between two pixels, i and j, is then calculated using the following equation

$$d(i,j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \; (i \in A, j \in A^c) \tag{5}$$

where A is the set of black pixels represented by 1 on I^B. A^c is the complement of A. Afterward, the Euclidean distance transformed crack image (I^E) can be obtained by

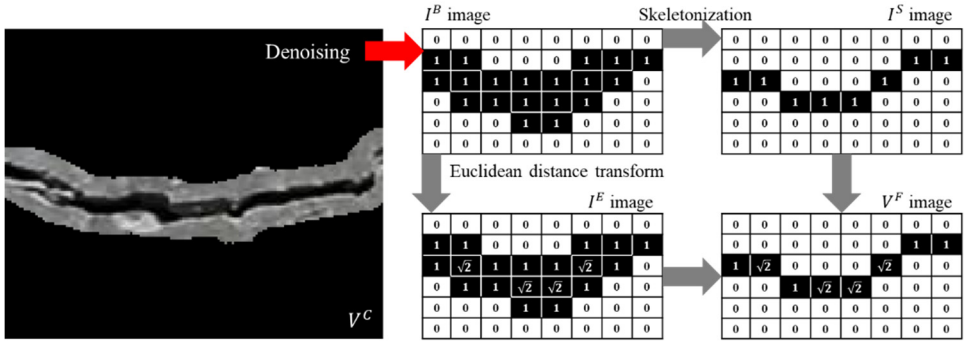


Fig. 5 Overall crack quantification procedure: denoising, skeletonization, and Euclidean distance transform: V^C, I^B, I^S, I^E, and V^F are the RGB classified crack, binary crack, skeletonized, Euclidean distance transformed, and crack width quantified images, respectively

$$I^E(i) = min(d(i,j)|I^B(j) \in A^c) \quad (6)$$

Finally, the final image (V^F) can be obtained by multiplying I^E and I^S for each pixel value.

2.4 ConvLSTM-based crack growth prediction

To predict the crack growth using I^B, we employ a ConvLSTM-based deep learning model known as the Predictive Coding Network (PreCNet) (Straka *et al*. 2023). PreCNet is a hierarchical predictive network that has been adapted into a deep learning framework while preserving the fidelity of the original network (Rao and Ballard 1999). Fig. 6 illustrates the hierarchical structure and training procedure of PreCNet. In PreCNet, each level in the hierarchical predictive network generates prediction (P_t) of neural activity and sends them to the lower level through feedback connections, except for the lowest level. Conversely, the error signal (E_l) between P_t and the actual input (I_t) is propagated to the higher level through feedforward connections. Note that, since the first prediction ($t = 1$) is produced without any feedback information, E_l related to the first predicted input is ignored. This E_l is utilized by the predictive estimator at each level to adjust its current estimate of I_t and generate the subsequent P_t. In the predictive estimator, the feedback prediction of the representation state r^{td} from the higher level is sent to lower-level predictive estimator. The difference between r^{td} and the current state of r is sent back to the higher level through feedforward connections as E_l. Additionally, the same E_l with the opposite sign, $r^{td} - r$, is used as input to the ConvLSTM along with the representation states from the previous time step to update the current state of r. The updated representation states r is then decoded by convolutional layer and passed to the lower predictive estimator. On the feedforward pathways, the internal representation state r is updated by incorporating the error from lower predictive estimator into the ConvLSTM. Once the higher layer prediction $\hat{A}$ and the pooled representation state R are obtained, these two states are used to calculate the error representation states by subtracting them in both orders and concatenating them using the ReLU activation function.

For stable training of PreCNet, we used a total of 2 million visual sequence datasets, which are well known benchmark datasets with 128×160 pixels size, namely KITTI (Geiger *et al*. 2013) and BDD100K (Yu *et al*. 2020). Although the KITTI and BDD100K datasets are irrelevant to crack images, their time-variant image features are effective to train the proposed PreCNet. The training process consisted of 10,000 epochs with a learning rate of 0.0005 for the first 9,900 epochs. Then, the learning rate was decreased to 0.0001 for the last 100 epochs. The filter size of the ConvLSTM and convolutional decoding layer was set to 3. The network was trained by minimizing the weighted sum of E_l in each predictive estimator, using the following loss function

$$L_{train} = \sum_{m=1}^{M} L_{seq}(m), \quad (7)$$

$$L_{seq}(j) = \sum_{t=1}^{T} \mu_t \sum_{l=1}^{N} \frac{\lambda_l}{n_l} \sum_{i=1}^{n_l} E_l^t(i), \quad (8)$$

where $L_{seq}(m)$ represents the loss of $m-$th single sequence. M is the number of sequences in the training dataset, and T is the length of each sequence. $N + 1$ is the number of predictive estimators, and n_l is the number of error units in the l-th predictive estimator. $E_l^t(i)$ denotes the error of the i-th unit in the predictive estimator l at time t. μ_t and λ_l are the time and predictive estimator weighting factors, respectively. The mini-batch gradient descent method was employed for the minimization process. In this study, we constructed the time-series I_t with T = 10, and total 20,000 I_t data were used for PreCNet training. During the training process,

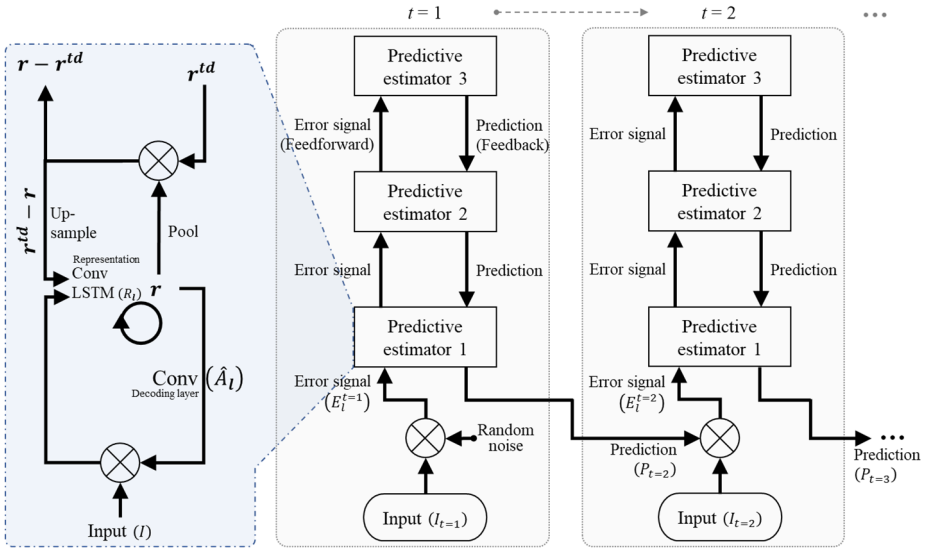


Fig. 6 Hierarchical structure and training procedure: $r, r^{td}, R, \hat{A}, E^t, P_t$ and I_t are the feedback prediction of current and representation state, pooled representation state, higher layer prediction, error, prediction and input data, respectively

$E_l^t(i)$ is sequentially updated.

For crack growth prediction tests, the predicted crack image ($P_{t=2}^B$) can be obtained in the first step of prediction with $I_{t=1}^B$ through the pre-trained PreCNet. Subsequently, the pre-trained PreCNet is updated for the crack growth prediction purpose by comparing the next $I_{t=2}^B$ with $P_{t=2}^B$. This network updating process is repeated while the last crack image ($I_{t=10}^B$) is used as input to the prediction test. In the last step, the final crack growth prediction image ($P_{t=11}^B$) is generated from PreCNet with updated crack growth features.

3. Experimental validation

The proposed deep ConvLSTM-based concrete growth prediction technique was experimentally validated using time-series crack growth digital images collected from a concrete mock-up specimen.

3.1 Experimental setup

Fig. 7 illustrates the experimental setup, which consisted of a reinforced concrete mock-up specimen, a digital camera, and a pressure machine. The dimensions of the mock-up specimen were 3×3.5×1.5 m^3 along the x, y, and z-directions. Bending experiments were conducted to study crack behavior, as they allowed for adjusting the focal length and ROI at each loading step, facilitating the validation of the stabilization procedure.

In the experiments, the concrete mock-up specimen was subjected to bending by the pressure machine, which pushed the specimen outward along the z-direction, as depicted in Fig. 7. The aim was to investigate the crack opening, and therefore, ROI containing surface cracks in the x- and y-directional were intentionally selected and

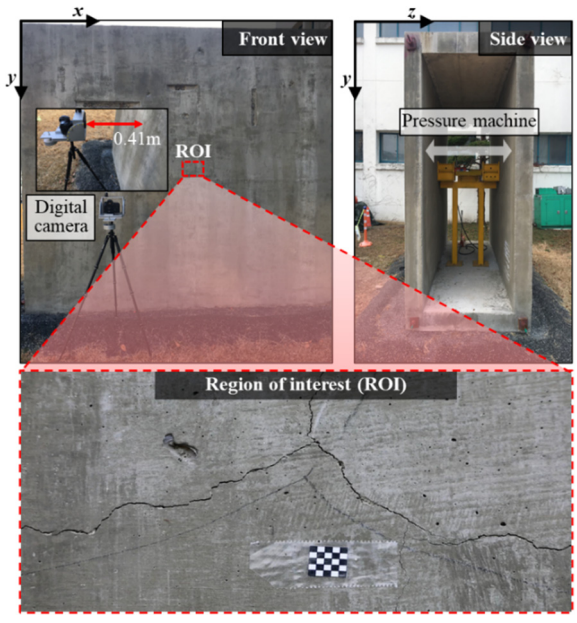
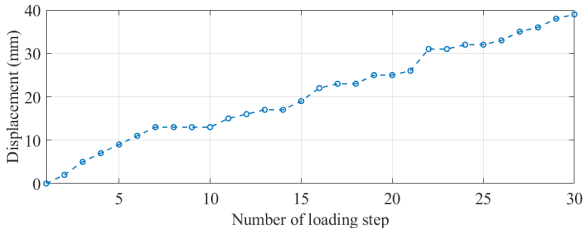
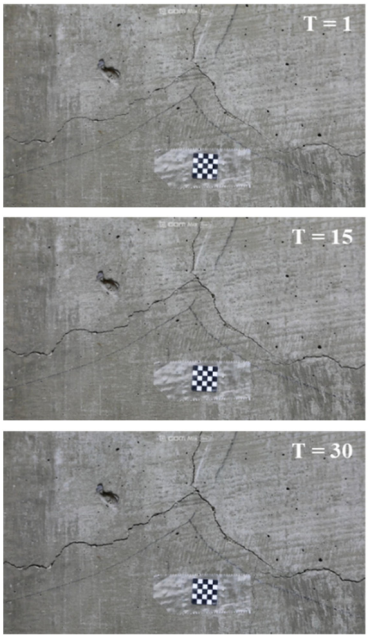


Fig. 7 Experimental setup with reinforced mock-up specimen, digital camera and pressure machine



(a) z-directional displacements corresponding to the number of loading step



(b) Representative crack growth at the 1, 15, and 30 loading steps

Fig. 8 Representative crack growth results and the corresponding z-directional displacements

examined. As the loading progressed, the selected crack width gradually increased, and a corresponding crack image was captured outside the specimen at each loading step.

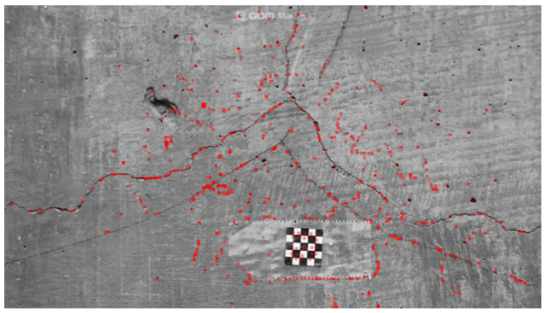
The digital camera used in the experiment was a Canon EOS 5D Mark 4 equipped with a 100 mm F 2.8 L macro IS USM lens. The camera settings were fixed at ISO 1000, F 16, and an exposure time of 0.01 seconds. The acquired digital images had a resolution of 1920×1080 pixels. The images were captured under outdoor conditions, with a working distance of 410 mm between the camera and the concrete mock-up specimen.

3.2 Experimental results

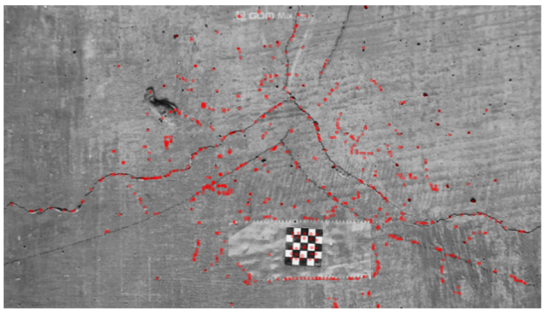
This section presents the experimental results in four steps. First, the acquisition of time-series digital crack images is demonstrated. Second, the acquired images are stabilized through feature detection, descriptor extraction, and matching. Next, deep learning-based crack detection and quantification are performed. Finally, the results of ConvLSTM-based crack growth prediction are provided.

3.2.1 Time-series digital image acquisition

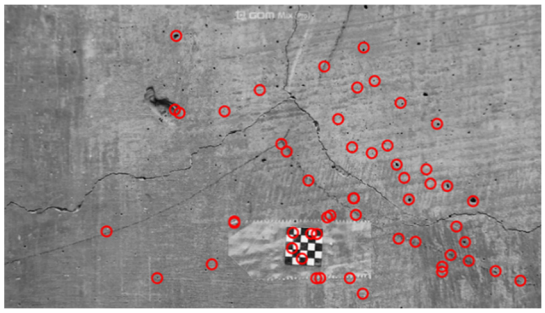
After capturing the digital images in the unloaded condition, the ROI containing cracks was intentionally



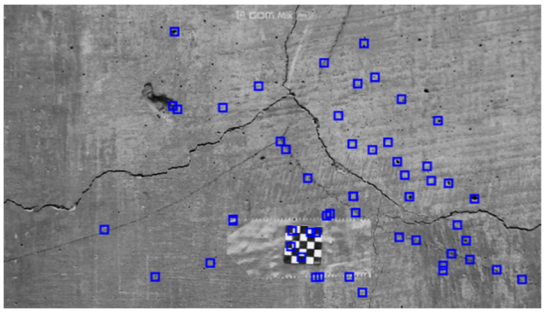
(a) Representative crack images at the loading step 1 with detected feature points



(b) Representative crack images at the loading step 30 with detected feature points

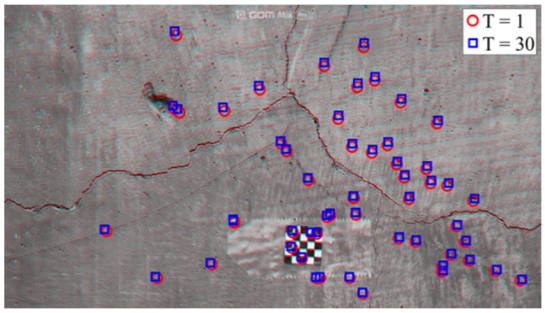


(c) The matched descriptors at the loading step 1

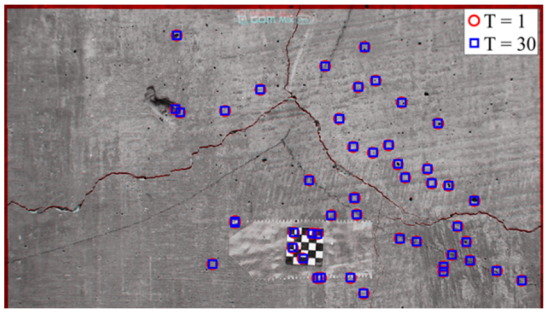


(d) The matched descriptors at the loading step 30

Fig. 9 Representative crack images obtained from the loading steps 1 and 30



(a) Descriptor matching results between the crack images at the loading steps 1 and 30



(b) V^S at the loading steps 1 and 30 using inlier descriptor matching

Fig. 10 Stabilization results

selected to investigate crack growth. The loading condition along the z-direction consisted of 30 loading steps, and the corresponding z-directional displacements are shown in Fig. 8(a). As the bending loads gradually increased and propagated, the assumption was made that the loads were compatible with the cracks in the ROI.

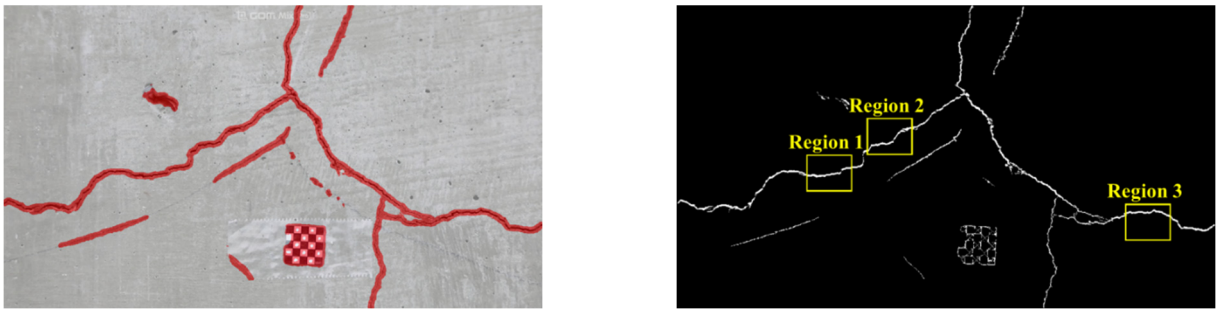
In the first step, no loading was applied, but a small crack was observed on the surface of the mock-up specimen due to environment effect, as shown in Fig. 8(b). This initial small crack did not pose a structural problem.

3.2.2 Stabilization of the time-series digital images

To achieve precise prediction of crack growth rate, it is necessary to stabilize all acquired crack growth images according to the loading step. Figs. 9(a) and (b) show representative crack images obtained from loading step 1 and 30, respectively, along with detected feature points. Each feature point was detected using Eq. (1) with a windowed patch size of 49. In the images, the detected feature points from crack image 1 and crack image 30 are marked as red circles and blue rectangles, respectively. The number of detected feature points were 766 and 852 for the two images.

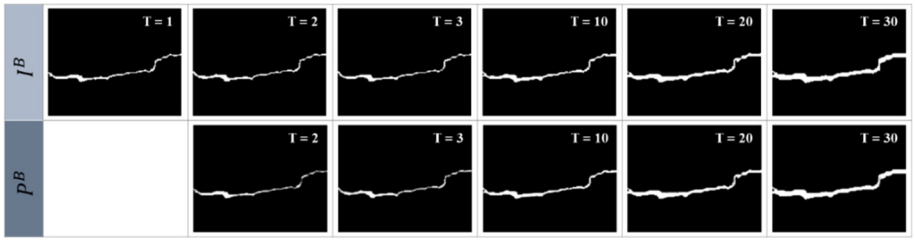
For matching the detected feature points, corresponding descriptors were extracted using the FREAK descriptor with Eqs. (2) and (3). Figs. 9(c) and (d) show the matched descriptors of the crack images, with only 54 matching descriptors marked as the red circles and blue rectangles, respectively.

Fig. 10(a) illustrates the crack image obtained from the loading step 1 overlaid with the reprojected crack image obtained from the loading step 30. However, once the descriptors were matched between the two images, specific outliers were eliminated using the MSAC algorithm. As a result, a total of 48 inlier descriptors remained to stabilize the timely-adjacent crack images. Fig. 10(b) displays V^S

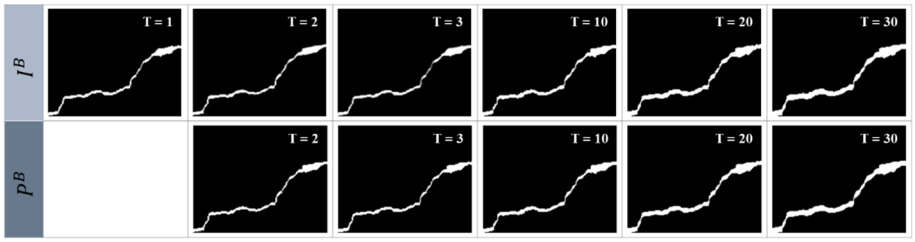


(a) Representative V^D (b) I^B with selected crack regions

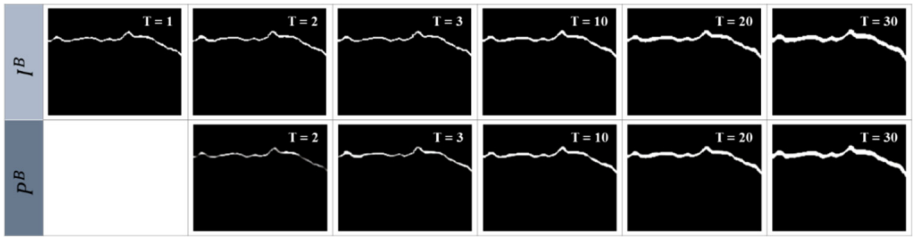
Fig. 11 Representative crack detection results using the deep learning-based crack detection network



(a) Region 1



(b) Region 2



(c) Region 3

Fig. 12 Representative crack growth prediction results with the corresponding ground truth images

along with the inlier descriptors. This stabilization procedure achieves precise alignment between the crack images produced by the different loading steps, and the results are acceptable for the subsequent crack detection and quantification.

3.2.3 Deep learning-based crack detection and quantification

After stabilizing the timely-adjacent crack images, the next step is to perform deep learning-based crack detection and quantification. Fig. 11(a) shows the representative V^D result using the deep learning-based crack detection network. This process allows for the extraction of crack information while eliminating surrounding environmental effects such as markings, illumination, and luminance.

Subsequently, statistical denoising process explained in Section 2.3 was sequentially performed for obtaining I^B shown in Fig. 11(b). To evaluate the crack width of I^B, Eqs. 5 and 6 were used, and the representative quantification results of Region 1 was listed in Table 1. It can be confirmed that the crack quantification results at each loading step is closely matched with the ground truth obtained by direct measurement.

Table 1 Crack quantification results on Region 1 (unit: mm)

Loading steps	0	10	20	30
Ground truth	0.055	0.063	0.084	0.106
Quantification	0.054	0.062	0.082	0.103

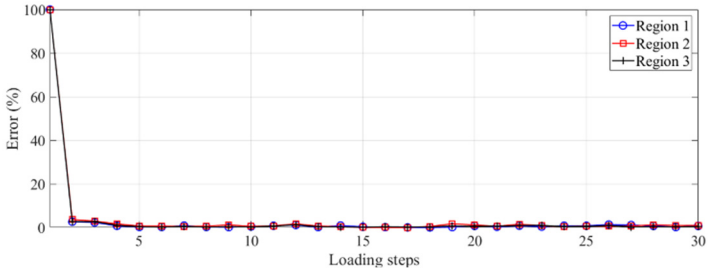
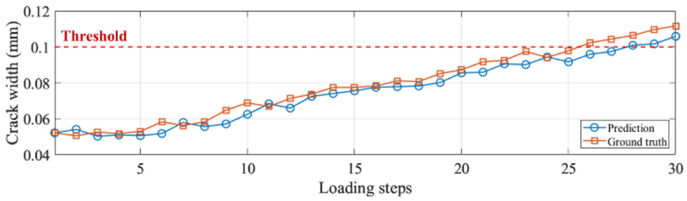
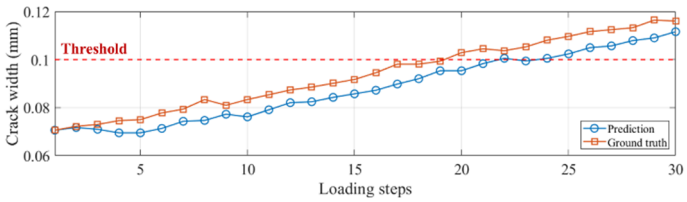


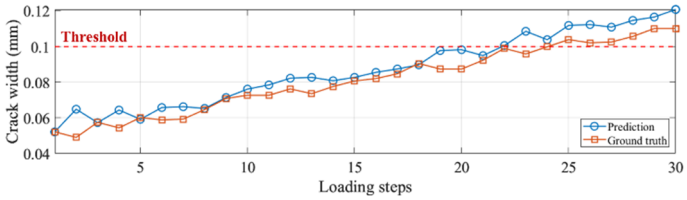
Fig. 13 E^t of the representative crack regions in accordance with every step updating



(a) Region 1



(b) Region 2



(c) Region 3

Fig. 14 Crack width quantification results of ground truth and prediction at Regions 1, 2 and 3

3.2.4 ConvLSTM-based crack growth prediction

For crack growth prediction, the representative crack regions denoted by Regions 1, 2 and 3 in Fig. 11(b) were selected. In ConvLSTM-based crack growth prediction, I_t^B was used as the input for the proposed PreCNet, as explained in Section 2.4. When $I_{t=1}^B$ is used as the first input, PreCNet produces the next frame prediction $P_{t=2}^B$. Then, the $I_{t=2}^B$ is used for updating the network by comparing with $P_{t=2}^B$. By repeating the updating procedure, the prediction results at each time step with respect to the representative crack regions were obtained, as shown in Fig. 12. It clearly shows that the crack widths of P^B gradually increase accordance with each loading step increment. By comparing with I^B, it can be observed that P^B in all the representative regions looks matched with I^B well.

E^t results corresponding to the sequential update steps are shown in Fig. 13, demonstrating that E^t gradually converges as the update progresses. It is importantly noted that the prediction accuracy can be improved by the sequential updating over time.

To quantitatively evaluate the performance of the predictive network, the crack widths of P^B and the corresponding I^B ones are quantified and compared in Fig. 14. Based on these quantification results on each region, the crack prediction accuracies with respect to Regions 1, 2, and 3 are calculated as 95.2 %, 94.1 %, and 92.6 %, respectively. Although the crack widths are under- or over-predicted in some loading steps, the overall crack growth tendency is well tracked in all tested regions. In addition, the loading step (or time) estimation performance corresponding to 100 μm crack width was tested to quantify the time prediction accuracy. Here, crack width of 100 μm is one of bridge crack management standards in South Korea based on the Act on the Safety Control and Maintenance of Establishments law (KLRI), which is denoted by threshold in Figure 14. In case Region 1, the crack width of P^B reaches the threshold at the loading step 28, while I^B over crosses the threshold right after the step 25, as shown in Fig. 14(a). Similarly, the proposed PreCNet predicts the maintenance time 3 step ahead, as displayed in Fig. 14(b). In contrast, 2 loading step late prediction results can be seen in Region 3, as shown in Fig. 14(c). These

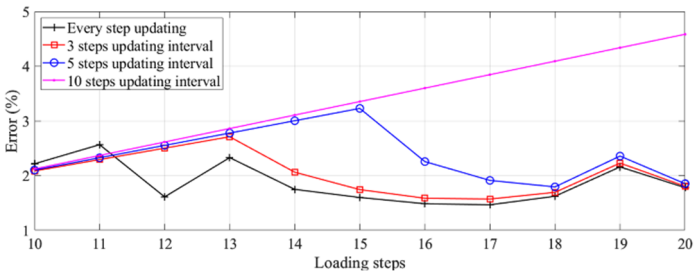


Fig. 15 Comparison of E^t at Region 1 among every 3, 5, 10 steps interval updating

maintenance time prediction results imply that the proposed technique can contribute predictive maintenance planning.

4. Discussion

The discussion focuses on the importance of updating errors for achieving accurate crack growth prediction results. The prediction performance of the proposed PreCNet is highly depending on the training dataset. However, the main advantage of PreCNet is that only the sequential-varying feature of the time-series training images are trained rather than the static training images themselves. Thus, the average prediction accuracy of around 94% was achieved in the experimental study, even though we used the KITTI and BDD100K datasets which are irrelevant to crack images. However, since time-dependent crack growth pattern has own characteristics which may not perfectly matched with the KITTI and BDD100K datasets, actual cumulative time-series crack growth dataset would be helpful to enhance the prediction performance.

Nevertheless, the discrepancy between the training dataset and test data can be minimized by employing the updating procedure. Fig. 15 illustrates the representative crack growth prediction errors according to different updating step intervals of Region 1. The black cross line represents E^t of every step prediction updating. The red rectangular, blue circular, and purple dot lines represent E^t with updating intervals of 3, 5, and 10 steps, respectively. Here, E^t results are compared from the 10^{th} step after E^t is converged. It can be observed that as the number of updating interval increases, E^t also increases. This suggests that more frequent updates of the predictive network can lead to higher prediction accuracy. The number of sequential crack growth images plays a crucial role in learning and updating the predictive network. In other words, shorter updating periods contribute to improved prediction accuracy.

This finding highlights the importance of continuously updating the predictive network with the error of inputted time-series crack images. By incorporating more recent crack growth information into the network, it becomes better equipped to make accurate prediction. Therefore, it is recommended to update the network at regular intervals to ensure optimal prediction performance.

5. Conclusions

This study presents a crack growth prediction technique for concrete structure aimed at predictive maintenance. The proposed technique consists of four steps: time-series crack image acquisition, target image stabilization, deep learning-based crack detection and quantification, and crack growth prediction. The key advantage of this technique is its reliance on data-driven time-series crack propagation digital images, allowing for accurate crack growth prediction without the need for explicitly defined features such as stress, length, or orientation of the crack. Experimental validation of the proposed technique was conducted using a concrete mock-up specimen subjected to step-wise bending loads to induce sequential crack growth. The results demonstrate a high accuracy of approximately 95% when compared to the ground truth. This accuracy level enables the establishment of a structural maintenance plan based on the predicted crack growth, in accordance with regular safety inspection criteria.

Since actual time-dependent crack growth pattern would be helpful to enhance the prediction performance, larger dataset of in-situ crack growth images is being cumulated from various concrete structures such as bridge, dam and building. Additionally, the proposed technique has the potential to be extended beyond crack growth prediction and applied to various types of damage assessment in the future.

Overall, this crack growth prediction technique holds promise for enabling effective predictive maintenance strategies for concrete structures, contributing to improved structural safety and longevity.

Acknowledgments

This work was supported by the faculty research fund of Sejong University in 2024 (20240311).

References

Alahi, A., Ortiz, R. and Vandergheynst, P. (2012), “FREAK: fast retina keypoint”, In: *2012 IEEE Conference on Computer Vision and Pattern Recognition*, Providence, RI, USA, June.

Bae, H. and An, Y.K. (2024), “Computer vision-based statistical crack quantification for concrete structures”, *Meas.*, **211**, 112632. https://doi.org/10.1016/j.measurement.2023.112632

Bae, H., Jang, K. and An, Y.K. (2021), “Deep super resolution

crack network (SrcNet) for improving computer vision-based automated crack detectability in in situ bridges", *Struct. Health Monit.*, **20**(4), 1428-1442. https://doi.org/10.1177/147592172091722

Chang, H., Shen, M., Yang, X. and Hou, J.X. (2020), "Uncertainty modeling of fatigue crack growth and probabilistic life prediction for welded joints of nuclear stainless steel", *Materials*, **13**, 3192. https://doi.org/10.3390/ma13143192

Ding, W., Yang, H., Yu, K. and Shu, J.P. (2023), "Crack detection and quantification for concrete structure using UAV and transformer", *Automat. Constr.*, **152**, 104929. https://doi.org/10.1016/j.autcon.2023.104929

Geiger, A., Lenz, P., Stiller, C. and Urtasun, R. (2013), "Vision meets robotics: the KITTI dataset", *Int. J. Robot Res.*, **32**, 1231-1237. https://doi.org/10.1177/0278364913491297

Griffith, A.A. (1921), "The phenomena of rupture and flow in solids", *Philos. T. R. Soc. Lond.*, **221**, 163-198. https://doi.org/10.1098/rsta.1921.0006

Harris, C. and Stephens, M. (1988), "A combined corner and edge detector", *Proceedings of the 4th Alvey Vision Conference*, Manchester, UK, August.

Holt, E. and Leivo, M. (2004), "Cracking risks associated with early age shrinkage", *Cement Concrete Compos.*, **26**, 521-530. https://doi.org/10.1016/S0958-9465(03)00068-4

Ioffe, S. and Szegedy, C. (2015), "Batch normalization: accelerating deep network training by reducing internal covariate shift", *Proceedings of the 32nd International Conference on Machine Learning*, Lille, France, July.

Ismail, M., Muhammad, B. and Ismail, M.E. (2010), "Compressive strength loss and reinforcement degradations of reinforced concrete structure due to long-term exposure", *Const. Build Mater.*, **24**, 898-902. https://doi.org/10.1016/j.conbuildmat.2009.12.003

Jang, K., An, Y.K., Kim, B. and Cho, S. (2021), "Automated crack evaluation of a high-rise bridge pier using a ring-type climbing robot", *Comput.-Aided Civ. Inf.*, **36**, 14-29. https://doi.org/10.1111/mice.12550

Jang, K., Jung, H. and An, Y.K. (2022), "Automated bridge crack evaluation through deep super resolution network-based hybrid image matching", *Automat. Constr.*, **137**, 104229. https://doi.org/10.1016/j.autcon.2022.104229

Jang, K., Park, S., Jung, H., Yoo, H. and An, Y.K. (2024), "Deep learning-based 3D digital damage map of vertical-type tunnels using unmanned fusion data scanning", *Comput.-Aided Civ. Inf.* **162**, 105397. https://doi.org/10.1016/j.autcon.2024.105397

Koh, E., Jin, S.S. and Kim, R.E. (2022), "Physical interpretation of concrete crack images from feature estimation and classification", *Smart Struct. Syst., Int. J.*, **30**(4), 385-395. https://doi.org/10.12989/sss.2022.30.4.385

Lai, J., Cai, J., Chen, Q.J., He, A. and Wei, M.Y. (2020), "Influence of crack width on chloride penetration in concrete subjected to alternating wetting-drying cycles", *Mater.*, **13**(17), 3801. https://doi.org/10.3390/ma13173801

Lam, L., Lee, S.W. and Suen, C.Y. (1992), "Thinning methodologies-a comprehensive survey", *IEEE T. Pattern Anal.*, **14**, 869-885. https://doi.org/10.1109/34.161346

Nair, V. and Hinton, G.E. (2010), "Rectified linear units improve restricted Boltzmann machines", *Proceedings of the 27th International Conference on Machine Learning*, Haifa, Israel.

Ni, F.T., Zhang, J. and Chen, Z.Q. (2019), "Pixel-level crack delineation in images with convolutional feature fusion", *Struct. Control Health Monit.*, **26**(1), e2286. https://doi.org/10.1002/stc.2286

Otsu, N. (1979), "A threshold selection method from gray-level histograms", *IEEE T. Syst. Man Cyb.*, **9**, 62-66. https://doi.org/10.1109/TSMC.1979.4310076

Paris, P. and Erdogan F. (1963), "A critical analysis of crack propagation laws", *J. Fluid Eng.*, **85**, 528-533. https://doi.org/10.1115/1.3656900

Pierson, K., Rahman, A. and Spear, A.D. (2019), "Predicting microstructure-sensitive fatigue crack path in 3D using machine learning framework", *Jom.*, **71**(8), 2680-2694. https://doi.org/10.1007/s11837-019-03572-y

Qi, Y., Yuan, C., Kong, Q., Xiong, B. and Li, P. (2021), "A deep learning-based vision enhancement method for UAV assisted visual inspection of concrete cracks", *Smart Struct. Syst., Int. J.*, **27**(6), 1031-1040. https://doi.org/10.12989/sss.2021.27.6.1031

Raja, A., Chukka, S.T. and Jayaganthan, R. (2020), "Prediction of fatigue crack growth behaviour in ultrafine grained Al 2014 alloy using machine learning", *Metals*, **10**(10), 1349. https://doi.org/10.3390/met10101349

Rao, R.P.N. and Ballard, D.H. (1999), "Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects", *Nat. Neurosci.*, **2**, 79-87. https://doi.org/10.1038/4580

Reagan, D., Sabato, A. and Niezrecki, C. (2018), "Feasibility of using digital image correlation for unmanned aerial vehicle structural health monitoring of bridges", *Struct. Health Monit.*, **17**(5), 1056-1072. https://doi.org/10.1177/1475921717735326

Rovinelli, A., Sangid, M.D., Proudhon, H. and Ludwig, W. (2018), "Using machine learning and a data-driven approach to identify the small fatigue crack driving force in polycrystalline materials", *Npj. Comp. Mater.*, **4**(35). https://doi.org/10.1038/s41524-018-0094-7

Safiuddin, M., Kaish, A.B.M., Woon, C.O. and Raman, S.N. (2018), "Early-age cracking in concrete: causes, consequences, remedial measures, and recommendations", *Appl. Sci.*, **8**, 1730. https://doi.org/10.3390/app8101730

Special act on the safety control and maintenance of establishments, Land, Infrastructure and Transport, *KLRI*.

Straka, Z., Svoboda, T. and Hoffmann, M. (2023), "PreCNet: next-frame video prediction based on predictive coding", *IEEE T. Neur. Net. Lear.*, **2023**, 1-15. https://doi.org/10.1109/TNNLS.2023.3240857

Torr, P.H.S. and Zisserman, A. (2000), "MLEASC: a new robust estimator with application to estimating image geometry", *Comp. Vis. Image Und.*, **78**, 138-156. https://doi.org/10.1006/cviu.1999.0832

Wang, X.Y. (2018), "Effects of crack and climate change on service life of concrete subjected to carbonation", *Appl. Sci.*, **8**, 572. https://doi.org/10.3390/app8040572

Wang, B.X., Zhao, W.G., Du, Y.L., Zhang, G.Y. and Yang, Y. (2016), "Prediction of fatigue stress concentration factor using extreme learning machine", *Comp. Mater. Sci.*, **125**, 136-145. https://doi.org/10.1016/j.commatsci.2016.08.035

Wu, P., Liu, A., Fu, J., Ye, X. and Zhao, Y. (2022), "Autonomous surface crack identification of concrete structures based on an improved one-stage object detection algorithm", *Eng. Strut.*, **272**, 114962. https://doi.org/10.1016/j.engstruct.2022.114962

Xu, Y., Fan, Y.L. and Li, H. (2023), "Lightweight semantic segmentation of complex structural damage recognition for actual bridges", *Struct. Health Monit.*, **22**(5). https://doi.org/10.1177/14759217221147015

Yu, F., Chen, H.F., Wang, X., Wian, W.Q., Chen, Y.Y., Liu, F.C., Madhavan, V. and Darrell, T. (2020), "BDD100K: a diverse driving dataset for heterogeneous multitask learning", In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Seattle, WA, USA, June.

Yu, S., Yan, C., Liu, C.Y. and Pu, J.P. (2023), "Fatigue life evaluation of parallel steel-wire cables under the combined actions of corrosion and traffic load", *Struct. Control Health Monit.*, **2023**, https://doi.org/10.1155/2023/5806751