Computer Assisted Methods in Engineering and Science
31(1): 3–28, 2024, doi: 10.24423/cames.1088

CMM 2022 & 42nd SolMech 2022

Convolutional Neural Networks in the SSI Analysis for Mine-Induced Vibrations

Maciej ZAJAC*, Krystyna KUZNIAR

Institute of Technology, University of the National Education Commission, Krakow, Poland
* *Corresponding Author e-mail: maciej.zajac@up.krakow.pl*

Deep neural networks (DNNs) have recently become one of the most often used soft computational tools for numerical analysis. The huge success of DNNs in the field of image processing is associated with the use of convolutional neural networks (CNNs). CNNs, thanks to their characteristic structure, allow for the effective extraction of multi-layer features. In this paper, the application of CNNs to one of the important soil-structure interaction (SSI) problems, i.e., the analysis of vibrations transmission from the free-field next to a building to the building foundation, is presented in the case of mine-induced vibrations. To achieve this, the dataset from in-situ experimental measurements, containing 1D ground acceleration records, was converted into 2D spectrogram images using either Fourier transform or continuous wavelet transform. Next, these images were used as input for a pre-trained CNN. The output is a ratio of maximal vibration values recorded simultaneously on the building foundation and on the ground. Therefore, the last layer of the CNN had to be changed from a classification to a regression one. The obtained results indicate the suitability of CNN for the analyzed problem.

Keywords: deep learning, convolutional neural networks, shallow neural networks, small data sets, soil-structure interaction, mine-induced vibrations.

1. Introduction

Deep neural networks (DNN) have recently become one of the most often used soft computational tools for numerical analysis [1–8]. By using the computational capabilities of graphics processing units (GPUs), these methods allow for quick processing of large amounts of data, making them an effective instrument, e.g., in image and speech processing [3, 4, 9].

The great success of DNN in image processing is closely tied to the deployment of convolutional neural networks (CNNs). Thanks to their distinctive structure, they allow for the effective extraction of multi-layer features, proving efficient and accurate at image classification and object detection [3, 4, 6]. In addition, compared to classical (shallow) fully connected neural networks (SNNs) with a similar number of neurons, CNNs generate fewer training parameters, which significantly reduces the learning time [8, 10].

Another advantage of using CNNs is the possibility of applying the machine learning technique called transfer learning [11, 12], which allows for adapting a model trained on one task to solve another one. This is especially helpful in the case of a small dataset – a common in experimental measurements [13].

Soil structure interaction (SSI) refers to the mutual influence between vibrations of the soil subgrade and those of the surface building structure. This interaction can have significant effects on, e.g., ground vibration transmission to a building and the actual response of the structure. Consequently, it is an important problem in the dynamic analysis and design of buildings and in the prediction of building vibrations. One of the main challenges in SSI investigation lies in the complexity of the soil-structure system [14–17].

The investigation of SSI effects is the most important for structures subjected to dynamic loads from ground motion (i.e., seismic-type influences), particularly in the context of earthquakes, as the soil can either amplify or reduce the intensity of the ground shaking [14, 16, 18, 19]. A similar problem arises with paraseismic rockbursts, with mine-induced shocks being the strongest among them [20–24].

In order to analyze and predict the very complex effects of SSI, one can use a variety of analytical and experimental tools and techniques. These include numerical modeling [25–28], laboratory testing [29, 30], and field measurements [21, 22, 31, 32], among others.

In this paper, the application of CNNs in an important soil-structure interaction (SSI) problem, i.e., the analysis of mine-induced vibrations' transmission from the free-field next to a building to its foundation, is presented.

One of the main reasons for choosing CNNs to analyze vibration transmission from the ground to the building was their ability to effectively train algorithms with a small number of patterns through the use of transfer learning. Such a learning technique is widely used for this type of architecture and is characterized by its efficiency and ease of implantation. In addition, focusing on a specific network allows for detailed comparative analyses of the most commonly used and recognized CNN network architectures such as VGG or AlexNet. Therefore, we have decided not to analyze the usefulness of, e.g., the long short-term memory recurrent neural network (LSTM RNN) at this stage of the analysis. The primary reason for this decision is that, in the case of LSTM networks, transfer learning is not widely employed. This is mainly because of the limited number of ready-made networks and the specificity of the data on which they were trained (the similarity of the input between the original and the new problem to be solved).

For the analysis, a small dataset from in-situ experimental measurements, comprising fewer than 500 ground acceleration 1D records, was converted to

2D spectrogram images using either Fourier transform or continuous wavelet transform. These images were next used as input for a pre-trained CNN. The output is a ratio ra as a fraction of the maximal values (amplitudes) of vibrations recorded simultaneously on the building foundation ($a_{f\,\max}$) and on the ground ($a_{g\,\max}$). Therefore, the last layer of CNN had to be modified from a classification to a regression one.

Additionally, the obtained results using CNN were compared with those of SNNs presented in [33].

The major, novel contributions of this paper are as follows:

- proving usefulness of selected CNNs in SSI analysis and verifying them as an alternative tool to SNNs,
- adapting the architecture of CNNs to the regression problem,
- analyzing the impact of the method of image input generation on the CNN prediction results,
- providing creative suggestions for fine-tuning the architecture analysis to improve the CNN prediction,
- verifying the suitability of CNNs in handling a small set of experimental data.

2. Application of artificial neural networks for regression problem

Artificial neural networks are one of the main tools of artificial intelligence (AI) used in machine learning [8, 34–36]. As the 'neural' part of their name suggests, they are brain-inspired systems which aim to mimic the learning processes observed in the human brain.

Artificial neural networks can be divided into two main types: the historically older ones (known as SNNs) and the very popular and recent DNNs. CNN is a subtype of DNN [1, 2, 8]. Figure 1 shows the appropriate diagram.

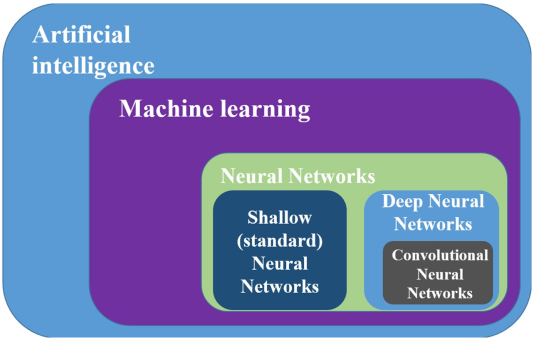

Fig. 1. The schematic subsets of AI.

SNNs, in contrast to deep learning ones, have one or a maximum of two hidden layers (see Fig. 2a), resulting in fewer parameters to optimize, what makes them generally simpler and easier to train. Their small size means that they do not need a lot of computing power. However, the simple network architecture has its disadvantages. One of them is limited model capacity that leads to limitations in extracting and analyzing complex patterns [8, 10, 13]. Despite these limitations, SNNs can still be effective in certain applications, particularly in scenarios where computational resources are limited [8].

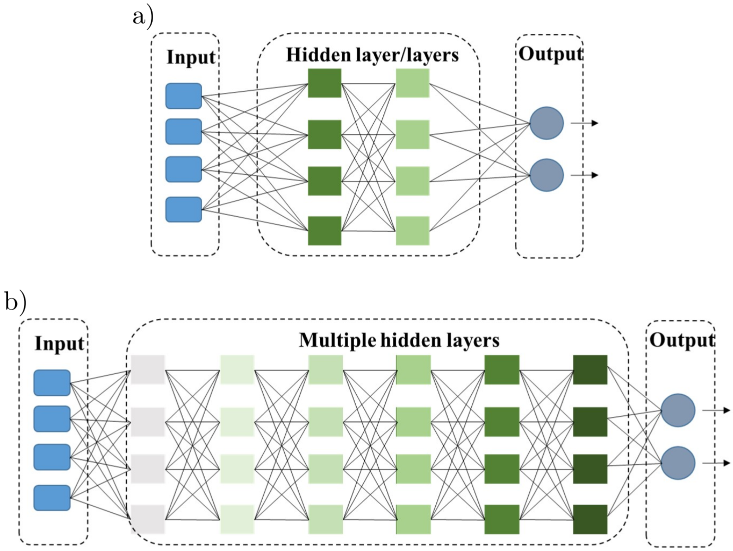

Fig. 2. Demonstration of architectures: SNN (a) and DNN (b).

A characteristic feature of deep neural networks is that they have many hidden layers (see Fig. 2b) that enable the analysis of complex relationships within data. A large part of deep learning algorithms is used for classification problem [3, 8, 35], it is worth noting that some of them can be applied to regression tasks to predict a value based on a given set of input data. One example is the multilayer perceptron (see Fig. 2b) where the input data is typically first processed through many fully connected layers, which are responsible for extracting a set of features from the data, and the final layer generates predictions for the output [13, 37, 38].

The development of neural network architectures allowed to extend their predictive capabilities, particularly in the field of regression. One of the new and promising algorithms is called the CNN. This type of architecture was originally designed for image recognition and processing. CNN architecture consists of multiple convolutional layers with learnable filters responsible for identification and extraction of specific features that may not be immediately visible to the human eye [3, 4, 6]. The basic architecture of a typical CNN is presented in Fig. 3.

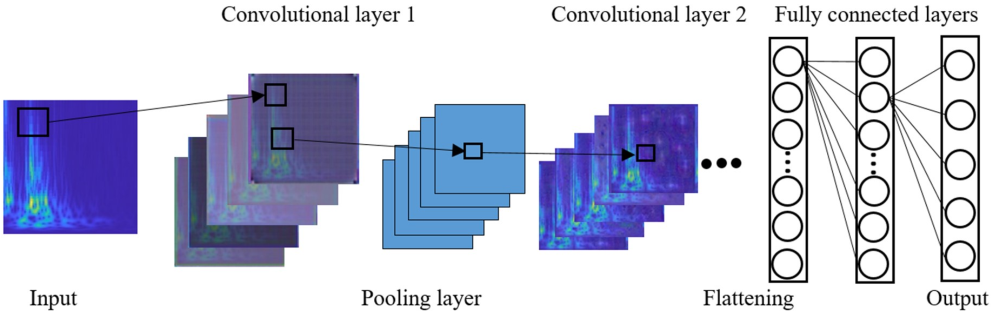

Fig. 3. Basic architecture of a typical CNN.

CNNs require a large amount of data to attain high prediction accuracy [3, 4]. This is due to their complex, deep architecture with millions of parameters whose values are computed during the learning process. For problems of classification or image recognition, there are ready-made databases with millions of images available to use [39, 40]. In addition, to increase the number of patterns, a technique called data augmentation is often applied [41].

Unfortunately, in the case of regression problems, data availability is often constrained, which makes it difficult to train deep networks and achieve satisfactory accuracy. There is also no possibility of generating new patterns, especially in the case of experimental results. In addition, training on a small training set can lead to overfitting or underfitting problem [2, 35, 36].

One way to use the advantages of CNNs in regression analysis with a small number of patterns is to apply techniques known as transfer learning and fine-tuning [11, 12, 42].

Transfer learning allows to use a previously trained network for a new problem or different data. This technique takes advantage of the feature extraction capabilities acquired during learning on a large pattern base and transfers this knowledge to solve another issue, helping to improve the performance and accuracy of the model on the new task [11, 12].

Many different approaches of transfer learning techniques exist. We can use the entire pre-trained model or only a part of it, and the rest built from scratch [11, 13, 42].

Fine-tuning is another powerful tool for improving the performance of neural networks on a specific task. This technique involves adjusting the hyperparameters of a pre-trained neural network to generate more accurate prediction for a new problem. Depending on the task, fine-tuning can be applied to the entire architecture, its parts or the final layers, responsible for repurposing the network [11, 13, 42].

The main stages in transfer learning including fine-tuning are shown in Fig. 4.

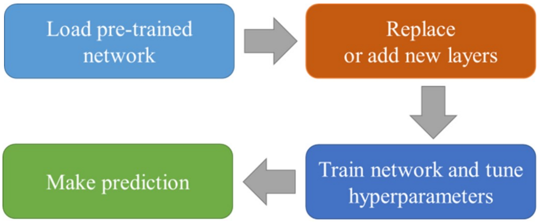
Load pre-trained network
Replace or add new layers
Train network and tune hyperparameters
Make prediction

Fig. 4. The main stages in transfer learning, including fine-tuning.

3. SSI measurement data for mine-induced vibrations

In this paper, we propose to use CNNs for the analysis of a difficult type of SSI problem in the context of mine-induced vibrations. This specific problem of SSI deals with significant differences in vibrations between building foundation and the adjacent free-field ground vibrations occurring simultaneously [21, 22]. Thus, the goal is to predict the transmission of these ground vibrations to the building basement.

The neural network input and output data are based on the results of full-scale measurements performed in the Legnica–Glogow Copperfield (LGC) – a copper ore mining area. A representative example of numerous similar residential buildings is taken into consideration. The building's plan and vertical section are schematically shown in Fig. 5. The figure also shows the positions

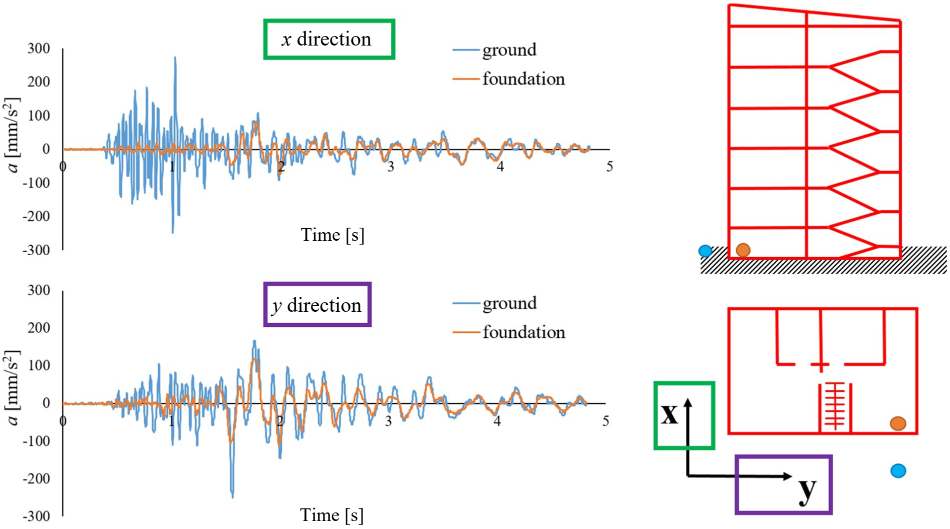

Fig. 5. The plan and vertical section of the building, positions of the ground and foundation seismic stations, and records of measured free-field and foundation vibration.

of the ground and foundation seismic stations (blue and orange dot, respectively) as well as demonstration records of measured free-field and foundation vibration (acceleration a). Accelerations in the time domain on the ground surface near to the analyzed structure and on the building foundations are registered concurrently using a seismic monitoring system. Specifically, 'an armed partition' accelerometers are employed, starting to work at the assumed signal level. These sensors are placed in rigid nodes of transverse-longitudinal structural elements in the building. The ground accelerometers are positioned approximately 5 m from the building structure. This distance ensures minimal or no impact from the existing building on the measured free-field vibrations. Such a distance (5 m) between the free-field and the building sensors is smaller than the distance usually recommended for natural seismic vibrations. However, due to the characteristic properties of mine-induced shocks, we can expect that the building vibrations with relatively low natural frequencies are not substantially affecting ground vibration records with very high predominant frequencies.

The experimental focus was on horizontal vibrations along the x and y directions, parallel to the crosswise and lengthwise building axis (see Fig. 5). The collected set of measurement data consists of 464 pairs of accelerations (in time domain) of ground and foundation vibrations [22, 43]. Randomly selected 50% of these patterns were intended for the learning set of the CNN. The validation and testing sets were created using equal portions – 25% for the validation set and 25% for the testing set.

4. CNNs for SSI problem

4.1. Input data pre-processing

For the neural network input, a small dataset from *in situ* experimental measurements, containing fewer than 500 ($P = 464$) ground acceleration 1D records, was converted to 2D spectrogram images using short-time Fourier transform (STFT) [44, 45] as well as continuous wavelet transform (CWT) [45, 46], see Fig. 6. A total of $P = 464$ spectrograms were generated for the x and y components of the recorded ground accelerations. As mentioned above, these spectrograms were randomly divided as the learning, validation and testing inputs for the neural network.

The use of both transformations allowed to generate spectrograms representing the frequency content of a signal over time. The horizontal axis corresponds to the duration of the shock and the vertical axis represents the frequency range from 0 to 100 Hz. After generating the spectrograms, the images were scaled to the size required by the appropriate neural network.

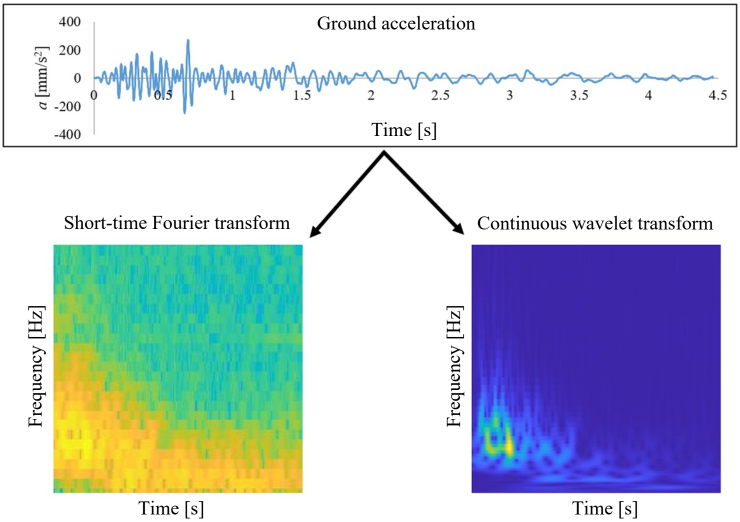

Fig. 6. Conversion of a ground acceleration 1D record to 2D spectrogram images using STFT and CWT.

All the analyzed networks have a three-channel input (RGB), which allowed to test the influence of spectrogram colors selection on neural prediction results. Figure 7 shows four colormaps that were used to generate inputs of designed variants of neural networks. Whereas Figs. 8 and 9 present spectrograms of the

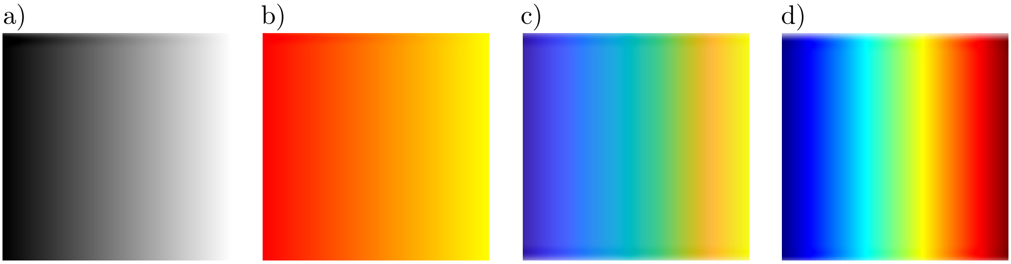

Fig. 7. Colormaps used to generate spectrograms:
a) Gray (G), b) Autumn (A), c) Parula (P), d) Jet (J).

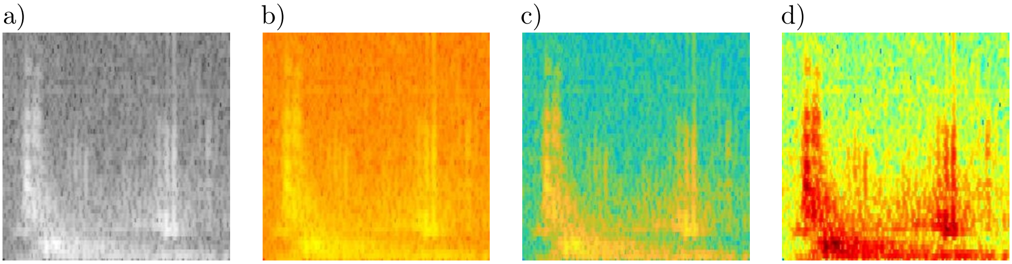

Fig. 8. STFT spectrograms of the same chosen accelerogram for different colormaps:
a) Gray, b) Autumn, c) Parula, d) Jet.

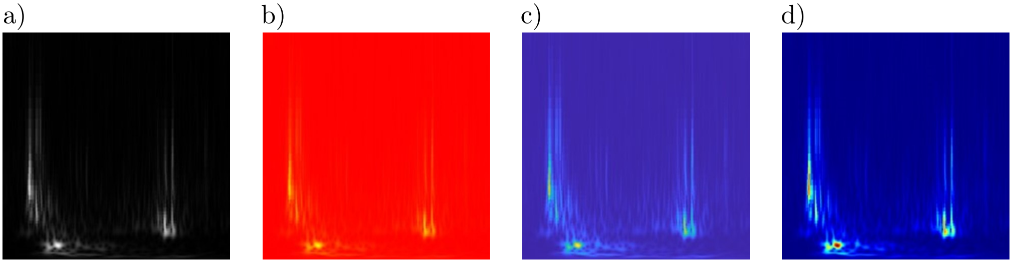

Fig. 9. CWT spectrograms of the same chosen accelerogram for different colormaps: a) Gray, b) Autumn, c) Parula, d) Jet.

same exemplary record of ground vibration in the case of various considered color palettes and transformation types (STFT in Fig. 8 and CWT in Fig. 9, respectively).

It is worth noting that generating inputs using STFT requires the selection of appropriate transformation parameters [44, 45]. Figure 10 provides an exemplary illustration of the possible differences in selected spectrogram generated when employing different window size and overlap values.

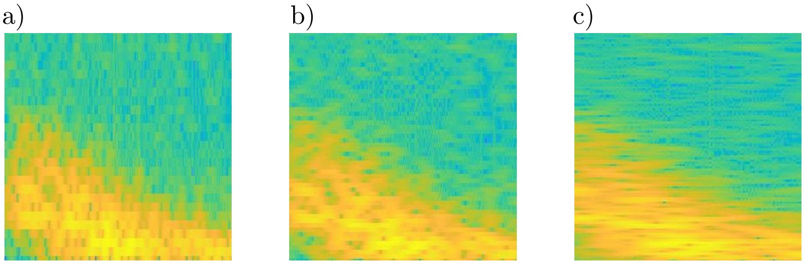

Fig. 10. STFT spectrograms of chosen accelerogram for different values of window size and overlap: a) 128, 120; b) 256, 250; c) 512, 500.

4.2. Output parameters

To estimate vibration transmission from the free-field to the building foundation, the ratio *ra,* expressed as a fraction of maximal values (amplitudes, peak accelerations) of accelerations in time domain simultaneously measured (recorded) on the building foundation ($a_{f\,\max}$) and at the free-field ($a_{g\,\max}$), was calculated for each pair of vibration, and used as the network output:

$$ra = a_{f\,\max}/a_{g\,\max}. \tag{1}$$

The comparison of peak values of vibrations recorded simultaneously on the ground and at the foundation level is the simplest method for evaluating differences between such ground and foundation records. This frequently used evaluation method of vibration transmission, however, does not take into account the important parameter of vibration – frequency. It is also worth noting that the peak ground and peak foundation accelerations generally do not occur simultaneously due to the fact that the ground vibration is usually modified at the contact between the soil and the foundation.

Figure 11 illustrates the idea of preparing CNN output parameters according to the formula (1) ($a_{f\,\max}$ and $a_{g\,\max}$ are marked with dots).

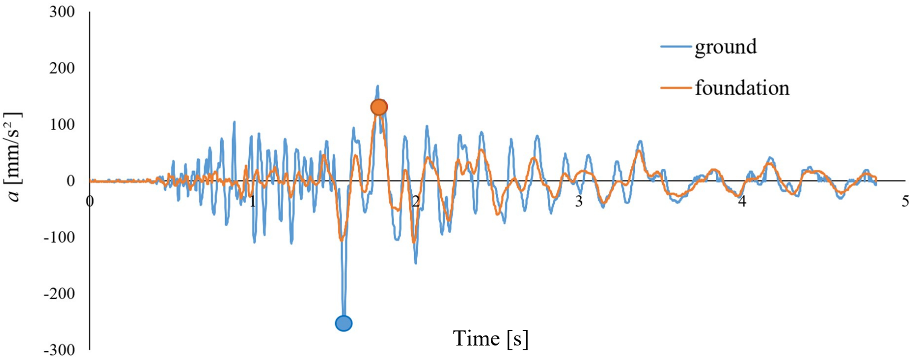

Fig. 11. Illustration of the idea of preparing CNN output parameters.

4.3. Architectures and parameters of chosen CNNs

Four architectures of CNNs were selected for the calculations, which, after modifying the final values, were employed to solve the regression problem. CNNs chosen for the analysis are marked in red in Fig. 12.

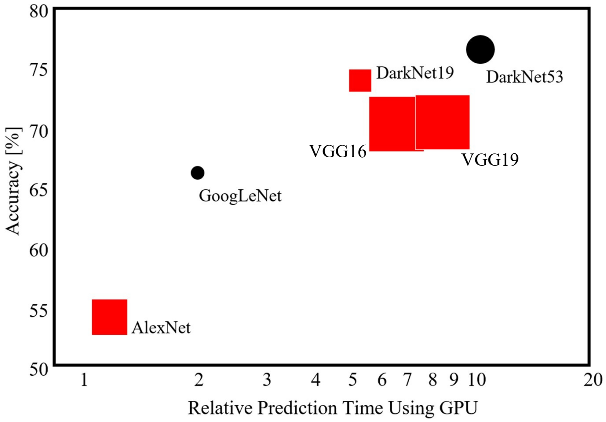

Fig. 12. Types of pre-trained networks (based on [47]).

The selected networks vary in depth, the number of convolutional layers, the number of learnable parameters, the size of the input image, accuracy as well as relative prediction time [3, 4, 48]. As a selection criterion, we choose the ability to adapt the neural network architecture to solve new regression task by removing the final layers responsible for classification and adding new ones, i.e., fully connected and regression layers. Some of the parameters of chosen and used types of CNNs are listed in Table 1.

Table 1. Parameters of chosen CNNs [3, 4, 48].

CNN	Total number of layers	Number of convolutional layers	Number of fully connected layers	Number of learnable parameters	Input size
AlexNet	25	5	3	61M	$227 \times 227 \times 3$
DarkNet19	64	19	0	21M	$256 \times 256 \times 3$
VGG16	41	13	3	138M	$224 \times 224 \times 3$
VGG19	47	16	3	144M	$224 \times 224 \times 3$

Figure 13 shows the AlexNet architecture (Figs. 13a, b, and c) generated in the Matlab Deep Learning Toolbox [47] and the modification scheme of the last layers to change the purpose of the network (Fig. 13d). Similar modifications were applied to the other considered networks.

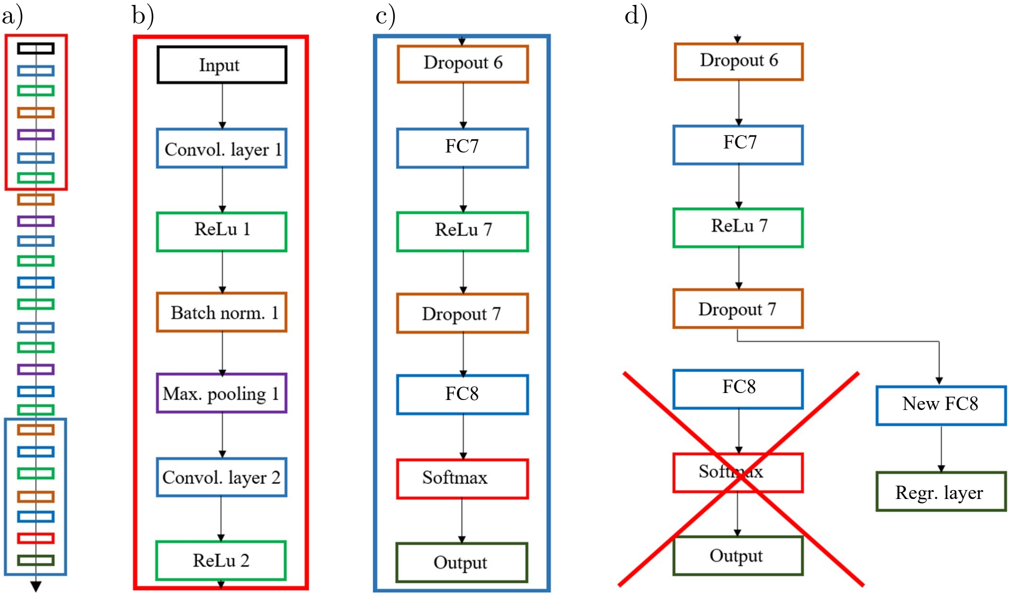

Fig. 13. Scheme of the original AlexNet architecture (a, b, c) and final layers replacement (d) in MATLAB Deep Learning Toolbox (based on [47]).

After modifying the network architecture, the new networks were named as follows: A (based on AlexNet), V16 (based on VGG16), V19 (based on VGG19), and D19 (based on DarkNet19). In addition, each neural network's name includes the type of signal transformation and the colormap used. For example, D19_STFT_P represents a network based on the DarkNet19 architecture, where the input is a spectrogram obtained through STFT in color Parula.

Finding the neural network with the best prediction is difficult due to the large number of hyperparameters defining the learning process. In this work, selected networks were first tested in terms of prediction quality due to the method of generating spectrograms from ground vibration accelerations (CWT and STFT) and color representation in three channels – red, green, blue.

In the first step of the analysis, all the considered networks were trained using the following parameters: 1000 learning epochs, a full batch size equal to 232 patterns, a learning parameter set to 0.0001, Adam optimizer and a new fully connected (FC) layer with a regression output.

All computations were carried out in the MATLAB environment using the Deep Learning Toolbox with GPU support [47].

For the estimation of neural network prediction accuracy, mean square errors (*MSE*), relative errors (*ep*), coefficients of linear correlation (r) for training (L), validating (V) and testing (T) were computed. Additionally, a success ratio *SR*(*ep*) [%] was determined for all neural network variants (in statistics: the cumulative curve illustrating the percentage of patterns with relative errors less than *ep*).

5. Results and discussion

A summary of the results for neural learning, validation and testing of different input transformation types and colormaps of CNNs for the selected networks is presented in Table 2.

Figures 14 to 17 show *SR* for A, V16, V19 and D19 network testing depending on the transform type (STFT, CWT) of ground acceleration in the time domain and colormaps.

One can see that the A network based on AlexNet, in terms of relative errors up to 30%, performs better for the input with CWT transformation and colormaps with a wide spectrum of colors (P – Parula, J – Jet). Whereas in the case of the V16 network based on VGG16, V19 network based on VGG19, and D19 network based on DarkNet19, better results in terms of small relative errors were obtained for the networks with the STFT transformation of network input information (i.e., ground vibration in time domain). At the same time, no particular influence of colormaps on the quality of results was observed.

A comparison of the percentage of success for learning, validation and testing in the case of networks with the parameters giving the best results listed in Table 2, is shown in Fig. 18. In addition, Fig. 19 shows the graphs of the output-target and the absolute error for testing for the V19_STFT_P network (STFT,

Table 2. Comparison of *MSE* and r (for learning, validation and testing) obtained for various input transformation types and colormaps of CNNs.

Network	Transform.	Colormap	*MSE*(L)	*MSE*(V)	*MSE*(T)	$r(L)$	$r(V)$	$r(T)$
A	CWT	Autumn	0.0262	0.0304	0.0320	0.8242	0.5964	0.5752
		Gray	0.0204	0.0323	0.0308	0.8469	0.5415	0.5828
		Jet	0.0279	0.0389	0.0363	0.8139	0.4910	0.5237
		Parula	0.0271	0.0315	0.0313	0.8575	0.5992	0.5991
	STFT	Autumn	0.0312	0.0578	0.0374	0.8379	0.3898	0.5412
		Gray	0.0397	0.0626	0.0437	0.8534	0.4196	0.5225
		Jet	0.0302	0.0390	0.0360	0.8013	0.4301	0.4804
		Parula	0.0224	0.0415	0.0383	0.8135	0.4893	0.4179
V16	CWT	Autumn	0.0386	0.0449	0.0356	0.8288	0.5736	0.6365
		Gray	0.0210	0.0293	0.0289	0.8861	0.6481	0.6283
		Jet	0.0184	0.0353	0.0302	0.8576	0.5441	0.6003
		Parula	0.0273	0.0359	0.0319	0.8587	0.5731	0.6185
	STFT	Autumn	0.0225	0.0351	0.0325	0.8536	0.4846	0.6446
		Gray	0.0232	0.0433	0.0325	0.8399	0.3034	0.5787
		Jet	0.0183	0.0396	0.0300	0.8968	0.4802	0.6328
		Parula	0.0174	0.0361	0.0339	0.8958	0.4419	0.5712
V19	CWT	Autumn	0.0213	0.0329	0.0267	0.8403	0.5717	0.6430
		Gray	0.0192	0.0313	0.0304	0.8911	0.6102	0.6222
		Jet	0.0154	0.0344	0.0286	0.8876	0.5408	0.6158
		Parula	0.0201	0.0345	0.0250	0.8857	0.5446	0.6845
	STFT	Autumn	0.0167	0.0401	0.0284	0.8928	0.5157	0.6256
		Gray	0.0275	0.0647	0.0372	0.8508	0.4485	0.5769
		Jet	0.0142	0.0353	0.0292	0.9053	0.5071	0.6026
		Parula	0.0142	0.0304	0.0235	0.9076	0.6013	0.7054
D19	CWT	Autumn	0.0083	0.0422	0.0424	0.9518	0.3650	0.3841
		Gray	0.0086	0.0444	0.0386	0.9397	0.4660	0.4620
		Jet	0.0095	0.0404	0.0414	0.9403	0.4874	0.4450
		Parula	0.0065	0.0365	0.0326	0.9641	0.5083	0.5967
	STFT	Autumn	0.0072	0.0485	0.0421	0.9517	0.2935	0.3729
		Gray	0.0105	0.0468	0.0432	0.9403	0.3261	0.3486
		Jet	0.0093	0.0434	0.0463	0.9449	0.3899	0.3856
		Parula	0.0086	0.0534	0.0360	0.9416	0.3297	0.4830

colormap – Parula – P). As one can see, in the case of learning, the best results were obtained for the D19 network, without the original fully connected layers. For validation and testing, for the error range up to 50%, all analyzed networks show similar predictive properties.

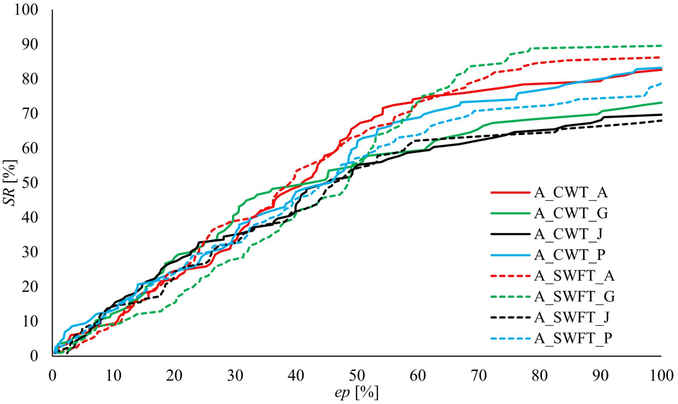

Fig. 14. *SR* in predicting the transmission of ground vibrations to a building for A network in testing depends on transform type and colormaps.

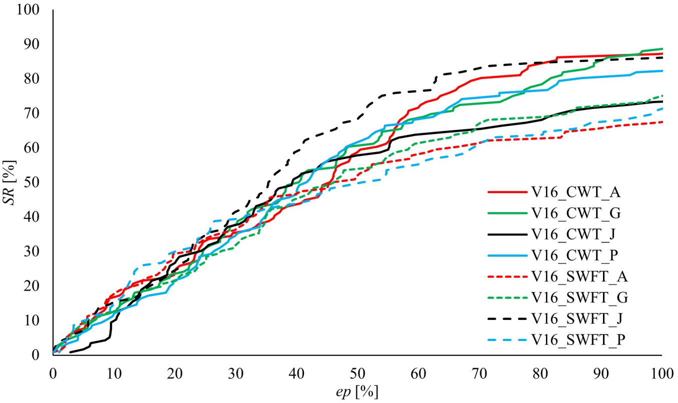

Fig. 15. *SR* in predicting the transmission of ground vibrations to a building for V16 network in testing depends on transform type and colormaps.

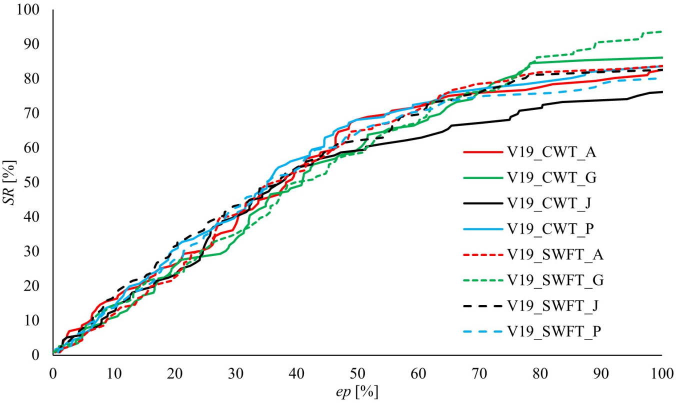

Fig. 16. *SR* in predicting the transmission of ground vibrations to a building for V19 network in testing depends on transform type and colormaps.

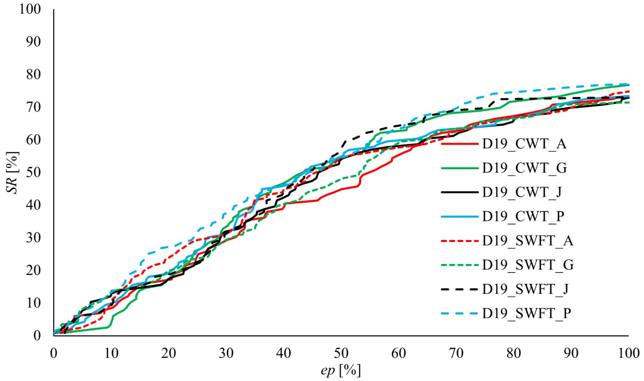

FIG. 17. *SR* in predicting the transmission of ground vibrations to a building for D19 network in testing depends on transform type and colormaps.

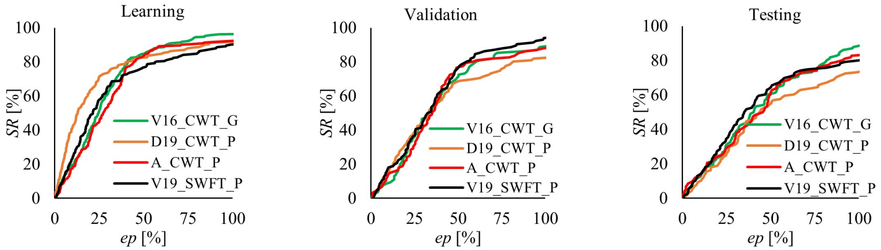

FIG. 18. *SR* in predicting the transmission of ground vibrations to a building for the chosen four networks with the best performance.

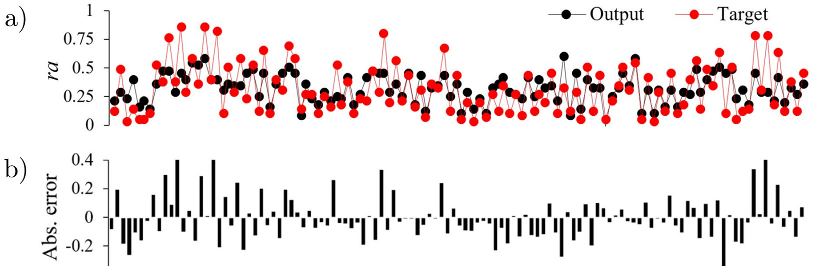

FIG. 19. V19_STFT_P network testing result: a) output-target relation, b) output-target absolute error distribution.

The results presented in Table 2 and Figs. 14–19 were used to select the networks with the best properties, which in the next step were subjected to another test aimed at improving them.

Because the original source networks were trained on a set of patterns different from the specific patterns of the analyzed problem, after adapting their architecture to the regression problem, they had to be additionally fine-tuned to the new data. This is done by unfreezing the appropriate number of layers, which means that it is possible to update the weights of neurons during training. The ability to fine-tune a network learned on a large dataset allows its properties to be used for a new problem with a small number of data, which is competitive to standard, shallow networks.

In the next step of the analysis, four various levels of layer freezing were used. The first case of layer freezing, denoted as F1, means that all the original layers of the basic network have been frozen and only the newly created fully connected layer is updated. The second case, denoted as F2, means thawing of all final layers including the original FC layers. The third case, denoted as F3, additionally unfreezes the last convolutional layer. On the other hand, networks denoted as FN are networks without layer freezing (where all parameters are updated). So, in the case of the D19 network, which originally does not have FC layers, and, subsequently, counting from the end, convolutional layers are unfrozen.

It should be noted that defrosting more and more layers entails an increase in numerical effort and thus the demand for computing power. The time of neural network learning is also significantly increased.

The results presented in Table 3 indicate that thawing successive layers improves predictive properties. This may be due to the difference in the content of the original and new training data and the need to adjust the parameters to the problem of regression instead of classification.

Table 3. Comparison of *MSE* and r (for learning, validation and testing) for different number of frozen layers of chosen CNNs.

Network	Frozen layers	$MSE(L)$	$MSE(V)$	$MSE(T)$	r_L	r_V	r_T
A_CWT_P	F1	0.0271	0.0315	0.0313	0.8575	0.5992	0.5991
	F2	0.0030	0.0278	0.0315	0.9934	0.6499	0.5755
	F3	0.0012	0.0273	0.0294	0.9970	0.6455	0.6062
	FN	0.0005	0.0297	0.0300	0.9987	0.6298	0.5990
V16_CWT_G	F1	0.0210	0.0293	0.0289	0.8861	0.6481	0.6283
	F2	0.0012	0.0241	0.0237	0.9967	0.7195	0.7029
	F3	0.0019	0.0239	0.0212	0.9968	0.7011	0.7306
	FN	0.0010	0.0254	0.0181	0.9983	0.6972	0.7920
D19_CWT_P	F1	0.0065	0.0365	0.0326	0.9641	0.5083	0.5967
	F2	0.0004	0.0338	0.0301	0.9979	0.5464	0.6136
	F3	0.0001	0.0341	0.0333	0.9993	0.5422	0.5555
	FN	0.0001	0.0367	0.0254	0.9992	0.6185	0.7234

Figure 20 presents the results of the percentage of testing success for three selected networks depending on the depth of layer freezing. It can be seen that in the case of large networks (V16, V19, D19), unfreezing all layers significantly improves the predictive capabilities of the networks.

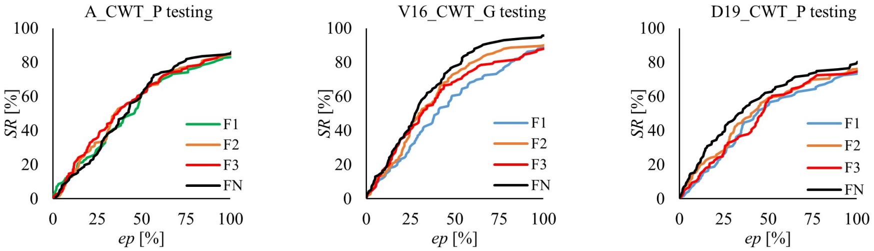

Fig. 20. *SR* in predicting the transmission of ground vibrations to a building for chosen networks in the case of various number of frozen layers.

In the next test, the influence of the number of FC layers on the quality of the results was investigated. The purpose of the FC layers in a convolutional network is to transform the data from a matrix to a vector and to introduce non-linearity to the model.

The D19 network based on the DarkNet19 network, which initially did not have such layers, was used for the above-mentioned analysis. In the most developed version of the D19 network, four FC layers were added with the following number of neurons: 1000-500-500-250.

The obtained results are presented in Table 4 and Fig. 21. Adding successive layers, and thus increasing the capacity of the model without increasing the number of training data, results in a deterioration of the network's learning ability. A similar effect is seen for validation. In the case of testing, for errors up to 50%, this effect is present but not so clearly visible.

Table 4. Comparison of *MSE* and r (for learning, validation and testing) for various number of fully connected layers for D19_CWT_P networks.

No. of FC layers	$MSE(L)$	$MSE(V)$	$MSE(T)$	r_L	r_V	r_T
1	0.0065	0.0365	0.0326	0.9641	0.5083	0.5967
2	0.0257	0.0533	0.0386	0.9756	0.5340	0.5953
3	0.0310	0.0519	0.0362	0.9805	0.5570	0.6180
4	0.0483	0.0596	0.0436	0.9774	0.5272	0.6212

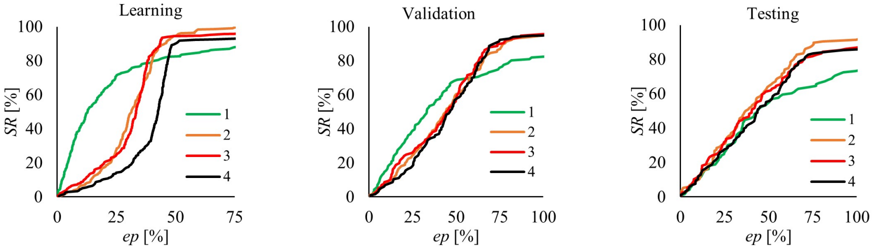

Fig. 21. *SR* in predicting the transmission of ground vibrations to a building for D19_CWT_P networks with different number of fully connected layers.

Effective training of deep networks requires a large amount of training data. During this process, the data is divided into so-called batches that constitute the entrance to the network. The division into batches increases the number of iterations during one epoch of learning, which may extend the analysis time, but at the same time reduces the demand for computing power and improves optimization using the stochastic gradient descent method. Thus, reducing the amount of input data is advisable in the case of calculations on hardware with relatively little computing power. Table 5 and Fig. 22 show the results of calculations using the V19_CWT_G network in the case of dividing the learning patterns into batches of various sizes, where the full batch contains 232 patterns.

Table 5. Comparison of *MSE* and r (for learning, validation and testing) for various learning batch sizes for V19_CWT_G network.

Batch size	$MSE(L)$	$MSE(V)$	$MSE(T)$	r_L	r_V	r_T
29	0.0963	0.1336	0.1422	0.7500	0.5120	0.5657
58	0.0599	0.0766	0.0678	0.8269	0.5324	0.5399
116	0.0222	0.0369	0.0361	0.8757	0.5854	0.5824
232	0.0192	0.0313	0.0304	0.8911	0.6102	0.6222

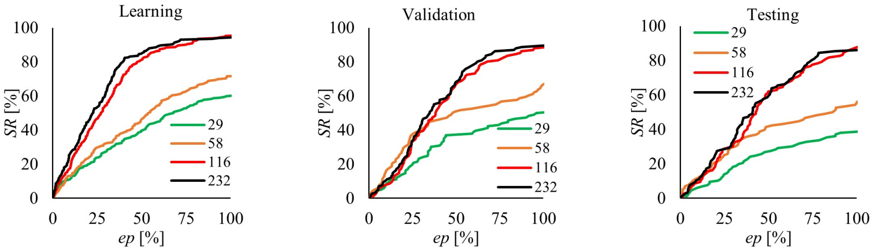

Fig. 22. *SR* in predicting the transmission of ground vibrations to a building for V19_CWT_G networks with different learning batch sizes.

The obtained results indicate that the batch size has a significant impact on the quality of the network in the case of analyzing the problem with a small number of learning patterns. For the V19_CWT_G network, better results are obtained with a set with more data (full batch and half full batch).

The use of CNNs for regression in the case of SSI analysis requires the transformation of data in the form of ground vibration accelerations into images with a size depending on the selected base network. One of the possible methods of this type of conversion is the STFT, which allows changing the signal into a spectrogram in the time domain. When constructing it, several parameters are given, the key of which is the size of the window, which determines the resolution of the spectrum in relation to time. Table 6 and Figs. 23 and 24 summarize the results of selected networks for input data generated using windows of widths of 128,

Table 6. Comparison of *MSE* and r (for learning, validation and testing) for various Fourier transform window size of chosen CNNs.

Network	Widow size	$MSE(L)$	$MSE(V)$	$MSE(T)$	r_L	r_V	r_T
A_STFT_G	128	0.0350	0.0523	0.0396	0.8129	0.5101	0.5255
	256	0.0397	0.0626	0.0437	0.8534	0.4196	0.5225
	512	0.0374	0.0542	0.0403	0.7629	0.4165	0.4843
A_STFT_P	128	0.0228	0.0302	0.0305	0.8335	0.5753	0.6178
	256	0.0224	0.0415	0.0383	0.8135	0.4893	0.4179
	512	0.0347	0.0556	0.0400	0.7589	0.4468	0.4839
V19_STFT_G	128	0.0228	0.0351	0.0286	0.8216	0.5805	0.6436
	256	0.0275	0.0647	0.0372	0.8508	0.4485	0.5769
	512	0.0238	0.0382	0.0291	0.8182	0.4324	0.6067
V19_STFT_P	128	0.0193	0.0242	0.0301	0.8701	0.6981	0.5980
	256	0.0142	0.0304	0.0235	0.9076	0.6013	0.7054
	512	0.0199	0.0339	0.0284	0.8570	0.5542	0.6142

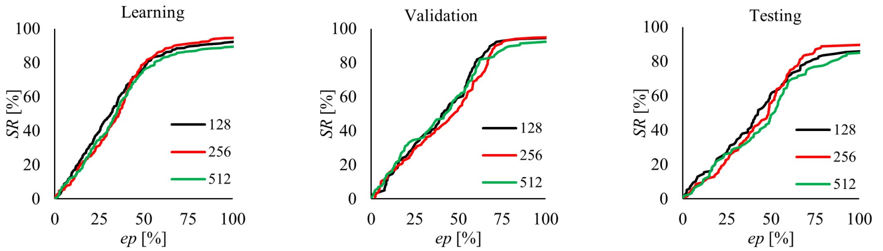

Fig. 23. *SR* in predicting the transmission of ground vibrations to a building for A_STFT_G networks with different STFT window sizes.

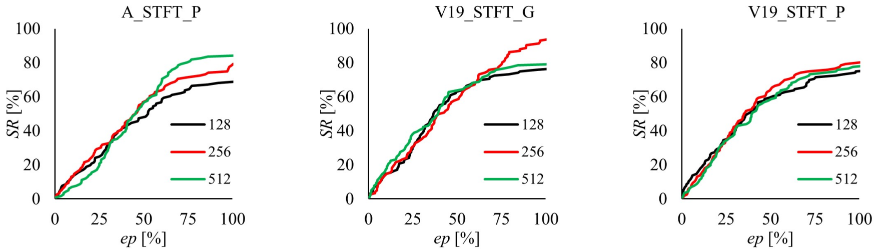

Fig. 24. *SR* in predicting the transmission of ground vibrations to a building for testing for chosen networks in the case of various STFT window sizes.

256 and 512. For example, for the A_STFT_G network (Table 6, Fig. 23), the window size does not significantly affect the quality of the results for learning, validation and testing. A similar tendency can be observed in the case of the network testing results presented in Fig. 24 and Table 6. The low sensitivity of

the network to the size of the window parameter facilitates the preparation of input data in the form of spectrograms.

Another problem deals with the number of CNN learning epochs. The choice of this parameter affects the generalizing properties of the network and allows protecting against underfitting or overfitting. In the case of training DNNs from scratch on a large data set, this number can be large, which means long-time training. For the analyzed case, with a small training set and the fine-tuning technique, the use of about 500 epochs allowed achieving satisfactory results. Table 7 and Fig. 25 show the prediction results for the V16_CWT_G_FN network depending on the different number of learning epochs.

Table 7. Comparison of *MSE* and r (for learning, validation and testing) for various number of training epochs – V16_CWT_G_FN network.

No. of epochs	$MSE(L)$	$MSE(V)$	$MSE(T)$	r_L	r_V	r_T
250	0.0052	0.0303	0.0229	0.9924	0.6718	0.7772
500	0.0024	0.0266	0.0195	0.9978	0.6863	0.7847
750	0.0009	0.0258	0.0181	0.9978	0.6972	0.7882
1000	0.0010	0.0254	0.0181	0.9983	0.6972	0.7920
1250	0.0018	0.0262	0.0183	0.9984	0.6862	0.7964

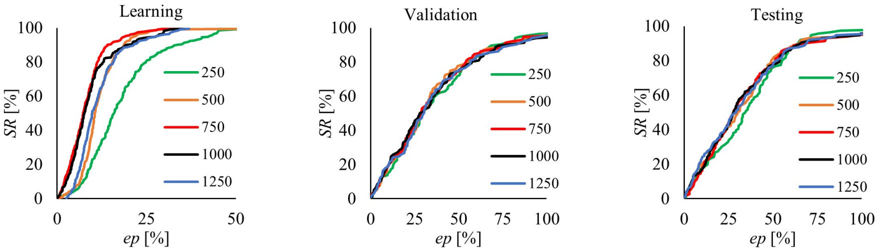

Fig. 25. *SR* in predicting the transmission of ground vibrations to a building for testing for various number of training epochs in the case of V16_CWT_G_FN network.

Early stopping is one of the techniques to prevent the network overtraining, which allows one to stop the training process when the learning error decreases with the increase in the number of epochs and the validation error after the initial decrease starts to increase. Stopping the learning process early also saves time and numerical analysis costs. For the V16_CWT_G_FN network and 1000 learning epochs, this is about 2.2932e+18 multiplication and addition operations. However, in the case of training deep neural networks, an early stop may not be optimal due to the occurrence of the 'double descent' phenomenon, in which, after an increase in the validation error caused by overfitting, it decreases with further training and thus improves the generalizing properties. Hence, it is necessary to analyze the sensitivity of the network to the number of learning epochs.

In the last step, the results of one of the discussed CNNs – V16_CWT_G_FN network and two SNNs (denoted as SNN1 and SNN2) [33] with the best predictive properties were compared.

This version of V16_CWT_G_FN network is built based on VGG16, where the inputs are spectrograms obtained by wavelet transformation in the gray spectrum, with no layers freezing and with 1000 learning epochs. Figure 26 shows the output-target plots of this network for training, validation, and testing. The red line marks the location of points for the coefficients of linear correlation equal to 1. It is visible that the differences in expected (target) and calculated (output) values are smaller for learning patterns than for validation and testing patterns. Nevertheless, it is worth noting that this CNN was selected as the best based on the results in Table 7, and this is the network for which the testing error is the smallest with a given number of learning epochs. Therefore, despite the differences in training and testing errors, the results in Table 7 do not indicate overfitting because of the constant value of the validation and testing errors. No upward trend was observed for the validation and testing curve.

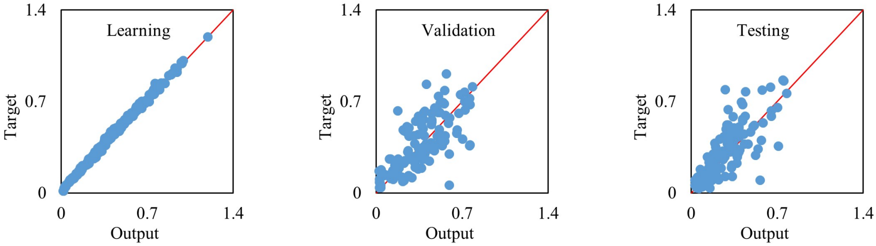

Fig. 26. Target vs. output relation for learning, validation and testing for V16_CWT_G_FN network.

Both the SNN1 and SNN2 were designed as multi-layer perceptrons with error backpropagation and learning according to the Levenberg–Marquardt algorithm. The input data vector contains tremor and ground vibration parameters such as the maximum ground acceleration value, tremor energy, and epicentral distance (three input parameters) in the case of SNN1. In the case of SNN2, the input information is additionally extended by two local seismological coordinates, a dominating frequency of free-field vibration, and the parameter describing the direction of vibration. Thus, there are seven input parameters in SNN2. Similar information to that in the input parameters of shallow networks is given indirectly in the input parameters of the convolutional networks in the form of pre-processed ground vibration 1D records. Both the SNN1 structure and the SNN2 structure have one hidden layer consisting of 25 and 4 neurons, respectively. Similarly to the CNNs, a ratio ra was used as a network output in the case of both above-mentioned SNNs.

The test results presented in Table 8 and Fig. 27 indicate similar prediction capabilities of the CNN and SNN2 networks, especially in the range of relative error up to 25%. The better training properties of the applied CNN result from the greater capacity of the model and the number of training epochs. Definitely worse results were obtained by the SNN1 network with very poor information in the input vector.

Table 8. Comparison of MSE and r (for learning, validation and testing) for chosen CNN and SNNs.

Network	$MSE(L)$	$MSE(V)$	$MSE(T)$	r_L	r_V	r_T
V16_CWT_G_FN	0.0010	0.0254	0.0181	0.9983	0.6972	0.7920
SNN1 (3-25-1)	0.0194	0.0237	0.0220	0.6815	0.4284	0.4903
SNN2 (7-4-1)	0.0129	0.0120	0.0103	0.8016	0.7384	0.7928

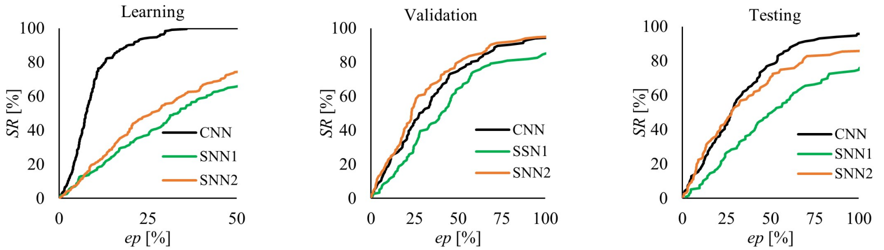

Fig. 27. SR in predicting the transmission of ground vibrations to a building for chosen CNN and SNNs.

6. Conclusion

The obtained results indicate the usefulness of CNNs in analyzing the prediction of transmission of vibrations from the ground to the building.

Thanks to the use of transfer learning and fine-tuning mechanisms, the user can easily apply the deep network architecture learned on millions of examples to a new, individual problem, even with a small set of available experimental data. In addition, networks of this type are easily transformed to solve regression problems, as exemplified by the above calculations.

Another advantage of using CNN in the SSI analysis is the ease of preparing input data, i.e., spectrograms of ground vibration accelerations, which already contain all information about the nature and parameters of the phenomenon. In the case of shallow networks, this input data can- not be so rich because of limitations in neural network architecture, and the input parameters are obtained in time-consuming preprocessing.

Of course, the disadvantage of using CNNs in relation to SNNs is the long calculation time and the need to have hardware with a powerful graphics card, especially for complex architectures and large batches of input data.

However, the obtained results show that using pre-trained DNN architecture, even in the case of a small data set, might be an alternative to SNNs.

The carried studies additionally confirm the complexity of the problem of dynamic soil-building interaction prediction.

Acknowledgments

We gratefully acknowledge Poland's high-performance computing infrastructure PLGrid (HPC Centers: ACK Cyfronet AGH) for providing computer facilities and support within the computational grant no. PLG/2022/015901.

References

1. Y. LeCun, Y. Bengio, G. Hinton, Deep learning, *Nature*, **521**: 436–444, 2015, doi: 10.1038/nature14539.

2. I. Goodfellow, Y. Bengio, A. Courville, *Deep Learning*, The MIT Press, Cambridge, Massachusetts, London, England, 2016.

3. A. Krizhevsky, I. Sutskever, G.E. Hinton, ImageNet classification with deep convolutional neural networks, [in:] F. Pereira, C.J. Burges, L. Bottou, K.Q. Weinberger [Eds.], *Advances in Neural Information Processing Systems*, Vol. 25, pp. 1097–1105, 2012.

4. K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, *arXiv*, 2014, arXiv:1409.1556.

5. K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, *arXiv*, 2015, arXiv:1512.03385.

6. S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: Towards real-time object detection with region proposal networks, *arXiv*, 2015, arXiv: 1506.01497.

7. I.J. Goodfellow *et al.*, Generative adversarial networks, *arXiv*, 2014, arXiv:1406.2661.

8. H. Wang, B. Raj, On the origin of deep learning, *arXiv*, 2017, arXiv:1702.07800.

9. D. Hepsiba, J. Justin, Computational intelligence for speech enhancement using deep neural network, *Computer Assisted Methods in Engineering and Science*, **29**(1–2): 71–85, 2022, doi: 10.24423/cames.397.

10. H. Mhaskar, Q. Liao, T. Poggio, When and why are deep networks better than shallow ones?, [in:] *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, Vol. 31, No. 1, pp. 2343–2349, AAAI Press, Palo Alto, California USA, 2017.

11. S.J. Pan, Q. Yang, A survey on transfer learning, *IEEE Transactions on Knowledge and Data Engineering*, **22**(10): 1345–1359, 2010, doi: 10.1109/TKDE.2009.191.

12. F. Yu, X. Xiu, Y. Li, A survey on deep transfer learning and beyond, *Mathematics*, **10**(19): 3619, 2022, doi: 10.3390/math10193619.

13. S. Feng, H. Zhou, H. Dong, Using deep neural network with small dataset to predict material defects, *Materials & Design*, **162**: 300–310, 2019, doi: 10.1016/j.matdes.2018.11.060.

14. F. Khosravikia, M. Mahsuli, M.A. Ghannad, The effect of soil–structure interaction on the seismic risk to buildings, *Bulletin of Earthquake Engineering*, **16**: 3653–3673, 2018, doi: 10.1007/s10518-018-0314-z.

15. J. Fu, J. Liang, M.I. Todorovska, M.D. Trifunac, Soil-structure system frequency and damping: Estimation from eigenvalues and results for a 2D model in layered half-space, *Earthquake Engineering and Structural Dynamics*, **47**: 2055–2075, 2018, doi: 10.1002/eqe.3055.

16. H. Güllü, M. Karabekmez, Effect of near-fault and far-fault earthquakes on a historical masonry mosque through 3D dynamic soil-structure interaction, *Engineering Structures*, **152**: 465–492, 2017, doi: 10.1016/j.engstruct.2017.09.031.

17. E. Ahmadi, Concurrent effects of inertial and kinematic soil-structure interactions on strength-ductility-period relationship, *Soil Dynamics and Earthquake Engineering*, **117**: 174–189, 2019, doi: 10.1016/j.soildyn.2018.10.043.

18. J.F. Semblat *et al.*, Seismic wave amplification: Basin geometry vs soil layering, *Soil Dynamics and Earthquake Engineering*, **25**(7–10): 529–538, 2005, doi: 10.1016/j.soildyn.2004.11.003.

19. S. Chopra, P. Choudhury, A study of response spectra for different geological conditions in Gujarat, India, *Soil Dynamics and Earthquake Engineering*, **31**: 1551–1564, 2011, doi: 10.1016/j.soildyn.2011.06.007.

20. O. Deck, M.A. Heib, F. Homand, Taking the soil–structure interaction into account in assessing the loading of a structure in a mining subsidence area, *Engineering Structures*, **25**(4): 435–448, 2003, doi: 10.1016/S0141-0296(02)00184-0.

21. E. Maciag, K. Kuzniar, T. Tatara, Response spectra of the ground motion and building foundation vibrations excited by rockbursts in the LGC region, *Earthquake Spectra*, **32**(3): 1769–1791, 2016, doi: 10.1193/020515EQS022M.

22. K. Kuzniar, T. Tatara, Full-scale long-term monitoring of mine-induced vibrations for soil-structure interaction research using dimensionless response spectra, *Case Studies in Construction Materials*, **16**: e00801, 2022, doi: 10.1016/j.cscm.2021.e00801.

23. T. Li, M.F. Cai, M. Cai, A review of mining-induced seismicity in China, *International Journal of Rock Mechanics and Mining Sciences*, **44**(8): 1149–1171, 2007, doi: 10.1016/j.ijrmms.2007.06.002.

24. A.V. Lovchikov, Review of the strongest rockbursts and mining-induced earthquakes in Russia, *Journal of Mining Science*, **49**: 572–575, 2013, doi: 10.1134/S1062739149040072.

25. M.E. Hoseny, J. Ma, W. Dawoud, D. Forcellini, The role of soil structure interaction (SSI) on seismic response of tall buildings with variable embedded depths by experimental and numerical approaches, *Soil Dynamics and Earthquake Engineering*, **164**: 107583, 2023, doi: 10.1016/j.soildyn.2022.107583.

26. Y.E. Ibrahim, M. Nabil, Finite element analysis of multistory structures subjected to train-induced vibrations considering soil-structure interaction, *Case Studies in Construction Materials*, **15**: e00592, 2021, doi: 10.1016/j.cscm.2021.e00592.

27. A. Mikami, J.P. Stewart, M. Kamiyama, Effects of time series analysis protocols on transfer functions calculated from earthquake accelerograms, *Soil Dynamics and Earthquake Engineering*, **28**(9): 695–706, 2008, doi: 10.1016/j.soildyn.2007.10.018.

28. S.F. Ghahari, F. Abazarsa, O. Avci, M. Çelebi, E. Taciroglu, Blind identification of the Millikan Library from earthquake data considering soil–structure interaction, *Structural Control and Health Monitoring*, **23**(4): 684–706, 2015, doi: 10.1002/stc.1803.

29. D. Pitilakis, M. Dietz, D.M. Wood, D. Clouteau, A. Modaressi, Numerical simulation of dynamic soil–structure interaction in shaking table testing, *Soil Dynamics and Earthquake Engineering*, **28**(6): 453–467, 2008, doi: 10.1016/j.soildyn.2007.07.011.

30. F. Xu *et al.*, Shaking table test on seismic response of a planar irregular structure with differential settlements of foundation, *Structures*, **46**: 988–999, 2022, doi: 10.1016/j.istruc.2022.10.090.

31. J.E. Luco, M.D. Trifunac, H.L. Wong, Isolation of soil-structure interaction effects by full-scale forced vibration tests, *Earthquake Engineering and Structural Dynamics*, **16**: 1–21, 1988.

32. F. Gara, M. Regni, D. Roia, S. Carbonari, F. Dezi, Evidence of coupled soil-structure interaction and site response in continuous viaducts from ambient vibration tests, *Soil Dynamics and Earthquake Engineering*, **120**: 408–422, 2019, doi: 10.1016/j.soildyn.2019.02.005.

33. K. Kuzniar, L. Chudyba, Evaluation of the influence of some mining tremors and ground vibrations parameters on vibrations transmission from the ground to building [in Polish], [in:] Aktualne problemy wpływów sejsmicznych i parasejsmicznych na budowle, Tom II: Badania wstrząsów górniczych i drgań komunikacyjnych, K. Stypula [Ed.], Vol. 477/2, Seria Inżynieria Lądowa, Wyd. Politechniki Krakowskiej, Krakow, pp. 23–37, 2015.

34. I. Nametevs, K. Sudars, A. Dobrajs, Interpretability versus explainability: Classification for understanding deep learning systems and models, *Computer Assisted Methods in Engineering and Science*, **29**(4): 297–356, 2022, doi: 10.24423/cames.518.

35. C.M. Bishop, *Pattern Recognition and Machine Learning*, Springer-Verlag, Berlin, 2006.

36. P.C. Jackson, *An Introduction to Artificial Intelligence*, Dover Publications, New York, 2006.

37. F. Rosenblatt, The perceptron: A probabilistic model for information storage and organization in the brain, *Psychological Review*, **65**(6): 386–408, 1958, doi: 10.1037/h0042519.

38. S.I. Gallant, Perceptron-based learning algorithms, *IEEE Transactions on Neural Networks*, **1**(2): 179–191, 1990, doi: 10.1109/72.80230.

39. A. Krizhevsky, *Learning Multiple Layers of Features from Tiny Images*, Technical Report TR-2009, University of Toronto, Toronto, Ontario, 2009.

40. O. Russakovsky *et al.*, ImageNet large scale visual recognition challenge, *arXiv*, 2014, arXiv:1409.0575.

41. C. Shorten, T.M. Khoshgoftaar, A survey on image data augmentation for deep learning, *Journal of Big Data*, **6**: 60, 2019, doi: 10.1186/s40537-019-0197-0.

42. Z. Zhou, J. Shin, L. Zhang, S. Gurudu, M. Gotway, J. Liang, Fine-tuning convolutional neural networks for biomedical image analysis: Actively and incrementally, [in:] *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4761–4772, Honolulu, HI, USA, 2017, doi: 10.1109/CVPR.2017.506.

43. K. Kuzniar, T. Tatara, The ratio of response spectra from seismic-type free-field and building foundation vibrations: the influence of rockburst parameters and simple models of kinematic soil-structure interaction, *Bulletin of Earthquake Engineering*, **18**: 907–924, 2020, doi: 10.1007/s10518-019-00734-w.

44. J.B. Allen, L.R. Rabiner, A unified approach to short-time Fourier analysis and synthesis, *Proceedings of the IEEE*, **65**(11): 1558–1564, 1977, doi: 10.1109/PROC.1977.10770.

45. S. Scholl, Fourier, Gabor, Morlet or Wigner: Comparison of time-frequency transforms, *arXiv*, 2021, arXiv:2101.06707.

46. J.P. Antoine, P. Carrette, R. Murenzi, B. Piette, Image analysis with two-dimensional continuous wavelet transform, *Signal Processing*, **31**(3): 241–272, 1993, doi: 10.1016/0165-1684(93)90085-O.

47. Transfer Learning for Deep Learning, MATLAB & Simulink, MathWorks, 2022, https://www.mathworks.com/discovery/transfer-learning.html.

48. P. Jiang, D. Ergu, F. Liu, Y. Cai, B. Ma, A review of Yolo algorithm developments, *Procedia Computer Science*, **199**: 1066–1073, 2022, doi: 10.1016/j.procs.2022.01.135.

Received February 1, 2023; revised version July 20, 2023;
accepted August 3, 2023; published online November 29, 2023.