Crack Detection Method for Tunnel Lining Surfaces using Ternary Classifier

Jeong Hoon Han[1], In Soo Kim[2], Cheol Hee Lee[2], and Young Shik Moon[1*]
[1] Department of Computer Science and Engineering, Hanyang University
Ansan, 15588, Republic of Korea
[2] Deep Inspection,
Beopwon-ro 11 Gil, 25, A-1314&1315,
Songpa-Gu, Seoul, Republic of Korea
[e-mail:bghan@visionlab.or.kr, iskim@deepinspection.ai, chlee@deepinspection.ai, ysmoon@hanyang.ac.kr]
*Corresponding author: Young Shik Moon

*Received April 16, 2020; revised June 8, 2020; accepted July 29, 2020;
published September 30, 2020*

Abstract

The inspection of cracks on the surface of tunnel linings is a common method of evaluate the condition of the tunnel. In particular, determining the thickness and shape of a crack is important because it indicates the external forces applied to the tunnel and the current condition of the concrete structure. Recently, several automatic crack detection methods have been proposed to identify cracks using captured tunnel lining images. These methods apply an image-segmentation mechanism with well-annotated datasets. However, generating the ground truths requires many resources, and the small proportion of cracks in the images cause a class-imbalance problem. A weakly annotated dataset is generated to reduce resource consumption and avoid the class-imbalance problem. However, the use of the dataset results in a large number of false positives and requires post-processing for accurate crack detection. To overcome these issues, we propose a crack detection method using a ternary classifier. The proposed method significantly reduces the false positive rate, and the performance (as measured by the F1 score) is improved by 0.33 compared to previous methods. These results demonstrate the effectiveness of the proposed method.

***Keywords*: Crack Detection, Convolutional Neural Network, Tunnel Lining Inspection**

This work was supported by the Technology Business Innovation Program (No. 18TBIP-C111806-03) grant funded by Ministry of Land, Infrastructure and Transport of Korean government.

http://doi.org/10.3837/tiis.2020.09.013 **ISSN : 1976-7277**

1. Introduction

The market size of facility safety diagnoses in South Korea is approximately 300 billion won, of which the safe management of tunnel facilities in Seoul accounts for almost 24% [1]. Moreover, considering the uniqueness of tunnel environments, the market share of tunnel and concrete facility management is expected to continue to increase. Crack detection is an important task in tunnel facility management, through which the structural conditions and safety of tunnels can be checked. Cracks in tunnels have various causes, and they are important indicators that can be used to predict the conditions and performances of structures. Structural cracks can be categorized into horizontal, vertical, and shear cracks, according to the direction of propagation. They can be further categorized into structural and non-structural cracks, depending on the cause. “Structural cracking” refers to cracks that have evolved and reached a state in which the load of the structure is no longer supported, meaning the structure is no longer functional. Structural cracking can occur as a result of design errors, external loads exceeding the design load, poor construction, or a lack of reinforcing bars. Non-structural cracks refer to those cracks that do not fit into this first category. These cracks can cause a deterioration in durability and the stability of the structure, and are the result of factors such as the corrosion of reinforcing bars [2]. To minimize potential damage due to cracking, periodic safety checks of tunnels and structures, as well as structural evaluations, must be repeatedly conducted, and repair and reinforcement measures are required for cracks of or exceeding a certain size.

Naked-eye crack inspections require inspectors to block the tunnel lane, photograph the inside of the tunnel using a work vehicle, and visually check the captured images to identify the cracks then check the condition of the crack in real tunnel. This method determines whether or not cracks are present using the subjective opinions of each inspector; thus, it struggles to ensure objectivity and requires considerable time and resources. To resolve this problem, many crack detection methods based upon image processing techniques have been proposed for the images acquired in tunnel scanning procedures. However, crack detection methods using traditional image processing techniques suffer from the limitation that they only function well in a restricted environment. Furthermore, their performances are not robust due to the difficulties of evaluating the images taken in tunnels; for instance, they struggle to distinguish between the color of the concrete itself and the actual cracks discolored by concrete pouring marks and soot.

To resolve this problem, crack detection methods using convolutional neural networks (CNNs) have been proposed [3], [4], [5], [6]. However, these approaches require a large quantity of data, and are at a disadvantage in that accurate ground truth data are required for accurate predictions. In turn, expert-annotated data are required to generate these ground truth data, which consumes time and resources. Despite these shortcomings, many previous crack detection methods have attempted to identify cracks using, well-annotated small datasets. These are suitable for crack detection methods employing supervised learning; however, they require large well-annotated ground truth data. Moreover, the crack containing areas of the images are extremely small, and this class imbalance(crack and non-crack) degrades the crack detection performance considerably. As such, there is a great demand for effective crack detection networks.

Identifying the crack shapes is important for automatic tunnel inspection systems because detecting the crack precisely and accurately is necessary for measuring the crack thickness and shape. However, accurate crack detection is challenging, and the aforementioned data imbalances and insufficiently small well-annotated datasets exacerbate this. Previous crack detection methods have attempted to overcome this through post processing. However, post-processing methods require a large computational burden on the tunnel inspection system and it is dependent on the result of the crack detection method.

Here, we propose a crack detection method to overcome these shortcomings. The contributions of this paper are as follows:

- We propose a CNN networks for crack detection and demonstrate its superior performance through comparisons with previous methods.
- In the weakly annotated dataset environment, we propose a training strategy that combines a ternary classifier, to detect precise and accurate crack results.

The remainder of this paper is as follows. Section 2 describes the traditional and CNN crack detection methods and identifies the limitations thereof. Section 3 describes the proposed method. The corresponding experimental results are described in Section 4. Finally, Section 5 concludes the paper, reviewing the results obtained in Section 4.

2. Related Work

2.1 Crack Detection Methods based on Traditional Image-Processing Techniques

In terms of traditional crack detection methods employing image processing, numerous methods using the intensity differences between cracks and their surroundings have been studied. A representative approach for detecting cracks in concrete is that of using the image edges [7], [8], [9], [10]. These edge-based methods identify the darker intensity values of cracks compared to the surrounding material and employ edge information obtained using, for instance, Sobel or Canny edge-detection procedures. Furthermore, numerous methods based on domain transformations such as Fourier and Wavelet transforms have been studied [11], [12], [13]. However, methods based on domain conversions and edge detection only function well in a restricted environment, such as by setting a threshold according to the contrast and the crack. To solve this problem, model- and fuzzy-based methods have been proposed [14], [15], [16], [17]; these aim to distinguish noises and cracks in existing edge-based methods and minimize crack breakage. However, this requires repeated operation depending on the environment; thus, they are disadvantaged in that their execution time is long, and they are only effective in restricted environments.

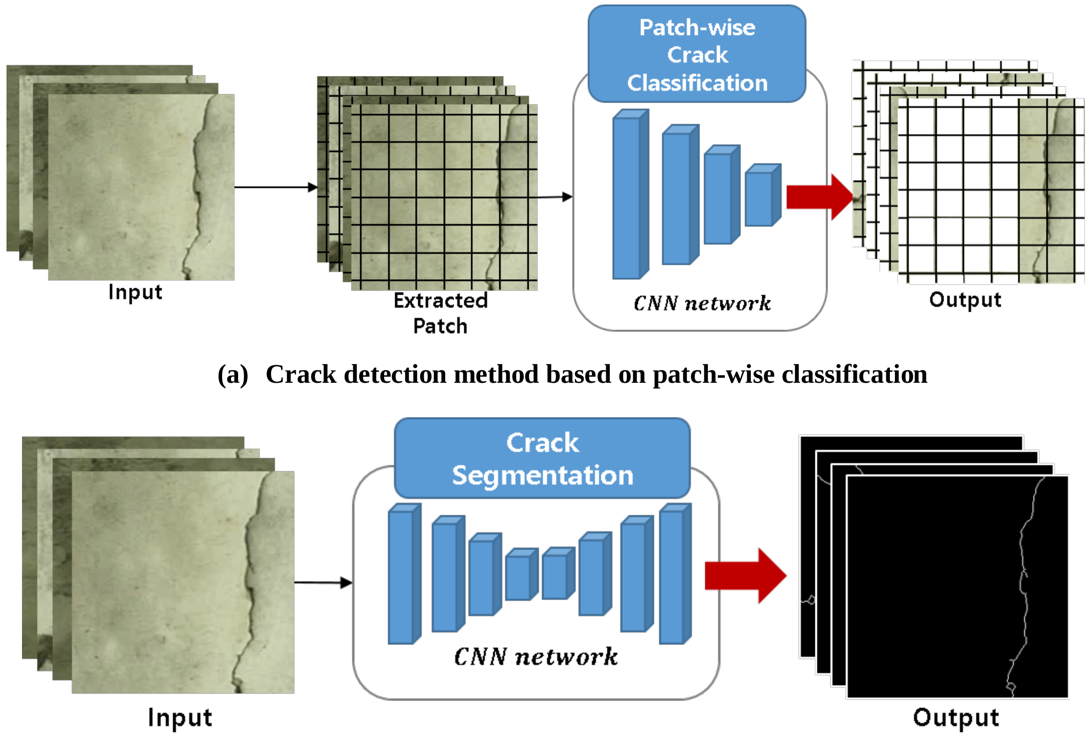
2.2 Crack Detection Methods based on Convolutional Neural Networks

CNNs are a form of feedforward network, they are widely used in many computer vision fields, such as face detection and recognition, image segmentation, and image classification. They mimic the visual processing of creatures and are capable of performing recognition tasks even when the sizes and positions of the patterns change. To solve problems in the computer vision field, it is necessary to design and extract the features of images; however, CNNs can automatically extract the features of objects in an image. As a result, CNNs are being used to solve many computer vision problems.

Crack detection approaches using CNN can be largely divided into two types: patch classification-based and image segmentation-based methods. **Fig. 1** illustrates both methods.

Patch classification-based crack detection methods [4], [5], [18] convert the input image into patch units and then classify the crack image. These methods have the advantage that the resources consumed in generating crack ground truth data are relatively small. However, to accurately detect the crack, the size of the patch must be set experimentally. Furthermore, when detecting the position of the crack in the patch, post-processing is required, and the patch classification-based methods suffer from the class-imbalance problem.

Compared to patch-based methods, image segmentation-based methods [3], [6], [19] have the advantage of identifying the crack area at the pixel level. This helps identify the actual crack characteristics (thickness, length, and overall depth). However, compared to the patch-based method, the network depth required is relatively large, and there are further disadvantages in that more resources are consumed in generating ground truth data. Moreover, the image segmentation-based methods suffer from the same problems of data imbalance as the patch-based methods.



(a) Crack detection method based on patch-wise classification

(b) Crack detection method based on binary class image segmentation

Fig. 1. Crack detection methods based on classification and segmentation

2.3 Class-Imbalance Problem

Class-imbalance problems have many adverse effects in deep learning. For example, if a severe data imbalance occurs in supervised learning-based network training, data with a small number of classes may be excluded from the final prediction stage or may cause noise. To resolve these problems, multi-task [20], semi-supervised [21], and weakly supervised learning methods [22] have been proposed.

Class imbalances in crack detection also cause serious problems, leading to network predictions of zero (background class) in the training stage and inference. This is because the cracks on the concrete surface are extremely small and thus occupy only a small proportion of the entire image. To solve this problem, previous crack detection methods have used a label weight [23], [6], [19], [20] determined by the number of crack containing pixels in the ground truth data. The weight control, determined by the label weight during the training stage with only pure dataset, is stable. However, in the case of using both datasets, the class weight amplifies false positive. Therefore, while avoiding the zero conversion problem, to suppress the false positive method is required simultaneously.

3. Proposed Method

In this section, we describe the proposed method in detail. It consists of four elements: (1) the overall features of the crack images and collected datasets, (2) the network structure, (3) the loss function, and (4) the training strategy.

3.1 Overall Features of the Crack Images

The characteristics of images taken of tunnel-lining cracks are as follows:

1. On the surface of the tunnel lining, cracks have lower intensity values than the surrounding concrete. However, cracks that differ from their surroundings by a relatively small intensity (as a result of soot pollution around the crack) also occur.

2. Although differences can arise as a result of the distance between the tunnel lining and imaging device, most cracks are small and thin. Cracks on tunnel-lining surface images taken from a large distance have a thickness of 1–4 pixels, cracks in tunnel-ceiling images taken by an imaging device on the road have a thickness of 1–2 pixels. This is an extremely small proportion of the entire image.

3. Cracks may appear differently depending on the imaging technique employed and the shape of the tunnel-lining surface. Flaking/exfoliating may also occur in or around the cracks.

Fig. 2 shows the RGB values of the intensity line scan of two crack images. The purple box represents the point of contact between the scan line and the crack. The upper left-hand panel of **Fig. 2** shows a heavily soot-contaminated crack in the surface of the tunnel lining. As previously described in the characteristics of crack 1, it can be seen that in the left-hand purple box, the differences between the surrounding pixel values and the crack pixel values are small due to contamination. Furthermore, the right-hand box has a lower intensity value than the left-hand one; as described above, a single crack may exhibit characteristics of differing intensities. The bottom left-hand panel of **Fig. 2**. shows a soot-polluted concrete image. Compared to the upper left-hand image, the degree of contamination is relatively small; thus, the crack is visible to the naked eye, and the intensity difference of the crack is clear (as can be seen from the intensity distribution of the scan line). In the image, the left-hand purple box can be seen as a crack; however, in the results of the scan line, there are two minimum values. This is due to the presence of noise around the crack.

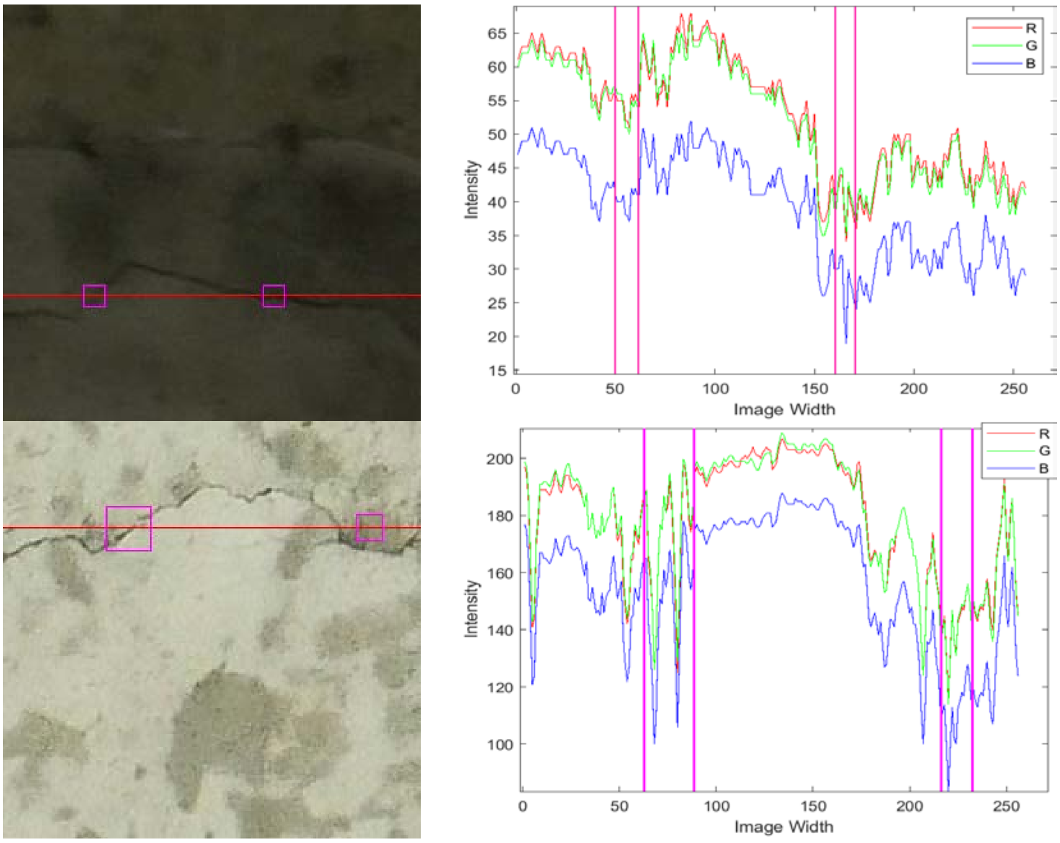


Fig. 2. Intensity values of the line scan

3.1.1 Data Collection

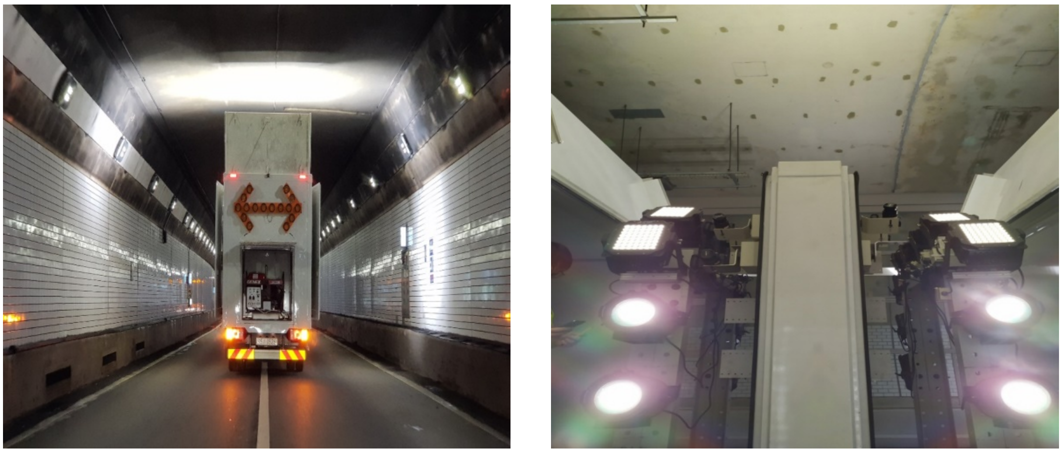


Fig. 3. Image-data collection vehicle (tunnel-lining scanning)

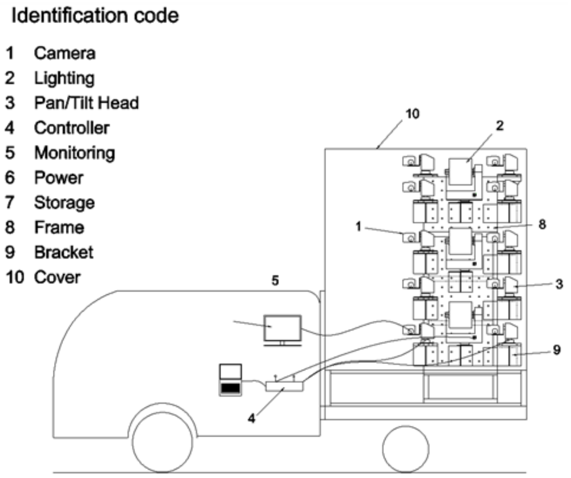


Fig. 4. Design of the scanning vehicle

Capturing the tunnel lining is the most important task in crack detection. The imaging machine must satisfy various conditions for obtaining a clear crack image. In addition, scanning of the tunnel lining can cause traffic congestion, thus requiring special vehicles. In order to satisfy the laws and regulations, to ensure the safety of the photographer and to shoot the lining accurately, we have produced a vehicle that meets all requirements [24]. The **Fig. 3**, **Fig. 4** show that the scanning vehicle and the modules of the scanning vehicle. For imaging the tunnel lining, the 24 of cameras were used. The **Table 1** shows the camera setting.

Table 1. Camera settings

Division / Shooting Area	Camera Set-up			Vehicle Speed
	ISO	Shutter	Diaphragm	
0.3 m × 0.3 m (2.6 m)	800–1,600	1/4000 –1/5000	2.8–5.6	20 km/h

3.1.2 Data Annotation

The collected images were annotated and the data divided into two categories according to the annotator's annotation format. The first was the case of pixel-by-pixel annotation. This was conducted by a safety expert, and the annotations referred to the presence and thicknesses of cracks. For convenience, this dataset is referred to as the "pure dataset" in this paper. The second type of data was coarse data, which indicated only the presence or absence of cracks; it was annotated by the public. This data did not contain the thicknesses of internal cracks and only tagged their approximate locations. The annotated pixel thicknesses were typically 7–13 pixels, and the cracks' center lines and annotations were not matched. This dataset is hereafter referred to as the "coarse dataset." Both datasets were used in this paper; they were produced in highly different proportions depending on the time and resources available for annotation work. For the pure data, an inspector measured the thicknesses of cracks in real tunnels and annotated the final pixel-wise image for the cracks identified. **Fig. 5** presents an example of the annotated crack dataset: (a) shows an example from the pure dataset, the crack is one pixel thick and the line follows it clearly; (b) shows an example from the coarse dataset, the annotation crack is thicker than the real crack of original input image.

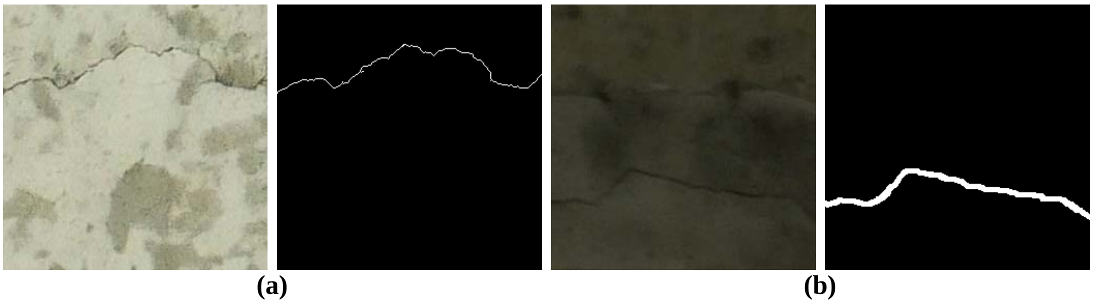


Fig. 5. Captured image and ground truth
(a) Pure dataset (b) Coarse dataset

3.2 Crack Detection Network Architecture

As previously mentioned, the class imbalance of the pure dataset results in a zero conversion, and using the both dataset results in a large false positive. To resolve these issues, we propose a crack detection method employing a crack detector and a ternary classifier. The overall flow of our method is shown in **Fig. 6**.

Our crack detection scheme was inspired by U-Net [25] based semantic segmentation networks, which are widely used in segmentation tasks. Other existing crack detection methods have followed the U-Net schema; that is, they have used local and global information extracted from each layer of the encoder and its end, respectively. The extracted features are fused in the decoder at each layer. Our detector was also built around this mechanism, although we changed the sequential decoders into parallel structures for faster inference.

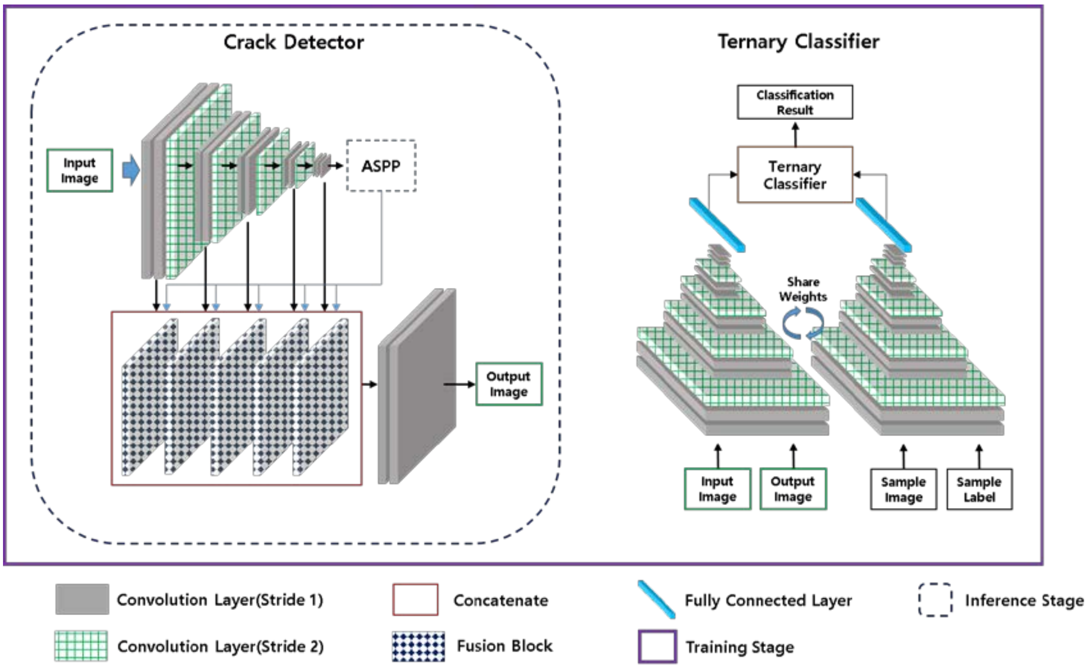


Fig. 6. Overview of proposed method

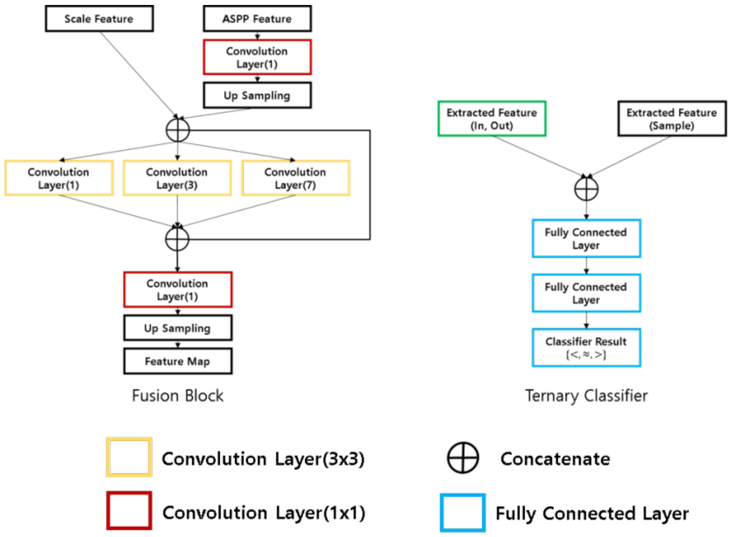


Fig. 7. Detail of the fusion block and the ternary classifier
The (n) in convolution layer of the fusion block denotes dilated rate of dilated convolution

The structure of our encoder follows the basic structure of VGG [26] with Short-cut mechanism [27], and the structure of our decoder uses one fusion block for each scale. The last convolution block of the encoder contains the global information of the input image. In the proposed method, an Atrous Spatial Pyramid Pooling (ASPP) [28] module was added at the end of the encoder to extract rich global information via an atrous convolution, and it was also added to each decoder layer. The fusion block of each layer fuses the global and local information; finally, the combined information generates results through two convolution blocks. The feature extractor of ternary classifier is built same as encoder of crack detector and the ternary classifier is consist of three fully connected layers. The detail of fusion block and ternary classifier is shown in The **Fig. 7**.

3.3 Loss Function

Given a training dataset containing N images as $S = \{(X^n, Y^n, Z^n), n = 1, \ldots, N\}$, where $X^n = \{x_i^{(n)}, i = 1, \ldots, I\}$ denotes the raw input image; $Y^n = \{y_i^{(n)}, i = 1, \ldots, I, y_i^{(n)} \in \{0,1\}\}\}$ denotes the ground truth corresponding to X^n; $Z^n = \{z_i^{(n)}, \in \{B, P, C\}\}$ denotes the ground truth belonging to; $\{B, P, C\}$ refers to the background, pure, and coarse datasets, respectively; I denotes the number of pixels in every image; and K denotes the decoder layer of the crack detector D, the extracted feature information of each decoder layer can be formulated as $F^{(k)} = \left\{f_i^{(k)}, i = 1, \ldots, I\right\}$. Furthermore, the result of the crack detector can be defined as $F^{(fuse)} = \{f_i^k, i = 1, \ldots, I\}$.

The crack detection problem is thus converted into one of binary classification, where the proportion of cracks in the whole dataset is extremely small. To overcome the data-imbalance problem, we adopted a weighted cross-entropy loss to measure the difference in layer information [6],[19],[20] :

$$l(F_i; W_D) = -\sum_{i \in Crack} Weight_0 * log\, Pr(F_i = 0|X_i, W_D) - \sum_{i \in Background} Weight_1 * log\, Pr(F_i = 1|X_i, W_D), \qquad (1)$$

$$L(W_G) = \sum_{i=1}^{I}\left(\sum_{k=1}^{K} l\big(F_i^{(k)};W_D\big) + l\big(F_i^{fuse};W_D\big)\right), \tag{2}$$

where $Pr(\cdot)$ refers to the probability of a positive or negative value for a pixel in the predicted map; W_D denotes the crack detector network; and $Weight_0$ and $Weight_1$ denote the crack and background class weights, respectively. Let C_0 and C_1 be the total number of background and crack, we set $Weight_0 = 1.0$ and $Weight_1 = \frac{C_0}{C_1}$ same as [20].

The ternary classifier converts the problem into one of 3-class classification [29]. When two variables x and y exist, the relationships between them are expressed as follows:

$$\begin{array}{ll} x > y, & if\ x - y > \tau \\ x \approx y, & if\ |x - y| \leq \tau \\ x < y, & if\ x - y < \tau \end{array} \tag{3}$$

where τ is the threshold. The relationships between samples $z^{(ref)}$ and $z^{(tar)}$ can also be expressed in a similar way:

$$\begin{array}{ll} z^{(ref)} < z^{(tar)}, & if\ z^{(ref)} \in P, z^{(tar)} \in C \\ z^{(ref)} < z^{(tar)}, & if\ z^{(ref)} \in B, z^{(tar)} \in C \\ z^{(ref)} < z^{(tar)}, & if\ z^{(ref)} \in B, z^{(tar)} \in P \\ z^{(ref)} \approx z^{(tar)}, & if\ z^{(ref)} = z^{(tar)} \end{array} \tag{4}$$

We use $q^{ref,tar}$ to denote the ground truth relationship between $z^{(ref)}$ and $z^{(tar)}$, and $p^{ref,tar}$ to denote the predicted relationship from the ternary classifier. The distance can be expressed as

$$\text{Distance} = L(W_{Ter}) = \sum_{i=0}^{ref}\sum_{j=0}^{tar}\sum_{k=0}^{2} q_k^{i,j} \log p_k^{i,j}, \tag{5}$$

where $q^{ref,tar} = \{(q_0^{ref,tar}, q_1^{ref,tar}, q_2^{ref,tar}), q^{ref,tar} \in \{0,1\}\}$ and $p^{ref,tar} = \{\,(p_0^{ref,tar}, p_1^{ref,tar}, p_2^{ref,tar}), p^{ref,tar} \in \{0,1\}\}$. Therefore, the crack detector loss function can be expressed as

$$L_{Crack\_Detector} = \lambda L(W_D) + (1-\lambda)L(W_{Ter}). \tag{6}$$

3.4 Training Strategy

The proposed training method consisted of two steps. The first was to train the two networks separately. The purpose of this step was to train the detector to recognize the features of the cracks and the classifier to determine their thicknesses and lengths from each ground truth. For this stage, the crack detector was trained using the dilated pure data, and the ternary classifier was trained using the entire dataset. A morphology operation was used to generate the expanded ground truths of the pure dataset.

We trained using the dilated pure ground truths, to avoid the zero conversions that can occur when only the pure dataset is used. It can be seen that the same occurs in coarse data. However, if the coarse data are used in the first step, the detector is trained the large number of false positives. In a similar way, training with eroded coarse data means that the detector is trained with incorrect labels; as such, it cannot guarantee that the eroded label contains the crack. The ternary classifier can control the large false positive rate in first step; however, the time taken to explore optimal weights was larger with the coarse dataset.

In the second stage, the crack detector was trained on the entire dataset using the ternary classifier. In this stage, the ternary classifier classified the relationships between the detector results and dataset samples. The classifier suppressed false positives and prevented the zero-conversion training problem of the detector. **Table 2** shows the second stage ground truth table of the ternary classifier.

Table 2. Ternary classifier ground truth table

Reference	Target	Expression	Meaning
Pure	Pure	$\approx$	Preserve
	Coarse	$<$	Thin
	Background	$>$	Thick
Coarse	Pure	$\approx$	Thin
	Coarse	$<$	Thin
	Background	$>$	Thick
Background	Pure	$<$	Thin
	Coarse	$<$	Thin
	Background	$\approx$	Preserve

For example, the reference image ground truth set was obtained from the coarse dataset, and the target was selected therefrom. The actual relationship between the two data was $z^{(ref)} \approx z^{(tar)}$. Because the desired result of the crack detector is “pure,” it was trained based on the relationship $z^{(ref)} < z^{(tar)}$. Furthermore, when the reference is from the background dataset and the target is from the pure one, the result of the detector is “background.” Thus, the detector was trained to satisfy $z^{(ref)} < z^{(tar)}$ and minimize the noise.

4. Experiments and Results

This section discusses the crack detection experiments. First, we describe the dataset and experimental settings; then, we compare the experimental results of the proposed method with those of existing crack detection methods. Finally, we compare and analyze the performances achieved using different settings for the proposed method.

4.1 Experimental Settings

1) Implementation: We implemented our network using Pytorch, a popular public deep-learning framework. A normalization layer was used between the convolutional layers of the encoder, decoder, and ternary classifier, and group normalization was applied to reinforce the learning of small batch sizes. The number of groups was set to 16. All convolution weights were initialized using the method developed by Kaiming He [30]. For up-sampling, bilinear interpolation was applied. Adam optimization [31] was used and the learning rate was set as 1e-4. For the dilation operation of the first training step, we used disk and five-pixel structure elements, the weight value of the loss function between the crack detector and ternary classifier was set to 0.5. The network was trained using eight images per mini-batch. The beta and weight decay used were 0.9 and 0.0005, respectively, and the networks were trained for 50 epochs. In this study, all networks were trained on a single NVIDIA TITAN-RTX.

2) Dataset: Datasets were taken from a total of five tunnels and used to compare the proposed method against existing ones. The dataset consisted of three small pure datasets and two large coarse datasets. **Table 3** describes each dataset.

Table 3. Dataset specifications

Dataset	Resolution	Dataset Size (Crack/ Background)	Annotation Style
Masung	256 × 256	1,857 / 13,798	Pure
Habuncheon	256 × 256	672 / 970	Pure
Sangock	256 × 256	408 / 847	Pure
Banggyo	256 × 256	5,015 / 34,258	Coarse
Gwangji	256 × 256	1,120 / 3,256	Coarse
Hwasan	256 × 256	6,300 / 15,727	Coarse

3) Metric: We used the recall, precision, F1-score, and inference time to evaluate the crack detection methods.

4) Performance-comparison methods: We measured and compared the performance of the proposed method with those of previous methods. The proposed method was compared to deep learning-based semantic-segmentation methods (U-Net [25], Att-UNet [33], and Deeplab v3+ [32]) and crack detection methods (Han's method [3], Liu's method [6], and Zou's method [19]). The previous methods and the proposed method were trained using the datasets of Masung, Habuncheon, Banggyo, Gwangji, and Hwasan, and performance was measured based on the SanGock and Habuncheon tunnels.

4.2 Evaluation Results

Table 4. Quantitative evaluation of the test dataset

Network	Sangock			Habuncheon			Processing Time (second)
	Recall	Precision	F1-Score	Recall	Precision	F1-Score	
U-Net	0.85	0.20	0.32	0.75	0.16	0.26	0.01
Att-UNet	0.87	0.17	0.28	0.81	0.18	0.29	0.013
Deeplab v3+	0.84	0.22	0.35	0.80	0.22	0.35	0.045
Han's method	0.80	0.25	0.38	0.71	0.19	0.30	0.013
Liu's method	0.84	0.18	0.30	0.77	0.17	0.28	0.0086
Zou's method	0.85	0.22	0.35	0.77	0.14	0.24	0.01
Proposed Method(A)	0.92	0.20	0.33	0.83	0.18	0.30	0.028
Proposed Method(B)	0.88	0.47	0.61	0.75	0.45	0.56	0.028

4.2.1. Overall Performance Comparison with Previous Deep Learning-based Methods.

In Table 4, we compare the proposed method with the previous semantic-segmentation and crack detection methods, using the Sangock and Habuncheon tunnel test data. To evaluate the performance of the proposed crack detection model, the differences in results obtained in the presence (proposed method B) and absence (proposed method A) of a ternary classifier were measured. To analyze the tendency of the crack detectors according to the data sets, previous methods were trained without the ternary classifier. As can be seen in **Table 4**, the previous methods and proposed method (A) showed high recall performance. However, these methods also achieved low-precision results. This is because the coarse dataset included a wider range of crack classes than the pure dataset. In this case, the pure dataset ground truth cannot be accurately used with the coarse dataset in the training step, and the crack detector produces a false positive. The results of the proposed method (A) in both tunnels exhibited the highest recall compared to the previous methods; however, it also showed low-precision results. On the other hand, when both the detector and classifier of the proposed method were used, the performance was improved by 0.08, 0.29, and 0.33 in terms of the recall, precision, and F1-score for the Sangock tunnel, respectively, compared with the previous methods. Furthermore, the results of the proposed method exhibited performance improvements of 0.04, 0.16, 0.32 in terms of the recall, precision, and F1-score, respectively, compared to previous methods for the Habuncheon tunnel. When both networks of the proposed method were used, the performance showed average performance differences of -0.06, 0.27, and 0.27 in terms of recall, precision, and F1-score, respectively, compared to the crack detector alone. This is significantly higher in terms of precision than the detector alone. In **Table 3**, the processing time indicates the inference speed of each frame. It shows similar speeds according to the similarity of the structures. The result of Liu’s method showed the highest processing speed; this method was built using an encoder and several up-sampling layers. U-Net and Zou’s method showed an almost identical processing time, Zou’s method is based around U-Net and changes only the skip-connection mechanism. The Deeplab v3+ exhibited the lowest speed in the table, this network structure has the largest architecture. The inference speed of the proposed method was slower than that of almost all previous methods, but it was faster than that of Deeplab v3+. In terms of overall performance, the proposed method showed a slightly slower processing speed than the other methods; however, it shows greatly improved results.

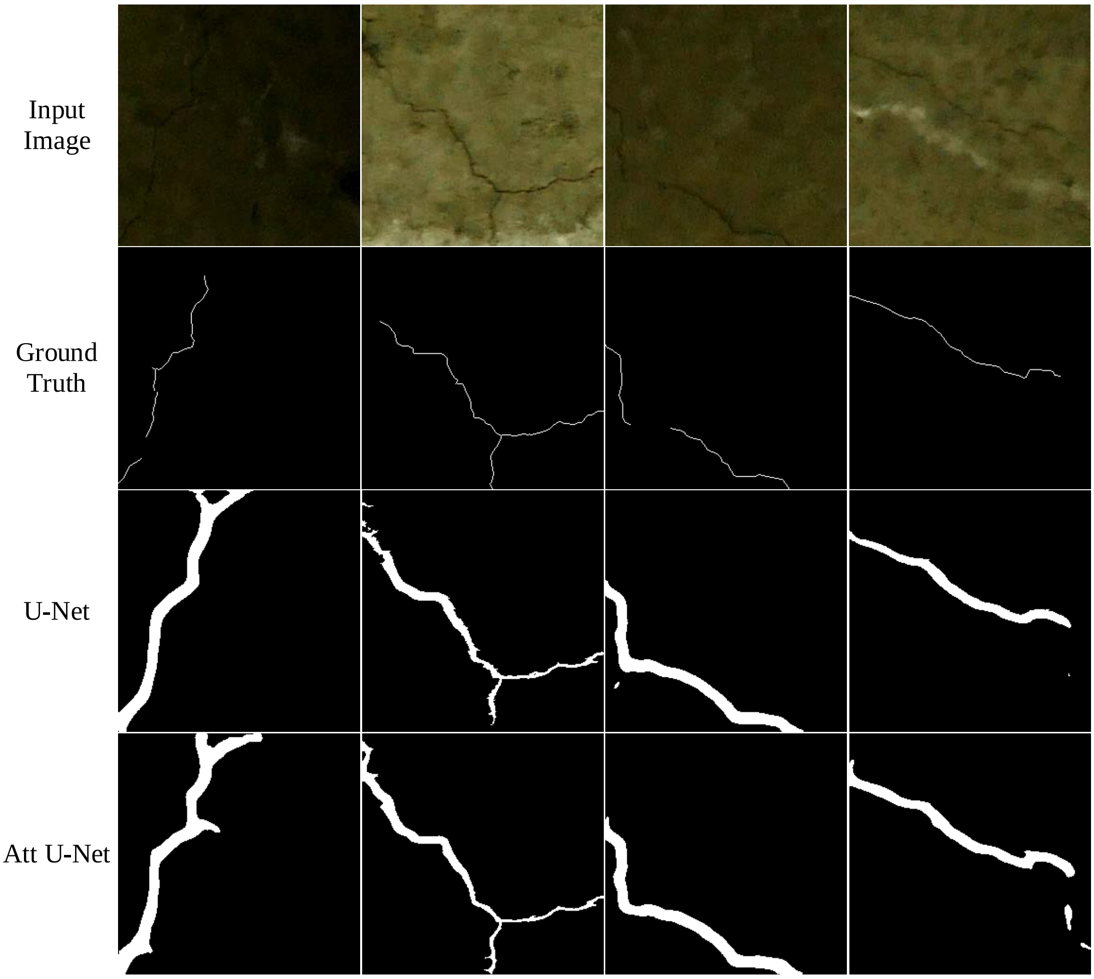
Figs. 8 and 9 show the crack detection results of the proposed method and the previous methods. The first row is the input image, and the second row is the ground truth. The ground truth image has a one-pixel crack thickness in **Figs. 8 and 9**. As previously mentioned, the ground truth was annotated by structure-safety experts, and the crack thickness was measured via naked-eye inspections.

In column (a) of **Fig. 8**, the input image exhibits low contrast. The results of previous methods and the proposed method (A) show a larger false positive rate than with the classifier and the ground truth. The results of previous methods show that the location of the crack is detected; however, three individual cracks are detected as one. On the other hand, the proposed method detected all cracks accurately; however, noise was detected at the top of the image. In the (b) column, all methods detected the middle and right-hand cracks but misdetected the left-hand soot as a crack. The results of our proposed method show a thinner crack than previous methods; however, the noise is misdetected on the left-hand side. In the middle of the image results, the proposed method and Deeplab v3+ show a disconnected crack. In the (c) column, the left- and right-hand cracks are disconnected in the ground truth. The results of the proposed method (A) and Han’s method show the disconnected cracks, the remainder all show the connected result. In the (d) column of **Fig. 8**, all methods detect the shear crack in the image; however, the results of Attention U-Net, Deeplab v3+, and the proposed method (B) show some noises along with the detected crack.

In column (a) of **Fig. 9**, the input image is blurred and the surface is discolored. The results of all methods (asides from the proposed method(B)) show a large false positive rate.

Furthermore, the results of Liu’s method show a misdetection result on the right-hand side of the image. In the (b) column of **Fig. 9**, the results of previous methods show thicker results than the proposed method, and Zou’s method shows the disconnected crack. The input image of column (c) contains soot and concrete formwork impressions. The input image contains a large quantity of noise; however, the results of Attention U-Net, Deeplab v3+, and the proposed method show the crack location more accurately than the others. In column (d) of **Fig. 9**, the input image contains three reinforcing bars. The previous methods detected the bars and cracks as one crack, without the proposed method. At the center of the crack (close to the reinforcing bar), soot is detected as a crack under the proposed method.

The results of the proposed method are shown in **Figs. 8** and **9**; they are significantly more precise than those of previous methods. These results show that the ternary classifier works effectively in crack detection. Previous methods detected the correct locations of cracks; however, their results showed large false positive, as occurs with the coarse dataset. Even though the results of column (b) in **Fig. 8** showed a difference in crack thickness between the left- and right-hand sides, these results were also thicker than the ground truth, and the remainder also exhibited a large false positive. On the contrary, the results of the proposed method (B) consistently show a thin and accurately detected crack. Thus, it can be seen that the ternary classifier controls the weights of the detector in the training step for all datasets.



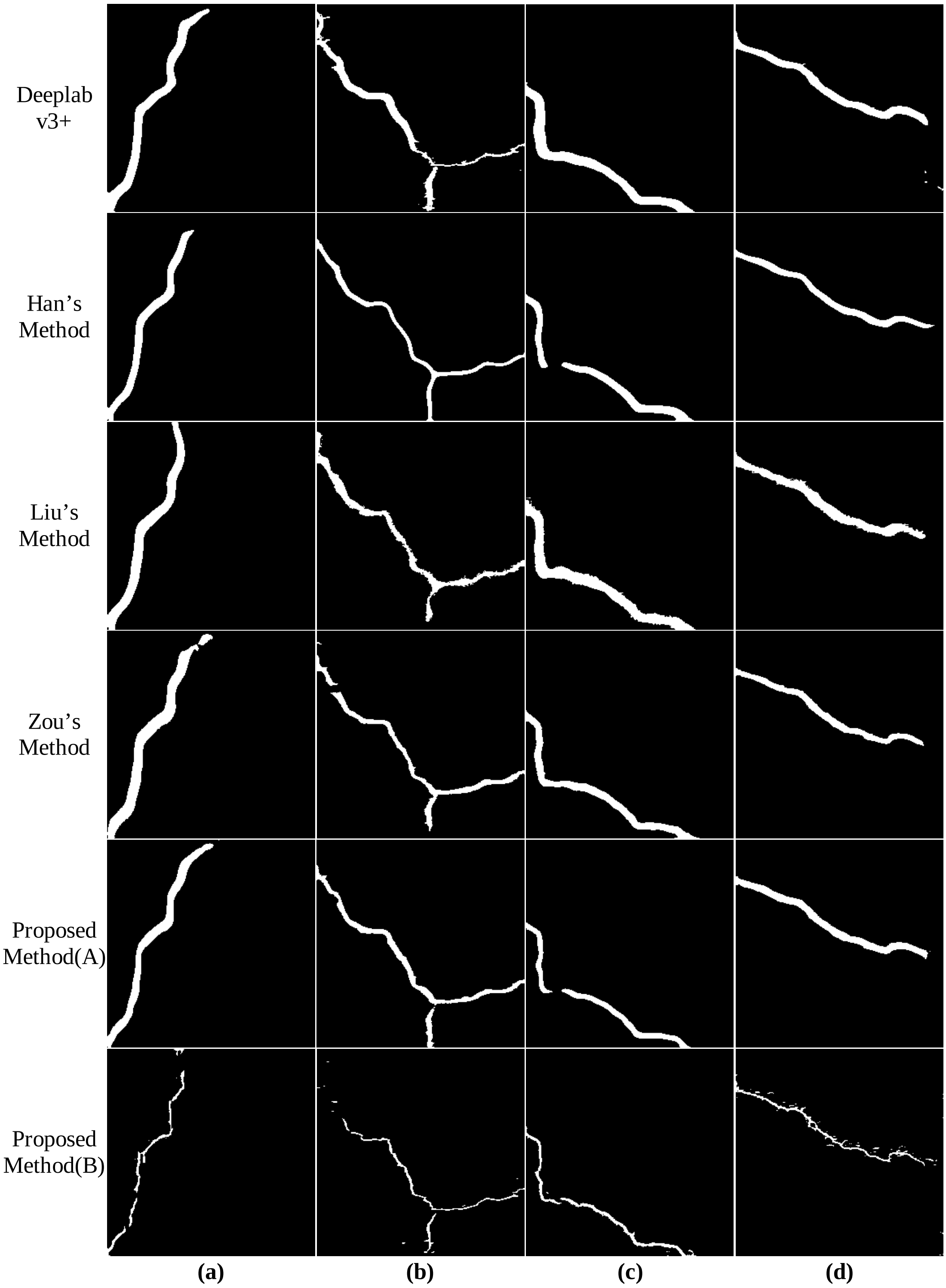
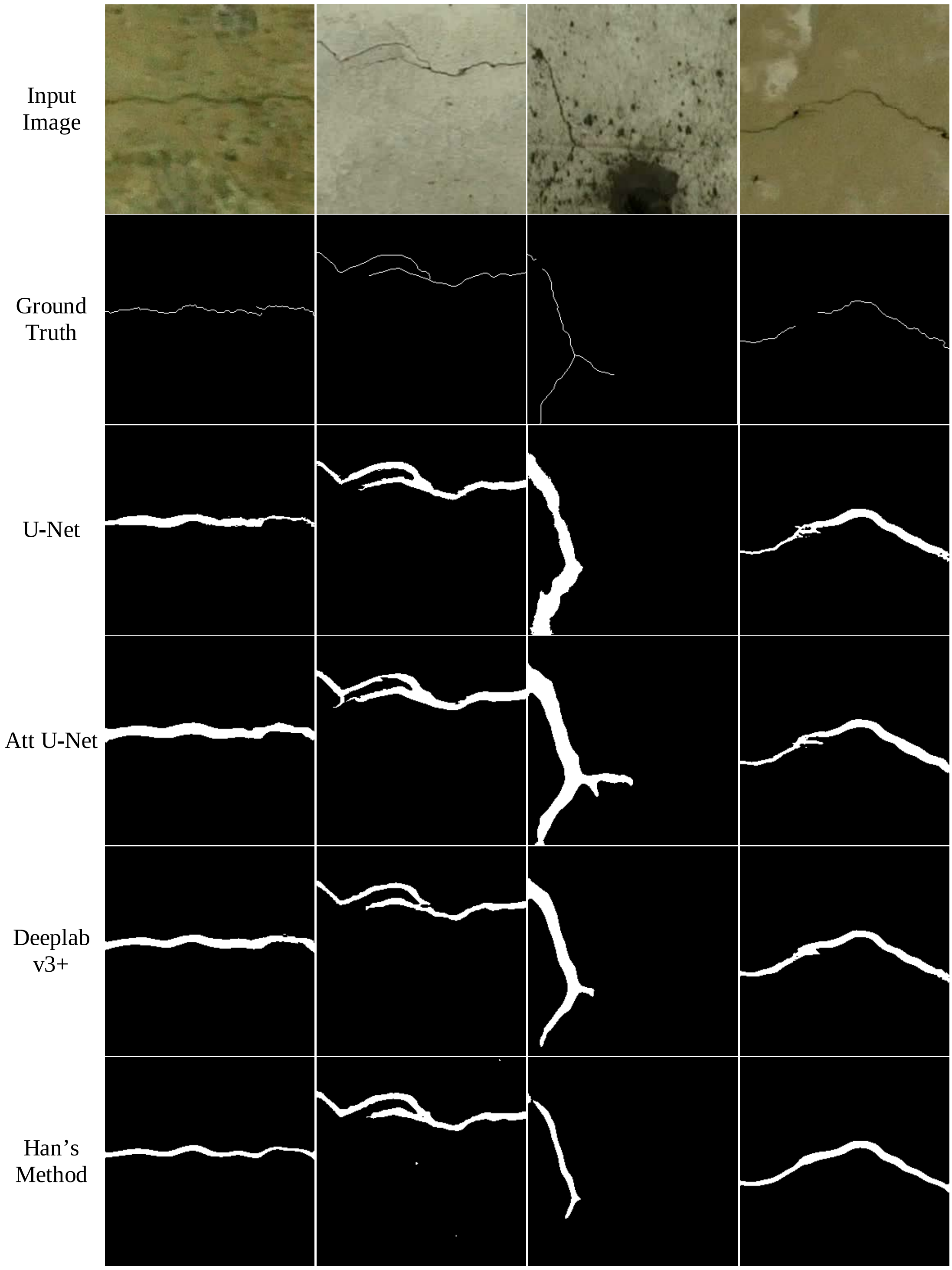


Fig. 8. Comparison of results obtained by different methods on four sample images from Sangock tunnel


Input Image
Ground Truth
U-Net
Att U-Net
Deeplab v3+
Han’s Method

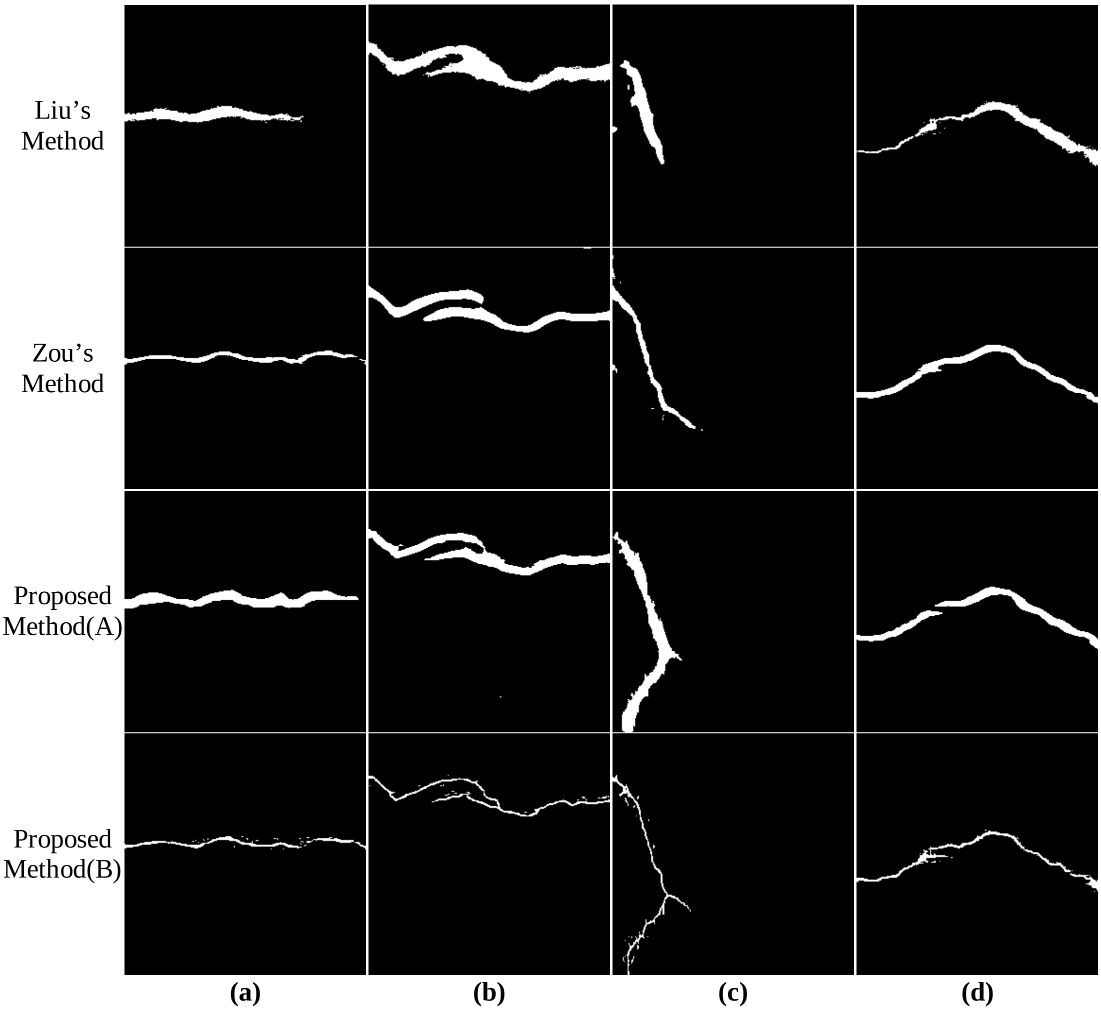


Fig. 9. Comparison of results obtained by different methods on four sample images from Habuncheon tunnel

4.2.2. Effects of Dilation on Pure Ground Truth Dataset.

In the first training stage, the crack detector was trained using only the dilated pure ground truths. The proportion of cracks in the ground truths of the pure dataset is extremely small it can result in zero conversion and may negatively impact the detector training in the second stage. To confirm the effects of dilated ground truths on the pure dataset, we conducted this experiment using different degrees of dilation. The detector of the proposed method used only, and it trained with Masung tunnel training set.

Table 5. Results of crack dilation experiment

Dilation	Recall	Precision	F1-score
None	0.08	0.78	0.15
3	0.22	0.45	0.3
5	0.4	0.31	0.35
7	0.62	0.23	0.34

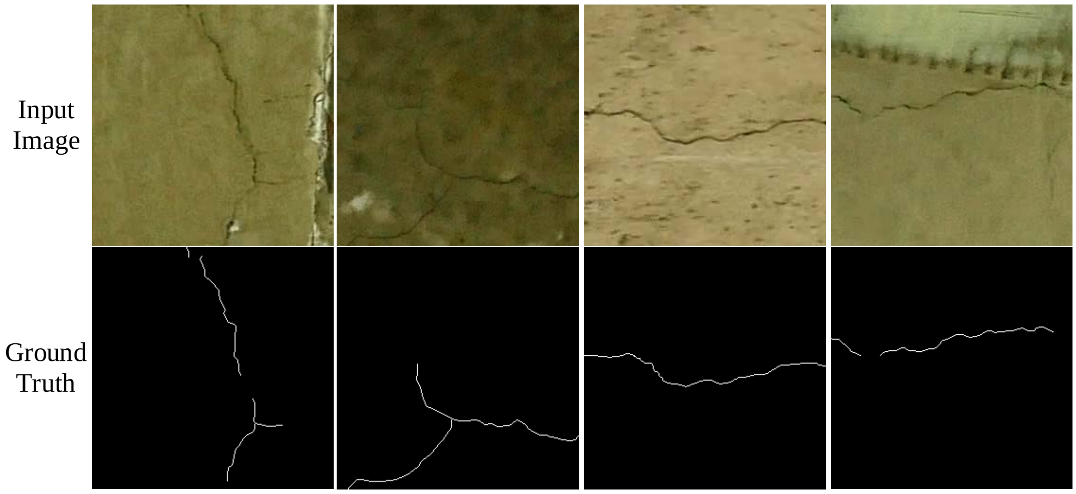
Table 5 shows the performance of the method (when trained using different dilation degrees) in terms of the recall, precision, and F1-score. As the dilation degree was increased, the performance of the recall increased, whereas the precision tended to decrease. For the detector training with a ground truth dilated by three pixels, the recall improved by 0.14, and the precision decreased by 0.33. Furthermore, when training with a ground truth dilated by seven pixels, the recall improved by up to 0.54, but the precision decreased by 0.55. This shows that the crack detection results using the dilated ground truths of the pure dataset exist in a trade-off relationship, and training the detector using only the dilated pure dataset is unsuitable.

4.2.3. Effect of the Weightings between Detector and Classifier. Thus far, the results of the experiments above highlight the role of the ternary classifier as a suppression controller in the training step. To confirm to what extent the ternary classifier affects the weighted detector (λ), we experimented using different weightings. In the training step, the weights control the sensitivity to crack thickness. In the first step of the experiment, 0.3, 0.5, and 0.7 weights were tested and a five-pixel disk structure element was used for dilation.

Table 5. Accuracy comparison for different weightings of the proposed method

Weight	Recall	Precision	F1-score
0.3	0.82	0.46	0.59
0.5	0.64	0.52	0.57
0.7	0.38	0.73	0.44

Table 6 shows the performance of the proposed method depending on the weightings. When the weight was set to 0.3, the recall showed the highest score; however, the precision was low. Conversely, when the weight was set to 0.7, the recall was lowest and the precision was highest. This suggests that the ternary classifier affects the crack detector according to weight, and both models exist in a trade-off relationship. When the weight was set to 0.3, 0.5, and 0.7, the performance improved compared to those of the previous methods in Table 3, and when the weight was set to 0.3, the detector exhibited optimal performance.



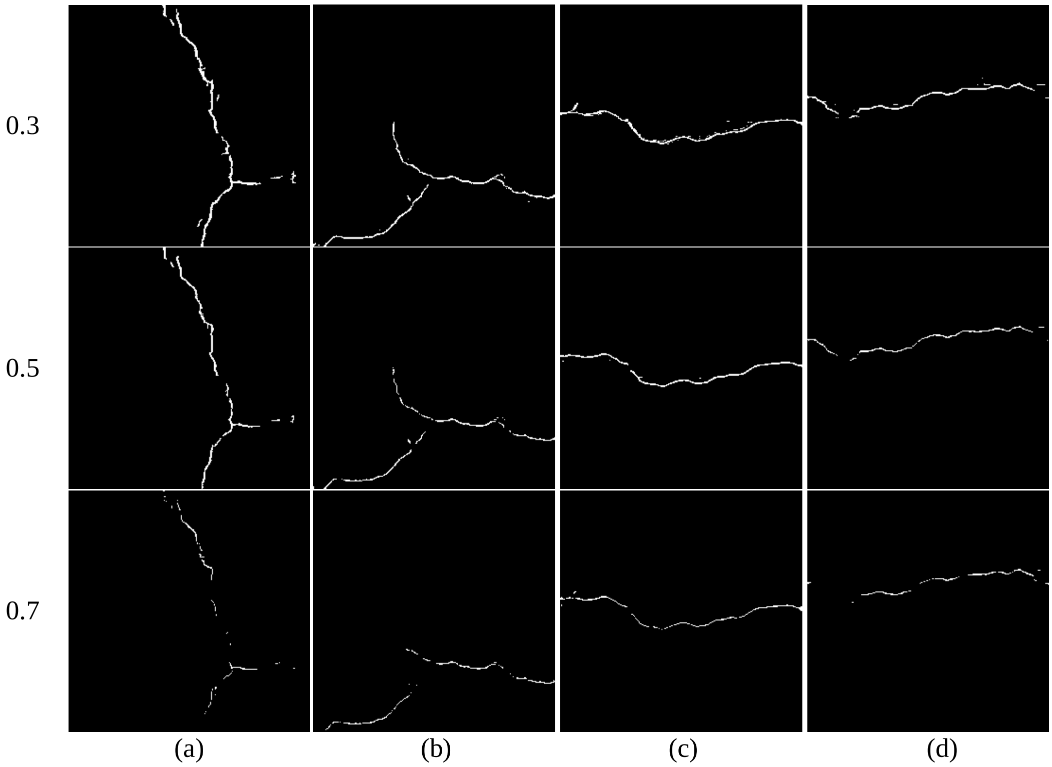


Fig. 10. Comparison of results obtained by different weights

Fig. 10 shows the crack detection results using different weights. When the weight was set to 0.3, the results were thicker than those found using 0.5 and 0.7 weights. This is due to the suppression ratio of the ternary classifier. Furthermore, under a weight of 0.7, the results showed the thinnest crack; however, the connectivity of the crack was broken. This suggests that the thickness of the crack is suppressed by the classifier. However, the result for 0.3 and 0.5 in **Fig. 10**, and the results of the (a), (b), (c), and (d) columns, show that the thickness is reduced but connectivity is preserved.

This experiment shows the effect of the weightings between detector and classifier. **Tables 3** and **5** show that the proposed method (using a ternary classifier) gives a better performance than previous methods; however, through this experiment, we confirm that the proposed method exists in a trade-off relationship, depending on the loss weight.

5. Conclusion

In this paper, we proposed a method of detecting cracks on the surface of tunnel linings. The network was designed based on image semantic segmentation and a ternary classifier, which was modified for faster inference speeds and improved performance. To solve the class-imbalance problem present in the acquired data, the proposed method used a dilated ground truth and ternary classifier, which improved the performance of the crack detector. Tunnel-lining images were collected from six tunnels in South Korea and used for training and evaluating the proposed method. The experimental results show that the performance of the proposed method is superior to that of the previous segmentation based and crack detection methods. In future research, we will endeavor to increase the performance and reliability of the algorithm by collecting more data. Furthermore, we will develop an algorithm that detects cracks, leaks, peeling flakes, and other defects that are dangerous to the safety of tunnels, based on the results of the method proposed here.

References

[1] Ministry of Land, Infrastructure and Transport, Department of Advanced Road Safety, "Road Bridge and Tunnel Status," *KOSIS Transportation · Information Communication*, 2018.

[2] Korea Infrastructure Safety Corporation, "Development of special specifications of crack evaluation method, repair, strengthening on concrete structures," *Korea Infrastructure Safety Corporation*, 1999.

[3] B. K. Han, H. S. Yang, J. M. Lee, and Y. S. Moon, "Crack Detection in Tunnel Using Convolutional Encoder-Decoder Network," *Journal of The Institute of Electronics and Information Engineers*, vol. 54, pp. 80-89, 2017. Article (CrossRefLink)

[4] C. J. Cha, W. R. Choi, and B. Oral, "Deep Learning-Based Crack Damage Detection Using Convolutional Neural Networks," *Computer-Aided Civil and Infrastructure Engineering*, vol. 32, pp. 361-378, 2017. Article (CrossRefLink)

[5] F. Chen and M. Jahanshahi, "NB-CNN: Deep Learning-based Crack Detection Using Convolutional Neural Network and Naïve Bayes Data Fusion," *IEEE Transactions on Industrial Electronics*, vol. 65, pp. 4392-4400, 2018. Article (CrossRefLink)

[6] Y. Liu, J. Yao, X. Lu, R. Xie, and L. Li, "DeepCrack: A deep hierarchical feature learning architecture for crack segmentation," *Neurocomputing*, vol. 338, pp. 139-153, 2019. Article (CrossRefLink)

[7] C. AbdelQader, O. Abudayyeh, and M. Kelly, "Analysis of edge-detection techniques for crack identification in bridges," *Journal of Computing in Civil Engineering*, vol. 17, no. 4, Oct. 2003. Article (CrossRefLink)

[8] Y. Fujita and Y. Hamamoto, "A robust automatic crack detection method from noisy concrete surfaces," *Machine Vision and Applications*, vol. 22, no. 2, pp. 245-254, 2011. Article (CrossRefLink)

[9] T. Nishikawa, J. Yoshida, T. Sugiyama, and Y. Fujino, "Concrete crack detection by multiple sequential image filtering," *Computer-Aided Civil and Infrastructure Engineering*, vol. 27, no. 1, pp. 29-47, 2012. Article (CrossRefLink)

[10] Y. R. Kim and T. M. Oh, "Multi-scale crack detection using scaling," *Journal of the Institute of Electronics and Information Engineers*, vol. 50, no. 9, pp. 194-199, Sep. 2013. Article (CrossRefLink)

[11] T. C. Hutchinson and Z. Chen, "Improved image analysis for evaluating concrete damage," *Journal of Computing in Civil Engineering*, vol. 20, no. 3, pp. 210-216, May, 2006. Article (CrossRefLink)

[12] H. Takeda, S. Koyama, K. Horiguchi, and T. Maruya, "Using image analysis and wavelet transform to detect cracks in concrete structures," *Report of Taise Technology Center*, no. 39, p. 25, 2006.

[13] H. G. Sohn, Y. M Lim, K. H. Yun, and G. H. Kim, "Monitoring crack changes in concrete structures," *Computer-Aided Civil and Infrastructure Engineering*, vol. 20, no. 1, pp. 52-61, 2005. Article (CrossRefLink)

[14] T. Yamaguchi, S. Nakamura, and S. Hashimoto, "An efficient crack detection method using percolation-based image processing," in *Proc. of the IEEE Conference on Industrial Electronics and Applications*, pp. 1875-1880, Jun. 2008. Article (CrossRefLink)

[15] T. Yamaguchi and S. Hashimoto, "Fast crack detection method for large-size concrete surface images using percolation-based image processing," *Machine Vision and Applications*, vol. 21, no. 5, pp. 797-809, 2010. Article (CrossRefLink)

[16] G. K. Choudhary and S. Dey, "Crack detection in concrete surfaces using image processing, fuzzy logic, and neural networks," in *Proc. of the IEEE International Conference on Advanced Computational Intelligence (ICACI)*, pp. 404-411, Oct. 2012. Article (CrossRefLink)

[17] Y. H. Noh, D. H. Koo, Y.M. Kang, D. G. Park, and D. H. Lee, "Automatic crack detection on concrete images using segmentation via fuzzy C-means clustering," in *Proc. of the International Conference on Applied System Innovation (ICASI)*, pp. 877-880, 2017. Article (CrossRefLink)

[18] Y. Xu, Y. Bao, J. Chen, W. Zuo, and H. Li, "Surface fatigue crack identification in steel box girder of bridges by a deep fusion convolutional neural network based on consumer-grade camera images," *Structural Health Monitoring*, vol.18, pp 653-674, 2018. Article (CrossRefLink)

[19] Q. Zou, Z. Zhang, Q. Li, and X. Qi, "DeepCrack: Learning Hierarchical Convolutional Features for Crack Detection," *IEEE Transactions on Image Processing*, vol. 28, pp. 1498-1512, 2019. Article (CrossRefLink)

[20] S. J. Park, W. J. Jeong, and Y. S. Moon, "X-ray Image Segmentation using Multi-task Learning," *KSII Transactions on Internet and Information Systems*, vol. 14, no. 3, pp. 1104-1120, 2020. Article (CrossRefLink)

[21] Y. Zhang, R. Yao, Q. Jiang, C. Zhang, and S. Wang, "Video Object Segmentation with Weakly Temporal Information," *KSII Transactions on Internet and Information Systems*, vol. 13, pp. 1434-1449, 2019. Article (CrossRefLink)

[22] C. Hu, X. Wu, and Z. Chu, "Bagging deep convolutional autoencoders trained with a mixture of real data and GAN-generated data," *KSII Transactions on Internet and Information Systems*, vol. 13, pp. 5427-5445, 2019. Article (CrossRefLink)

[23] S. Xie and Z. Tu, "Holistically-nested edge detection," in *Proc. of the IEEE International Conference on Computer Vision (ICCV)*, pp. 1395-1403, 2015. Article (CrossRefLink)

[24] I. S. Kim and C. H. Lee, "Development of Video Shooting System and Technique Enabling Detection of Micro Cracks in the Tunnel Lining while Driving," *Journal of the Korean Society of Hazard Mitigation*, vol. 18, pp. 217-229, 2018. Article (CrossRefLink)

[25] O. Ronneberger, P. Fischer, T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pp. 234-241, 2015. Article (CrossRefLink)

[26] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," in *Proc. of the International Conference on Learning Representations (ICLR)*, pp. 1-14, 2015. Article (CrossRefLink)

[27] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770- 778, 2016. Article (CrossRefLink)

[28] L. C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," *arXiv:1706.05587*, 2017. Article (CrossRefLink)

[29] K. S. Lim, N. H. Shin, Y. Y. Lee, and C. S. Kim, "Order Learning and Its Application to Age Estimation," in *Proc. of the International Conference on Learning Representations (ICLR)*, 2020. Article (CrossRefLink)

[30] Y. Wu and K. He, "Group normalization," in *Proc. of European Conference on Computer Vision (ECCV)*, 2018. Article (CrossRefLink)

[31] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," in *Proc. of the International Conference on Learning Representations (ICLR)*, 2015. Article (CrossRefLink)

[32] L. C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proc. of European Conference on Computer Vision (ECCV)*, pp. 801-818, 2018. Article (CrossRefLink)

[33] O. Oktay, J. Schlemper, L. Folgoc , M. Lee, M. Heinrich, K. Misawa, K. Mori, S. McDonagh, N. Y Hammerla, B. Kainz, B. Glocker, and D. Rueckert, "Attention U-Net: Learning Where to Look for the Pancreas," *Medical Imaging with Deep Learning*, 2018. Article (CrossRefLink)

Appendix

Automatic Crack Detection Method

In this appendix, we explain about the automatic crack detection method. The collected data using working vehicle is converted to the stitching image for generating whole tunnel image map. The image to be used for image stitching must be carefully selected as the camera may be out of focus when capturing the lining. One span image (one span ≈10m, one camera) usually has a resolution of 3000 × 38000 to 4000 × 45000 depending on the speed of the vehicle and the shape of the tunnel, and 24 stitching images (24 cameras). **Fig. 11** shows the single stitched image (single span, one camera)

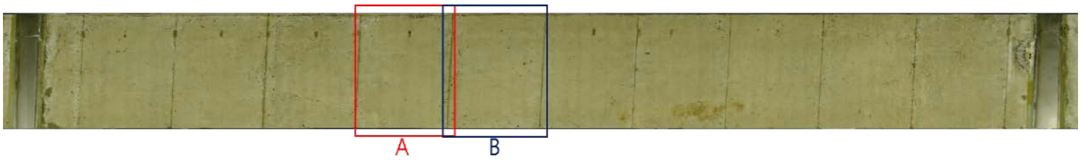
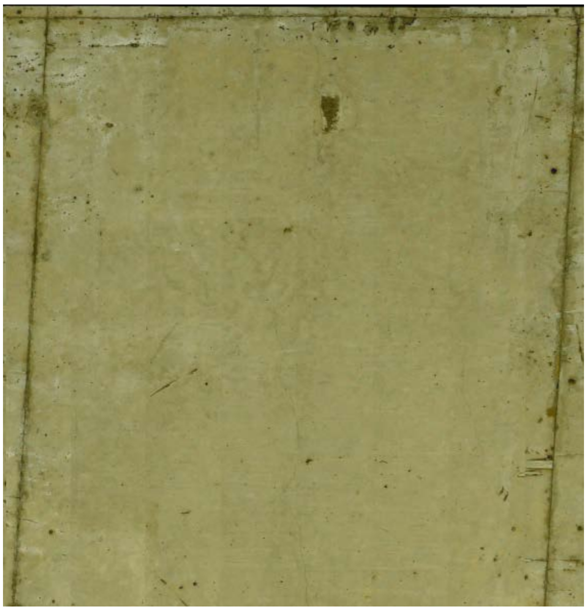


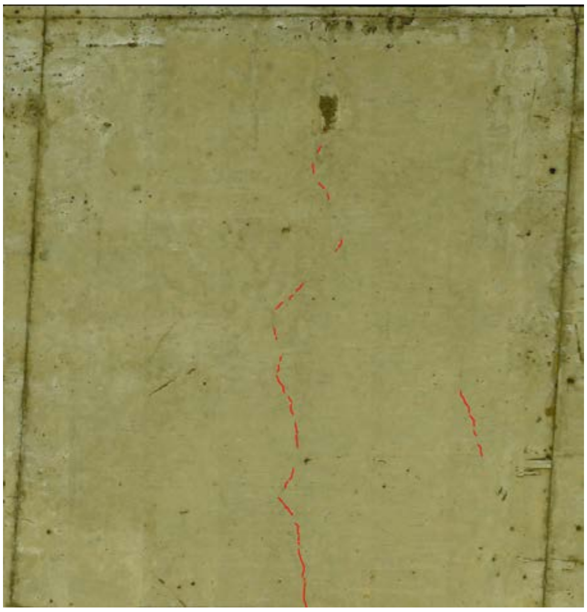
Fig. 11. Example of the single span stitched image

The resolution of a stitched image is too big to handle at once, the image divide into patch images (256 × 256 is used in this paper) and each patch is used for the crack detector in parallel way then they are used detection result. When the average tunnel length in South Korea is 739m, the execution time of each procedure must be considered and it is very important to analyze the shape of the crack showing the propagation direction of the crack. However, the basic information generation of a crack detection system using CNN is very expensive, and training a network using a coarse data set predicts the wrong crack results. Therefore, we proposed an efficient crack detection method using coarse and pure dataset generated at a low cost. **Fig. 12** is the prediction result of the proposed method using the images A and B of **Fig. 11**. The input data is sample span of Bongsan Tunnel in South Korea and the resolution of the A, B is 2880 × 3000. The results show that the shape of predicted crack and ground truth is simliar and the thickness is almost correct. When comparing the result with the previous method in **Figs. 8 and 9**, the efficiency of the proposed method can be confirmed.

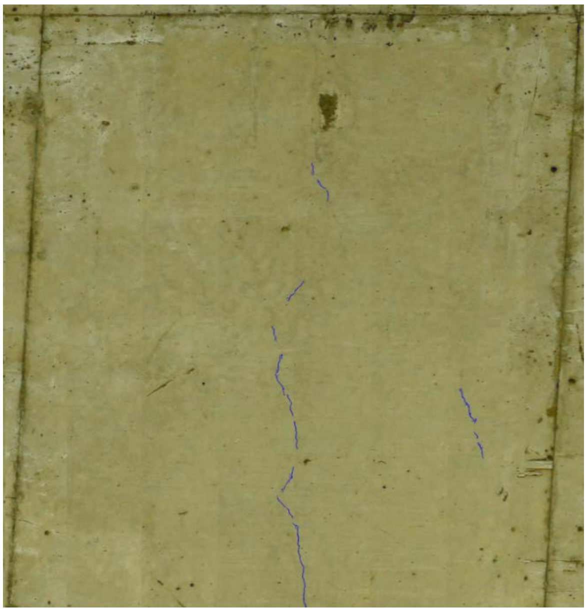
Input Image A



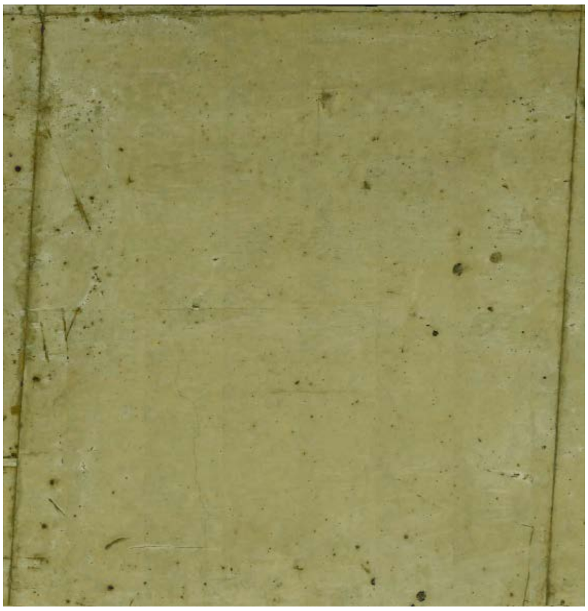
Ground Truth A

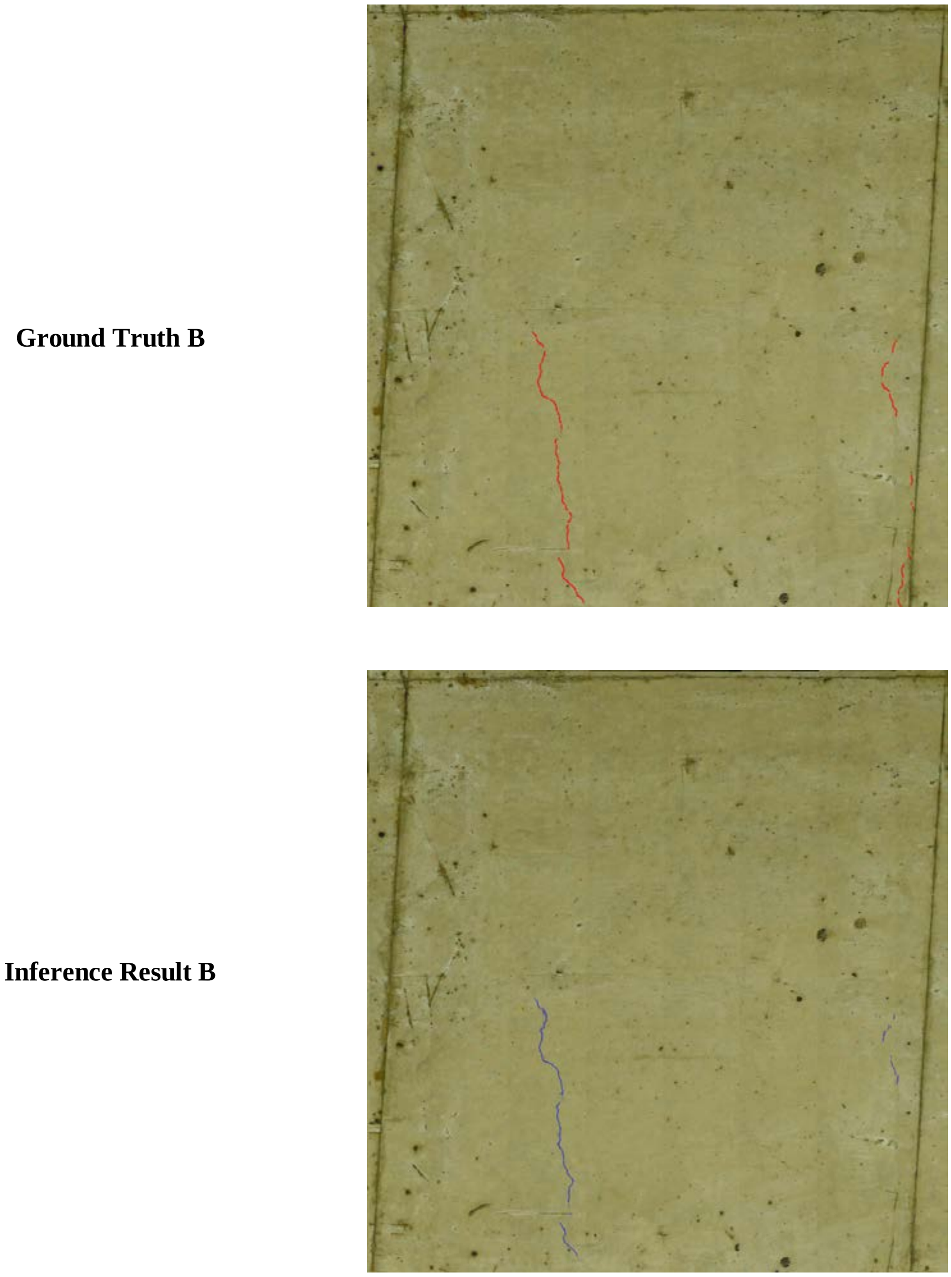


Inference Result A



Input Image B





Ground Truth B

Inference Result B

Fig. 12. Result of automatic crack detection

Jeong Hoon Han received his B.S. degree in the Department of Computer Science and Engineering from Hallym University, Korea, in 2016. He is currently working towards PhD. Degree at the Department of Computer Science and Engineering from Hanyang University, Korea, From 2016. His research interests include computer vision and machine learning.
Email : bghan@visionlab.or.kr

In Soo Kim received the B.S. and M.S. degrees in Civil Engineering from Chungbuk National University, Korea, in 1998 and 2000, respectively. From 2000 to 2003, he had been a researcher at the Korea Institute of Civil Engineering and Building Technology, Korea. From 2003 to 2015, he had been a researcher at the Hanmac Engineering, Korea. In 2015, he joined the institute of technology at Deep Inspection Inc., Korea, as a director of research.
Email : iskim@deepinspection.ai

Cheol Hee Lee received the B.S. and M.S. degrees in Civil Engineering from Chungbuk National University, Korea, in 1996 and 1999, respectively. He is currently working towards PhD. Degree at the Department of Civil Structure Engineering from Chungbuk National University, Korea, From 2005. From 1999 to 2004, he had been a researcher at the Unison, Korea. From 2006 to 2015, he had been a researcher at the Taemyung Engineering, Korea. In 2015, he founded Deep Inspection Inc., Korea, is currently the CEO.
Email : chlee@deepinspection.ai

Young Shik Moon received the B.S. and M.S. degrees in Electronics Engineering from Seoul National University and Korea Advanced Institute of Science and Technology, Korea, in 1980 and 1982, respectively, and PhD. degree in Electrical and Computer Engineering from the University of California at Irvine, CA, in 1990. From 1982 to 1985, he had been a researcher at the Electronics and Telecommunication Research Institute, Daejon, Korea. In 1992, he joined the Department of Computer Science and Engineering at Hanyang University, Korea, as an Assistant Professor, and is currently a Professor. Dr. Moon served as General Chair of 2014 IEEE International Symposium on Consumer Electronics, and worked as the President of the Institute of Electronics and Information Engineer, Korea.
Email : ysmoon@hanyang.ac.kr