Tunnel Lining Crack Detection Method by Means of Deep Learning

Masato UKAI
Image Analysis and IT Laboratory, Signalling and Transport Information Technology Division

Existing image processing programs for detecting structural damage such as cracks have required the fine-tuning of numerous parameters and experience-based expertise. A method for distinguishing different types of cracks applying deep learning has been developed using tunnel lining images. A classifier was created after learning from a large volume of images in two groups – either with “presence of a crack” or “absence of a crack.” The classifier successfully recognized the presence or absence of cracks in images at a rate of more than 90%. Using a color-coded pixelated image to show the position of probable cracks, this paper proposes a hybrid detection method for analyzing cracks with a focus on their location and direction of progress.

Keywords: tunnel, crack detection, machine learning, deep learning, image processing

1. Introduction

Maintenance and inspection of railway structures is usually performed “visually,” creating issues relating to work efficiency and inspection accuracy. It is hoped that image processing technology will be available for use as an alternative to visual inspections for maintenance as soon as possible.

RTRI has proposed a method for detecting cracks and other structural damage based on shading and outlines in images [1] (hereinafter “Conventional Method”), which was developed by finding an optimized algorithm on a trial-and-error basis from experience. As such, deep learning, that has garnered increasing interest over the past few years, was explored as another possible approach to extract images showing cracks, and a method was developed that produces detection results close to those of the human eye [2] by combining it with a new image analysis technique. This paper describes the deep learning processes applied, and reports on a sample of detection results.

2. Conventional crack detection method

Conventional crack detection involves adjusting a vast number of image processing parameters which need to be readjusted every time the image changes, which is very time consuming. Figure 1 shows a series of tunnel lining photographs showing cables, joints, stains, etc. and cracks. In the Conventional Method it is difficult to remove noise in these images because cracks with similar shading and outlines may also be identified as structural damage.

3. Crack identification using deep learning

3.1 Overview of deep learning

Deep learning is based on Neural Networks, modeled after the human brain’s neural network. The network structure consists of multiple hidden layers sandwiched between input layers and output layers as shown in Fig. 2. Processes are generally divided into “learning” and “inference.” The learning process generates discriminative models by repetitively learning a large quantity of labeled images and thereby finding features such as regularities and patterns. In the inference process, when unknown images are given to the learned discriminative models, the probability scores of the labels are returned as the inference results. Thus, a computer will eventually be able to acquire feature quantities automatically and analyze the data autonomously even without explicit human programming.

(a) Original photographed image  (b) Conventional Method  (c) Desirable extraction result

Fig. 1 Crack detection using Conventional Method

Input layer  Hidden layer  Output layer

Crack

Fig. 2 Structure of Deep Neural Network (DNN)
3.2 Concept of learning and training data set

This section describes the training data used for learning. If the image is large, it is more likely to have a mixture of joints and cables, which makes sorting by feature unclear. Furthermore, subsequent processing becomes necessary to identify the position of cracks in the images. Also, since each image pixel is equivalent to one input in the input layer, the larger the size of the image, the larger the input dimension, and the greater the computational complexity. If the image is too small however, it becomes impossible to discriminate between cracks and noise. Consequently, tests were carried out to find the size of image which would allow anyone to see the “crack,” without mistake. As a result of these tests, an image with a 128 mm square area compressed at a resolution of 2 mm per pixel (the image size was 64 pixels by 64 pixels) was prepared of an image of a tunnel span (in the unit split by construction joints).

Images in this study were categorized into two classes: those with cracks ("Crack") and those without cracks ("No crack"). Two data sets were prepared: set A consisting of approximately 80,000 ‘Crack’ images and approximately 250,000 images with ‘No crack’; and set B consisting of approximately 40,000 ‘Crack’ images (containing only major cracks of about 0.5 mm or wider in a different tunnel) and approximately 120,000 images with ‘No crack.’ The data for validation was chosen from different span images to the training data. Figure 3 shows some examples of the Crack training data.

The multiplication of images by processing was studied to increase the ‘Crack’ training data. As a result of experimenting on such processing as rotation, inversion, scaling and luminance correction, sufficient data was prepared by applying the rotation and inversion which were effective in improving the learning performance and multiplying the images by 16.

![Fig. 3 Examples of training data (‘Crack’ images)](image)

3.3 Creation of deep learning models

Caffe, which is often used in image recognition, was used as the framework for the deep learning. Table 1 shows the created models. GoogLeNet was used as the Neural Network, to which NAG (Nesterov Accelerated Gradient) was used for the optimization algorithm in the gradient descent method to minimize the error function. Additionally, GPU (Graphics Processing Unit) was used to speed up learning and recognition processing.

![Table 1 Created models for deep learning](image)

| Model No. | Type 00 | Type 01 | Type 02 |
|-----------|---------|---------|---------|
| Training data set (A/B) | Crack A | 80,000 images | — | 120,000 images (00 + 01) |
| | B | — | 40,000 images | |
| No crack A | 250,000 images | — | 370,000 images (00 + 01) |
| B | — | 120,000 images | |

3.4 Results of inference using discriminative models

The discriminative model output the cracking probability normalized to between 0 and 1.0. When the output results were checked, it was found that the images contained most of the cracks to be extracted if the cracking probability was set to at least 70%. Figure 4 shows the inference results of the model Types 00 and 01. The higher the probability of cracking present in a cell, the thicker the lines drawn, making response to cracks visible at a glance. Results suggest that the fouling and cables extracted in the Conventional Method were correctly recognized as background.

![Fig. 4 Results of crack inference using discriminative models](image)

(a) Recognition result by Type 00  (b) Recognition result by Type 01

3.5 Speeding up of inference

The possibility of speeding up inference processing was explored using parallel computation using GPU. Figure 5 shows the graph plotting the processing time according to the batch size for parallelization. In the same figure, the
Fig. 5 Parallelization and recognition processing time

Forward Time indicates the processing time for the part which could be parallelized, and Total Time indicates the processing time for the whole. When the cell images were processed one by one, it took about 15 seconds to obtain inference for the whole span. The time was successfully reduced to approximately 10 seconds by parallelizing the image processing program and setting the batch size to about 30 parallels. The difference between the Forward Time and the Total Time indicates the processing of the part which could not be parallelized, such as output of a probability value. It was also found time reduction plateaued if the number of parallels was over 30.

3.6 Performance evaluation of discriminative models

An ROC (Receiver Operating Characteristic) curve [3] is often used to evaluate discriminative models in machine learning. When the quantity of data which actually has and is predicted to have cracks is called true positive (hereinafter, TP (True Positive)), the quantity of data which actually has no cracks and is predicted to have no cracks in the background is called true negative (hereinafter, TN (True Negative)), the quantity of data which actually has no cracks but is predicted to have cracks is called false positive (hereinafter, FP (False Positive)), and the quantity of data which actually has cracks but is predicted to be the background is called false negative (hereinafter, FN (False Negative)), an ROC curve is plotted on the horizontal axis of false positive rate = FP/(FP+TN) (hereinafter, FPR (False Positive Rate)) and on the vertical axis of true positive rate = TP/(TP+FN) (hereinafter, TPR (True Positive Rate)).

It seems intuitively good to use the accuracy rate = (TP+TN) / (TP+TN+FP+FN) as an evaluation index. However, if there is a variation in data quantity among the classes as in this case, the overall accuracy rate would increase if the accuracy rate for No ‘Crack’ data in a large quantity is high even though the accuracy rate of ‘Crack’ data in a small quantity is low, which would not lead to proper evaluation. On the other hand, an ROC curve has a feature that it is not affected by a variation in data quantity among the classes. Figures 6 and 7 show the ROC curves for the discrimination results of model Types 00 and 01.

Fig. 6 ROC curves of model type 00

Fig. 7 ROC curves of model type 01

The AUC (Area Under the Curve) based on these ROC curves was set as the evaluation standard for the discriminative models. The measure can range from 0 to 1. On full discrimination the area is one (1). In other words, it is considered that the larger the area, the better the prediction. Generally, an AUC values of between 0.7 and 0.9 indicates medium accuracy, whereas values over 0.9 indicate high accuracy. Accordingly, good results were obtained from all the discriminative models. However, attempts were made to distinguish even fine cracks of Type 00 which may have led to identification of stains by mistake. Therefore, a fall in the AUC was seen in the recognition of tunnel data set B, which was different from the dataset used for learning. On the other hand, Type 01 images seemed to be accurately labeled. A high AUC was observed similarly in the recognition of the different data set A. With Type 02, both datasets were re-learned using the learned recognition models. As a result, the AUC remained over 0.95, which indicated the validity of additional learning.
4. Discussion of inference results

While deep learning may produce highly accurate results, the drawback is that it does not show why the results are accurate nor what the rationale was for decisions. Although some argue that good accuracy is all that matters, it is also important to understand the rationale behind decisions. Various investigations were therefore conducted to elucidate this problem.

One investigation used the Grad-CAM [4] method, in which a magnitude of change in probability of output against input is visualized in a heat map, to identify elements in the images that could show what decisions are involved in the deep learning process. An evaluation is made by taking out the coefficient that represents the magnitude of change caused in the probability score when a minor change is added to the whole crack image. This method focuses on the fact that parts in the image that have most influence on classification decisions have a larger differential coefficient for the probability score.

A heat-map output program was designed for this study using Grad-CAM and used to perform an evaluation experiment. First, relatively clean images showing only cracks were processed among 'Crack' images. Figure 8 shows the visualization results. The upper images are the input images, and the lower images indicate the results of the visualization using Grad-CAM. The parts colored in red are the areas that have heaviest influence on the scores. In addition, their discrimination results indicated that the probabilities that they would be classified among Crack images were at least 98%. It was also found that the crack centers, around the edges and in the surrounding areas differed significantly in luminance from the background.

![Fig. 8 Grad-CAM results of 'Crack' images -1](image1.png)

![Fig. 9 Grad-CAM results of 'Crack' images -2](image2.png)

Next, from a number of 'Crack' images that included other objects in addition to the cracks were processed. Figure 9 shows the visualization results. It was found that there was no reaction to the noise of other objects such as cables and stains, whereas the cracked portions were properly identified. Good discrimination results were obtained, with probabilities that the image included a Crack being approximately 70–90%.

Figure 10 shows the results of processing False Negative images where the cracks were discriminated as background. The heat maps of both Crack and No crack images are shown below for confirmation. As the cracks were discriminated as background, while the cracks framed in red were not colored at all in the 'Crack' image heat maps, areas other than the cracks, such as stains and cables, were colored in the same way as in the No 'Crack' images. It seems that the cables and blackish stains in the image were observed and discriminated as “background.” It is true that the original images showed wide cracks. However, the final decision was made using discriminative models. Therefore, it was found that ambiguous labeling that suggested two results would lead to a decrease in recognition performance. The following countermeasures against the above issue can be considered:

- The volume of training data should be slightly reduced to avoid mixing cracks with other objects as far as possible.
- A “Pending” class of images that could fall into either category, should be added. These images can be re-sorted in the light of surrounding classes in the later-described detailed image analysis.

These countermeasures will be the focus of future study.

Using a visualization technique such as Grad-CAM, helped validate the rationale that “AI makes decisions by observing an area, although it is unknown why it observed that area.” Having identified the parts underlying the rationales for discrimination, it was considered that they could be utilized for creating training data to achieve a higher performance in crack detection, to study the learning methods and for other purposes.
5. Detailed image analysis to compensate for inference results

5.1 Detection of crack skeleton lines

Investigations into incorrect inference results indicated that in many cases some parts of a series of cracks could not be detected or were generated by the system reacting to blackish stains. For practical crack detection, it is important to perform post-processing, such as joining the broken cracks and removing isolated points. Therefore, a hybrid crack detection method was proposed for detailed analysis of shaded images by regarding cracking probabilities as pixel values.

The bitmapped images were generated with probabilities between 0 and 1.0 converted on a linear scale into between 0 and 255, and the images were analyzed by focusing on locations and directionalities. Since the cracks tended to be connected in the principal axis direction, the line components in proximity were connected after applying expansion processing, and then line segments shorter than the given length were removed. Furthermore, the remaining areas were contracted and smoothed, and then the skeleton lines were extracted. Thus a method was developed to identify them as final cracks. Figure 11 shows the results of crack detection using the proposed method. As shown in Fig. 12, the cracks behind the cables and fouling were correctly detected.

Figure 13 shows the same location in the hand-drawn interior elevation with structural damages and in the detection result using the proposed method. Their comparison indicates that the practical detection results using the proposed method were close to what a human could detect.

5.2 Detailed detection of cracks

In the method described in the preceding section, a crack is considered to be detected on the basis of skeleton lines in the cell area containing cracks. However, there is a need for detection results to also accurately show traced cracks. Therefore, the original images that correspond to the cells as a result of inferring the discriminative models were extracted, and a mode was designed to allow cracks to be detected using blocks. This type of image processing produces results which faithfully trace cracks, although the processing time somewhat increases, because the images can be corrected to those with the optimum contrast by absorbing the brightness difference with the blocks. Figure 14 shows the results of the detailed detection.
5.3 Crack measurements with sub-pixel accuracy

The developed crack detection algorithm [5] was improved: Excessive noise extraction was suppressed to ensure that significantly thicker cracks were extracted. First, the original images were smoothed, and then the images were split by dynamic threshold processing. The images were expanded after removing the particles smaller than the given area, and then the areas were narrowed down to search for cracks. Since the outline of a crack is located in the part which is darker inside compared to the surrounding area as shown in Fig. 15, a filter to extract the parts with higher differential values (a differential means a difference in luminance value in case of a digital image) in the image was designed to act on this search area, and the crack outlines were extracted as shown in Fig. 16. The sub-pixel processing enables measurements with approximately 10 times higher accuracy than in photography resolution. Figure 17 shows a case of measuring the maximum widths of cracked areas enclosed within outlines.

The three parameters (smoothing coefficients, and upper and lower limit thresholds of differential values) used in this filter are important values that affect detection performance. However, they were set on a trial-and-error basis, which was an issue for achieving stable detection. This method provided internal calculations from the maximum width of cracks to be extracted and the contrasts of images.

This method can contribute to automation of parameter tuning, which should make it easier to apply in practice. Finally, for cracks under observation that had connected adjoining extremities, components shorter than the given length were removed, allowing final cracks to be detected.

6. Conclusion

This paper proposes a hybrid crack detection method by creating discriminative models to extract the areas where cracks are present, using deep learning, and focusing image analysis on locations and directionalities that correspond with acquired extraction results. The detection results obtained with this method were close to what is achieved with human inspection. Additional work aims to extend the scope of crack detection to cover other structural damage such as water leakage, as basic technologies to allow automatic creation of interior elevations with struc-
tural damage in the future.

References

[1] Ukai, M. and Nagamine, N., Misaki, N., Sakamoto, Y., “Development of High Precision Image Processing method for tunnel deformation extraction,” *Cybernetics Quaterly report*, Vol.17, No.2, 2012.

[2] Ukai, M. and Misaki, N., Yasuda, M., Sugahara, A., Tanaka,T., “Development of crack detection method using machine learning,” *Cybernetics Quaterly report*, Vol.22, No.2, 2017.

[3] Richard Szeliski, “*Computer vision - algorithms and applications*,” Kyoritsu Shuppan, 2013.

[4] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, Dhruv Batra, “Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization,” ArXiv e-prints, Mar 2017.

[5] Ukai, M. and Nagamine, N., “Accuracy improvement of structural inspection method by image processing technology,” RTRI Report, Vol.28, No.11, 2016 (in Japanese).

Author

Masato UKAI
Senior Chief Researcher, Head of Image Analysis and IT Laboratory. Signalling & Transport Information Technology Division
Research Areas: Image Processing, Computer Vision