Tunnelling and Underground Space Technology 124 (2022) 104472

Contents lists available at ScienceDirect

Tunnelling and Underground Space Technology incorporating Trenchless Technology Research

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

Automatic tunnel lining crack evaluation and measurement using deep learning

L. Minh Dang [a,c,1], Hanxiang Wang [a,1], Yanfen Li [a], Yesul Park [a], Chanmi Oh [a], Tan N. Nguyen [b], Hyeonjoon Moon [a,*]

[a] Department of Computer Science and Engineering, Sejong University, 209 Neungdong-ro, Gwangjin-gu, Seoul 05006, Republic of Korea
[b] Department of Architectural Engineering, Sejong University, 209 Neungdong-ro, Gwangjin-gu, Seoul 05006, Republic of Korea
[c] Department of Information Technology, FPT University, Ho Chi Minh city 70000, Viet Nam

ARTICLE INFO

Keywords:
Tunnel
Lining crack
Deep learning
U-Net
Segmentation
Measuring

ABSTRACT

A tunnel is an imperative underground passageway that supports fast and uninterrupted transportation. Over time, various factors, such as ageing, topographical changes, and excessive force, slowly affect the tunnel's internal structure, which causes tunnel defects that can reduce the structure's stability and eventually lead to enormous damage. Therefore, the tunnels need to be checked regularly to detect and fix the cracks promptly. Earlier inspection approaches mainly relied on the operators who directly observed videos to detect the cracks and determine their seriousness, which is laborious, error-prone, and tedious. This research suggests a deep learning-based tunnel lining crack segmentation framework for tunnel images taken by high-resolution cameras. The primary contributions are (1) a lining crack segmentation framework, which is motivated by U-Net architecture, where the encoder is replaced by a ResNet-152 model, (2) the automated measurement of the segmented cracks, which include length, thickness, and type, and (3) a huge lining crack segmentation database. The experimental results showed that the framework obtained comparable performance compared to existing crack segmentation models and supported the automated measurement of the segmented cracks.

1. Introduction

Tunnels are a fundamental public infrastructure that ensures essential civilian or military services and stimulates the economic growth of every community (Huang et al., 2021). Their status has gradually changed from the newly built structure to mainly maintenance in many developed countries. A delay in tunnel maintenance can cause sudden structural and functional failure, which demands a huge maintenance budget and potentially causes unwanted accidents (Attard et al., 2018). As a result, extensive and timely tunnel inspection is required to ensure the tunnel's structural stability and fundamental functionality through the operational time.

Current crack evaluation systems primarily focused on human visual inspection to ensure the tunnel's structural and functional requirements. However, those approaches hold severe shortcomings (Xu & Yang, 2020), such as prejudice in the inspector's decision, high labor cost, high time-consumption, and shortened operational uptime of the tunnel. To overcome such challenges, other non-destructive approaches, including visual imaging (Kaise et al., 2020), ultrasound tomography (Zhan et al., 2020), acoustic emission (Wang et al., 2020), and ground-penetrating radar (Liu et al., 2021), have been increasingly implemented for the tunnel inspection. These approaches usually demonstrate low performance and still mainly rely on human resources.

Recently, a new generation of scanning systems and robots equipped with high-resolution image acquisition devices and vision-sensing technologies has been increasingly used in order to provide a high-quality inspection (Loupos et al., 2018). The huge volume of multimedia data motivates the development of automatic inspection systems that aid the tunnel investigation and deliver a reliable structural evaluation (Y. Liu et al., 2019; Menendez et al., 2018). For example, a self-developed moving tunnel inspection robot equipped with multiple line-scan cameras to synchronously obtain raw data of the tunnel surface (Zhao et al., 2021; Zhao et al., 2020). An advanced multi-degree-of-freedom (multi-DOF) robot includes a moving vehicle, an extending crane, and a high-accuracy robot arm was made recently (Menendez et al., 2018). In addition, many ultrasonic sensors and a 3D vision system

* Corresponding author.
E-mail address: hmoon@sejong.ac.kr (H. Moon).
[1] These authors contributed equally to this work.

https://doi.org/10.1016/j.tust.2022.104472
Received 8 April 2021; Received in revised form 4 March 2022; Accepted 5 March 2022

are put on the front of the robot to identify various defect features efficiently.

Lining cracks have a major role in tunnel safety because they are an early indicator that reveals the structure load conditions (Lei et al., 2021; N. Zhang et al., 2019). They can expose the concrete's inner material to environmental factors, such as water and air, which lead to further degradation and corrosion of the structure (Dang et al., 2022). Moreover, harsh noise and uneven lighting that frequently happens during the data collection process can also significantly influence the performance of automatic tunnel inspection systems (Attard et al., 2018).

Most traditional tunnel crack detection methods are based on edge detection, which identifies major local intensity shifts on the boundary of two distinct regions in an image. Standard edge detection algorithms are Canny edge detector and Laplacian of the Gaussian (Wang et al., 2018). Another common approach is the threshold segmentation method, which is well-known for its simplicity (Kamaliardakani et al., 2016). For instance, a tunnel-lining crack segmentation framework, which utilized edge detection, adaptive partitioning, and the OTSU thresholding to obtain a more reliable segmentation performance, was proposed recently (Attard et al., 2018). The introduction of a customized denoising method allowed the model to achieve superior segmentation results with low error rates. However, the conventional methods can only recognize a series of unconnected crack edges that are required to be linked together using some post-processing techniques to extract the complete crack edges and often struggle to find the cracks in the high-clutter and low-contrast images.

In the past few years, deep learning has offered remarkable performances on crack segmentation applications, which can automatically learn distinctive features from the training data (Minh et al., 2021). Through the suggestion of the fully convolutional networks (FCN) (Long et al., 2015) and the U-Net architecture (Ronneberger et al., 2015), it becomes feasible to obtain the spatial features using the transposed convolution operation, which is the core concept in image segmentation models. For example, a defect segmentation of shield tunneling based on the fully convolutional network (FCN) was trained for a huge number of iterations to perform the forward and backward learning process efficiently (Huang et al., 2018). The model demonstrated state-of-the-art crack segmentation performance with a low error rate. In other research, a real-time deep learning-based crack segmentation framework (SDDNet) that efficiently dealt with challenging problems, such as crack-like objects and complicated backgrounds, was proposed (Choi & Cha, 2019). The model contained regular convolutional layers, dense layers, a customized atrous spatial pyramid pooling (ASPP) block, and a decoder block, which obtained the average mean intersection-over-union (mIoU) of over 0.8 on the test set. Moisture marks, formed by cracks inside the tunnel, are viewed as one of the most critical structural issues for the tunnel. As a result, a Mask R-CNN-based instance segmentation system was proposed to recognize them automatically (Zhao et al., 2020). The authors revealed that the IoU, F1 score, and accuracy of the model were better than previous state-of-the-art structures, such as FCN and region growing algorithm (RGA). Finally, the path aggregation network (PANet) was recently improved to perform tunnel crack segmentation effectively (Zhao et al., 2021). The authors integrated an A* algorithm into the PANet to perform crack segmentation and quantification in a single step. The experimental results proved that the proposed model was superior at reducing crack disjoint issues and skeletonization error. The listed approaches each have particular strengths and weaknesses. The main advantage is that the features are extracted automatically as part of the learning process without any handcrafted parameters. However, those models demand a massive dataset and a tedious labeling process that limits adaptation in existing systems. In addition, they wrongly segment a complete lining crack as a series of disconnected cracks during the testing process, which is referred to as the *disjoint* problem (Xu & Yang, 2020).

Some previous studies have tried to compute the segmented crack's length and thickness in order to offer the engineers with more useful statistics (Wang et al., 2021). For example, an efficient approach was introduced to measure the cracks to get the width and thickness properties (Yang et al., 2018). The experimental results showed that the outputs are correct for simple cracks, but the crack length was wrongly miscalculated when the segmented crack had the *disjoint* problem. Still, other important crack properties, such as thickness, type, and area, were ignored.

By pointing out the weakness of previous related work, there is a pressing need to develop a robust lining crack segmentation framework that can segment the tunnel crack with high accuracy as well as support the analysis of the crack's essential features, such as length, thickness, and type. Motivated by recent successes of deep learning on various applications (Dang et al., 2018; Hassan et al., 2019), this study offers a customized encoder-decoder-based structure based on the U-Net structure to effectively segment tunnel cracks using the crack's abstract features. After that, various post-processing methods are introduced to significantly reduce the noise in the segmented images. Finally, additional information about the segmented crack, such as crack length, thickness, and crack type, is extracted. The main objectives of this paper include:.

1. An introduction of a huge tunnel lining crack segmentation dataset with about 170,000 crack images and 170,000 mask images.
2. The proposed model replaces the U-Net encoder with a ResNet model, which significantly improved its performance compared to the U-Net in reducing noise.
3. Several post-processing methods were introduced to reduce noise generated during the segmentation process.
4. The extraction of crucial knowledge related to the segmented cracks, such as crack length and thickness.
5. A method to determine the lining crack type based on the crack intersection.

The work is divided as follows. Section 2 explain the creation of the lining crack segmentation dataset. Next, the crack segmentation framework based on the U-Net structure is introduced in Section 3. Moreover, the post-processing process and crack measurements are also described. In Section 4, various experiments are performed to assess the proposed framework's performance and robustness on both the crack segmentation and the crack measurements. Section 6 then summarizes the research by discussing the proposed model's strengths and weaknesses and finally showing the future work.

2. Tunnel lining crack dataset

2.1. Dataset collection

Initially, a Tunnel Deep Scanner truck was controlled to collect images of 5 different tunnels across Korea, which include Hwasan, Deugseong, Gamcheon, Banggyo, and Bongsan from January 11th, 2020, to August 27th, 2020. After the data collection process, 6810 high-resolution images were collected.

The inspection speed of the truck is kept stable at approximately 3–5 km/h to keep a high inspection precision. The truck was equipped with night 4 K line-scan cameras (4000 × 3000) that use the charge-coupled device (CCD) sensors to ensure that high-quality images are captured with low noise (Song et al., 2019). The settings of the line-scan cameras guarantees that all images are collected orthogonal to the lining surface. In addition, fifteen LED lights (100 W ~ 300 W) are equipped to offer enough lighting to ensure the coherency of images (Nguyen et al.). The truck's inspection range is about 180 degrees excluding the bottom, and the filming distance is kept stable at approximately 2 m from the tunnel surface.

2.2. Data stitching and data labeling

After the data collection step, an image stitching algorithm was

implemented to thoroughly inspect the tunnel lining cracks. The image stitching process involves some basic steps. Firstly, keypoints were detected using keypoint detectors, such as Difference-of-Gaussian (DoG) or Harris. Local invariant descriptors, such as scale-invariant feature transform (SIFT) or speeded-up robust features (SURF), were then computed from the input images. Secondly, the descriptors between the images were matched. Next, a homography estimation method was applied using the matched keypoints to estimate a homography matrix. Finally, a warping transformation algorithm was applied to process the obtained homography matrix to create the final stitched image.

In this study, a mosaic algorithm proposed by (Brown & Lowe, 2007) was utilized to stitch the collected tunnel images. It addressed the main weaknesses of the previous image stitching approaches and produced high-quality stitched images by using the invariant local features that were insensitive to image order, orientation, and illumination changes. The SIFT algorithm was first implemented to extract the invariant local features according to the description from the paper. The feature matching phase was followed, which detected and extracted a set of images with over 6 matches between them. A random sample consensus (RANSAC) was then performed to compute the homography, and finally, gain compensation and image blending were implemented. After the image stitching process, a total of 663 stitched images were generated, with the image sizes vary from 10441×2910 to50739×3140, as described in Table 1.

A total of 5 experts from a deep inspection company ("Deep inspection, https://www.deepinspection.ai/," 2021) were assigned a one-month labeling task, and averagely, 5 images were labeled individually per day. The labeling tool is a computer vision-aided crack annotation tool developed using Python and PyQt5, which implemented edge detection and morphological techniques to identify the cracks automatically. The experts then manually evaluated and corrected the predictions with all the lining cracks highlighted in red. At the end of this process, the original dataset includes 663 original images (Fig. 1(a)) and 663 corresponding labeled images (Fig. 1(b)) were created.

2.3. Segmentation dataset description

Fig. 1 visualizes the process of creating the segmentation dataset from the stitched database. The original stitched image and the corresponding labeled image with the lining cracks highlighted in red are displayed in Fig. 1(a) and Fig. 1(b), respectively. After that, a color-based thresholding approach was applied to Fig. 1(b) to produce a binary image, as displayed in Fig. 1(c), where the crack's regions were assigned the pixel value of 255, and the background areas were assigned the pixel value of 0 in order to minimize the influence of the color channel during the training process. The generated binary images were huge, with the image size ranging from 10441×2910 to50739×3140. Therefore, a sliding window with the window size of 448×448 and a stride ratio of 0.3 was used to slide over the original stitched image (Fig. 1(a)) to obtain a list of original images and the generated binary image (Fig. 1(c)) to extract the corresponding mask images. The windows that contained only the background pixel were removed automatically.

Table 1
Description of the proposed tunnel crack segmentation dataset including original dataset (original images labeled images) and segmentation dataset (crack images, mask images).

Location	Date	Original dataset		Segmentation dataset	
		Original	Labeled	Crack	Mask
Hwasan	January 11st, 2020	8	8	1529	1529
	January 17th, 2020	50	50	7763	7763
	March 04th, 2020	26	26	7124	7124
Deugseong	March 10th, 2020	50	50	7182	7182
	March 27th, 2020	37	37	6151	6151
	April 10th, 2020	33	33	3895	3895
	April 30th, 2020	30	30	3241	3241
Gamcheon	May 01st, 2020	40	40	6792	6792
	May 08th, 2020	72	72	10,763	10,763
Bongsan	May 16th, 2020	56	56	6234	6234
	June 20th, 2020	79	79	21,193	21,193
	June 25th, 2020	40	40	16,934	16,934
Banggyo	July 12th, 2020	23	23	16,545	16,545
	July 18th, 2020	20	20	12,349	12,349
	July 26th, 2020	36	36	7954	7954
	August 23rd, 2020	30	30	6770	6770
	August 27th, 2020	33	33	27,920	27,920
Total		663	663	170,339	170,339

After completing the operation as mentioned above, a collection of 170,339 original images and 170,339 corresponding mask images were obtained. The number of images for each tunnel is described in Table 1. The holdout approach used to divide the dataset into training/testing data is as follows. 80% of the data (136,271 images) were randomly picked as the training set, whereas 20% of the data (34,068 images) were used as the testing set to assess the model's robustness. The training dataset was further divided into the training set (100,204 images) to train the model and the validation set (34,067 images) to fine-tune the hyper-parameters.

3. Methodology

Fig. 2 describes the main processes of the tunnel crack segmentation framework, which includes three main steps. 1) Initially, a Tunnel Deep Scanner truck is controlled to take pictures of different tunnels across South Korea. After that, the stitched images are generated using the collected images. Next, the cracks in the stitched images are manually highlighted by experts. A lining crack segmentation dataset for the tunnel is produced at the end of the dataset preparation phase. 2) A customized deep learning model, which is based on the U-Net and ResNet architecture, is then proposed to perform the lining crack segmentation. 3) Several post-processing methods are implemented to reduce potential noise from the predicted images. Finally, a skeletonization operation is applied to facilitate the analysis of the crack length, the crack thickness, and the crack type.

The crack segmentation model is discussed in Section 3.1. The post-processing process is then analyzed in Section 3.2. Finally, Section 3.3 introduces the crack measurements process.

3.1. Deep learning-based crack segmentation

Inspired by the remarkable performances of the U-Net architecture (Ronneberger et al., 2015) and the ResNet model (He et al., 2016), this work introduces a novel TunnelURes model in order to recognize the tunnel cracks with high performance and efficiency. The U-Net model was introduced to segment the cells, so the transfer learning process did not work well on the tunnel crack segmentation topic. Moreover, it was undiscovered whether using such a deep segmentation architecture effectively recognizes the lining cracks at a pixel level. As a result, it was trained from scratch using the collected lining crack segmentation dataset. In addition, the encoder of the original U-Net model was replaced by the pre-trained ResNet-152 model on the ImageNet dataset with the two layers at the head removed (pooling layer and softmax classifier) in order to enrich the feature extraction process.

A series of RGB images with the image size of 448×448 is fed into the TunnelURes model. They then go through a series of layers, and the decoded output is the 448×448 binary images, where each pixel correlates with a two-dimensional vector showing the likelihood that a pixel is a crack or a background. The vertical rectangles depict various intermediate layers, whereas the arrows indicate common deep learning operations, such as max-pooling and copy. The main role of each layer is described below.

- Conv# symbolizes the convolutional layer.
- Max-pooling means the Max-pooling layer.

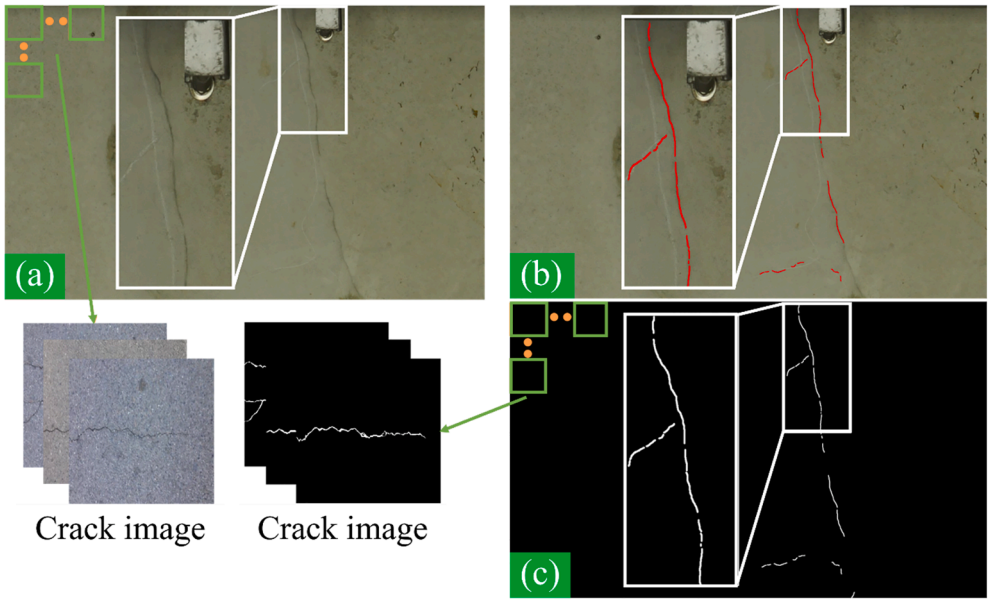

Fig. 1. Step-by-step explanation of the dataset creation process. Note: There are three main components: (a) stitched RGB images, (b) labeled images, containing cracks highlighted in red, and (c) binary images, which show the cracks in white pixels and the background in black pixels. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

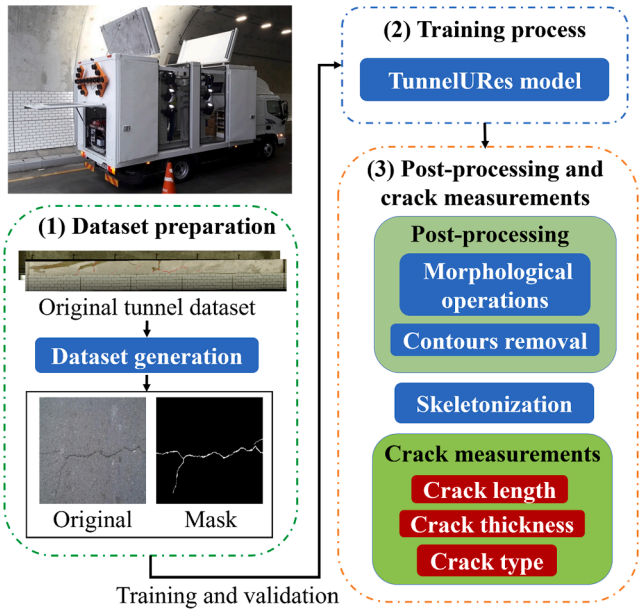

Fig. 2. Overall architecture of the tunnel lining crack segmentation framework. Note: There are three main modules, including data preparation, training process, and post-processing & crack measurements.

- Concatenate indicates that two layers are combined at the channel level as one layer.

The ResNet-152 encoder starts with a convolutional layer, which has the 7×7 kernel size and stride of 2. The first layer was then followed by a batch normalization (BN) layer (Santurkar et al., 2018), a rectified linear unit (ReLU) layer, and a max-pooling layer. Next, the output features go through a series of 3, 8, 36, and 3 repeated residual blocks, as indicated by the blue arrow in Fig. 3. Each residual block includes convolutional layers, BN layers, and ReLU activations (Schmidt-Hieber, 2020).

The convolutional layer is a vital component of deep learning, which contains a collection of adjustable convolution kernels. In this work, the padding of zero is applied to all the convolutional operations to manage the image's output size and keep the boundary data. From the mathematical viewpoint, the convolutional layer reduces the number of CNN parameters and gains weight sharing and weight sparseness better than the fully connected layer. Although the output feature maps of the convolutional layer demand to be processed further, it is challenging to process all extracted features, because the number of parameters explodes when the model becomes deeper and the possibility of overfitting rises significantly. As a result, the pooling layer is used to decrease the computational complexity and the overfitting problem. The most used pooling operation is the max-pooling, which picks the highest value of a 2×2 distinct area in each image channel to shrink the output ($w \times h \times ch$ becomes $\frac{w}{2} \times \frac{h}{2} \times ch$ after applying the max-pooling operation).

Different from the encoder side, the deconvolution layer, or transposed convolutional layer, is the main component of the decoder side. It serves as the opposite of the convolutional layer. The decoder uses several upsampling blocks, which are indicated in the green arrow (Fig. 3), in order to reconstruct the original input size by enlarging the spatial dimension of the output twice and decreasing the number of feature maps by half. Each upsampling block includes a BN layer, a ReLU layer, and a transpose convolution layer (kernel size 2×2 and a stride of 2). In addition, a skip connection is used at every block of the decoder to obtain precise positional features. It first merges the output feature maps from the transposed convolution layers and the output layer from the encoder at the same level, followed by a 1×1 convolution for every downsampling block. The mentioned skip connection operation can be seen from Fig. 3 as the grey horizontal arrows that join the output feature maps at each level.

The batch normalization layer (Santurkar et al., 2018) normalizes each mini-batch input layer by narrowing each layer's output to a fixed distribution to dramatically lower the number of training epochs and stabilize the network. The batch normalization process for a collection of training batches with n images $B[x_1, x_1, \cdots x_n]$ is described as follows.

$$\mu B = \frac{1}{m}\sum_{i=1}^{m} x_i$$

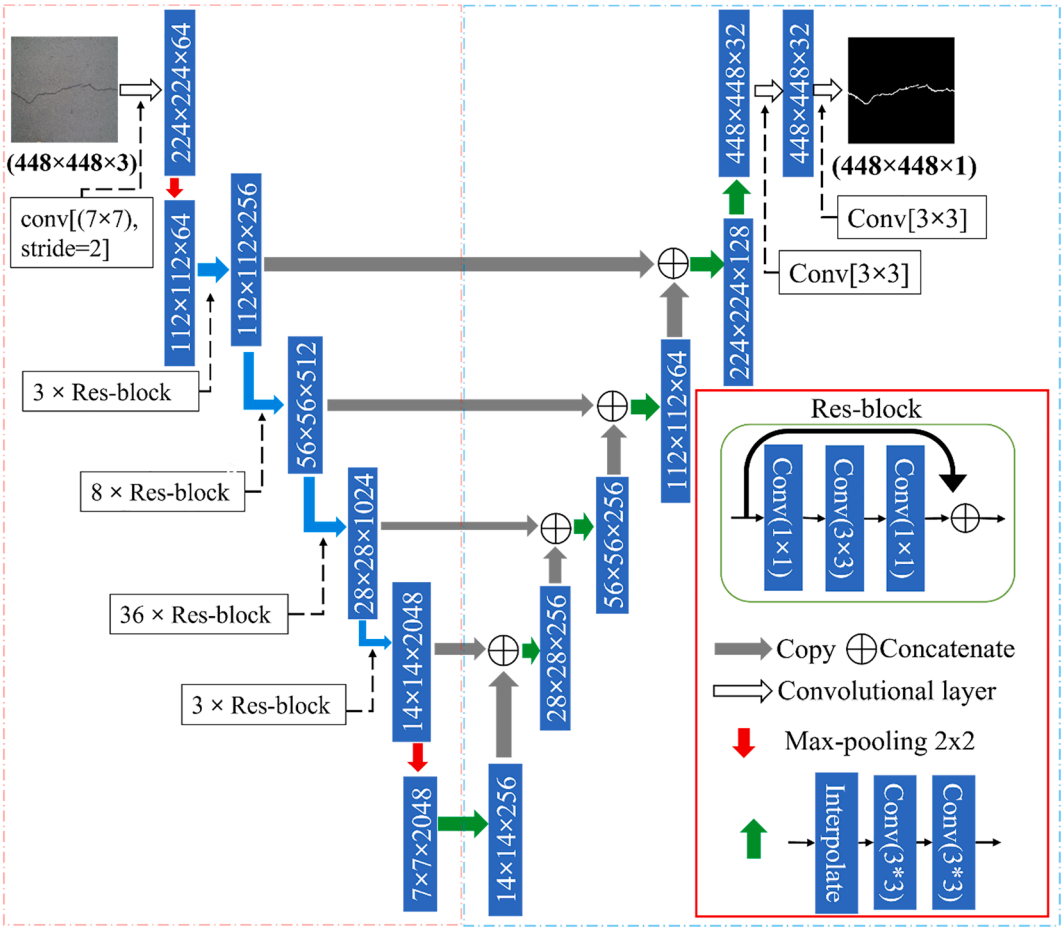

Fig. 3. Detailed structure of the proposed TunnelURes model.Note: Original U-Net's down-sampling path is replaced by the ResNet-152 model.

$$\sigma_B^2 = \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu B)^2$$

$$\widehat{x_i} = \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}}$$

$$y_i = \gamma \widehat{x_i} + \beta \tag{1}$$

where μB depicts the mean and σ_B^2 represents the variance of the current training batch. $\widehat{x_i}$ is the normalized output of deducting the mean μB and varianceσ_B^2. y_i is the scaling and translation result, where the scale factor γ and the translation coefficient β are learnable parameters.

Activation functions are a group of functions, which have monotonic and nonlinear properties, applied to map the input to the output and help the deep learning model adapt to the nonlinear functions and learn the complex relationships and patterns in the training data. ReLU (Schmidt-Hieber, 2020) is a standard activation function described as follows.

$$h = \max(0, a), where a = Wx + b \tag{2}$$

The most significant advantage of ReLU is that it lowers the possibility of the vanishing gradient, which occurs when$a > 0$, because the gradients are always kept at a stable value leading to a faster learning process. On the contrary, the gradients of the sigmoid function get smaller as the modulus of x increases. One more advantage of the ReLU activation is sparsity, which happens if$a \leq 0$, because many of these units appear in a layer lead to a sparser representation. Sigmoid function likely generates various non-zero values, which leads to dense representations.

The last layer in the deep learning model is the output layer that yields the actual probability scores for each class label. For example, the output of any classification CNN model can be represented by an n-dimensional vector$\vec{z}$, where the probability that each element z_j belonging to $\vec{z}$ corresponds to the j-th class. The most common function is the Softmax activation (Ronneberger et al., 2015), which gives at least a minimal amount of probability to all elements in the output vector, normalizes the classification prediction output range between 0 and 1, and forces their sum as 1. When an input is negative or small, the softmax function converts it into a small probability. In contrast, the softmax function turns the input into a large probability if it is large. The equation of the softmax activation is described as follows.

$$softmax(z_j) = \frac{e^{z_j}}{\sum_{k=0}^{c} e^{z_k}} \tag{3}$$

The model's current error must be estimated repeatedly as part of the deep learning model's optimization process, which is based on the choice of an error function (loss function). The loss function's primary objective is to calculate the current loss so that the learnable parameters can be updated to decrease the loss on the next run. The loss function of the original U-Net model is the cross-entropy loss, which converts the segmentation problem into a multiclass classification. The main task of the model is to classify each pixel into one of the classes. The author proved that the cross-entropy loss performed very well compared to the traditional loss functions (Ronneberger et al., 2015). The lining crack segmentation involves two main classes$C = 2$, which are the crack and the background. Thus, the binary cross-entropy loss is applied to the TunnelURes. Moreover, a sigmoid layer is added before the binary cross-entropy loss (BCE) to ensure more numerical stability because it took

advantage of the log-sum-exponential trick (Ge et al., 2019). The BCE with the sigmoid function added is calculated as follows.

$$BCE = -\sum_{i=1}^{C=2} t_i \log\left(\frac{1}{1+e^{-s_i}}\right) \quad (3)$$

where s_i and t_i are the score and the ground truth label for the classC_i..

3.2. Post-processing image heuristics

3.2.1. Post-processing process

After the segmentation process, there can be some discontinued cracks or noise in the output binary image. In this study, several post-processing methods are implemented to refine the segmented cracks, which include 1) filling small holes and 2) reducing noise. These post-processing methods were done mainly based on the opening and closing morphological operations, as described in Fig. 4. First of all, the small holes were filled using a closing operation that is expressed as follows.

$$closing = ((f \oplus \varphi) \ominus \varphi) \quad (4)$$

where $\ominus$ represents the erosion operation, which is a mathematical morphology that shrinks the pixels of the object boundaries, whereas the dilation operation, which is depicted as$\oplus$, adds the pixels to the object boundaries. The erosion calculates a local minimum, whereas the dilation computes a local maximum over the area of a given kernel.

After the first pre-processing process, the noise from the segmented images was eliminated using the morph opening operation as explained in Equation 5, which is an opposite process of the morph closing operation, where the erosion operation was followed by the dilation operation.

$$opening = ((f \ominus \varphi) \oplus \varphi) \quad (5)$$

Assuming that the noise was outside the segmented crack, the opening morphological operation omitted noise by first performing the erosion operation with a particular structural kernel, followed by the dilation operation to recover the pixels to their correct shape. On the other hand, if the noisy pixels were inside the crack, the dilation operation was applied to warp out the noisy pixel and remove them. According to the initial prediction outputs of tunnel crack segmentation, the number of noisy pixels was limited. Therefore, to prevent possible negative influences, this study implements the closing operation and the opening operation using a cross-shaped kernel to eliminate noise from the pixel-level crack segmentation and extract raw crack pixels. A cross-shaped kernel was used because most of the cracks were lining cracks, so it captured the cracks (especially transverse and longitudinal cracks) well and reduced noise significantly.

3.2.2. Skeletonization process

The primary purpose of skeletonization is to transform a crack into one-pixel wide visualization or crack skeleton, which precisely represents the crack's topology. The crack skeletons can be used to measure different aspects of a crack. There have been many algorithms introduced to obtain the skeleton image from a binary segmented image, such as medial axis (Marie et al., 2016), morphological thinning (Changxian & Yulong, 1998), digital patterns thinning (Zhang & Suen, 1984), and 3-D medial surface thinning (Lee et al., 1994).

As displayed in Fig. 5(b), the skeletons generated by the morphological thinning algorithm (Changxian & Yulong, 1998) were blurry, whereas the outputs produced by the other algorithms were almost similar for a simple crack with a large thickness value. The created skeletons of the crack with multiple branches (the second row of Fig. 5) are similar to the simple case (the first row of Fig. 5), because all the skeletons were adequately created, except for the morphological thinning that wrongly created many additional branches of the crack.

Zhang (Zhang & Suen, 1984) and Lee's algorithms (Lee et al., 1994) required a huge computation time if the cracks were complicated because they involved several iterative processes. Each algorithm has its advantages and disadvantages, so the experts can evaluate them separately and decide the most appropriate one for a specific dataset. For example, Zhang's skeletonization method is prioritized for applications that require accurate extraction with an acceptable delay. This study chose the median-axis skeletonization method (Marie et al., 2016) as the skeletonization algorithm, because it showed a good performance on the

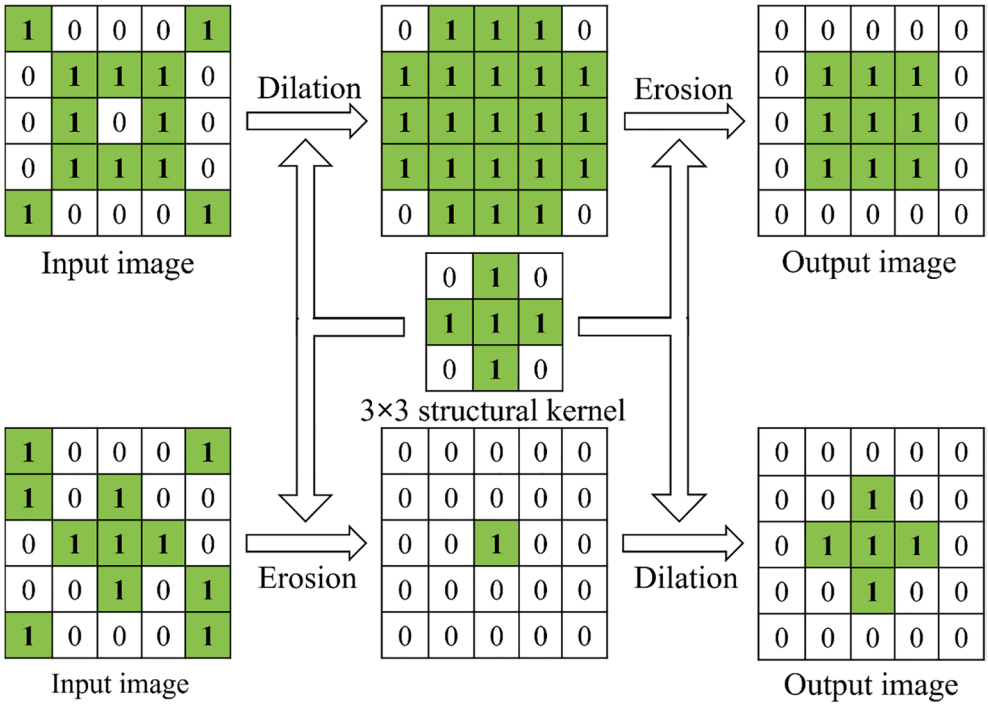

Fig. 4. Explanation for the morph closing and morph opening operations.

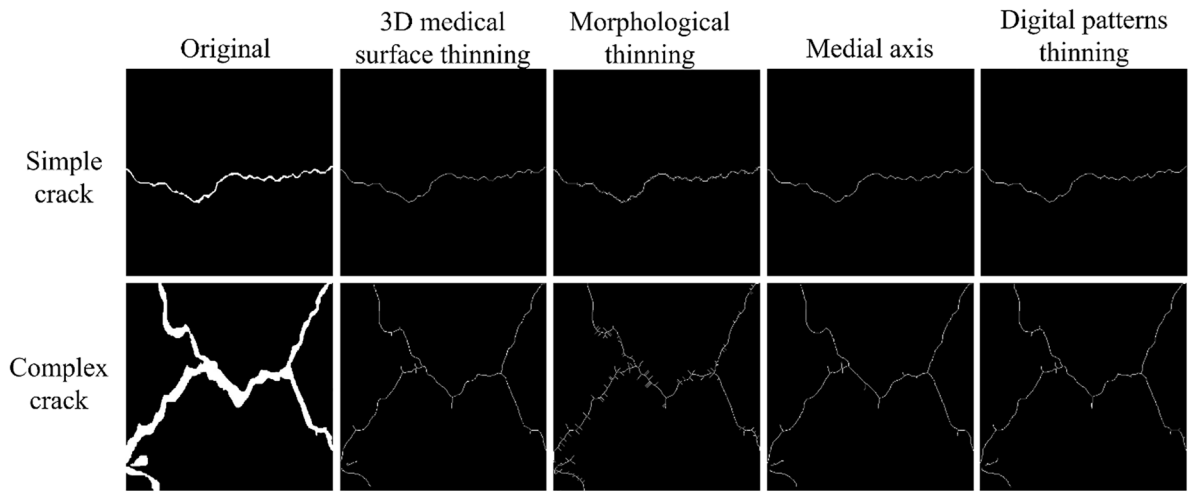

Fig. 5. Performance comparison of four different skeletonization algorithms for both simple and complex cracks.

tunnel crack dataset, performed well on the cracks with fewer branches, and can be executed in real-time.

3.2.3. Crack measurements

According to the previous research (Qiu et al., 2017; Yang et al., 2018), after the crack skeletons were obtained, the length of the crack L can be computed as follows.

$$L = \int_c Gdl \cong \sum Gdl \tag{6}$$

where G represents the geometric calibration. dl is the finite-length ofL. Originally, G is proposed as a calibration parameter for the pixel displacements in the segmented images. The dataset proposed in this work is assumed to contain images that do not have the geometric distortion. Therefore, G is set to 1 to enable L to be calculated by summing up the total pixels of the skeletons directly. After that, the average thickness of the cracks can be calculated.

$$\overline{D} = \frac{\int_s G^2 dS}{L} \cong \frac{\sum G^2 dS}{\sum Gdl} \tag{7}$$

where L represents the previously computed crack length and dS is the finite-area of the skeleton.

When the camera calibration parameters, image scale, or resolution are provided, the pixel-based thickness and length of the crack can be converted into real-world length and thickness measurements (Yang et al., 2018). Moreover, the crack's coverage rate is evaluated quantitatively by calculating the portion of the crack pixels to the total pixels in an image. Such measurements give the experts the analytical references to determine the tunnel's condition.

According to (Roy & Sarkar, 2017; N. Zhang et al., 2019), the lining crack is the primary tunnel defect that can cause various problems, such as 1) reducing the endurance capacity of the structure to the surrounding rocks, 2) leading to leaking, which easily causes corrosion of steel bars and facilities in the tunnel, and 3) excessive deformation of the cracks can reduce the spatial clearance and affect the safety of various types of traffic. Three main types of tunnel lining cracks are longitudinal, transverse, and inclined cracks, as displayed in Fig. 6. The detailed information for each type is described as follows.

- A longitudinal crack refers to the lining cracks that appear along the arch shoulder in the tunnel's longitudinal axis. This crack can lead to the stress intensity at the arch and the sidewall junction.
- A transverse crack refers to the lining cracks that occur along the tunnel axis's vertical direction due to the compressional seismic P-wave propagation.

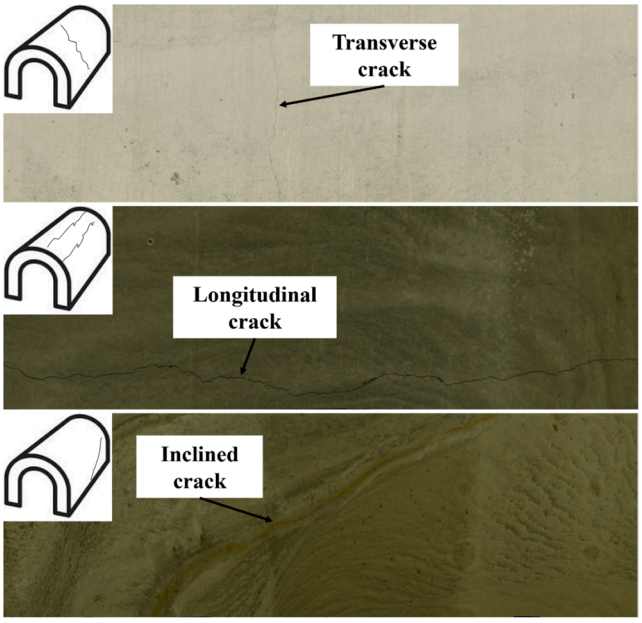

Fig. 6. Three main patterns of the lining crack, including transverse crack, longitudinal crack, and inclined crack.

- An inclined crack refers to the lining slant cracks along the tunnel axis of 40–70 degrees, around the angle arch and sidewall due to the tensile forces.

The crack type can be determined by calculating the rotation angle r in degrees of the segmented crack based on the property of each type of crack, as discussed in (Zhang et al., 2018).

$$Type = \begin{cases} Transversecrackifr = [80, 100] \\ Longitudinalcrackifr = [140, 180] \\ Inclinedcrackifr = [40, 70] \end{cases} \tag{8}$$

4. Experimental results

In this study, all experiments are conducted on an Ubuntu 16.04 Linux machine equipped with four Titan X 12 GB GPUs and 64 GB of DDR4 RAM. Section 4.1 explains various evaluation metrics applied to check the model's performance. Section 4.2 then illustrates the effectiveness of the proposed automatic lining crack segmentation compared

to the other well-known methods. After that, the proposed model's robustness is tested under various challenging scenarios in Section 4.3. Finally, the skeletonization approach and crack measurement using the generated skeletons are described in Section 4.4.

4.1. Evaluation metrics

Tunnel crack segmentation can be considered a binary classification problem because the output image contains crack pixels and background pixels. Therefore, three primary components of the confusion matrix, including the true positive (TP), false negative (FN), and false positive (FP), can be extracted to allow the calculation of PA and IoU (Li et al., 2022), which are the two standard evaluation metrics for the crack segmentation. A high PA value may lead to a false evaluation when the crack pixels (positive class) are minor compared to the background pixels (negative class), because it is biased toward assessing how well the model identified the negative samples. Therefore, the IoU is computed as an additional measurement.

PA measures the proportion of accurately segmented pixels to the total pixels, whereas the IoU metric quantifies the overlap percentage between the ground truth and the prediction. PA and IoU can be formulated as follows.

$$PA = \frac{TP + TN}{TP + TN + FP + FN} \tag{9}$$

$$IoU = \frac{TP}{TP + FN + FP} \tag{10}$$

As an additional evaluation metric for the binary problem, a receiver operating characteristic (ROC) curve can be plotted in order to show the binary segmentation's performance at different thresholds. The ROC curve involves two primary parameters, which are the true-positive rate (TPR) and the false-positive rate (FPR).

$$TPR = \frac{TP}{TP + FN} \tag{11}$$

$$FPR = \frac{FP}{TN + FP} \tag{12}$$

4.2. Tunnel crack segmentation

75% of the data were randomly chosen and fed into the TunnelURes model, whereas the remaining 25% of the data was used to evaluate the proposed model using the created training dataset (Section 2.3). The learning rate is a crucial hyperparameter that is computed to optimize and minimize the neural network's loss by controlling how quickly weights or the coefficients get updated. The model achieves varying performances under different learning rates, so selecting the right learning rate has a critical role in the convergence of loss function (He et al., 2016). In this study, two learning rates (10^{-3} and 10^{-4}) were selected to train the CNN model through 50 epochs.

The model's validation losses at different learning rates are represented in Fig. 7, where the solid orange line and the solid blue line indicate the validation loss for the learning rate 10^{-3} and10^{-4}, respectively. When the learning rate equals10^{-3}, the validation losses fluctuated widely, and the validation loss was at 0.065 after 50 epochs. On the other hand, the validation loss was stable as the validation loss steadily decreased and reached 0.061 at epoch 50th for the learning rate of10^{-4}. The result is in line with the findings from (Smith, 2017), which stated that the loss decreases slowly with a small learning rate, whereas the training process diverges when the learning rate gets too big.

In this part, the model's robustness is examined and compared with the existing state-of-the-art models using the collected test dataset that contains about 34,068 images. There are a total of six models implemented for comparison. We evaluate two extensions of the proposed model before (TunnelURes) and after applying the post-processing process (TunnelURes_PO). In addition, four previously proposed models, which include U-Net (Ronneberger et al., 2015), DeepCrack (Y. Liu et al., 2019), Fully Convolutional Network (Long et al., 2015), and CrackNet (A. Zhang et al., 2019), are selected for comparison. They were implemented based on the descriptions from the original papers, with the hyper-parameters configured the same as those described in the reference papers.

Table 2
Crack segmentation performance of the proposed models (TunnelURes, TunnelURes_PO) compared to previous approaches using the collected testing dataset.

Approach	PA	IoU
U-Net (Z. Liu et al., 2019)	0.849	0.643
DeepCrack (Y. Liu et al., 2019)	0.923	**0.853**
FCN (Yang et al., 2018)	0.824	0.836
CrackNet (A. Zhang et al., 2019)	0.817	0.735
TunnelURes	0.848	0.726
TunnelURes_PO	**0.928**	0.847

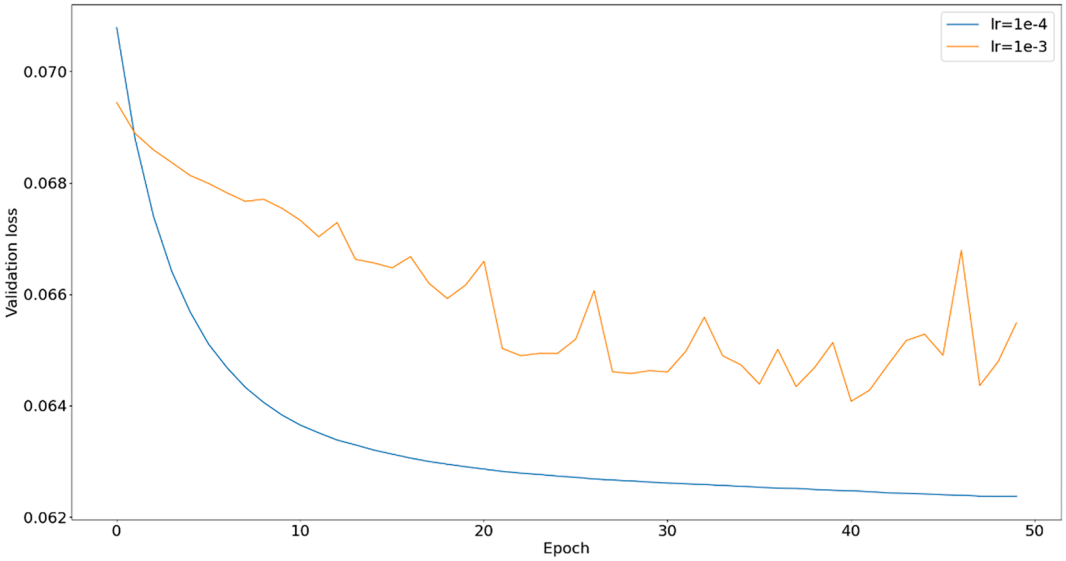

Fig. 7. Corresponding validation losses of the proposed model for two different learning rates.

PA and IoU, as mentioned previously in Section 4.1, are computed to compare the model's performance. Table 2 shows the performances of various crack segmentation models. In general, all models obtained high PA values of over 0.8. The proposed TunnelURes_PO model and the DeepCrack model achieved the highest PA value of 0.92. The U-Net, CrackNet, and TunnelURes models got relatively low IoU values of 0.643, 0.735, and 0.726. By contrast, the proposed TunnelURes_PO and DeepCracks model obtain significantly higher IoU scores at 0.84 and 0.85, respectively. The statistics indicate that the proposed TunnelURes_PO outperformed most of the previous approaches in terms of the lining crack segmentation.

Fig. 8 shows the AUC values for different models on the collected dataset. The highest goodness-of-fit based on the testing dataset (AUC = 0.97) was recorded on the proposed TunnelURes_PO. The DeepCrack model also achieved comparable results with the AUC value of 0.96, which implies that, at the current setup, these two models could properly distinguish between the crack pixels and the background pixels. The two models were followed by the U-Net (AUC = 0.94), TunnelURes (AUC = 0.89), FCN (AUC = 0.84), and CrackNet (AUC = 0.82) models. This means that the ability of the proposed TunnelURes to segment the lining crack on the proposed dataset was better than the previous models.

In addition, it is noticeable that the original TunnelURes without the post-processing process obtained lower performance (AUC = 0.89) than the DeepCrack model (AUC = 0.97). The main reason is that the TunnelURes created various noises, which lowered the AUC value. Therefore, the post-processing step significantly improved the performances of the TunnelURes (PA was improved by 0.11, IoU was improved by 0.12, and AUC value was improved by 0.08).

Some normal testing images are presented in Fig. 9 in order to check the proposed model's outputs. The first row of Fig. 9 shows the sub-images cropped from the original images that contain cracks. The second row represents the corresponding ground truth masks that were labeled by humans. The segmentation performances of the TunnelURes model are listed in the third row. In general, the proposed model accurately localized the cracks. In addition, even though the ground truth images contain several disjoint cracks, the proposed model significantly reduced them due to the post-processing process.

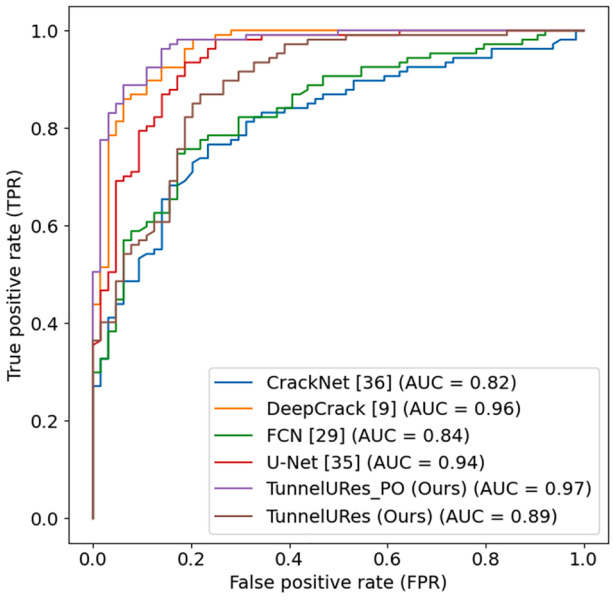

Fig. 8. AUC values of our model compared to other approaches using the testing dataset.

4.3. Model's robustness on challenging cases

Although the cracks inside the tunnel are the main goal of this research, the inner surface of the tunnel usually contains challenging elements that resemble the crack or affect the crack segmentation performance, such as tunnel lighting, joints, wires, and patchworks. As a result, this section is dedicated to checking whether the proposed model can segment the lining cracks for the challenging images with various types of noise.

Fig. 10 shows four different models' performances on four types of noises, which include dark & writing, light & wire, spots on the wall, and wall joint. From the first column to the last column are the original image, the corresponding ground truth, crack segmentation results of the U-Net model (Z. Liu et al., 2019), DeepCrack model (Y. Liu et al., 2019), TunnelURes model, and TunnelURes_PO model. The image is challenging for the dark image and handwriting case because the image is dark, and the writing itself resembles the crack. The U-Net model incorrectly segmented the handwriting as cracks, which proved that the model was sensitive to noise. By contrast, the cracks were correctly differentiated from the handwriting using the TunnelURes and the DeepCrack models. The proposed TunnelURes identified more lining cracks compared to the DeepCrack model. Moreover, the TunnelURes_PO showed the effectiveness of the post-processing process, which significantly reduced the noise.

The crack is difficult to be segmented in the second case, because the wires are similar to the cracks (shape and color). Moreover, the lighting can also affect the performance of the models. The U-Net model barely recognized the cracks, the wire pixels were segmented as crack pixels, and many noises were also misclassified as cracks. On the other hand, the TunnelURes and DeepCrack models segmented the cracks correctly and were robust against the noise.

Spots on the wall, described in the third row, are also a common type of noise that usually appears on the tunnel surface that affects the model's performance if the cracks appear inside these spots. The U-Net and the DeepCrack model missed most of the crack regions that appeared inside the spots, whereas the TunnelURes and TunnelURes_PO models successfully segmented the cracks, even they were inside the spots.

The wall joint is also a common noise in the form of a straight line that is identical to the lining crack. The DeepCrack and TunnelURes models categorized it as background with some noise. After the post-processing process, the TunnelURes_PO model reduced the noise significantly. The preliminary experimental results proved that the TunnelURes model performed robustly for the lining crack segmentation.

The experiments proved that the TunnelURes model was robust against various types of noises, because the pre-trained ResNet-152 encoder helped the model obtain richer features that presented a good distinction between noises, cracks, and background. Therefore, it can be integrated into the existing crack segmentation applications efficiently.

However, it is worth noting that the models could not recognize the cracks in specific scenarios. The failed predictions of the proposed model can be classified into two main types, which include crack discontinuity and false alarm. The first error is illustrated in Fig. 11(a), where the model missed some parts of the tunnel crack. The continuous crack from the ground truth mask was predicted as two separated cracks in the output image. The error usually occurs during the prediction process when the model misclassifies some crack pixels as background pixels. As a result, it could be solved by performing various post-processing techniques.

For the second case (Fig. 11(b)), the red circles show two false alarms that were misrecognized as the crack skeletons. According to the segmentation output, although the proposed model achieved adequate validation performance, the medial-axis skeletonization algorithm created many errors because it was susceptible to the crack intersection and the image edges where the crack's representation changed. In order

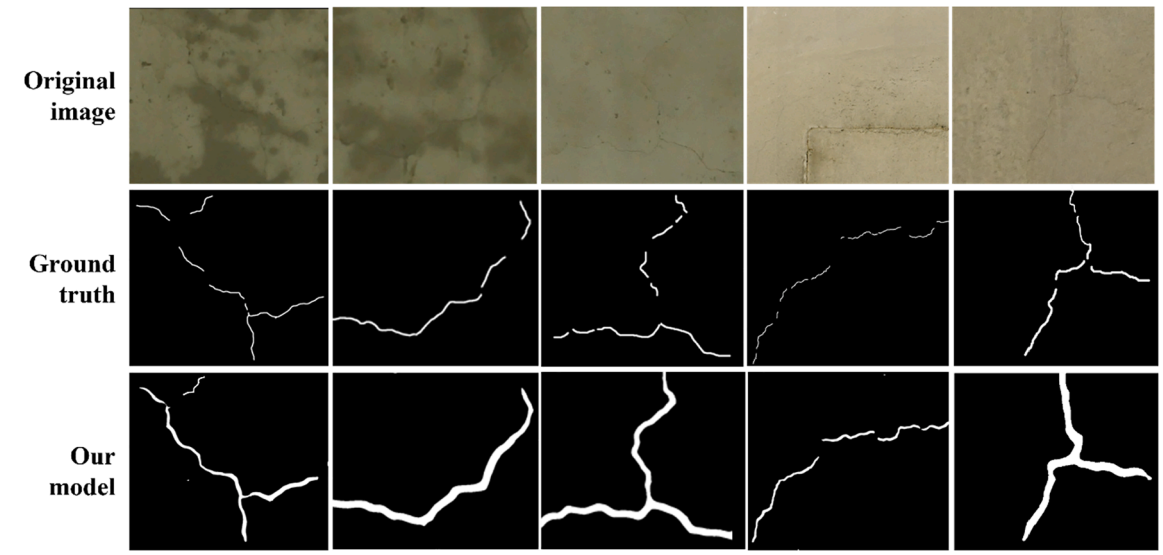

Fig. 9. Outputs of the proposed TunnelURes model compared to the ground truth on various crack types.

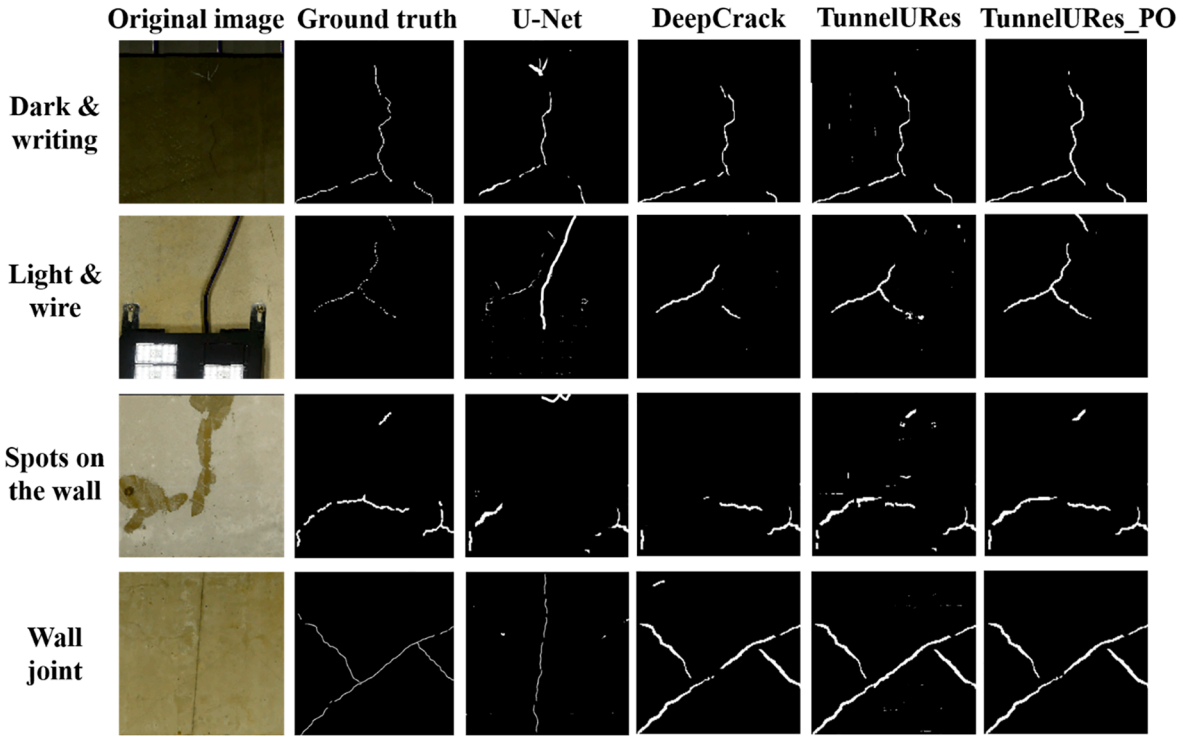

Fig. 10. Performance comparison between the U-Net, DeepCrack, TunnelURes, and TunnelURes_PO models on challenging scenarios, including dark & writing, light & wire, spots on the wall, and wall joint.

to completely solve the mentioned issue and create a more accurate skeleton, the conventional skeletonization algorithm needs to be improved, or a novel algorithm is required.

4.4. Crack measurement

After the segmented images were generated, it is crucial to investigate the crack length, thickness, and crack type in order to offer additional statistics regarding the cracks to the moderators. The measurements can be computed by first extracting the crack skeleton, which converts pixel-level crack into the thickness of the single-pixel level skeleton to show the crack's structure. As explained earlier, the median-axis skeletonization algorithm was implemented to extract the skeleton for each segmented crack. For comparison, we also implemented the skeletonization method on the ground truth image and considered it the ground truth skeleton.

Fig. 12 shows the segmentation results for three different types of lining cracks. The first column indicates the original RGB images, the next column is the corresponding ground truth images, the third column displays the generated crack skeletons for the ground truth, the outputs of the TunnelURes are displayed in the fourth column, and finally, the crack skeletons of the predicted images are presented in the fifth column. Fig. 10 shows that the TunnelURes model segmented different types of lining cracks robustly, even for the complex inclined crack. It

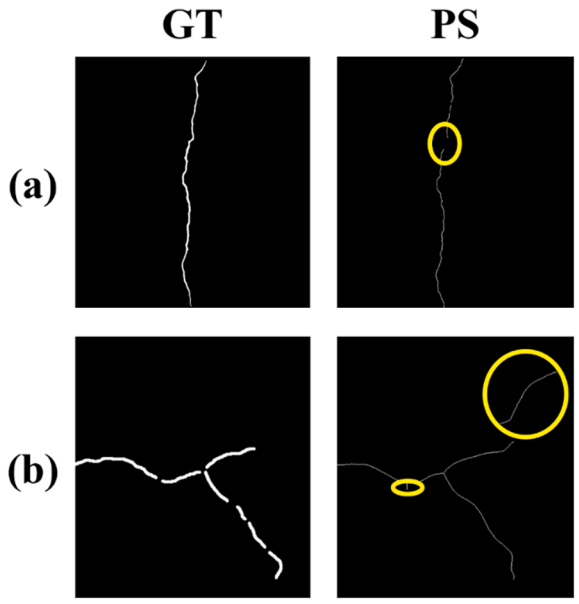

Fig. 11. Two types of false alarms appear in the output image.Note. Ground truth (GT), Predicted skeleton (PS)). The yellow circles indicate the wrong part. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

can be seen that the skeletons of the predicted segmented images are identical to the ground truth skeletons by inspecting the generated skeletons. As a result, the proposed algorithm has shown that it effectively segmented different kinds of cracks, and the medical-axis skeletonization algorithm also efficiently created the crack skeletons.

In order to check the crack measurement performance, a total of 400 sub-images were randomly picked from the testing dataset. Next, the medial-axis skeletonization was conducted on both the ground truth and the predicted images. Finally, the crack information, such as length, thickness, and areas, were computed based on the generated skeletons.

Fig. 13 represents three regression graphs for the three corresponding features (length, area, and thickness) to demonstrate the relationship of the ground truth crack and the predicted crack created by the TunnelURes model. The x-axis indicates the ground truth and the prediction measurement values in pixels. After the corresponding points are drawn, the linear regression is applied to evaluate the slope of the measurements. The dashed lines show the correct prediction (the predicted value is equal to the ground-truth value), whereas the solid lines depict the linear fitting results. The slope of crack length was 1.1, and its standard error was 0.01. The slope of the area and thickness were 0.85 and 1.16, respectively, and the standard errors were 0.01 and 0.03. The statistics indicate that the computed measurements of the segmented cracks were slightly smaller than those of the ground truth measurements. The main reason is that the crack edge pixels contained features identical to the background pixels, so they were ignored.

In practice, the crack thickness usually varies due to the positions and structural causes. However, it is interesting to note that most of the cracks were in the same thickness range. This is the primary limitation of the dataset used in this research, because it contains the images, which are mostly collected from the three tunnels (Banggyo, Bongsan, and Deugsong) with similar structural status and conditions.

In addition, p-value, R^2 statistic, and the F-statistic are computed to evaluate the linear regression effectiveness, as displayed in Table 3.

The R^2 statistic and the p-value demonstrate the fitting degree of the linear regression. The fitting degree becomes higher when the R^2 value is larger, and the p-value <0.001, which means that the linear fitting is efficient.

5. Conclusion

This paper introduces a novel deep learning-based tunnel crack segmentation framework to segment the lining cracks that usually appear on the tunnel surface. The proposed framework, which is called TunnelURes, utilizes the ResNet-152 model as the encoder, and the decoder mainly contains deconvolutional layers. Such a hybrid structure enabled the proposed model to segment cracks efficiently and robustly against noise. We also introduced a huge lining crack segmentation dataset containing 170,339 crack images and 170,339 mask images, which were labeled manually by the experts.

The experimental results on various learning rates showed that the learning rate of 0.001 significantly reduced the validation loss to$<$0.06. The post-processing step improved the PA and IoU of the original model to 0.848 and 0.928, respectively. The proposed model outperformed the previous state-of-the-art models, such as CrackNet, FCN, and U-Net, and was comparable to the state-of-the-art DeepCrack model. Moreover, the model produced accurate and robust results on challenging cases, such as noises, which had confused the previous models. Finally, crucial

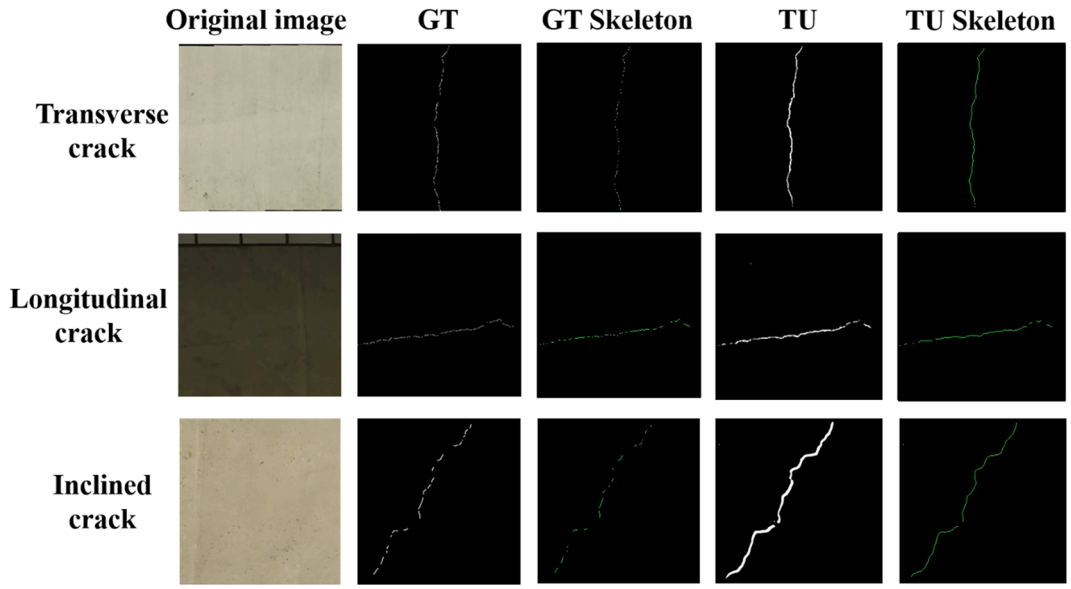

Fig. 12. Skeletonization results for three types of the lining cracks.Note: GT stands for Ground truth and TU represents the TunnelURes model output.

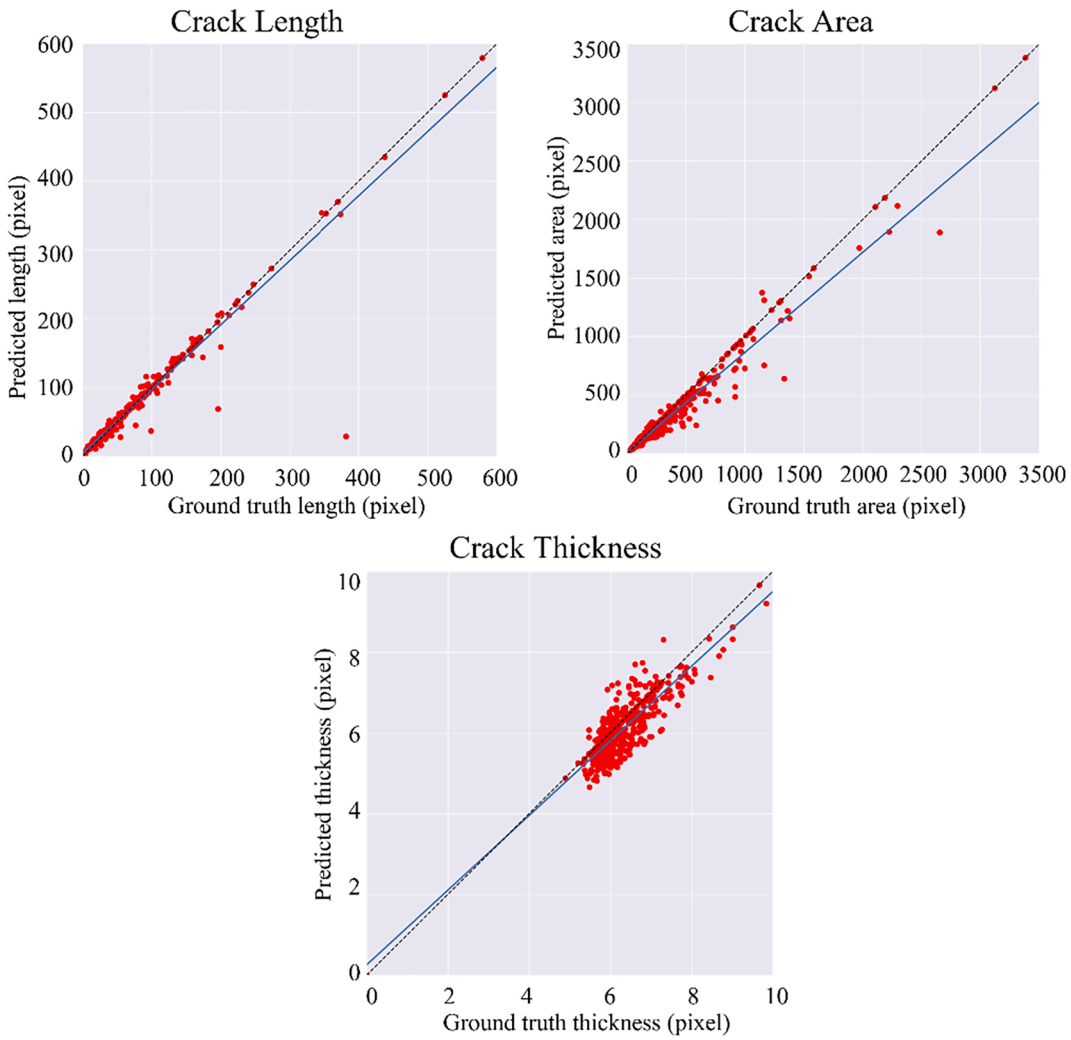

Fig. 13. Scatter diagrams for three different crack measurements, including crack length, crack area, and crack thickness using the proposed TunnelURes model.

Table 3
Linear fitting analysis of crack measurement statistics.

	Length	Area	Thickness
Slope	0.936	0.854	0.924
R^2 statistic	0.942	0.949	0.757
p-value	<0.001	<0.001	<0.001

information about the crack, such as crack length, thickness, and crack type, was automatically calculated with the standard error between 0.01 and 0.03, demonstrating the proposed method's robustness.

The experimental results proved that the proposed lining crack segmentation framework was convenient and efficient. It receives the input image and produces the segmented crack with all the necessary measurements for each crack. The model's performance can be further improved when more challenging images (noise, unbalanced lighting condition) are provided in future research. In addition, a customized medial-axis skeletonization algorithm can be introduced for the tunnel crack segmentation to enhance the crack measurement process.

CRediT authorship contribution statement

L. Minh Dang: Conceptualization, Writing – original draft. **Hanxiang Wang:** Visualization. **Yanfen Li:** Data curation, Writing – review & editing. **Yesul Park:** Resources, Writing – review & editing. **Chanmi Oh:** Writing – review & editing. **Tan N. Nguyen:** Data curation. **Hyeonjoon Moon:** 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.

Acknowledgment

This work was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (2020R1A6A1A03038540) and National Research Foundation of Korea (NRF) grant funded by the Korea government, Ministry of Science and ICT (MSIT) (2021R1F1A1046339) and by a grant(20212020900150) from "Development and Demonstration of Technology for Customers Bigdata-based Energy Management in the Field of Heat Supply Chain" funded by Ministry of Trade, Industry and Energy of Korean government.

References

Attard, L., Debono, C.J., Valentino, G., Di Castro, M., 2018. Tunnel inspection using photogrammetric techniques and image processing: A review. ISPRS Journal of Photogrammetry and Remote Sensing 144, 180–188.

Brown, M., Lowe, D.G., 2007. Automatic panoramic image stitching using invariant features. International journal of computer vision 74 (1), 59–73.
Changxian, S., & Yulong, M. (1998). Morphological thinning based on image's edges. ICCT'98. 1998 International Conference on Communication Technology. Proceedings (IEEE Cat. No. 98EX243),.
Choi, W., Cha, Y.-J., 2020. SDDNet: Real-time crack segmentation. IEEE Transactions on Industrial Electronics 67 (9), 8016–8025.
Dang, L.M., Hassan, S.I., Im, S., Mehmood, I., Moon, H., 2018. Utilizing text recognition for the defects extraction in sewers CCTV inspection videos. Computers in Industry 99, 96–109.
Dang, L.M., Wang, H., Li, Y., Nguyen, T.N., Moon, H., 2022. DefectTR: End-to-end defect detection for sewage networks using a transformer. Construction and Building Materials 325, 126584. https://doi.org/10.1016/j.conbuildmat.2022.126584.
Deep inspection, http://www.deepinspection.ai/. (2021). http://www.deepinspection.ai/.
Ge, Z., Mahapatra, D., Chang, X., Chen, Z., Chi, L., Lu, H., 2019. Improving multi-label chest X-ray disease diagnosis by exploiting disease and health labels dependencies. Multimedia Tools and Applications 1–14.
Hassan, S.I., Dang, L.M., Mehmood, I., Im, S., Choi, C., Kang, J., Park, Y.-S., Moon, H., 2019. Underground sewer pipe condition assessment based on convolutional neural networks. Automation in Construction 106, 102849. https://doi.org/10.1016/j.autcon.2019.102849.
He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. Proceedings of the IEEE conference on computer vision and pattern recognition.
Huang, H.-W., Li, Q.-T., Zhang, D.-M., 2018. Deep learning based image recognition for crack and leakage defects of metro shield tunnel. Tunnelling and underground space technology 77, 166–176.
Huang, M.Q., Ninić, J., Zhang, Q.B., 2021. BIM, machine learning and computer vision techniques in underground construction: Current status and future perspectives. Tunnelling and underground space technology 108, 103677. https://doi.org/10.1016/j.tust.2020.103677.
Kaise, S., Ito, T., Maeda, Y., Yagi, H., Maeda, K., & Shinji, M. (2020). Possibility on Substitution Close Visual Inspection for Imaging Technology to Tunnel Lining Health Monitoring. *Journal of Japan Society of Civil Engineers, Ser. F1 (Tunnel Engineering)*, *76* (1).
Kamaliardakani, M., Sun, L.u., Ardakani, M.K., 2016. Sealed-crack detection algorithm using heuristic thresholding approach. Journal of Computing in Civil Engineering 30 (1), 04014110. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000447.
Lee, T.C., Kashyap, R.L., Chu, C.N., 1994. Building skeleton models via 3-D medial surface axis thinning algorithms. *CVGIP*. Graphical Models and Image Processing 56 (6), 462–478.
Lei, M., Liu, L., Shi, C., Tan, Y., Lin, Y., Wang, W., 2021. A novel tunnel-lining crack recognition system based on digital image technology. Tunnelling and underground space technology 108, 103724. https://doi.org/10.1016/j.tust.2020.103724.
Li, Y., Wang, H., Dang, L.M., Jalil Piran, M.d., Moon, H., 2022. A Robust Instance Segmentation Framework for Underground Sewer Defect Detection. Measurement 190, 110727. https://doi.org/10.1016/j.measurement.2022.110727.
Liu, B., Ren, Y., Liu, H., Xu, H., Wang, Z., Cohn, A.G., Jiang, P., 2021. GPRInvNet: Deep Learning-Based Ground-Penetrating Radar Data Inversion for Tunnel Linings. IEEE Trans. Geosci. Remote Sensing 59 (10), 8305–8325.
Liu, Y., Yao, J., Lu, X., Xie, R., Li, L.i., 2019a. DeepCrack: A deep hierarchical feature learning architecture for crack segmentation. Neurocomputing 338, 139–153.
Liu, Z., Cao, Y., Wang, Y., Wang, W., 2019b. Computer vision-based concrete crack detection using U-net fully convolutional networks. Automation in Construction 104, 129–139.
Long, J., Shelhamer, E., Darrell, T., 2015. Fully convolutional networks for semantic segmentation. Proceedings of the IEEE conference on computer vision and pattern recognition.
Loupos, K., Doulamis, A.D., Stentoumis, C., Protopapadakis, E., Makantasis, K., Doulamis, N.D., Amditis, A., Chrobocinski, P., Victores, J., Montero, R., Menendez, E., Balaguer, C., Lopez, R., Cantero, M., Navarro, R., Roncaglia, A., Belsito, L., Camarinopoulos, S., Komodakis, N., Singh, P., 2018. Autonomous robotic system for tunnel structural inspection and assessment. International Journal of Intelligent Robotics and Applications 2 (1), 43–66.
Marie, R., Labbani-Igbida, O., Mouaddib, E.M., 2016. The delta medial axis: a fast and robust algorithm for filtered skeleton extraction. Pattern Recognition 56, 26–39.
Menendez, E., Victores, J.G., Montero, R., Martínez, S., Balaguer, C., 2018. Tunnel structural inspection and assessment using an autonomous robotic system. Automation in Construction 87, 117–126.
Minh, D., Wang, H.X., Li, Y.F., Nguyen, T.N., 2021. Explainable artificial intelligence: a comprehensive review. Artificial Intelligence Review 1–66.
Nguyen, T. N., Lee, J., Dinh-Tien, L., & Dang, L. M. Deep learned one-iteration nonlinear solver for solid mechanics. *International Journal for Numerical Methods in Engineering*.
Qiu, S., Wang, W., Wang, S., Wang, K.C.P., 2017. Methodology for accurate AASHTO PP67-10–based cracking quantification using 1-mm 3D pavement images. Journal of Computing in Civil Engineering 31 (2), 04016056. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000627.
Ronneberger, O., Fischer, P., Brox, T., 2015. U-net: Convolutional networks for biomedical image segmentation. International Conference on Medical image computing and computer-assisted intervention.
Roy, N., Sarkar, R., 2017. A review of seismic damage of mountain tunnels and probable failure mechanisms. Geotechnical and Geological Engineering 35 (1), 1–28.
Santurkar, S., Tsipras, D., Ilyas, A., & Madry, A. (2018). How does batch normalization help optimization? Advances in Neural Information Processing Systems,.
Schmidt-Hieber, J., 2020. Nonparametric regression using deep neural networks with ReLU activation function. Annals of Statistics 48 (4), 1875–1897.
Smith, L.N., 2017. Cyclical learning rates for training neural networks. 2017 IEEE winter conference on applications of computer vision (WACV).
Song, Q., Wu, Y., Xin, X., Yang, L.u., Yang, M., Chen, H., Liu, C., Hu, M., Chai, X., Li, J., 2019. Real-time tunnel crack analysis system via deep learning. Ieee Access 7, 64186–64197.
Wang, G., Tse, P.W., Yuan, M., 2018. Automatic internal crack detection from a sequence of infrared images with a triple-threshold Canny edge detector. Measurement Science and Technology 29 (2), 025403. https://doi.org/10.1088/1361-6501/aa9857.
Wang, H., Li, Y., Dang, L.M., Lee, S., Moon, H., 2021. Pixel-level tunnel crack segmentation using a weakly supervised annotation approach. Computers in Industry 133, 103545. https://doi.org/10.1016/j.compind.2021.103545.
Wang, S., Liu, C., Shao, Z., Ma, G., He, C., 2020. Experimental study on damage evolution characteristics of segment structure of shield tunnel with cracks based on acoustic emission information. Engineering Failure Analysis 118, 104899. https://doi.org/10.1016/j.engfailanal.2020.104899.
Xu, X., Yang, H., 2020. Vision measurement of tunnel structures with robust modelling and deep learning algorithms. Sensors 20 (17), 4945.
Yang, X., Li, H., Yu, Y., Luo, X., Huang, T., Yang, X.u., 2018. Automatic pixel-level crack detection and measurement using fully convolutional network. Computer-Aided Civil and Infrastructure Engineering 33 (12), 1090–1109.
Zhan, H., Jiang, H., Jiang, R., 2020. Three-dimensional images generated from diffuse ultrasound wave: detections of multiple cracks in concrete structures. Structural Health Monitoring 19 (1), 12–25.
Zhang, A., Wang, K.C.P., Fei, Y., Liu, Y., Chen, C., Yang, G., Li, J.Q., Yang, E., Qiu, S., 2019a. Automated pixel-level pavement crack detection on 3D asphalt surfaces with a recurrent neural network. Computer-Aided Civil and Infrastructure Engineering 34 (3), 213–229.
Zhang, N., Zhu, X., Ren, Y., 2019b. Analysis and study on crack characteristics of highway tunnel lining. Civil Engineering Journal 5 (5), 1119–1123.
Zhang, T.Y., Suen, C.Y., 1984. A fast parallel algorithm for thinning digital patterns. Communications of the ACM 27 (3), 236–239.
Zhang, X., Jiang, Y., Sugimoto, S., 2018. Seismic damage assessment of mountain tunnel: A case study on the Tawarayama tunnel due to the 2016 Kumamoto Earthquake. Tunnelling and underground space technology 71, 138–148.
Zhao, S., Zhang, D., Xue, Y., Zhou, M., Huang, H., 2021. A deep learning-based approach for refined crack evaluation from shield tunnel lining images. Automation in Construction 132, 103934. https://doi.org/10.1016/j.autcon.2021.103934.
Zhao, S., Zhang, D.M., Huang, H.W., 2020. Deep learning–based image instance segmentation for moisture marks of shield tunnel lining. Tunnelling and underground space technology 95, 103156. https://doi.org/10.1016/j.tust.2019.103156.