17th World Conference on Earthquake Engineering, 17WCEE
Sendai, Japan - September 13th to 18th 2020

SMARTPHONES AND DEEP LEARNING INTEGRATED SYSTEM FOR LOW-COST REAL-TIME BRIDGE VIBRATION STATUS REALIZATION

A. Shrestha [(1)], J. Dang [(2)]

[(1)] *International Division, Civil Engineering Department, Hazama Ando Corporation, Akasaka, Japan, ashishduwaju@hotmail.com*
[(2)] *Department of Civil and Environmental Engineering, Saitama University, Saitama City, Japan, dangji@mail.saitama-u.ac.jp*

Abstract

Safety of ageing infrastructures is one of the primary objectives of Structural Health Monitoring systems. Obtaining information of structure by means of vibration data is, therefore, important in disaster prevention and seismic evaluation of structures. Data driven approach in form of long-term vibration measurement is one of the straight forward method to evaluate structural integrity in terms of daily use against environmental loads, traffic loads, fatigue, etc. However, the data measured and recorded during monitoring is generally heterogeneous in nature and generating real-time insights out of such heterogeneous data with regard to its expanding volume is often a tedious and time-consuming process that hinders real-time evaluation of measured data for structural assessment. Therefore, the-state-of-art literature demands a method for auto realization of vibration data and eliminate the bottlenecks of processing large amount data to promote stable observation of structural conditions. With great success of deep learning, there exists a possibility to automatically learn the feature representation from time series for time series classification of recorded data which is comprehensively studied in this work. Automation of the manual process of looking after the raw data and realizing the vibration status promotes high-density observation as an important aspect for the state-of-art structural vibration monitoring.

In this study, a simple and customizable Convolution Neural Network framework was used to train a vibration classification model which can be integrated into the measurement application in order to realize accurate and real-time bridge vibration status on mobile platforms. The inputs for the network model are basically the multichannel time series signals acquired from built-in accelerometer sensor of smartphones while outputs are the predefined vibration categories. To verify the effectiveness of the proposed framework, data collected from long-term monitoring of bridge was used for training a model and its classification performance was evaluated on the test set constituting the data collected from the same bridge but not used previously for training. An iOS application program was developed for incorporating the trained model with predefined classification labels, so that it can classify vibration dataset measured on any other bridges in real-time. Results show an excellent classification accuracy up to 95%.

Keywords: Smart Devices; Convolution Neural Network; Long-Term Monitoring; Real-Time Vibration Classification

1. Introduction

Vibration measurement data is the basic source of information for structural dynamic analysis in the field of Structural Health Monitoring (SHM). Accordingly, obtaining information of structure by means of vibration data is important in disaster prevention and seismic evaluation of structures. SHM has traditionally relied on a structural identification paradigm using physics-based models where the goal is to use measurements to update a numerical (e.g. finite element) model of the structure and then deploy the model to make predictions on structural behavior. While such models are important for understanding structural behavior, such models are unlikely to be readily available for the majority of bridges. Their generation can also be resource and time intensive. Furthermore, computed models may not replicate behavior of real structures due to uncertainties and approximations in the modelling process. Therefore, robust approaches for measurement interpretation that are generic and readily applicable without requiring detailed prior knowledge of structures have tremendous value in the context of extracting information from monitoring for bridge management.

Data driven approaches, which rely purely on the collected measurements for measurement interpretation, offer great promise for long term continuous monitoring. Long-term vibration measurement is one of the straight forward method to evaluate structural integrity in terms of daily use against environmental loads, traffic loads, fatigue, etc. in addition to seismic loadings. Further, to incorporate low-cost measurement, especially for long-term purpose, Wireless Smart Sensor Networks (WSSNs) are increasingly applied in SHM around the world [1-6]. However, in most of the cases the long-term vibration data inevitably contains nonstationary noise which usually occurs due to sensor induced faulty signals. These sensor faults often occur in sensor data obtained by using low-cost WSSN due to hardly-avoidable reasons like changing temperature and battery issues. Since those noise would significantly affect analysis results, the SHM output may be untrustworthy. Moreover, for post earthquake bridge status and immediate occupation function evaluation, the environmental vibration and device noise will trigger the false threshold alarm too often during seismic observation in practice. Therefore, time-series data blocks appropriate for analysis needs to be detected manually, which is often a tedious and time consuming process that hinders real-time evaluation of measured data for structural assessment. Also, the system needs to find some automatic method to look after the raw data and realize the vibration status by labelling and make them prepare to be used. Automation of this manual process promotes high-density and long- term observation which is one of the major concerns currently [7,8]. By developing a method for auto realization of time-series blocks in long-term vibration records, the bottlenecks of processing large amount data will be eliminated, and stable observation of structural conditions can be promoted. The problem of auto detection can be formulized as a general classification problem and can be solved using simple neural network-based architecture or a more advanced deep learning model such as Convolutional Neural Networks (CNNs). With the tens of thousands of recorded data, which will continue to grow exponentially and indefinitely during long-term measurement; combining the powerful algorithms of CNN can make that data become the most valuable asset for structural condition assessment.

In recent years, convolutional neural networks have led to impressive results in object recognition [9] face verification [10] and audio classification [11]. Nevertheless, with great success of deep learning, there also exists a possibility to automatically learn the feature representation from time series for Time Series Classification (TSC) as well. However, there have not been many research efforts in the area of time series to embrace deep learning approaches in a 1D framework. Some studies that implement 1D CNNs includes classification of electrocardiogram (ECG) [12], fault detection in high power engines [13], and Human Activity Recognition (HAR) problems [14-16], etc. However, to the best knowledge of the authors, all these previous works that utilizes CNN although offers a robust framework for data classification but does not provide a solution to perform those task in real-time and using low-cost consumer grade devices like smart devices. Moreover, as the network grows deeper, the model complexity also increases correspondingly and even a relatively small network, involves millions of parameters to classify the time series data. Such costly computation complexity makes the deployment of CNN models unaffordable for common PCs and mobile devices, and has not been practiced in real-time vibration classification purpose at the same time. This paper, therefore, proposes a novel data driven approach to solve this challenge. Smart device is a powerful device that have built-in sensor infrastructures to inspect multiple parameters such as acceleration, displacement, angle, force, etc. The real-time data communication and computation adds even more advanced capabilities to the device. With remote monitoring and real-time data mining, users can solve more applications related to structural safety assessment, save inventory costs, and help prevent unplanned downtime. Therefore, realizing the advancement in the application of smart devices for SHM, these are used as a quick bridge vibration status monitoring kit in this study. For this purpose, first of all a simple yet computationally efficient convolution neural network-based framework has been investigated. Then, the framework is integrated into measurement application that performs a real-time classification of measured records from smart devices by exploiting relationships in sensor inputs and automatically extracting distinct features in time domain.

2. System Design

The system architecture framework consists of three main layers. The first layer is responsible for data preparation, including data collection and data preprocessing. The second layer is the key layer in which feature extraction is performed using customizable Convolutional Neural Network. The last layer is the classification layer, in which the trained classifier is integrated with iOS smart device operating system for real-time and in-device classification.

2.1 Data Collection

There currently exists no publicly available dataset of labelled time series showing different vibration categories and sensor induced faulty signals. Thus, a new database had to be assembled from scratch to test the proposed method. Smart devices' recorded data during long-term vibration measurement of bridge [17] was utilized to provide the necessary resources for creating a dataset. The bridge has been instrumented with six smart devices (iPhone 5s) inside of the box girder at six different locations, continually measuring acceleration vibrations as shown in Fig.1

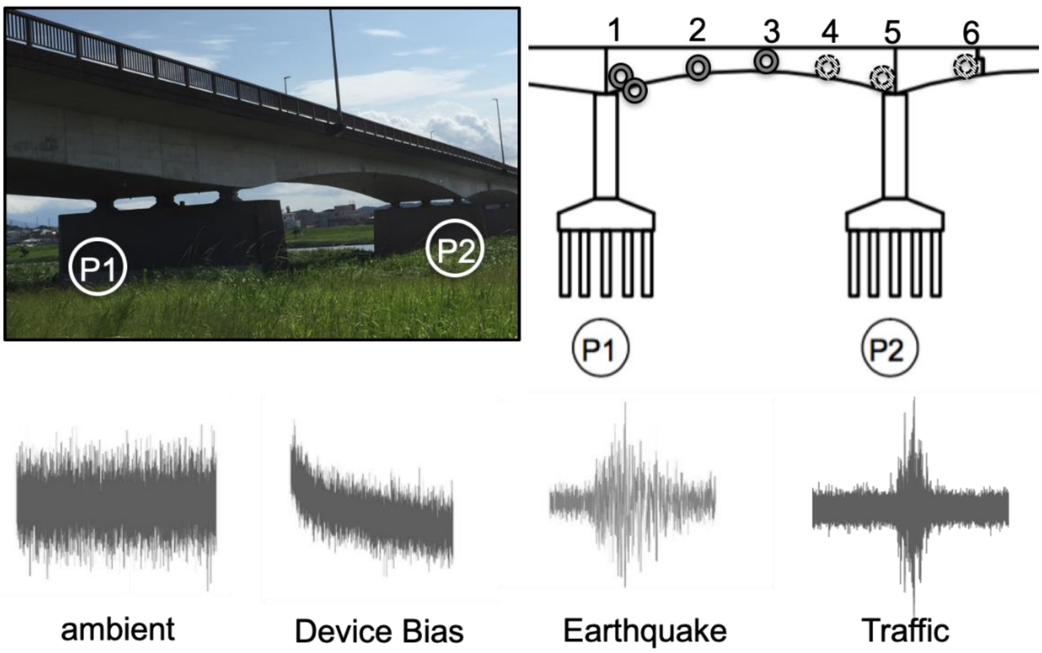

Fig. 1 – Vibration Data measured at the bridge

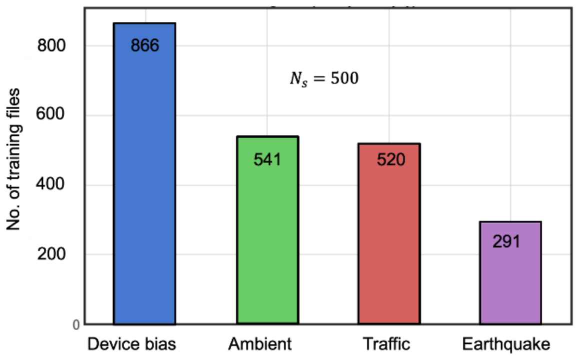

Fig. 2 – Distribution of vibration categories used as training set

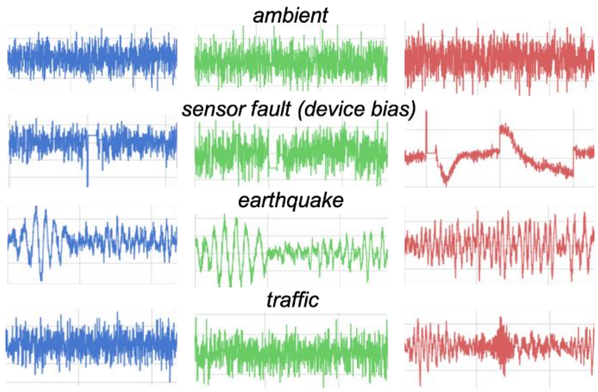

Fig.3 – Typical waveforms illustrating different vibration types

The vibration data collected from smart devices installed at different location of bridge mainly constitute four different types of raw acceleration data that corresponds to ambient data, sensor induced faulty data (drifts and spikes), traffic induced vibration data, and few earthquake records respectively. The database includes a total of randomly selected 2213 vibration data with each data containing 500 sample points N_s, corresponding to 5 seconds of acceleration record sampled at 100 Hz. All the vibration signals were hand labelled by the authors. Fig.2 shows the distribution of different types of vibration signals in the total database, while Fig.3 gives a visual illustration of different vibration types. Some common issues faced when applying classification task are the imbalance of the dataset and the lack of enough training data. Since, real earthquake event is a rare phenomenon, it is justifiable that earthquake class contained the least data number amongst other.

2.2 Data Preprocessing

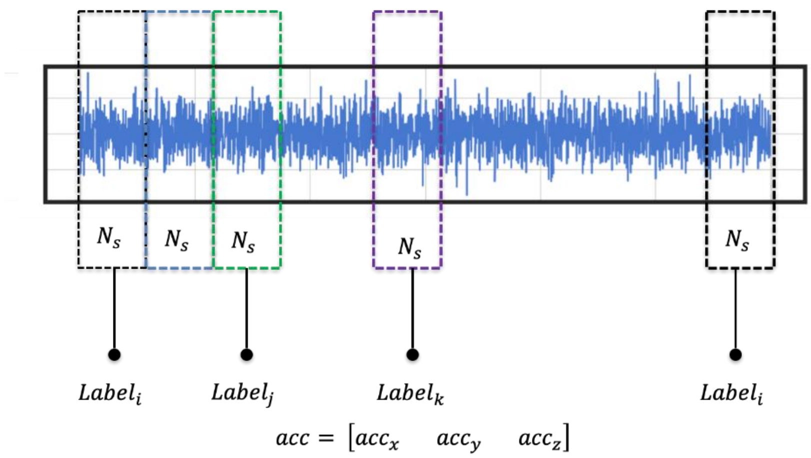

Fig. 4 – A typical time-sliced preprocessing of raw acceleration

After creating the database, since the raw data is never available in the format that is needed for training it needs to be preprocessed. Preprocessing includes splitting the whole waveform into small segments of predefined data length N_s and then manually assigning the appropriate label for each kind of vibration as shown in Fig.4. In this study, the data length for each sample, N_s is taken as 500. After dividing the training data into small segments, normalization is performed. Normalization with absolute maximum value amongst the training examples in a given sample is performed in this study. Similarly, since smart devices provide tri-axial acceleration measurement, the final training example for a given sample labelled with an appropriate vibration class contains acceleration values from all three axis concatenated together. This means that the final data length of a training sample is adjusted to $3 * N_s$

Another preprocessing step is to separate the whole dataset into a training and a test set. The data is split in such a way that the information from the test set does not bleed into the training set. This is generally the approach for evaluating the overall performance of the model during training and then validating against the test set. The real idea behind data splitting is that the network should predict the vibration characteristics from the data it has not seen before (i.e. data not used during training). Therefore, in this work, out of the total data, 80% of the data are randomly taken for training purpose while the remaining 20% data which do not overlap with the training data, are used for testing purpose.

2.3 Network Architecture

In this study a 1D CNN configuration is used in order to fuse feature extraction and learning (vibration classification) phases of the raw accelerometer data obtained from smart devices. A machine learning model involves a lot of complex code, manipulating arrays and matrices. But since machine learning has been around for a long time, researchers have created libraries that make it much easier to create machine learning models; many of which are written in Python, SAS, MATLAB and other software. In this study, "*Keras*"- a Python based machine learning framework has been implemented for utilizing the deep learning model for the purpose of training a classifier for vibration data classification. Keras provides a consistent and simple API for building models that can be trained on one backend while deploy in another. Another reason to use Keras is due to its support to integrate "*coremltools*", which directly creates iOS compatible trained model that can be used in smart device application for real-time and in-device classification tasks (Keras Documentation [18]).

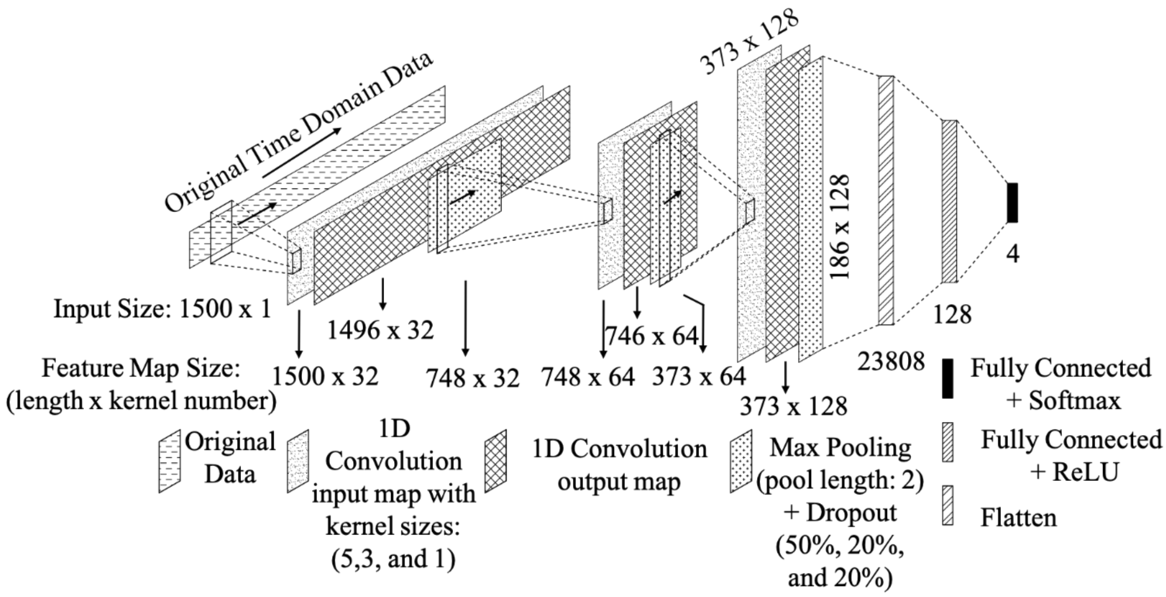

Fig. 5 – Detail map of CNN architecture for vibration classification

The 1D CNN architecture used in this work consists of three convolution layer with kernel size 5, 3, and 1 respectively, and each followed by MaxPooling layer and a dropout layer with drop percentage 50%, 20% and 20% respectively. The pooling layer halves the convolution layer's width and height while dropout controls overfitting of the neurons. The use of pooling and dropout layers after each convolution layers significantly reduces the number of parameters in the fully connected layers and training becomes faster. The lower layers use small no. of filters while higher layers use broader filters to process more complex parts of the input. Finally the top layers in CNN are stacked by two fully connected neural networks. These fully connected neural network are expected to combine different local structures in the lower layers for the final classification purpose. Table 1 provides the detail overview of the CNN model while Fig.5 illustrates the same in form of a network map.

Table 1 – Detail overview of CNN model architecture

Layer (type)	Output Shape	Param #
Conv1D (32x5x1)	1496x1x32	192
Max Pooling	748x1x32	0
Dropout (50%)	748x1x32	0
Conv1D (64x3x1)	746x1x64	6208
Max Pooling	373x1x64	0
Dropout (20%)	373x1x64	0
Conv1D (128x1x1)	373x1x128	8320
Max Pooling	186x1x128	0
Dropout (20%)	186x1x128	0
Flatten	23808	0
Fully Connected 1	128	3047552
Fully Connected 2	4	516
Total Parameters: 3,062,788		

From Table 1 it can be inferred that the total number of parameters while building a deep learning model depends on the number of variables that determines the network structure and how the network is trained. These variables are called hyperparameters. These hyperparameters are usually tweaked to find an optimum tradeoff between network accuracy and network training time. Convolution layer and pooling layer determines each layer's output shape and number of parameters to be trained. Eq. (1) and (2) defines the output shape and no of parameters involved in the training respectively.

$$Output\ Shape = W_i - W_f + 1,\ H_i - H_f + 1,\ N_f \tag{1}$$

$$No\ of\ Parameters = N_f * (W_f * H_f * D_i + 1) \tag{2}$$

W_i: input length, W_f : length of filter, H_i : input height
H_f : height of filter, N_f : filter number, and D_i : input depth

The weights from the convolutional layers are flattened and then goes to the fully connected layer. The output shape from the last convolutional layer and before the fully connected layer is (186, 1, 128). Therefore, the result of flattening is an array with 23808 elements as an input to the first fully connected layer. The first fully connected layer has 128 neurons which means that every neuron interacts with 23808 elements to produce a 128-neuron output layer. Finally, the 128 neurons are passed as an input to second fully connected layer to output as many neurons as the number of classes. In this model, since there are four classes, so the final output is four classes, each holding their probability of classification. The total number of trainable parameters therefore sums up to be 3,062,788. In this study, the categorical cross entropy loss function is used to calculate the loss as a function of difference between the true measure and predicted measure while Stochastic Gradient Descent (SGD) algorithm minimizes the loss function. Batch size and number of epochs are arbitrarily chosen to be 200 and 30 respectively. Each epoch generally improves loss and accuracy measurement. More epochs produce a more accurate model, but training takes longer and sometimes may also lead to overfitting.

3. Network Accuracy and Results

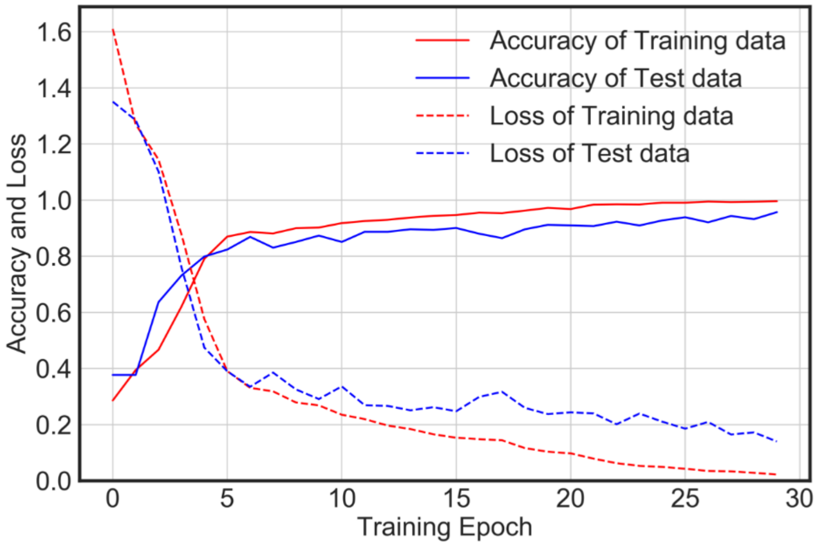

Fig. 6 – CNN model accuracy and loss for training and test dataset

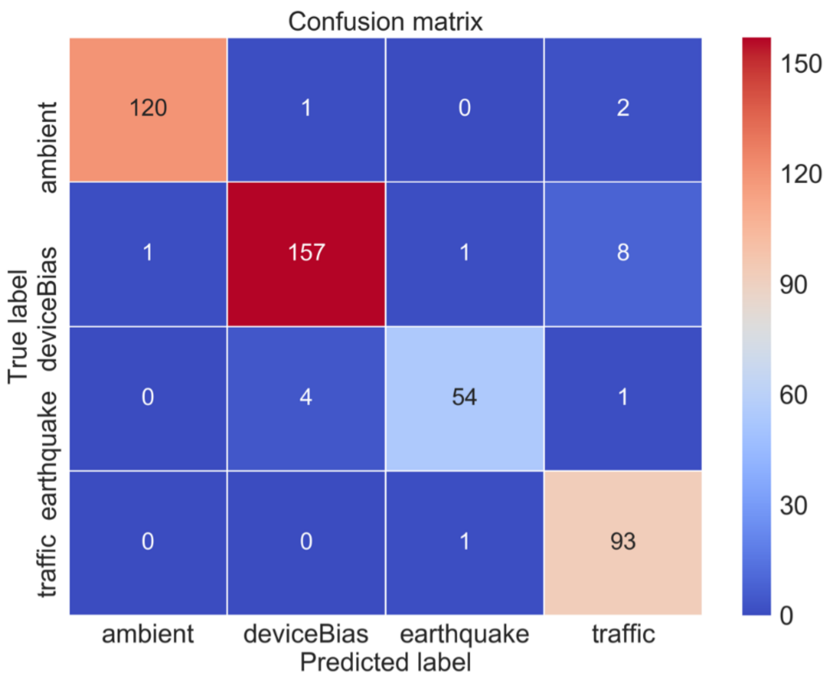

Fig.7 – Confusion matrix for CNN model with original dataset

The network architecture described in above section were implemented on the generated database to test the efficacy of their performance for the purpose of automated vibration classification. From Fig. 6 it is observed that with increase in epoch, loss value decreases and accuracy increases. There is no significant improvement in loss or accuracy after epoch 25. Thus in overall 30 epochs were performed. The training accuracy is around 99% while the test accuracy is around 95%. This means that the model generalizes well for the vibration data it has not seen before. To analyze the data in more detail, confusion matrix is shown in Fig.7. The confusion matrix indicates that many of the prediction error are due to confusion between three classes: bias, earthquake, and traffic. This is probably because these vibrations are relatively similar as compared to ambient vibration. Further, the performance of the model based on metrics such as Precision, Recall, and f1-score is excellent as shown in Table 3. These metrics are generally expressed mathematically using "Confusion Metrics" as shown in Table 2.

Table 2 – Confusion metrics parameters

	Predicted Class		
True Class		Class = Yes	Class = No
	Class = Yes	True Positive (t_p)	False Negative (f_n)
	Class = No	False Positive (f_p)	True Negative (t_n)

Table 3 – Precision, recall, and f1-score for original dataset

	Precision(P) $\frac{t_p}{t_p + f_p}$	Recall(R) $\frac{t_p}{t_p + f_n}$	f1-score $\frac{2 * R * P}{R + P}$	No.
ambient	0.99	0.98	0.98	123
device bias	0.97	0.94	0.95	167
earthquake	0.96	0.92	0.94	59
traffic	0.89	0.99	0.94	94
avg / total	0.95	0.96	0.95	443

Convolutional nets often tend to overfit when dealing with smaller datasets. The overall accuracy of the model although being high around 95%, could still be improved further with hyperparameter tuning and increasing the dataset especially using data augmentation method.

4. Real-Time Auto Classification of Records in Smart Devices

In this study, for realizing end-to-end processing from raw observation data to analysis result, a framework for real-time auto realization of smart device recorded bridge vibration signals was developed. The iOS application program developed for acceleration measurement [19] was further extended to incorporate the trained model with predefined classification labels, so that it can classify vibration dataset measured on any other bridges in real-time. The integration of powerful machine learning models into Apps on iOS devices is possible due to Apple's straight forward machine learning framework known as "*Core ML*" (Core ML Apple developers [20]).

The trained model as described in previous section is converted to iOS compatible model which is in Core ML format by utilizing *Keras* support for "*Coremltools*".

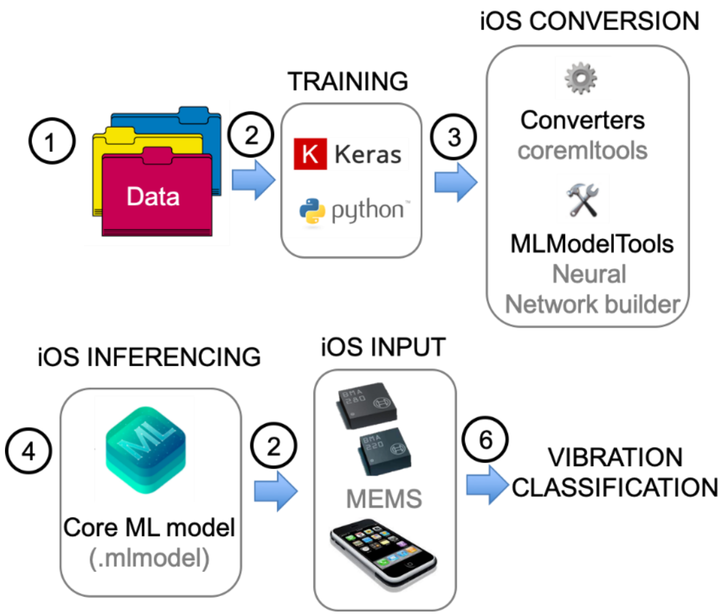

Fig.8 – Core ML integration into smart devices for real-time vibration classification

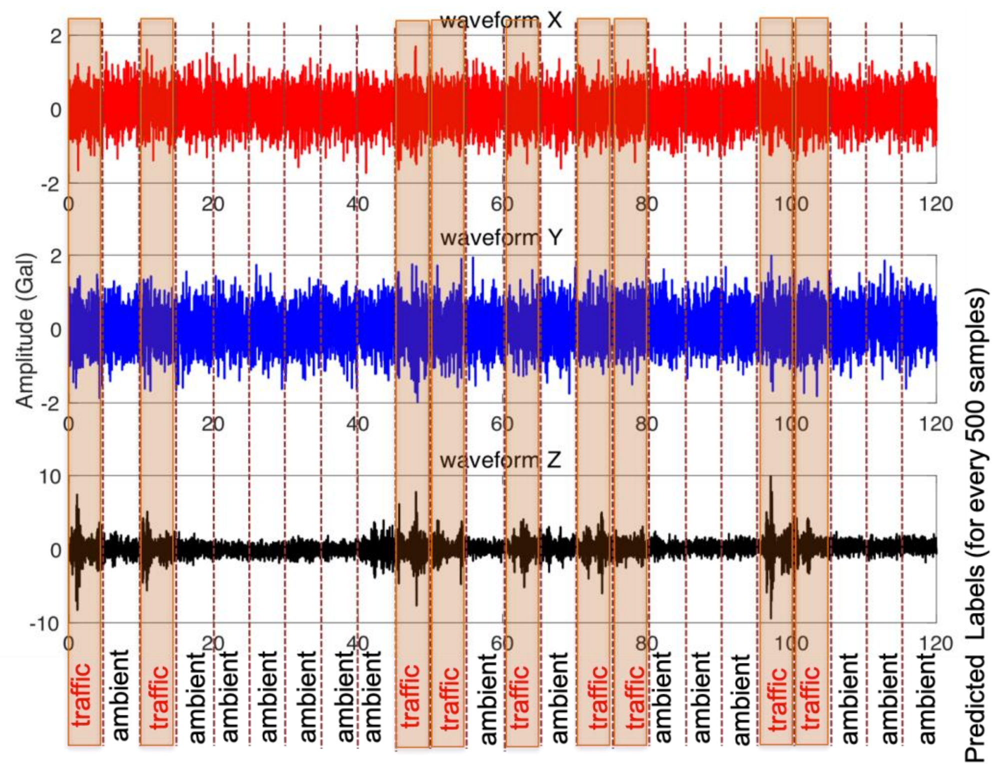

Fig.9 – In-device prediction (classification) of smart device recorded vibration data

Core ML is the machine learning framework used across apple products i.e. iOS, macOS, watchOS, and tvOS. Core ML delivers fast performance with easy integration of trained machine learning models. The machine learning prediction is calculated on the device itself due to which real-time classification performance is achievable. Core ML is optimized for on-device performance, which minimizes memory footprint and power consumption.

There are basically three advantages of using Core ML as machine learning on the device:

1. *Low Latency and near Real-Time Results*: No need to make network API call and wait for the response. This means that such framework is beneficial for applications such as processing the videos on successive frames.
2. *Offline availability*: The application runs without network connection.
3. *Cost*: No network connection, no API cost, and no model stored in the cloud.

However, there are some disadvantages as well. By adding the model to the device, the size of the app increases and creating an accurate model sometimes can be huge. Prediction and inference on the mobile devices involve lot of computation which increases battery power usage and some of the older devices may have difficulty in performance. The model on the device will need to be continually trained in most cases and any change to the model results in the app needing to update on the device.

The Core ML integrated iOS application program that measures device acceleration was then applied to verify the proposed method by collecting vibration data from random bridges. A sample example of a random three axis acceleration measurement on bridge using smart devices for a period of 120 seconds and its predicted classification is shown in Fig.9. Since the inputs for the trained CNN model are the multichannel time series signals each of 500 data points acquired from built-in accelerometer, it can be seen that the classifier predicts vibration categories (labels) to each 500 data points along the whole data set. Among the four pre-defined labels in the trained classifier, two of the categories (i.e. ambient and traffic) are rightly predicted. The portion of record with significant peaks in the vertical direction corresponds to influence of traffic while those without peaks corresponds to just ambient vibration, and the classifier predicts the classification accordingly. In this sample record, there are neither any faulty records nor an earthquake record, which therefore are not predicted by the classifier. These results thus demonstrate the accuracy and viability of autonomous bridge vibration realization using smart devices.

5. Conclusion

In this study, a framework for real-time autonomous bridge vibration realization using smart devices has been investigated. The framework integrates convolutional neural networks to exploit different types of relationships in sensor inputs, and automatically extract robust and distinct features in time domain to effectively carry out classification tasks in smart device itself. The effectiveness of the CNN framework, was also verified with data collected from long-term monitoring with results showing an excellent classification accuracy up to 95%. The experience of using smart devices for machine learning paradigm provided with valuable insights and promising guidelines, opportunities and potential of such low-cost devices for structural monitoring in a rapid, remote, automated, and quantified framework. Robust implementation of this low-cost and automated system can radically influence future advancements in smart, sustainable, and resilient infrastructure.

6. Acknowledgements

The authors would like to thank and express gratitude to Association for Miyazaki Infrastructure Monitoring System (MIMOS) and Miyazaki City for their continuous help in setting up the field work at the bridge.

7. References

[1] Lynch, J.P., Law, K.H., Kiremidjian, A.S., and Carryer, E. (2004): Design and performance validation of a wireless sensing unit for structural monitoring applications. *Structural Engineering and Mechanics*, Vol. 17, No. 3-4.

[2] Jo, H., Rice, J.A., Spencer. Jr, B.F., and Nagayama, T. (2010): Development of a High-sensitivity Accelerometer Board for Structural Health Monitoring. *Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems*, Vol. 7647, 764706, doi: 10.1117/12.848905.

[3] Cho, S., Yun, C.B., Lynch, J.P., Zimmerman, A.T., Spencer. Jr, B.F., and Nagayama, T. (2008): Smart Wireless Sensor Technology for Structural Health Monitoring of Civil Structures. *Steel Structures* Vol. 8, pp. 267-275.

[4] Zhu, Li., Fu, Y., Chow, R., Spencer Jr., B.F., Park, J.W., and Mechitov, K. (2018): Development of a High-Sensitivity Wireless Accelerometer for Structural Health Monitoring. *Sensors*, 18, 262; doi:10.3390/s18010262.

[5] Spencer Jr., B.F., Sandoval, M.R., Kurata, N. (2004): Smart sensing technology: opportunities and challenges. *Struct. Control Health Monit*, 11:349–368 doi: 10.1002/stc.48

[6] Bedon, C., Bergamo, E., Izzi, M., and Noè, S. (2018): Prototyping and Validation of MEMS Accelerometers for Structural Health Monitoring - The Case Study of the Pietratagliata Cable-Stayed Bridge. *J. Sens. Actuator Netw*, 7, 30; doi:10.3390/jsan7030030 .

[7] Hata, Y., Goto, H., and Yoshimi, M. (2016): Preliminary analysis of strong ground motions in the heavily damaged zone in Mashiki town, Kumamoto, Japan, during the main shock of the 2016 Kumamoto earthquake (Mw7.0) observed by a dense seismic array, *Seismological Research Letters*, 87:5, 1044-1049.

[8] Wapenaar, K. and Fkkema, J. (2006): Green's function representations for seismic interferometry, *Geophysics*, 71:4, SI33-SI46.

[9] Krizhevsky, A., Sutskever, I., and Hinton, G.E. (2012): Imagenet classification with deep convolutional neural networks. *In NIPS.*

[10] Schroff, F., Kalenichenko, D., and Philbin, J. (2015): Facenet: A unified embedding for face recognition and clustering. In arXiv preprint arXiv:1503.03832.

[11] Lee, H., Pham, P., Largman, Y., and Ng, A.Y. (2009): Unsupervised feature learning for audio classification using convolutional deep belief networks. *In NIPS.*

[12] Kiranyaz, S., Ince, T., and Gabbouj, M. (2016): Real-time patient-specific ECG classification by 1-D convolutional neural networks, *IEEE Trans. Biomed. Eng*. 63, 664–675, http://dx.doi.org/10.1109/TBME.2015.2468589.

[13] Ince, T., Kiranyaz, S., Eren, L., Askar, M., and Gabbouj, M. (2016): Real-time motor fault detection by 1-D convolutional neural networks, *IEEE Trans. Ind. Electron.* 63, 7067–7075, http://dx.doi.org/10.1109/TIE.2016.2582729.

[14] Yang, J.B., Nguyen, M.N., San, P.P., Li, X.L., and Krishnaswamy, S. (2015): Deep Convolutional Neural Networks On Multichannel Time Series For Human Activity Recognition. *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence , IJCAI*, pp. 3995-4001.

[15] Zeng, M., Nguyen, L.T., Yu, B., Mengshoel, O.J., Zhu, J., Wu, P., and Zhang, J. (2014): Convolutional Neural Networks for Human Activity Recognition using Mobile Sensors. *6th International Conference on Mobile Computing, Application and Services, MobiCASE*, doi: 10.4108/icst.mobicase.2014.257786, pp. 197-205.

[16] Hammerla, N.Y., Halloran, S., and Plotz, T. (2016): Deep, Convolutional, and Recurrent Models for Human Activity Recognition Using Wearables. *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI*, pp. 1533-1540.

[17] Shrestha A., Dang J., Wang X., and Matsunaga S. (2019): Smart Phone based Bridge Seismic Monitoring System and Long-Term Application Tests. *ASCE Journal of Structural Engineering*. Doi: 10.1061/(ASCE)ST.1943-541X.0002513.

[18] Keras Documentation. Available Online at: https://keras.io/ja/why-use-keras/

[19] Shrestha A., Dang J., and Wang X. (2018): Development of a smart-device-based vibration measurement system: Effectiveness examination and application cases to existing structure. *Struct Control Health Monit.* Vol. 25: e2120. https://doi.org/10.1002/stc.2120.

[20] Core ML (2018): Available Online at: https://developer.apple.com/documentation/coreml.

