Identification of three-dimensional defect topology in concrete structures based on self-attention network using hammering response data

Masaya Shimada¹, Takahiko Kurahashi¹*, Yuki Murakami², Fujio Ikeda³ and Ikuo Ihara¹

¹ Department of Mechanical Engineering, Nagaoka University of Technology, 1603-1, Kamitomioka, Nagaoka, Niigata 940-2188, Japan.
² Department of Civil Engineering, National Institute of Technology, Nagaoka College, 888 Nishikatakai-machi, Nagaoka-shi, Niigata 940-8532, Japan.
³ Department of Mechanical Engineering, National Institute of Technology, Nagaoka College, 888 Nishikatakai-machi, Nagaoka-shi, Niigata 940-8532, Japan.

*Corresponding author: kurahashi@mech.nagaokaut.ac.jp

Received September 27, 2021, Accepted October 18, 2021

Abstract

The ageing of concrete structures in Japan is becoming an increasingly serious issue. Periodic inspection is necessary to prevent accidents caused by ageing. One of the methods used to inspect concrete is the hammering test. In this study, we aim to develop a system using machine learning for identifying the topology of defects in concrete, based on acceleration response data obtained from the hammering test. As part of the machine learning method, we constructed a neural network based on self-attention. In addition, we evaluated the effect of changing the size of the estimation domain using the machine learning model.

Keywords hammering test, defect detection, machine learning, self-attention, wavelet analysis

Research Activity Group Mathematical Design

1. Introduction

The ageing of concrete structures in Japan, which were constructed rapidly during the period of high economic growth, is becoming an increasingly serious issue. Therefore, periodic inspection is necessary to prevent accidents caused by ageing. As a method of evaluating concrete structures, the hammering test is used to diagnose structures based on the responses such as displacement and sound when the structure is struck with a hammer. However, this method depends on the senses and skills of the operator. Therefore, there is a demand for an automated method for evaluating the internal conditions of concrete. In this study, we developed a machine learning model to estimate the defect topology (position and size) in concrete. The inputs are scalograms of the acceleration response obtained from the hammering test. For tasks to which recurrent neural networks (RNN) and convolutional neural networks (CNN) have been applied, some studies replace them with simpler models based on attention to improve performance [1]. We built a network based on self-attention for these tasks.

2. Dataset

In this study, we used a dataset of hammering tests on concrete plates with artificial defects [2]. The dataset contains the acceleration responses of strikes on two concrete floor plates with different embedding depths of 10 artificial defects. The plate size was 1700 × 1700 × 180 (mm³). The impact points were placed 50 mm apart, and there were 32 × 32 points per plate. The plate is represented by a non-dimensional density tensor of size (32 × 32 × 36), as shown in Figs. 1 and 2. Elements with a non-dimensional density of one are not shown in the figures.

In this study, two pre-processing steps were performed for the dataset. The first step was to normalise the acceleration response waveform. Because the hammering test for this data was done manually, the magnitude of the impact force varied depending on the impact point. To reduce the effect of the difference in the impact force between each point, each acceleration response was divided by its maximum impact force. The second pre-processing step generates the scalograms. Scalograms are time-frequency representations of time-series data, which are mainly created by continuous wavelet transforms. The acceleration response waveform is converted into a form that contains more information, before being used as input. For the continuous wavelet transform, we used Ricker wavelets of 32 different widths. From the scalograms corresponding to the impact points at the four corners of the domain to be estimated, we estimated the non-dimensional density distribution just below the points.

3. Machine learning model

\[ y_i = \sum_{j \in \mathcal{K}(i)} \alpha(x_{\mathcal{K}(i)}_j) \odot \beta(x_j), \]  

(1)
Fig. 1. A non-dimensional density tensor of the floor plate A.

Fig. 2. A non-dimensional density tensor of the floor plate B.

Fig. 3. Self-attention block.

\[
\alpha(x_{\mathcal{R}(i)}) = \gamma(\delta(x_{\mathcal{R}(i)})),
\]

\[
\delta(x_{\mathcal{R}(i)}) = [\varphi(x_i), [\psi(x_j)]_{\forall j \in \mathcal{R}(i)}].
\]

Fig. 3 shows the structure of the self-attention block [3], and (1), (2), (3) show its formulas. \(\alpha\) is the attention weight; \(\varphi, \psi\) and \(\beta\) are the learnable linear mapping; \(R(i)\) is a set of neighbouring elements of element \(i\); \(\gamma\) is the mapping function; \(\delta(X_{\mathcal{R}(i)})\) is the relation function; \(\odot\) is the Hadamard product. The block learns the residuals; therefore, input \(x\) is directly added to (1) and it becomes the output. The structure of the self-attention network is presented in Table 1. In this table, ‘SA’ is the self-attention block, ‘Trans.’ is a transition layer, ‘n-d sa’ is the self-attention with n-dimensional output, ‘n-d linear’ is the n-dimensional linear mapping, ‘[d1, d2] max pool’ is the max pooling where the stride and kernel size are [d1, d2], ‘[n × n] GA pool’ is n × n global average pooling and ‘σ’ is the sigmoid function. \(\varphi, \psi\) and \(\beta\) correspond to the query, key and value in the general self-attention framework, respectively. The left stream of the block calculates the attention weight from the query and the key.

4. Numerical experiments

Fig. 4 shows the computational flow. First, we obtain the acceleration response waveforms of four points using the hammering test and then perform normalisation and continuous wavelet transform to obtain four scalograms. These are then input to a machine learning model that outputs a non-dimensional density distribution under the domain enclosed by the four points. The machine learning model consists of four stacked pairs of self-attention block and transition layers, and finally the output tensor passes through the sigmoid function; thus each element of the output takes a value from 0 to 1. Four of the 32 columns in the dataset were used as test data, and the rest were randomly divided by train : validation = 8 : 2 to train on the training data, determine convergence on the validation data and evaluate the estimation accuracy on the test data. In addition, we shifted the positions of the four columns, created training and validation data in a similar way, learned and made predictions for the test data (Fig. 5) eight times.

### Table 1. Structure of machine learning model.

| Layer       | Output size | Functions                      |
|-------------|-------------|--------------------------------|
| Input       | 32 × 2048 × 4 | -                              |
| SA          | 32 × 2048 × 4 | \([3 \times 3, 4-d sa, 4-d linear] \times 2\) |
| Trans.      | 16 × 512 × 16 | \([2, 4] \text{ max pool } \rightarrow 16-d \text{ linear}\) |
| SA          | 16 × 512 × 16 | \([3 \times 3, 8-d sa, 16-d linear] \times 2\) |
| Trans.      | 8 × 128 × 32 | \([2, 4] \text{ max pool } \rightarrow 32-d \text{ linear}\) |
| SA          | 8 × 128 × 32 | \([3 \times 3, 8-d sa, 32-d linear] \times 4\) |
| Trans.      | 4 × 16 × 64  | \([2, 8] \text{ max pool } \rightarrow 64-d \text{ linear}\) |
| SA          | 4 × 16 × 64  | \([3 \times 3, 16-d sa, 64-d linear] \times 4\) |
| Output      | n × n × 36   | \([n \times n] \text{ GA pool, 36-d linear, } \sigma\) |

Fig. 4. Computational flow.
We used the weights with the smallest validation loss among 2000 iterations.

We expanded the size of the estimation domain at a time to reduce the number of impact points required. Because four columns are used for the test data, we set the estimation domain size to $1 \times 1, 2 \times 2$, and $4 \times 4$ so that the estimation domains do not overlap. The larger the estimation domain size, the more difficult the estimation and the lower the accuracy is expected to be. When the estimation domain size is $n \times n$, the size of the dataset that can be used for training is $(29 - n)(33 - n) \times 2$. In general, the smaller the size of the dataset is, the lower the accuracy of the machine learning model. Therefore, expanding the estimation domain could reduce the estimation accuracy. To unify the size of the dataset for training for all estimation domain sizes, data augmentation was performed. For the data with a set of a non-dimensional density distribution and four scalograms, as shown in Fig. 6, we create a $90^\circ$ side view of each and add it to the dataset. This process increases the number of data in the dataset by a factor of four because data with $0^\circ, 90^\circ, 180^\circ$ and $270^\circ$ rotations are now available. Thus, the size of the dataset used for all estimation domain sizes, data augmentation was performed. For the data with a set of a non-dimensional density distribution and four scalograms, as shown in Fig. 6, we create a $90^\circ$ side view of each and add it to the dataset. This process increases the number of data in the dataset by a factor of four because data with $0^\circ, 90^\circ, 180^\circ$ and $270^\circ$ rotations are now available. Thus, the size of the dataset used for all estimation domain sizes is unified to the maximum number of data that can be prepared for the estimation domain size $4 \times 4 ((29 - 4)(33 - 4) \times 2 \times 4 = 5800)$.

Adam [4] was used to update the training parameters. The other calculation conditions are listed in Table. 2.

The estimation results for the floor plate A at each estimation domain size are shown in Figs. 7, 8 and 9. For the estimation domain sizes $1 \times 1$ and $2 \times 2$, the overall defect distribution, including the smallest defects, is accurately estimated. For an estimation domain size of $4 \times 4$, the smallest defect can no longer be estimated, and some defects are missing. However, the overall trend is that the estimation is correct.

The estimation results for the floor plate B at each estimation domain size are shown in Figs. 10, 11 and 12. With an estimation domain size of $1 \times 1$, estimation is possible even for the smallest defects, although they are underestimated. With an estimation domain size of $2 \times 2$, the smallest defect can no longer be estimated, and the second smallest defect is partially missing. With an estimation domain size of $4 \times 4$, the overall trend can be estimated, although it is slightly fuzzy.

These results were evaluated using the mean square error (MSE) (Table. 3) from the non-dimensional density distribution of the correct answer. In all cases, the estimation of plate A was more accurate than that of plate B. This indicates that the deeper the defect, the more difficult it is to estimate. In almost all cases, the accuracy was better for $1 \times 1, 2 \times 2$, and $4 \times 4$ in that order. We observed that the results are consistent with the expectation that the larger the estimation domain, the more difficult it is to estimate. The exception is that the best estimation results are obtained for plate A-$2 \times 2$, but additional verification suggests that it is probably a coincidence owing to randomness in learning. In conclusion, our method is a trade-off between the estimation domain size and the estimation accuracy; thus, it can be used for both rough and high-precision estimations.

5. Conclusions

In this study, the location and size of defects in the concrete plates were estimated using the self-attention network for scalograms of the acceleration response waveform obtained using the hammering test. It was confirmed that even if the estimation domain was expanded, the defect could be estimated with a certain accuracy. This method can be used to achieve a trade-off between the estimation domain size and the estimation accuracy.
Acknowledgments

This work was supported by grants from Japan Science and Technology Agency, 'A-STEP tryout: Digital transformation of skill up process of hammering test technique by self-organized maps' and Nagaoka University of Technology. The computations were mainly carried out using the computer facilities at Kyushu University’s Research Institute for Information Technology. We wish to thank all the persons who assisted us with this study.

References

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser and I. Polosukhin, Attention is all you need, arXiv:1706.03762 [cs.CL].

[2] A. Nouchi, Y. Murakami, T. Iyama and F. Ikeda, Evaluation of defect area inside concrete structures by self-organizing map in hammering test with a fixed acquiring response signal position, in: Proc. of the Japan Concrete Institute 2018, pp. 1755–1760, 2018.

[3] H. Zhao, J. Jia and V. Koltun, Exploring self-attention for image recognition, in: Proc. of CVPR 2020, pp. 10076–10085, 2020.

[4] D. P. Kingma and J. Ba, Adam: A method for stochastic optimization, arXiv:1412.6980 [cs.LG].