Exploring the Potential of Transfer Learning Applications for Structural Damage Classification

BURAK DURAN and SAEED EFTEKHAR AZAM

ABSTRACT

Structural Health Monitoring (SHM) is essential for ensuring the safety and maintaining the functionality of structures and infrastructure systems, and Machine Learning (ML) techniques have shown great potential in rendering SHM as an automated process. However, unlike many other application areas of ML, when dealing with infrastructure SHM, there is little to no data available from damaged states to be used for ML training. To cope with this issue, a typical Finite Element Model (FEM) of a representative bridge beam structure is used for generating a damaged dataset. Secondly, creating a generalized network that can successfully work and perform a classification task for a variable set of structures seems to be one of the biggest challenges in the field of structural damage detection. In this work, the capability of Transfer Learning (TL) via Feature Extraction (FE) and Joint Training (JT) in generalizing the network is explored. Modeling uncertainties for supervised damage classification can be mitigated by utilizing a Deep Neural Network (DNN) comprised of Fully Connected (FC) and Convolutional (CONV) layers. In this regard, FEMs of three simply-supported beam structures with varying lengths were constructed, and acceleration time-history data was obtained from the interior nodes under the applied load history at the midpoint. The damage states were represented by the change in flexural stiffness within a range from 10% to 90%. Subsequently, the potential knowledge transfer was successfully implemented from the source domain to the target domains via TL and JT. Then, the proposed two-dimensional (2D) CNN network was tested with a target dataset that was not included in the training from another beam with a different length. The results indicate that FEMs could generate numerous source domains to achieve a generalized model even with including uncertainties.

Burak Duran, Department of Civil and Environmental Engineering, University of New Hampshire, Durham, NH 03824, US
Saeed Eftekhar Azam, Department of Civil and Environmental Engineering, University of New Hampshire, Durham, NH 03824, US

INTRODUCTION

Infrastructure such as bridges serves an important role in facilitating transportation, services, and connecting communities. These structures are subjected to numerous environmental factors and dynamic loading conditions, which can result in gradual deterioration or sudden failure at the local or global scale. Therefore, the damage detection framework of bridges is of utmost importance to ensure their structural integrity, prevent catastrophic failures, and take appropriate action to prioritize maintenance and repair efforts. In this regard, Structural Health Monitoring (SHM) has emerged as a valuable tool enabling continuous monitoring of bridge behavior and offering early warning signs of potential damage. SHM contains the deployment of sensors to collect data on structural responses, including vibrations, strains, and displacements. By analyzing this data, the health of the bridge can be assessed, and any indicators of damage or deterioration can be detected. Conventional vibration-based techniques have been widely employed in SHM, relying on the analysis of the frequency characteristics to identify changes indicative of damage. However, these techniques might have certain limitations in terms of accuracy, sensitivity, the time-consuming model generating, calibration, and feature extraction procedure, and the ability to detect subtle damage that might not be visible to the naked eye [1].

The integration of Machine Learning (ML) techniques in SHM has the potential to revolutionize damage detection in bridges. ML algorithms can automatically learn from the vast amount of sensor data and identify patterns. ML methods can be divided into two categories; Unsupervised and Supervised. Unsupervised methods, in which the labeled datasets are not available, play a role in detecting the existence of damage based on statistical analysis and novelty detection. On the other hand, Supervised methods utilize labeled datasets to train algorithms, enabling them to classify data and make predictions. Using this technique not only offers to identify the occurrence of damage but also makes it possible to gain deeper insights into its characteristics and location. Overall, this integration of SHM into ML frameworks allows for more accurate, robust, fast, and computationally efficient damage detection in bridges.

In traditional ML approaches, classification tasks heavily rely on the user's input and are often performed using time-consuming feature extraction techniques. Contrary, Deep Learning (DL) methods enhance the autonomy of the process since it has the ability to automatically learn feature representations from raw data, eliminating the need for manual feature extraction steps required in traditional ML methods [2]. One highly effective category of DL methods is Convolutional Neural Networks (CNNs).

Within the scope of this study, a potential application of TL via feature extraction (FE) and joint training (JT) by integrating SHM into DL by investigating a representative simply-supported bridge beam is presented. To that end, a finite element model (FEM) of the beam was subjected to white-noise load time history, and acceleration data was obtained from the interior nodes of the beam. Then, the obtained structural response was transformed into gray-scale images, and a two-dimensional (2D) CNN was developed. A successful CNN implementation through FE and JT showed the potential of knowledge transfer in the field of structural damage identification and classification framework.

DEEP NEURAL NETWORK (DNN)

Deep neural networks (DNNs) are a type of artificial neural network (ANN) that have gained significant attention in recent years due to their noticeable performance in various fields including SHM. It consists of several layers of interconnected neurons between the input and output layers (see Figure 1).

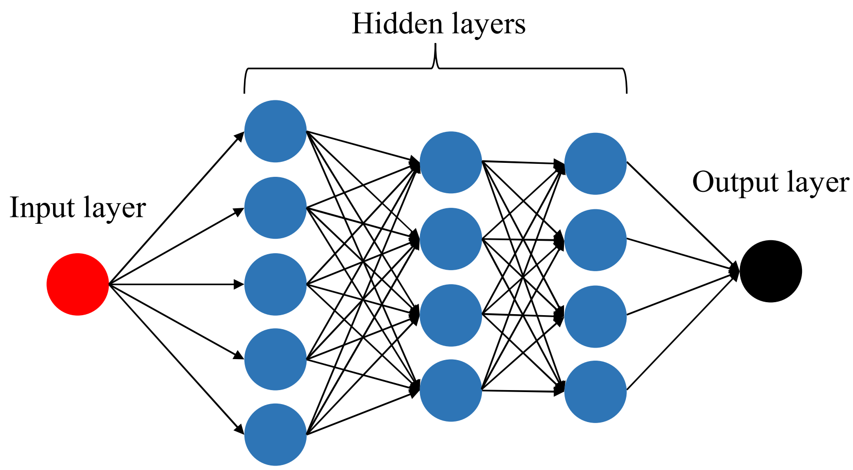

Figure 1. Layered representation of an example DNN

In a DNN, each neuron is represented as a mathematical function that takes input from the previous ones and produces an output. A layered configuration of DNNs allows for hierarchical feature extraction, where each layer and neuron learns progressively more abstract representations of the input data. Convolutional neural networks (CNNs) are a specialized type of DNN that is particularly designed for processing grid-like data such as images or time series. Rather than traditional ML techniques, CNNs automatically extract the relevant features and handle the spatial information directly from raw image pixels due to their unique architectural design. Thus, they can be adapted and generalized to different datasets and are well-suited for tasks such as image classification, object recognition, etc. A typical CNN is composed of convolutional (CONV), pooling (POOL), and fully connected (FC) layers. In a sequentially layered configuration, a CNN involves an optimization algorithm called backpropagation, which is a gradient descent algorithm and iteratively adjusts the network's parameters to minimize the difference between the predicted outputs and the ground truth labels. During the training procedure, the network updates the weights, which are the connections between the neurons, based on the computed errors. In conclusion, CNNs eliminate the need for manual feature extraction and offer an automated way for several tasks.

METHODOLOGY

A finite element model of an S3x5.7 steel section was generated using the OpenSeesPy [3] platform. The area and moment of inertia of the given section used in the model are 1077.42 mm^2 and 1.05x10^6 mm^4, respectively.

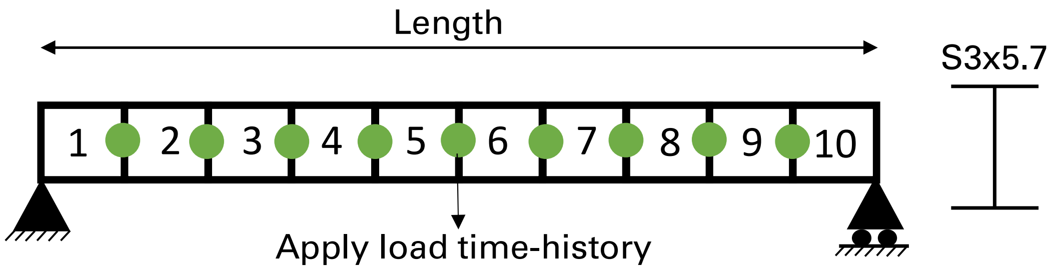

Figure 2. Representation of the beam in FEM.

Then, the modulus of elasticity is $200x10^3$ N/mm^2 and the density of the steel material is 7670 kg/m^3. As can be seen in Figure 2, the beam was discretized into ten individuals to accommodate time-history loading. A total of 1000 random Gaussian-based load time-history profiles were applied vertically at the mid-span.

Each excitation consisted of 1000 data points with a time step of $5x10^{-3}$ s, and the total duration of the analysis is 5 seconds per loading scenario. Acceleration outputs were recorded at each interior node of interest, represented by green solid circles in Figure 2. Each recorded data sample has a size of 1000 rows and 9 columns, capturing the response of the beam under the applied loading conditions. It is assumed that the damage occurs close to the point where the load is applied. Based on this assumption, the flexural stiffness (EI) of the fifth beam (Member-5) was sequentially reduced by 10% from a completely healthy case. Overall, there are nine unhealthy and one healthy labels. TABLE I provides an overview of the reduction percentage of EI and corresponding damage labels ranging from D1, representing the heaviest damage, to UN, which is the undamaged case. The EI reduction ratio gradually decreases from D1 to UN, indicating a decrease in the severity of the damage.

After simulating the structural damage by the variation in flexural stiffness and the recording of acceleration responses at the interior nodes, each corresponding non-image (tabular) data was converted into grayscale images to feed the proposed CNN. The aim of this process is to leverage the classification ability of CNNs while maximizing modeling efficiency and create computationally inexpensive way when dealing with large amount of sensor data [4].

TABLE I. LABEL INFORMATION

Labels	Reduction of EI (%)	EI value used in the analysis (%)	Label explanation
D1	90	10	Damage level - 1
D2	80	20	Damage level - 2
D3	70	30	Damage level - 3
D4	60	40	Damage level - 4
D5	50	50	Damage level - 5
D6	40	60	Damage level - 6
D7	30	70	Damage level - 7
D8	20	80	Damage level - 8
D9	10	90	Damage level - 9
UN	0	100	No damage

To facilitate this transformation, the numerical data were first normalized to ensure their suitability for image conversion. The normalization step is required to produce enhanced results and mitigate convergence issues for ML algorithms by establishing a standardized and higher-resolution scale. This process involved determining the global maxima and minima across the entire dataset and then applying Eq. 1 to achieve normalization.

$$z_i = \frac{x_i - \min(x)}{\max(x) - \min(x)} \tag{1}$$

In the above formulation, z_i is the normalized acceleration, x_i denotes the original acceleration, and $\min(x)$ and $\max(x)$ are the minimum and maximum acceleration readings of acceleration recorded in the entire dataset. After undergoing the transformation procedure, the acceleration outputs were rescaled to a range between 0 and 1, where 0 refers to the maximum negative acceleration and 1 represents the maximum positive value. An example of the input image extracted from the mid-portion of the highest level of damage (D1) is illustrated in Figure 3. It is apparent that the image depicts a discernible variation in pixel intensity, ranging from white to black, corresponding to the numerical values contained within each pixel. This grayscale representation effectively visualizes the spatial distribution and intensity of the structural response, providing valuable insights into the nature and extent of the observed damage.

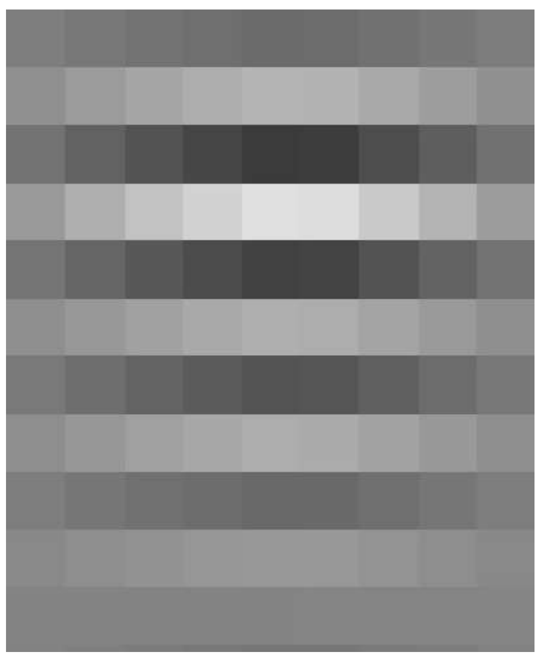

Figure 3. Monochromic grayscale example of the input image (Label is D1)

TRANSFER LEARNING AND CNN IMPLEMENTATION

In the scope of structural damage identification and classification framework, three beam cases were developed with varying length parameters. A beam whose length is

1000 mm (Beam A) was assumed to be a source domain. On the other hand, two additional datasets for the length of 1100 mm (Beam B) and 1200 mm (Beam C) were created for the target domains. Moreover, a dataset with a length of 1050 mm was generated for only testing purposes (Beam D), which was not included in any training. The overall objective is to transfer the knowledge gained from one task and apply it to another related or distinct one. In this way, TL via feature extraction (FE) includes utilizing pre-trained models that are trained on large datasets and learning general features of the data. Without a need for developing an entirely new network for each new task, TL focuses on accelerating the learning process, improving model performance, and overcoming limitations such as the need for large labeled datasets. In the FE method, all layers before the dense layers were frozen while the classification layers at the end were untouched to extract the meaningful features. Additionally, in this study, joint training (JT), also known as multi-task learning, is delved into. This technique involves training a single model to perform multiple interrelated tasks simultaneously. It functions as an upper bound, indicating the maximum level of performance attainable [5]. Rather than training separate models for each task, the model is jointly trained on a combined dataset that incorporates examples from all tasks. The shared representation layers in the model capture common features across tasks, allowing for knowledge sharing and transfer between tasks. It is important to underline that JT does not explicitly utilize a pre-trained model, yet it still benefits from the idea of knowledge transfer and leveraging shared representations. Thus, JT can be considered as a form of TL, specifically within the realm of multi-task learning.

Both FE and JT techniques employed the same CNN architecture with consistent hyperparameters (Figure 4). Following a hyperparameter tuning process, an optimal 2D CNN architecture included four convolutional layers (CONV), two pooling layers (POOL), three fully connected (FC), and a dense SoftMax output layer. The CONV layers utilized Rectified Linear Unit (ReLU) activation function to introduce non-linearity into the network. The number of filers in the CONV layers followed an ascending order of 32, 64, 128, and 256. At the end of the network, three FC-dense layers were included, each comprising 64 neurons. Moreover, the pooling operation was chosen to be maximum and the filter size in the network was selected to be 3x3. Adam optimization with a learning rate of 10^{-4} was used. To mitigate the potential overfitting issue, L1 regularization with a regularization factor of 0.001 was added to the objective function. The network was trained for a total of 300 epochs, which was determined to be the optimal number of training iterations for the proposed network.

The feature extraction method in this study can be summarized as follows; knowledge transfer was initially implemented from Beam-A, which is a pre-trained model (Model-1), to Beam-B through feature extraction (Model-2). Subsequently, the model was re-trained with Beam-C to finalize the network (Model-3). All datasets were split into 70% training, 15% testing, and 15% validation. Throughout this procedure, the hyperparameters remained consistent, while the dense layers were used to extract the features. On the other hand, the joint training approach involved collecting data samples from each dataset. The same split ratios were maintained, resulting in a random selection of 70% from each training set, 15% from both validation and testing sets. Consequently, the JT network had a total of 15,000 images for training and 3,000 images for the validation and testing datasets.

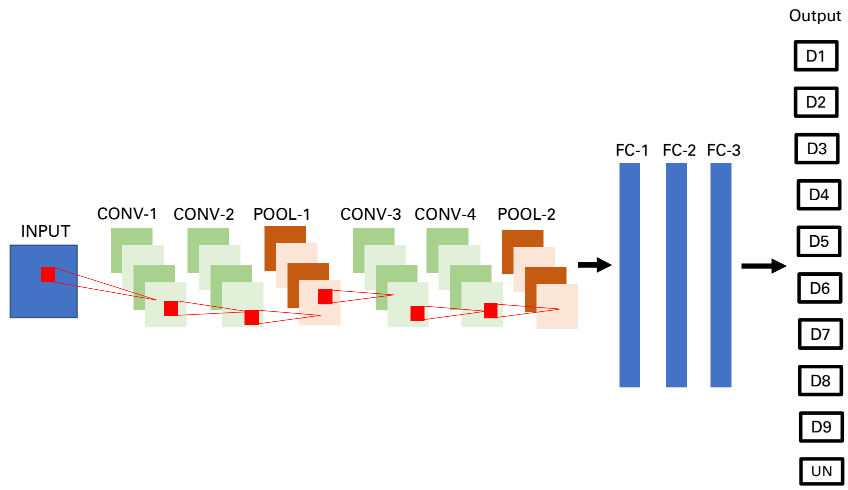

Figure 4. Applied CNN architecture for both FE and JT

RESULTS

Before implementing feature extraction, Model-1 achieved 100% accuracy for 10-label multi-class classification. At this step, when the model was tested with Beam B, Beam C, and Beam D data, the accuracy results are 0.35, 0.14, 0.33 without re-training. It is important to note that Beam D was not included in any training and can be assumed to be a dataset coming from the field experiment. Then, the TL via FE was applied, and re-training was performed by feeding the CNN with the Beam B dataset. Following this, Model-2 attained 0.28 and 0.61 accuracy for Beam C and Beam D, respectively. Afterward, Model-2 was re-trained using the Beam C dataset with the same technique. Model-3 reached 0.35 accuracy for Beam D. As expected, Model-2 achieved better accuracy for the target domain of Beam D since the length parameter is the midpoint of Beam A and Beam B. In case of using FE, the length of the beams might cause a shift in the distribution and degrades the performance of the model.

TABLE II. JOINT TRAINING - CLASSIFICATION REPORT FOR TESTING OF BEAM D

	Precision	Recall	f1-score	Support
D1	1.00	1.00	1.00	150
D2	1.00	1.00	1.00	150
D3	0.98	1.00	0.99	150
D4	0.99	0.98	0.98	150
D5	0.96	0.99	0.97	150
D6	0.93	0.96	0.94	150
D7	0.89	0.93	0.91	150
D8	0.84	0.89	0.86	150
D9	0.66	0.83	0.73	150
UN	1.00	0.57	0.73	150
Accuracy			0.91	1500
Macro avg.	0.92	0.91	0.91	1500
Weighted avg.	0.92	0.91	0.91	1500

The training and testing accuracy attained 1.0 in the JT task. Then, the model was tested with the unseen Beam D data. The results indicate that JT reached 0.92 to predict the Beam D classes successfully. The classification report is presented in TABLE II. It provides an overview of the performance of the classification model. As can be seen, the precision scores for the highest damage levels reached 1.0, it slightly goes down when the damage level gradually decreases to D9. This is because the difference in the pixel intensities is severe for the heavily-damaged cases.

CONCLUSIONS

This study is dedicated to investigating the potential applications of knowledge transfer in the context of detecting and classifying structural damage, particularly for bridge infrastructure. To that end, representative simply-supported bridge beams of various lengths were modeled in the OpenSees. Then, a Gaussian load time history was subsequently applied vertically to the beam at its midpoint, and the resulting acceleration outputs were recorded at each interior node. Through this process, for each loading case and damage scenario, 1000x9 tabular data entries were obtained. The damage was induced in Member-5 by gradually decreasing the flexural stiffness from a fully healthy case to a heavily damaged state with 10% of the original EI. In this way, ten labels were achieved with a total of 10,000 datasets per beam. Following linear FEM analysis, each of the tabular data was transformed into gray-scale images to be used as inputs for the proposed CNN, benefiting from a high capability of the classification task. Two forms of TL were presented in this study; TL via feature extraction and JT. The TL via feature extraction approach involved utilizing pre-trained models to extract relevant features from the input data, which were then fed into the CNN. On the other hand, JT involved training a single model to simultaneously handle multiple tasks. The results indicate that the JT outperforms FE. The possible reason why the accuracy goes down in FE as the training progressively occurs from source to target domains might be the non-homogenous distribution of the data and catastrophic forgetting. However, JT does not directly suffer from those issues and it can handle the task successfully. In FE method, fine-tuning and replay-based continual learning methods could be the ways to achieve better results. For generalization purpose and avoiding overfitting, JT seems to be a better option.

REFERENCES

1. C. S. N. Pathirage, J. Li, L. Li, H. Hao, W. Liu, and P. Ni, "Structural damage identification based on autoencoder neural networks and deep learning," *Engineering Structures*, vol. 172, pp. 13–28, Oct. 2018, doi: 10.1016/j.engstruct.2018.05.109.
2. D. Hajializadeh, "Deep learning-based indirect bridge damage identification system," *Structural Health Monitoring*, vol. 22, no. 2, pp. 897–912, Mar. 2023, doi: 10.1177/14759217221087147.
3. M. Zhu, F. McKenna, and M. H. Scott, "OpenSeesPy: Python library for the OpenSees finite element framework," *SoftwareX*, vol. 7, pp. 6–11, Jan. 2018, doi: 10.1016/j.softx.2017.10.009.
4. H. Khodabandehlou, G. Pekcan, and M. S. Fadali, "Vibration-based structural condition assessment using convolution neural networks," *Structural Control and Health Monitoring*, vol. 26, no. 2, p. e2308, 2019, doi: 10.1002/stc.2308.
5. B. Bagus and A. Gepperth, "An Investigation of Replay-based Approaches for Continual Learning," in *2021 International Joint Conference on Neural Networks (IJCNN)*, Jul. 2021, pp. 1–9. doi: 10.1109/IJCNN52387.2021.9533862.