UDC 004.8:624

CONVOLUTIONAL NEURAL NETWORKS FOR THE CRACK DIAGNOSTICS IN CONCRETE STRUCTURES

Danishevskyy Vladyslav [1], Dr. Sc., Prof., **Gaidar Anastasia** [2], PhD, Ass. Prof.
Prydniprovska State Academy of Civil Engineering and Architecture
[1] vladyslav.danishevskyy@pdaba.edu.ua; [2]nastuel_gaidar@pdaba.edu.ua

Problem statement. Millions of dollars are spent annually in the world on technical diagnostics of buildings and structures. Natural disasters such as floods and earthquakes, along with numerous negative man-made impacts lead to serious damage to building structures. The problem of diagnostics of the buildings and structures became extremely urgent after the aggression of the russian federation in Ukraine, which led to large-scale damage and destruction of industrial projects, housing stock and infrastructure projects such as roads, bridges, tunnels, etc. An important and urgent problem of Civil Engineering is to automate the processes of diagnostics of buildings and structures and to develop new methods for identifying building defects in building structures that would save human resources and reduce the dependence of survey results on subjective factors.

The aim of the study is to develop artificial neural networks for the identification and classification of cracks in vertical elements of building structures (e.g., concrete and reinforced concrete walls). The solution of this problem is complicated by the fact that cracks may be often visually similar to surface defects. The images of the studied structures can vary greatly depending on the texture of the surfaces, paint, light intensity, photography angle, etc. Cracks can be also irregular. These factors cause significant difficulties in training and testing ANN models [1–3].

Image dataset. We use SDNET 2018 digital photo collection [4] for training and testing the ANNs. The collection consists of 56 092 images of concrete structures with and without cracks, taken by a Nikon camera with a matrix resolution of 16 megapixels. To form the collection, 230 construction objects were used, which belong to three types: roads (104 objects), walls (72 objects) and bridges (54 objects). Images are scaled down to 256 by 256 pixels and have 3 colour channels with 256 luminance levels each (24-bit color).

Let us consider an ANN model that will diagnose cracks in concrete walls and classify them as vertical or horizontal. In practice, vertical cracks can usually indicate foundation settlement. Horizontal cracks may occur due to insufficient bearing capacity or overloading of the structure. The selected image dataset contains 1 086 photographs evenly divided into three groups: vertical cracks, horizontal cracks, and undamaged structures. Accordingly, each group consists of 352 photos.

Subsequently, the image dataset is divided into two groups. 80−85 % of images (training samples) are used to train the ANNs, that is, to establish optimal connections between neurons that allow the model to make correct decisions on data classification. The remaining 15−20 % of images (test samples) do not take part in the training process, but are used to test the model and check how effectively it can process data that it encounters for the first time.

Development of the ANN using Teachable Machine. Teachable Machine [5] is a free Google cloud tool that makes it easy to create machine learning models. Interaction with the tool is carried out through the web interface. Using the Teachable Machine, it is possible to train ANNs to recognize images, sounds and poses. Teachable Machine is based on a JavaScript machine learning library TensorFlow.js.

It should be noted that the developed model may not always work as expected. Training a neural network is a heuristic procedure, which effectiveness depends on a large number of parameters. In most cases, the optimal values of these parameters cannot be predicted in advance and must be determined experimentally for each individual task.

To develop the model, we use the image dataset described above. Teachable Machine allows one to modify the following training parameters:

1. Epochs. The number of epochs determines the duration of the training.
2. Batch Size. It determines the number of samples used in one training iteration.
3. Learning Rate. This coefficient controls the level of correction of the model parameters at each iteration.

In our study, the best results are obtained for Epochs = 50, Batch Size = 64, Learning Rate = 0.001. The accuracy of the model is displayed in Fig. 1. The loss function is given in Fig. 2. Here and in the sequel, blue and orange curves correspond to the training and to the test samples accordingly.

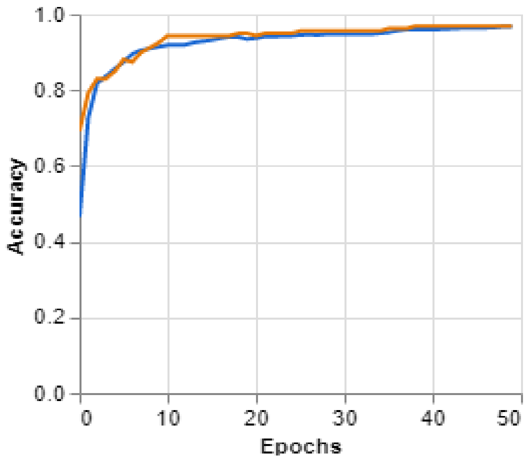

Fig. 1. Accuracy of the model

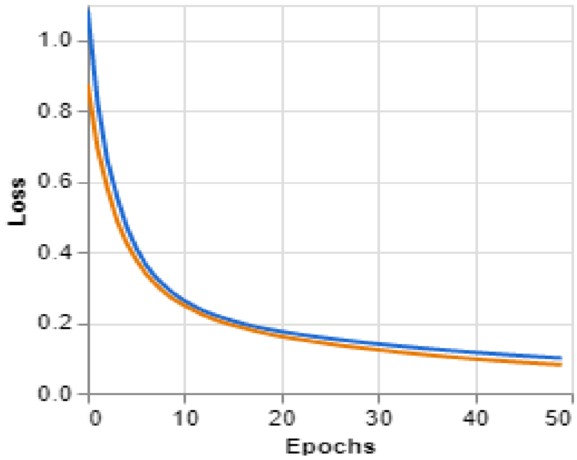

Fig. 2. Losses of the model

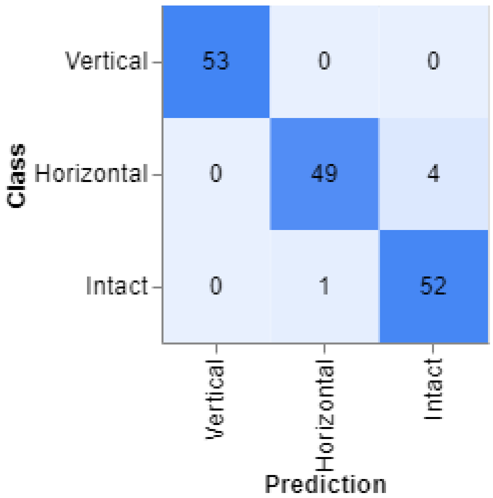

Fig. 3. Confusion matrix

The confusion matrix is presented at Fig. 3. It shows that the developed model recognizes correctly 53 vertical cracks, 49 horizontal cracks and 52 intact walls. However, there are 5 cases of misclassification: 4 horizontal cracks are interpreted as intact walls and 1 intact wall is interpreted as a horizontal crack.

The developed model is posted on the Internet and is available for a free use: https://teachablemachine.withgoogle.com/models/6RsGrbES2/.

Development of the ANN using TensorFlow. A significant limitation of the Teachable Machine is the inability to change the internal architecture of the ANNs, which is predefined automatically depending on the type of the project. In order to develop an ANN with its own architecture, we employ the open sources software library TensorFlow [6], developed by Google for machine learning and creating artificial intelligence models. We use Python and Colaboratory (Colab) [7] cloud development environment.

We conduct a series of computational experiments and determine the rational ANN architecture, which allows us to achieve a high accuracy and to minimize losses of the model.

The developed model is a multilayer convolutional ANN. The first layer is the input image layer with size 256×256×3, where 256×256 is the pixel size of the analyzed images, and 3 is the number of colour channels for one pixel. This layer also normalizes the input data, which brings all images to the same brightness level.

Next, there are 6 blocks located sequentially, each of which includes:

- 2D convolution layer tf.keras.layers.Conv2D, which detects the main features of the image;
- subsampling layer tf.keras.layers.MaxPooling2D, which reduces the size of the previous layer by compressing the image and discarding minor details;
- exclusion layer tf.keras.layers.Dropout, which randomly excludes a certain proportion (20 % in our project) of the neurons of the previous layer. This reduces the total number of weight coefficients and prevents the model from overtraining.

The tf.keras.layers.Flatten layer then reduces the dimensionality of the data by converting the 2D matrix input into a 1D vector output.

Next there are three conventional, fully connected neural layers tf.keras.layers.Dense, which classify the received data. The size of these layers decreases gradually. The last one has a size of 3, which equals to the number of classes that the model recognizes (vertical cracks, horizontal cracks, intact walls).

Thus, the model includes 23 layers (among them, 9 training layers) and 67 875 weight coefficients. For each layer, the nonlinear ReLU activation function is employed.

The model is trained for the following parameters: Epochs = 50, Batch Size = 64, Learning Rate = 0.001. The accuracy and the losses of the model are shown in Figs. 4, 5. It should be noted that the developed ANN classifies correctly all the examples that was misclassified by the Teachable Machine model. This allows us to conclude that the individual tuning of the ANN architecture for every specific task allows achieving significantly better results than the use of some general universal solutions.

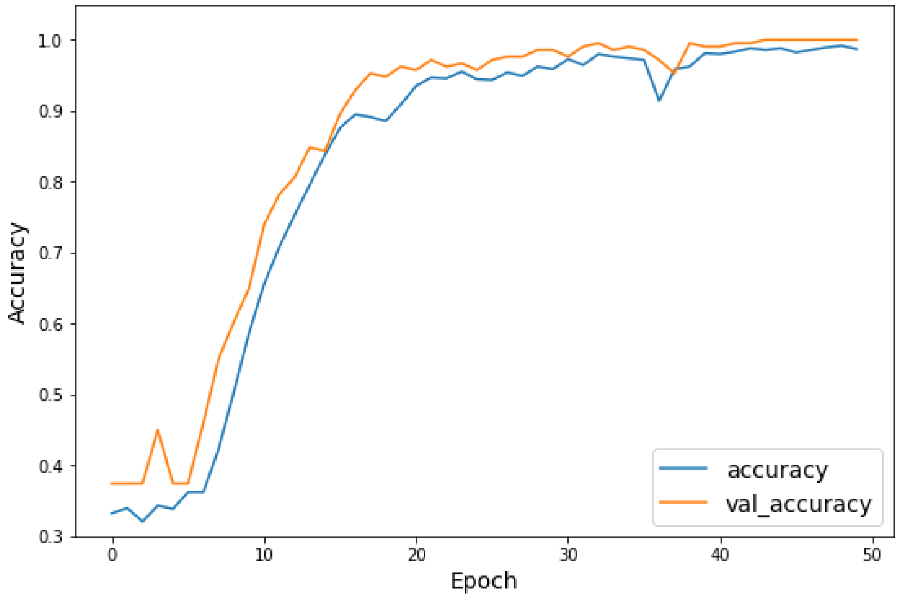

Fig. 4. Accuracy of the model

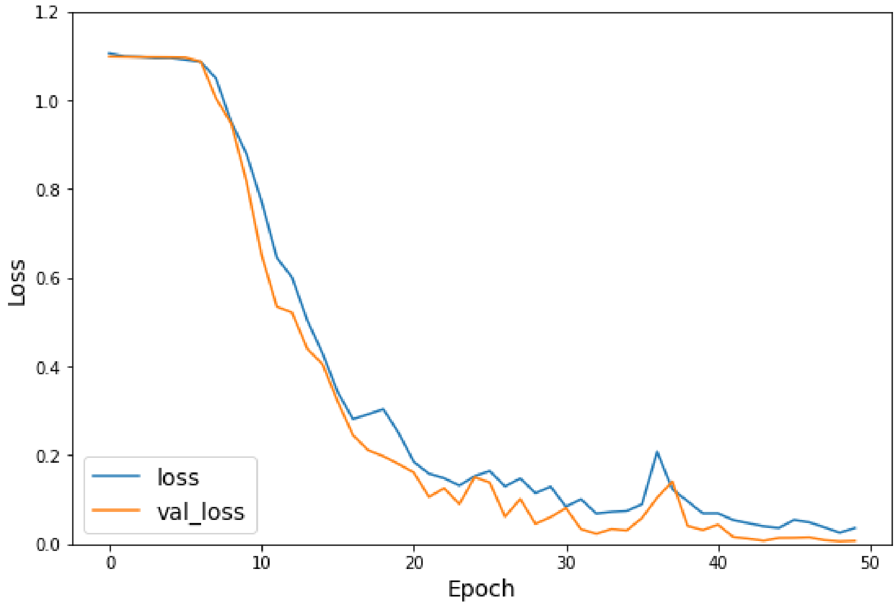

Fig. 5. Losses of the model

Conclusions. Two ANN models are developed for detecting and classifying cracks in concrete walls. We study how the magnitudes of the training parameters affect the accuracy of the models and determined the optimal values of these parameters. When developing the ANN using TensorFlow, a series of computational experiments is carried out and it is studied how the network architecture (number and types of layers, number of neurons, etc.) affects the performance of the model. As a result, the rational internal architecture of the ANN is determined ensuring the highest accuracy and the lowest losses.

The obtained results confirm a high efficiency of the methods of artificial intelligence for the diagnostics of buildings and structures. The proposed approaches can be further extended to diagnose and to classify a wide variety of building defects.

Acknowledgement. This work is supported by the European Commission under HORIZON 2020 project “A novel decentralized edge-enabled PREsCriptivE and ProacTive framework for increased energy efficiency and well-being in residential buildings”, grant agreement ID: 958284.

References

1. Perez H., TahJ. H.M., Mosavi A. Deep learning for detecting building defects using convolutional neural networks. *Sensors.* Vol. 19, 2019, id 3556 (22 pages).
2. Rajadurai R.S., Kang S-T. Automated vision-based crack detection on concrete surfacesusing deep learning. *Appl. Sci.* Vol. 11, 2021, id 5229 (14 pages).
3. Chun P-J., Yamane T., Maemura Yu. A deep learning-based image captioning method to automatically generate comprehensive explanations of bridge damage. *Comput. Aided Civ. Inf.* Vol. 37, 2022, pp. 1387–1401.
4. Maguire M., Dorafshan S., Thomas R.J. SDNET2018 : a concrete crack image dataset for machine learning applications. *Utah State University*. 2018. URL: https://doi.org/10.15142/T3TD19
5. Cloud tool Teachable Machine. URL: https://teachablemachine.withgoogle.com
6. Open-source software library TensorFlow. URL: https://www.tensorflow.org
7. Cloud development environment Colab. URL: https://colab.research.google.com