Development of a Deep learning system with small amount of data for Bridges Health Monitoring

Kiyoshi Hayakawa\(^a\)*, Syusaku Tomita\(^b\), Kenta Matsunaga\(^a\), Takeshi Wada\(^a\), and Takashi Otaba\(^c\)

\(^a\)Osaka Prefecture University College of Technology, 26-11 Saiwai-cho, Neyagawa-shi 572-8572, Japan
\(^b\)Nara Institute of Science and Technology University, 8916-5 Takayama-cho, Ikoma-shi 630-0192, Japan
\(^c\)Hokkai-Gakuen University, 4-1-40 Asahi-cho, Sapporo-shi 062-8605, Japan

*Corresponding Author: hayakawa@osaka-pct.ac.jp

Abstract

This paper shows the deep learning system for the health monitoring system of bridges. It checks two factors in the bridges, clacks and corrosions (or rusts). It decides damage rank using the deep learning system with small amount of data (about thousands of data). The deep learning system has several CNN models using fine tuning. Some of the CNN modes are accuracy to employ fine tuning with small training data. The system achieved 86.75% average accuracy in Xception fine tuning model for clacks and 83% accuracy for corrosions. Furthermore, the CNN models embedded in Jetson Nano which is one of the embedded boards.

Keywords: Deep Learning, CNN Model, Crack Damage Rank, Fine Tuning, Corrosion.

1. Introduction

Most building structures, such as office buildings, bridges, high ways, and so on, were built in 1970’s and 1990’s in Japan. Deterioration of the building structures in 1970’s becomes a social problem. Fig 1 shows percentage of the number of bridges which built over 50 years ago in Japan. In 2013, 18% of entire bridge in Japan are built over 50 years ago. In future, 2033, it will increase to 67%.

It is important to check and manage the infrastructures on a daily basis. Ordinary, the infrastructures are checked by looking and hammering test of specialists. There are two problems of the checking. One is a shortage of the specialists, the other is place regulation of infrastructure such as the bridges and highways. We have to develop a damage identification monitoring system by which everyone can check the building structures without the specialists of checking and hammering test of the infrastructures\(^(1)\).

On the other hand, it is also important to check damage points visually on the infrastructures. In Japan, inspectors who got a license of the checking are ranking damage of the bridges. However, the number of the inspectors is decreasing. We assume that they will need a support system of the ranking of the damage on the infrastructures. Therefore, we are also developing a health monitoring system which ranks damage level on the bridges using deep learning. Deep learning methods based on convolutional neural network (called CNN). It allows to dominate the speech recognition, vision based pattern recognition\(^(2)\), fault-tolerant control systems\(^(3)\), and so on. In particular, CNNs have brought breakthroughs toward object detection and recognition\(^(4)\). A variety of CNN applications have been constructed using a huge amount of annotated data for training of the mode such as crack detection system on steel surfaces\(^(5)\), concrete cracks and corrosions.

This paper will propose the rank decision system of crack damages and detection of corrosions using CNN model. A machine learning system is made by 3000 crack original images and 1770 corrosion original images.

Fig. 1. Bridges built over 50 years in Japan.

DOI: 10.12792/iciae2021.007

© 2021 The Institute of Industrial Applications Engineers, Japan.
2. Organization of the System

2.1 Overview of the Health Monitoring System

A health monitoring system have been developed (see Fig 2). It consists of measurement modules, a damage decision camera, and a cloud server. This paper refers the deep learning system in the cloud server.

The measurement modules are set on the infrastructures. The measurement modules measure vibration, then send the Fourier spectrum of the vibration to the cloud server. The base module sends Fourier spectrums received from many measurement modules to the cloud server by Ethernet. It manages data of the infrastructures in all of our country and finds damage of them. We are able to check the waves and results of damage identification through Web.

The damage decision camera takes pictures of the infrastructure, then find the damage and make decision of its damage rank automatically. In order to make decision of the damage rank, the cloud server has deep learning system using CNN. The damage decision camera embeds CNN constructed by the cloud server.

3. Deep Learning Data

3.1 Classification of damages

Supervised learning is employed to construct a neural network. There are several types of damage on the infrastructure such as crack, corrosion, deformation, free lime, leakage, and so on. This paper focused on cracks and corrosions on the bridges. The deep learning system enables judgement of damage rank for crack. The crack damage rank is classified from 1 to 5. Rank 1 means slight damage. Rank 5 means heavy damage. On the other hand, the corrosions damage has 2 rank, corrosions or no.

3.2 Contents of the Deep Learning Data for Cracks

The data of deep learning are provided by Hokkai-Gakuin University. The data consist of damage photograph file and csv file.

Fig. 3 and Fig. 4 show example of the crack damage photograph (JPG file) for rank of 2, 3, and 5. Red square areas are inserted in these photographs to indicate the damage area (Deep learning data didn’t include the red square). There are cases that one photograph has several damage areas (See Fig.3.). The deep learning system decides on a rank for a damage area. The size of photograph is different from each other. Minimum width size of photograph is 521[pixel], and that of maximum size is 2413[pixel]. Minimum height size of photograph is 379[pixel], and that of maximum size is 1697[pixel].

The CSV files consist of photograph file name, photograph’s file size, damage parts, the number of the damage, damage area, kind of damage, and damage rank.

Fig. 3. Example of photo. of crack damage (Rank 2 and Rank 3).

Fig. 4. Example of photo. of crack damage (Rank 5).
In case that data set (photo. and CSV file) have a deviation of the amount of data, the deep learning system doesn’t study precisely. For example, in case that the number of rank 1 damage data is 1/1000 smaller than the number of another rank, no damage area may decide on rank 1 damage. Therefore, the data set for training have a few deviations of the amount of data.

600 data among over 3000 data (600x5) on each rank were selected, and they increased to 600 data to 6000 by rotation, reflection and so on. Fine tuning is employed to have less training data.

3.3 Contents of the Deep Learning Data for Corrosion

Fig.5. shows the image data of corrosion. RGB image of the corrosion damage cuts out of the area which was unified into a 200×200 [pixel]. Total 1700 data provided by Hokkai-Gakuin university are selected. In addition, 70 data got by Internet were added into the data set. They also increase image data 8 times using same way as cracks image.

Fig. 5. Example of corrosion damage image.

4. The CNN Models

Six CNN models were employed for evaluation of the crack monitoring system. Then we evaluated the CNN models and selected the best CNN models among them. The selected CNN model was retrained using the corrosion data. This section explains the CNN models.

4.1 MobileNet

MobileNet(6) split the convolution layer into two sub tasks. Fig.6 shows that there is a depthwise convolution layer that filters the input, followed by a pointwise convolution layer that combines these filtered values to create new features.

4.2 ResNet50

When the deeper network starts to converge, a degradation problem has been exposed with the network depth increasing, accuracy gets saturated and then degrades rapidly. Such degradation is not caused by overfitting or by adding more layers to a deep network leads to higher training error. The deterioration of training accuracy shows that not all systems are easy to optimize.

To overcome this problem, Microsoft introduced ResNet(7) which is a deep residual learning framework. Instead of hoping every few stacked layers directly fit a desired underlying mapping, they explicitly let these layers fit a residual mapping. ResNet is Based on a plain network, a shortcut connection is inserted (See Fig.7). The identity shortcuts can be directly used when the input and output are of the same dimensions.

Fig. 6. MobileNet : convolution layers.

Fig. 7. ResNet50 : Shortcut connection.

4.3 Inception V3

The Inception deep convolutional architecture was introduced as GoogLeNet, here named Inception-v1. Later the Inception architecture was refined in various ways, first by the introduction of batch normalization (Inception-v2). The inception module described in Fig.8. Inception-V3(8) refined additional factorization ideas. InceptionV3 repeats an inception module deeper.
4.4 InceptionResNetV2

Inspired by the performance of the ResNet, a hybrid inception module was proposed. Fig.9 shows that there is a shortcut connection at the left of each module. This shortcut connection has been a kind of proved that it can help go deeper in ResNet\(^9\).

4.5 NASNet Mobile

NASNet Mobile\[^10\] employs a re-current network to generate the model descriptions of neural networks and train this RNN with reinforcement learning to maximize the expected accuracy of the generated architectures on a validation set (See Fig.10).

4.6 Xception

There is a much stronger hypothesis that cross-channel correlations and spatial correlations can be mapped completely separately. Xception is an extreme version of an Inception module based on this stronger hypothesis. It first uses a 1x1 convolution to map cross-channel correlations. And it then uses separately map the spatial correlations of every output channel described in Fig.11.

5. Accuracy of The Models for Cracks

From Fig.12 to Fig.17 show the results of accuracy of the CNN models with Confusion Matrix. Confusion Matrix is a performance measurement for machine learning classification. It is a table with 25 different combinations of predicted and actual values. diagonal element of the matrix means percentage of the correct answer.

Fig. 8. InceptionV3 : inception module.

Fig. 9. InceptionResNetV2 : Inception module & shortcut connection

Fig. 10. NASNet Mobile : Re-current network.

Fig. 11. Xception : architecture of CNN.

Fig. 12. The accuracy results of MobileNet.
Tab 1 shows the results of average, maximum, and minimum accuracy on the CNN models.

| Model name      | Average accuracy | Maximum accuracy | Minimum accuracy |
|-----------------|------------------|------------------|-----------------|
| MobileNet       | 84.77%           | 94.94%           | 76.11%          |
| ResNet50        | 88.04%           | 96.91%           | 67.18%          |
| InceptionV3     | 82.78%           | 95.83%           | 74.80%          |
| InceptionResNetV2 | 84.77%       | 97.33%           | 66.67%          |
| NASNet Mobile   | 84.77%           | 94.95%           | 76.11%          |
| Xception        | 86.75%           | 95.00%           | 82.65%          |
In terms of average accuracy, RetNet50 is the best performance among at 88.04%. In terms of maximum accuracy, InceptionResNetV2 is the best performance at 97.33%. But in terms minimum accuracy, it is the worst accuracy at 66.7%. There is unbalance among the damage rank on InceptionResNetV2. In terms of minimum accuracy, Xception is the best accuracy at 82.65%. Xception only achieved over 80% on average, maximum, and minimum accuracy.

6. Accuracy of The Models for Corrosions

Xception is good accuracy on cracks. Xception was applied only to corrosions. CNN model for corrosion was made using fine tuning.

Xception model achieved 83% accuracy on test data provided by Hokkai-Gakuin University and 80% accuracy on them provided by internet.

7. Prototype System using Jetson Nano

Fig.18. shows a prototype of health monitoring system. It consists of Jetson Nano with camera and monitor. It allows to low power and high speed to check damage. The trained model was embedded into Jetson Nano.

Python and Keras were installed into Ubuntu Linux system. CNN trained models programed by Python. The system takes the image of bridge image data by camera, analyzes it with the CNN model, and outputs the results of analysis on a display in less than one second.

8. Conclusions

This paper described development of health monitoring system, deep learning system and embedded system. The system allows to determine the degree of crack and corrosion damage using deep learning.

The CNN models are more than 80% accurate with small data (thousands of data). The deep learning system is able to find cracks and corrosions and determine the degree of damage on the cracks. In the evaluation, Xception model is best performance among the six CNN models for the damage of the bridges.

To make remote health monitoring system, deep learning system is embedded into Jetson Nano. It allows to find damages in less than one second.

In Future, we will put the Jetson Nano on a drone. We will implement IoT system using cloud service, such as AWS or GCP.

Acknowledgment

This work is supported in part by Grant No. 19K02991 (Development of the practice educational program of next-generation IoT system) from Japan Society for the Promotion of Science.

References

(1) Kiyoshi Hayakawa, Masashi Fujiwara, Takeshi Wada, and Takashi Obata : “Development of a damage identification monitoring system for building structures”, the 4th IIAE International Conference on Industrial Application Engineering, pp.340-347, 2016
(2) Y. LeCun, Y. Bengio, and G. Hinton : “Deep learning,” Nature, vol. 521, DOI 10.1038/nature14539, no. 7553, pp. 436–444, May. 2015
(3) S. Yin, H. Luo, and S. X. Ding : “Real-Time Implementation of Fault-Tolerant Control Systems With Performance Optimization,” IEEE Trans. Ind. Electron., vol. 61, DOI 10.1109/TIE.2013.2273477, no. 5, pp. 2402–2411, May. 2014
(4) C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going Deeper with Convolutions,” in Proc. 2015 IEEE Conf. Comput. Vision Pattern Recognition (CVPR’15),http://doi.org/10.1109/CVPR.2015.7298594, 2015
(5) D. Soukup and R. Huber Mork, “Convolutional Neural Networks for Steel Surface Defect Detection from Photometric Stereo Images,” in Proc. Int. Symp. on Visual Computing (ISVC’14), pp. 668–677, Dec. 2014.

(6) Takashi Obata: “Experimental study of damage identification health monitoring system by using piezoelectric element”, Journal of Structural Engineering, Vol. 60A, pp. 165-174, 2014

(7) Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, Hartwig Adam, “MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications”, arxiv:1704.04861, 2017

(8) Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, “Deep Residual Learning for Image Recognition”, arXiv:1512.03385, 2015

(9) Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, Zbigniew Wojna, “Rethinking the Inception Architecture for Computer Vision”, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2818-2826, 2016

(10) Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, Alex Alemi, “Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning”, arXiv:1602.07261, 2016

(11) Barret Zoph, Vijay Vasudevan, Jonathon Shlens, Quoc V. Le, “Learning Transferable Architectures for Scalable Image Recognition”, arXiv:1707.07012, 2018

(12) François Chollet, “Xception: Deep Learning with Depthwise Separable Convolutions”, arXiv:1610.02357, 2017