UDC 624.012.45:625:004.8

APPLICATION OF CONVOLUTIONAL NEURAL NETWORK IN STRESS CALCULATIONS OF REINFORCED CONCRETE SLABS OF ROAD PAVEMENTS

V. V. Molosh[1], A. E. Zheltkovich[2], K. G. Parchotz[3], I. G. Tomashev[4]

[1] Candidate of Technical Sciences, Associate Professor, Associate Professor of the Department of Theoretical and Applied Mechanics, Brest State Technical University, Brest, Belarus, e-mail: m.vic@rambler.ru
[2] Candidate of Technical Sciences, Associate Professor, Associate Professor of the Department of Theoretical and Applied Mechanics, Brest State Technical University, Brest, Belarus, e-mail: gelpek@mail.ru
[3] Programmer engineer, Belarus, e-mail: konstantinparhoc@gmail.com
[4] Senior Lecturer, Department of Theoretical and Applied Mechanics, Brest State Technical University, Brest, Belarus, e-mail: m.vic@rambler.ru

Abstract
The design of rigid reinforced concrete slabs of foundations, slabs, road surfaces is based on calculation models, which are developed on a relatively limited number of experimental studies, in most cases requiring quite large material and time costs. The complex stress-strain state occurring in stiff reinforced concrete base and pavement slabs under load, especially under cyclic dynamic loading, can often lead to cracking and failure of the slabs. In this paper, reinforced concrete slabs of a container yard pavement were investigated for load bearing from the wheels of a reach stacker (container loading vehicle) travelling on the surface. Existing models for the design of such slabs typically consider the slab loaded by a single local load applied to an edge or corner of the slab from the wheels of a moving vehicle. In fact, there may be two wheels on the slab, resulting in more unfavorable conditions. The application of the finite element method in such problems is quite laborious as it requires highly skilled design engineers and considerable time, making the design routine and of limited use. This paper investigates an alternative approach based on the application of an artificial convolutional neural network (CNN) with U-Net architecture, which provides a reasonably accurate prediction of stresses in the slab much faster and simpler compared to the finite element method. The paper presents the architecture of the neural network with an indication of the features and stages of its training. Statistical analysis of the calculation results is performed, which allowed us to assess the reliability of the neural network model for determining stresses in reinforced concrete slabs on an elastic base.

Keywords: reinforced concrete slab, computational model, convolutional neural network, U-Net architecture.

ПРИМЕНЕНИЕ СВЁРТОЧНОЙ НЕЙРОННОЙ СЕТИ В РАСЧЁТАХ НАПРЯЖЕНИЙ ЖЕЛЕЗОБЕТОННЫХ ПЛИТ ДОРОЖНЫХ ПОКРЫТИЙ

В. В. Молош, А. Е. Желткович, К. Г. Пархоц, И. Г. Томашев

Реферат
В основу проектирования жёстких железобетонных плит фундаментов, перекрытий, дорожных покрытий положены расчётные модели, которые разработаны на относительно ограниченном количестве экспериментальных исследований, в большинстве случаев требующих достаточно больших материальных и временных затрат. Сложное напряжённо-деформированное состояние, возникающее в жёстких железобетонных плитах фундаментов и дорожных покрытий под нагрузкой, и в особенности под циклической динамической нагрузкой, часто может приводить к образованию трещин и разрушению плит. В работе исследовались железобетонные плиты покрытия контейнерной площадки, которые воспринимают нагрузку от колёс перемещающегося по поверхности ричстакера (транспортного средства для погрузки контейнеров). Существующие модели для проектирования таких плит рассматривают как правило плиту, загруженную одной локальной нагрузкой, приложенной на краю или в углу плиты, от колеса передвигающегося транспорта. Фактически на плите могут располагаться два колеса, что приводит к более неблагоприятному состоянию. Применение метода конечных элементов в таких задачах является достаточно трудоёмким, так как требует высокого уровня квалификации инженеров-проектировщиков и значительных временных и трудовых затрат, что делает проектирование рутинным и мало целесообразным. В данной работе исследован альтернативный подход, основанный на применении искусственной свёрточной нейронной сети (CNN) с архитектурой U-Net, позволяющий получить достаточно точное предсказание напряжений в плите значительно быстрее и проще в сравнении с методом конечных элементов. В работе приведена архитектура нейронной сети с указанием особенностей и этапов её обучения. Выполнен статистический анализ результатов расчёта, позволивший оценить достоверность нейросетевой модели определения напряжений в железобетонных плитах на упругом основании.

Ключевые слова: железобетонная плита, расчётная модель, свёрточная нейронная сеть, архитектура U-Net.

1 Introduction

Reinforced concrete pavement slabs have a non-linear behavior under load with a complex stress-strain condition due to the inhomogeneous anisotropic structure of the composite material. In the design of such slabs, simplified design models are used, which are based on a number of assumptions and simplifications, and are most often developed based on the results of experimental tests of reduced slab fragments.

Traditionally, the design of reinforced concrete slabs has been based on mathematical models, finite element methods (FEM) and experimental tests. Keeping a balance between safety and economic feasibility, over the last decades, researchers and engineers have proposed many calculation models [2–9] for reinforced concrete slabs, based on which various design standards have been developed and are used worldwide [1–6].

Mathematical models of the resistance of rigid reinforced concrete slabs do not allow taking into account a large number of variables simultaneously due to the complexity and labor-intensive nature of this approach [1]. As a rule, such models take into account the behavior of each individual element of the structure, which in general for a structural system leads to the calculation of several equations, especially when the influence of more than one parameter on the resistance is taken into account, and complicates the complexity and duration of the calculation [11].

The laboriousness and considerable duration of analytically solutions using mathematical models or the finite element method in the design of structures and the experimental determination of the behavior of structural elements under load indicate, according to the authors [11], the need for reliable alternative prediction.

Due to the advances in computer science, many researchers have proposed to use soft computing methods to solve complex engineering problems in the last two decades [10–23]. The most popular of them are artificial neural networks, response surface methodology, fuzzy logic, particle swarm optimization and genetic algorithms [24].

In 1992, J. H. Garrett [25] reported that modelling with neural networks is much easier than with traditional mathematical models. Despite the fact that in neural networks the interconnections between its nodes (neurons) and minimization of the training error have a mathematical essence, mathematical formulas are not explicitly present in them. Artificial neural networks can be used to predict the strength of concrete and resistance of concrete structures with an error of less than 10 %.

It is also noted in [11] that neural networks can be used as an alternative to mathematical models or experimental tests at the initial design stage to obtain a quick prediction of the behavior of reinforced concrete slabs under load, determining the magnitude of resistance and deflections of resistance and deflections.

Neural networks are information processing systems whose architecture is based on the endeavor to replicate the structure of biological neural systems [26]. Unlike traditional computer programs, in which information is received and processed digitally in a sequential manner, neural networks store data in some way between individual neurons of the network by means of selected weighting coefficients. Neural networks do not contain any algorithms to process the data. They are 'trained' to find relationships, often not fully realized, that create a structure of causal interactions between input parameters and the result obtained.

Neural networks are able to model the behavior of systems with limited design costs and provide fast and reasonably accurate solutions in complex, uncertain and individual situations [27, 28]. Such prediction can be useful for a structural engineer in the preliminary design phase to determine the initial serviceability of a particular structure or to estimate the load carrying capacity of an in-service structure.

O. Moselhi in 2002 [28] highlighted the following characteristics of neural networks that make them useful for solving different types of engineering and scientific problems:

- neural networks are based on algorithms in which computational procedures are performed in parallel and decentralized rather than sequentially, as in conventional computer programs, resulting in fast data processing;
- they have a distributed memory represented by weight coefficients in the links distributed over all elements of the network;
- neural networks remain functional even after several network elements are damaged and fall out of network operation;
- they have the ability to learn from examples;
- allow predicting the behavior of systems with limited modelling capabilities;
- allow fast and reasonably accurate solutions to complex, uncertain and unusual situations.

Thus, it can be noted that the use of neural networks in engineering and scientific tasks allows to simplify and speed up the calculation procedure.

The main purpose of this work is:

- approbation of convolutional neural network in the problems of calculation of rigid reinforced concrete slabs of covering of container yards erected constructed on the ground base, determination of stresses in the design of such slabs caused by external influences from the loading vehicle (reach stacker) moving on the surface;
- to show the possibility of using 'soft computing' with the application of deep learning in tasks related to the design of building structures;
- to show the advantages of convolutional neural networks in comparison with other models in determining the stresses in reinforced concrete slabs on the base under different variants of concentrated loads from reach stacker wheels;
- evaluate the accuracy of stress values obtained using convolutional neural network.

2 Problem formulation and choice of neural network type

The design of reinforced concrete slab foundation on the soil base, for which the calculation was performed, consisted of two reinforced concrete slabs, of which the lower one modelled the reinforced concrete base slab of the container yard with a thickness of h = 100 mm, the upper one – the reinforced concrete cover slab of the container yard with a thickness of h = 250 mm.

In calculations for the base slab was taken concrete class $C^{12}/_{15}$ according to [29] with modulus of elasticity E = 19000 MPa, for the cover slab - concrete class $C^{32}/_{40}$ according to [29] with modulus of elasticity E = 38000 MPa. The Poisson's ratio for both slabs was assumed μ = 0.2.

The interaction of the upper and lower slabs with each other was modelled by means of elastic bonds of finite stiffness.

Calculation of the slabs for vertical loads from the wheels of a (reach stacker) travelling on the surface was initially performed by the LIRA PC.

The work of the elastic base was taken into account by means of the algorithm 'Soil Model' built into LIRA PC, which takes into account the elastic work of each layer of the soil base. Characteristics of soils in the layers were taken on the basis of engineering-geological surveys on the territory of the container site in the transshipment park of Brest-Northerly station in Brest.

The calculation was carried out for the action of constant load from the own weight of the base slabs and the container yard covering and short-term load on each front wheel from the reach stacker FERRARI F500-RS2. The area of load application from the reach stacker wheels, according to its technical specifications, was assumed to be A = 0,36 m^2. The value of the load from the reach stacker wheels was varied in the range from 150 to 900 kN to form a database for training the neural network.

The classification of neural networks developed to date is quite extensive. A perceptron with one hidden layer is a universal approximater, i.e. it is capable of approximating any continuous function with any degree of accuracy if a continuous, monotonically increasing, bounded function is used as the activation function of neural elements of the hidden layer [30]. Multilayer perceptron can be used for pattern classification, prediction and control tasks. Recurrent networks can be used for processing dynamic data, temporal patterns, solving prediction problems, system identification, speech recognition, natural language processing and control. Convolutional neural networks (CNN), which are a further development of multilayer perceptron, are widely used for image processing, and, unlike multilayer perceptron, they allow to take into account image topology and retain predictive properties in case of shifts, scaling and other distortions of the input image. Many other types of neural networks are also known.

Since the distribution of stresses, strains deformations or vertical displacements on the slab surface has similarity with the image, the authors of the paper decided to use convolutional neural network (CNN) to achieve the goal.

3 Data for neural network training

The dataset used in this study to train the CNN was obtained through parametric modelling in LIRA PC. A total of 125 numerical simulations (images) were performed with the inherent ability to vary at different levels three parameters: 1) the magnitude of the load from the wheels travelling on the surface of the container yard slab of the reach stacker with the container; 2) the location of the two front most loaded wheels on the surface of the slab; 3) the shape of the container yard cover slab in plan. The 125 numerical calculations were divided into two groups: 100 calculations were designed to train the CNN to predict the magnitude of stresses in the slab distributed over its surface; 25 calculations were designed to evaluate the accuracy of the developed CNN model.

The initial data intended for the formation of CNN feature maps characterizing the recognized image were generated using two different methods.

In the first case (model 1), the raw data were fed as four digital feature maps of dimensionality 56×56.

The first feature map, the shape feature, is designed to describe the shape of the slab by means of zeros and units. In this map, the body of the slab is described by units and the empty space by zeros.

The second feature map is the load map. The area to which the wheel load is applied is marked in the map with elements indicating the magnitude of the load applied to the corresponding area. The non-loaded area of the slab in the map is marked with zero elements.

The third and fourth feature maps describe the location of the reach stacker wheel load on the slab surface. In each of these maps, the center of the reach stacker wheel load area is described by a zero, and all other elements of the map represent the distance from this center to the corresponding point on the slab surface. The third map describes the load position of the first wheel and the fourth map describes the load position of the second wheel.

In the second case (model 2), the input data were fed as a single digital feature map of dimension 56 × 56, which displayed the coordinates of the slab points in 0.1 m increments and the magnitude of the vertical concentrated load on the slab at each coordinate.

The output was to obtain the stresses in the slab distributed over its surface. The stresses obtained for CNN training in LIRA PC were formed into a stress map, in which each element represented the stresses in the corresponding point of the reinforced concrete slab.

4 Neural network architecture and algorithm

A convolutional neural network (CNN) with U-Net architecture was used to predict the stress distribution on the surface of a reinforced concrete slab. U-Net is considered one of the standard CNN architectures for image segmentation tasks, when it is necessary not only to define the whole image class, but also to segment its regions by class, to create a mask that will divide the image into several classes. [31]. There is a perception that a very large number of annotated training samples are required to successfully train deep neural networks. However, the study [31] presents a CNN with U-Net architecture that relies on the active use of additional data to make better use of the available annotated samples. The authors of [31] show that the network they developed can be trained on a very small number of images and outperforms the previous best method for a number of neuron segmentation tasks on electron microscopic tubes.

The architecture of the convolutional neural network used by the authors of this paper is shown in Figure 1. It consists of two parts – encoder («convolutional») and decoder («unfolding deconvolution»). Encoder converts the input image into a multidimensional feature representation. It performs the feature extraction function. Decoder creates a segmented image based on the features extracted from the convolutional part of the network.

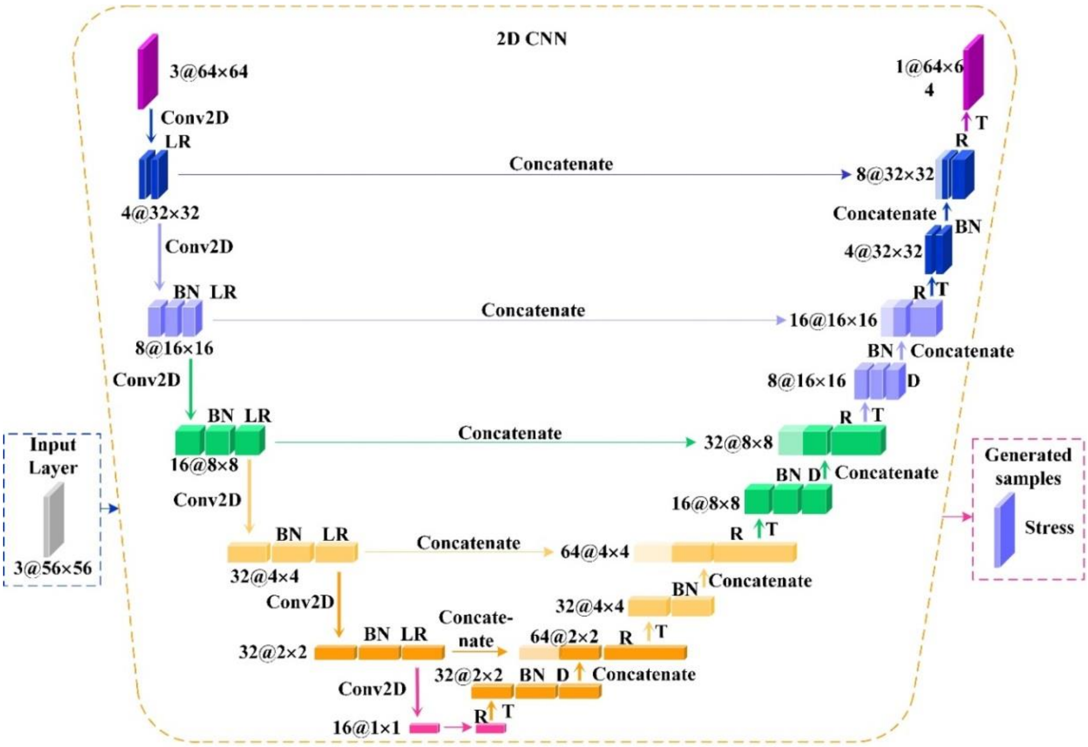

Figure 1 – Schematic of the applied CNN model with U-Net architecture with a single pixel at the lowest resolution. Each colored parallelepiped corresponds to a multi-channel feature map. The number of channels is indicated by the first digit (before @) in the map parameter signature. The map dimensions are indicated behind the @ sign in the map parameter caption. Transparent fields represent copied feature maps for the concatenate operation. The following parameters are labelled with letters: Conv 2D – Convolution 2D – 2D convolution; BN – Batch Normalization – Batch Normalization of data; LR – Leaky ReLU – activation function; R – ReLU – activation function; T – Conv2DTranspose – 2D transposed convolution layer; D – Dropout – regularization method designed to reduce network overfitting

The input was an image of an externally loaded slab with dimensions of 5,6 × 5,6 m, which were converted into three feature maps of dimensions 64 × 64. The feature map responsible for the shape of the slab was not considered at this stage of research as it was unchanged. The convolution represents a digital filter in which training was performed using the sliding window method [30] by means of weighted summation of values in the map cells (neurons) and weighting coefficients – coefficients of the convolution kernel. A sliding window is otherwise referred to as a local receptive field or filter kernel for the corresponding (usually one) neuron in the feature map (each receptive field in the input image space is mapped to a different neuron in each feature map). The total number of different synaptic connections in the convolutional layer is:

$$V(C_1) = M\left(p^2+1\right), \tag{1}$$

where C_1 is the convolutional layer designation and its number, p^2 is the total number of elements of the receptive field (kernels).

From expression (1) follows the peculiarity that the use of convolutional network reduces the total number of tunable customizable synaptic connections of a convolutional network in comparison with multilayer perceptron due to the use of identical neurons in each feature map [30].

The result of ‘sliding’ the kernel, in this paper sized 4 × 4 in steps of 2, across the entire image is written into a new image (a new feature map). At each layer, the coding block collapses the three-dimensional matrix, reducing the number of sampling points of the map by half and increasing the number of features (channels) responsible for the characteristic features (stress magnitude) of individual nodes of the network. To preserve the dimensions of the feature map output and capture extreme values, we added rows and columns to the right and left, as well as top and bottom, filled with zeros in the feature map (padding procedure).

If we represent the pixels of the input image in one-dimensional space, then the output value of the ij-th neuron for the k-th feature map in the convolutional layer is defined as [30]

https://doi.org/10.36773/1818-1112-2024-135-3-24-30

$$y_{ij}^k = F\left(S_{ij}^k\right), \tag{2}$$

$$S_{ij}^k = \sum_c \left| w_{cij}^k x_c - T_{ij}^k \right|, \tag{3}$$

where $c = \overline{1, p^2}$; F is the activation function; S_{ij}^k – weighted sum of the *ij*-th neuron in the *k*- th feature map; w_{cij}^k – weighting factor between the *c*-th neuron of the input layer and the *ij*-th neuron in the *k*-th feature map; T_{ij}^k – threshold value of the *ij*-th neuron in the *k*-th feature map.

Each feature map obtained by convolution reflects the same local features in all parts of the image. It represents a set of neurons, each of which has the higher value, the more the associated image fragment resembles a kernels.

At each stage of convolution, we performed batch normalization of the obtained data (Batch Normalization – BN in Figure 1), which allows to improve performance and stabilize the network, and rectification by a linear activation block Leaky ReLU (LR in Figure 1).

The second part of the network, «decoder» is a mirror image of the first. The image size needs to be restored to the original image size. To this end, up-sampling layers are used in combination with convolutional layers. Each layer in up-sampling represents the process of inverse convolution of the feature map, accompanied by doubling of its size and halving of the number of feature channels followed by batch normalization. The dropout layer following the batch normalization turns off at random (temporarily excludes from training) a certain percentage of neurons in the network at each training step, which helps to prevent overdependence of the model on specific paths and nodes in the network leading to overtraining [33]. The probability p with which each neuron will be excluded is typically between 0.2 and 0.5. The feature map obtained in this layer is concatenated with the corresponding trimmed feature map from the convolution layer and straightened by the linear activation unit ReLU. The last layer uses convolution 1 × 1 to map each 64-component feature vector to the correct number of classes.

5 Neural network training

It should be noted that in this study the number of trained parameters, which was 98673 in model 1 and 98545 in model 2, exceeds the number of images on which the model was trained, which, according to many researchers, is a drawback of the model.

When training in convolutional neural networks, the whole image or local regions (patches) around an image pixel can be provided as input data. The authors of [31] note that when optimizing computations by stochastic gradient descent, the method of convolution over the whole image is identical to training over local areas (patch). The authors of [31] did not find that patch training provides faster or better convergence for dense prediction, while whole-image training is, in their opinion, quite efficient and effective.

In [34], the authors are based on an elegant architecture, the so-called fully convolutional network proposed in [31]. They modify and extend this architecture so that it works with a small number of training images, by dividing images into local regions (patches), and performs more accurate image segmentation. The main idea of [31] is to supplement the usual convolutional network with sequential layers, in which the pooling operators are replaced by up-sampling operators (up-sampling or sampling operator). Thus, according to the authors, these layers increase the resolution of the output signal. For localization, the high-resolution features obtained during narrowing are combined with up-sampled output data (concatenate). As a result, the subsequent convolution layer receives more accurate input data.

An important change in the neural network architecture presented in [34], the authors note the presence of a large number of feature channels in the up-sampling procedure, which allow the network to transfer contextual information to layers with higher resolution. As a consequence, the expanding path (decoder) is more or less symmetric to the narrowing path (encoder) and yields a U-shaped architecture. The network does not have fully connected layers and uses only the valid part of each convolution, i.e., the segmentation map contains only those pixels for which full context is available in the input image. This strategy, according to the authors of [34], allows seamless segmentation of arbitrarily large images. To predict pixels in the boundary region of the image, the missing context is extrapolated by mirroring the input image. This procedure is important for applying the network to large images, since otherwise the resolution would be limited by the CPU memory.

The authors [34] emphasize that their U-Net CNN can recognize local area, has a much larger amount of training data in the form of local areas (patch) than the number of training images and has a high predictive ability.

Thus, it can be noted that the existing opinion that the training sample size should be equal to the size of the trained parameters in order to prevent overtraining of the network is currently debatable. The architecture of modern neural networks can be configured in such a way as to allow training on a small training sample size and achieve a sufficiently high predictive ability.

In this work, the authors used a two-dimensional convolutional neural network (CNN) with U-Net architecture to predict the stress distribution on the surface of a reinforced concrete slab using the stochastic gradient descent (SGD) method for training.

The training of the neural network, as mentioned above, is carried out using the Batch Normalization method by the error back propagation method according to the provisions given in [32]. In the batch normalization method, some layers of the neural network are fed with preprocessed data having zero mathematical expectation and unit variance.

The rectification function was used as neuron activation functions, by which neuron output values can be calculated as:

$$y_{ij}^k = F(S_{ij}^k) = \begin{cases} S_{ij}^k, S_{ij}^k > 0, \\ kS_{ij}^k, S_{ij}^k \le 0, \end{cases} \tag{4}$$

where S_{ij}^k – is the same as in formula (3); in convolution layers the coefficient k = 0.02 was taken, in unwrapped (deconvolution) layers – k = 0.

In both models (model 1 and model 2), 300 training epochs were assigned. To train the neural network, 100 samples were used with the ratio between training and validation samples being 97 % to 3 %. To test the prediction accuracy of the models, an additional test sample of 25 samples not used in the training of the network was used.

When testing the neural network, the mean absolute error with L1 norm [35, 36] was used, as this metric reflects well the accuracy of the prediction result. The loss function was defined as:

$$E = \frac{1}{n}\sum \left| Y_{target} - Y_{predicted} \right|, \tag{5}$$

where n – number of training images, Y_{target} – training data, $Y_{predicted}$ – predicted data.

All stages of CNN creation, training and validation were implemented using the Python programming language and the open-source machine learning software library developed by Google to solve the tasks of building and training the Tensorflow neural network [37].

6 Results of calculations and their analysis

As a result of this study, it was found that the input data layout in model 1 was better than in model 2. This is evidenced by the speed and quality of training of the neural network, as well as by the numerical statistical evaluation of the reliability of the coincidence between the training values obtained by finite element calculation and the stress values predicted by the neural network. A rather close coincidence of training and validation sampling errors in Model 1 is observed after 120 epochs of training and is maintained until the end of training (Figure 2a). The divergence of training and validation sampling errors is about 17 %. In model 2, the training and validation sampling errors are quite different. While the training sampling error decreases to 15 % during the training process, the validation sampling error fluctuates within 50 % almost during the whole training process (Figure 2b). In addition, the curve of variation of the training sampling error obtained by model 1 is smoother than by model 2, which indicates a greater stability of neural network training in the first case. The validation sampling error variation curve obtained by model 1 is also smoother.

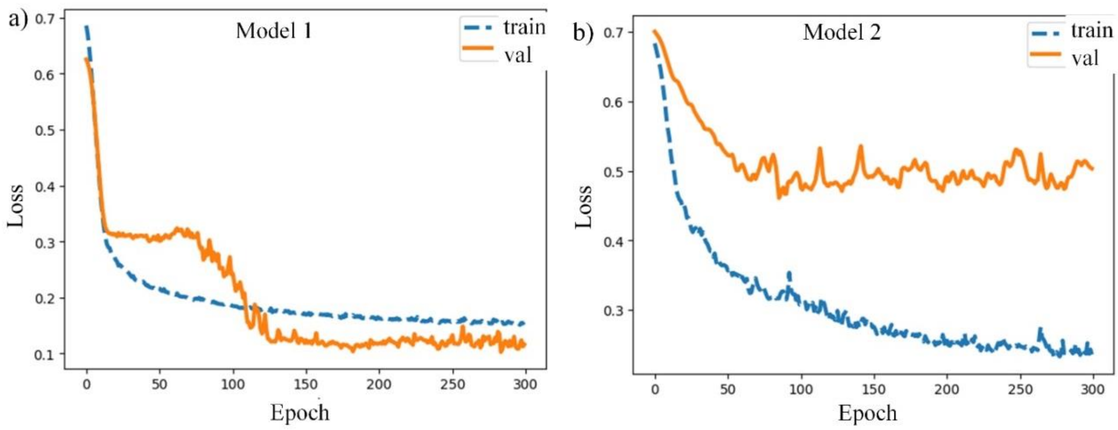

Figure 2 – Parameters of neural network training according to model 1 (a) and model 2 (b)

To evaluate the accuracy of stresses in reinforced concrete slabs predicted using neural network, we used well known mathematical statistics: mean absolute error (MAE), standard deviation (RMSE), Pearson correlation coefficient (*r*), coefficient of determination (R^2). In addition, we determined the value of correction factor *b* for the mean deviation of training and predicted stress values, the mean error value of models – Δ, obtained from the error vector δ, and the coefficient of variation V_δ (of the error vector δ), calculated according to the procedure given in Appendix D of CH 0.01.01 [38]. The specified statistical parameters are given in Table 1.

Table 1 – Statistical parameters characterizing the degree of accuracy of the developed neural network models

Model	*RMSE*	*MAE*	*r*	R^2	*b*	Δ	V_δ
Model 1	0,478	0,305	0,924	0,854	0,928	0,428	0,015
Model 2	0,637	0,424	0,862	0,733	0,853	0,658	0,146

The overall distribution of training and model 1 and model 2 predicted stresses in reinforced concrete slabs is shown in Figures 3 and 4, and the ratio of training and model 1 and model 2 predicted stresses in reinforced concrete slabs is shown in Figure 5.

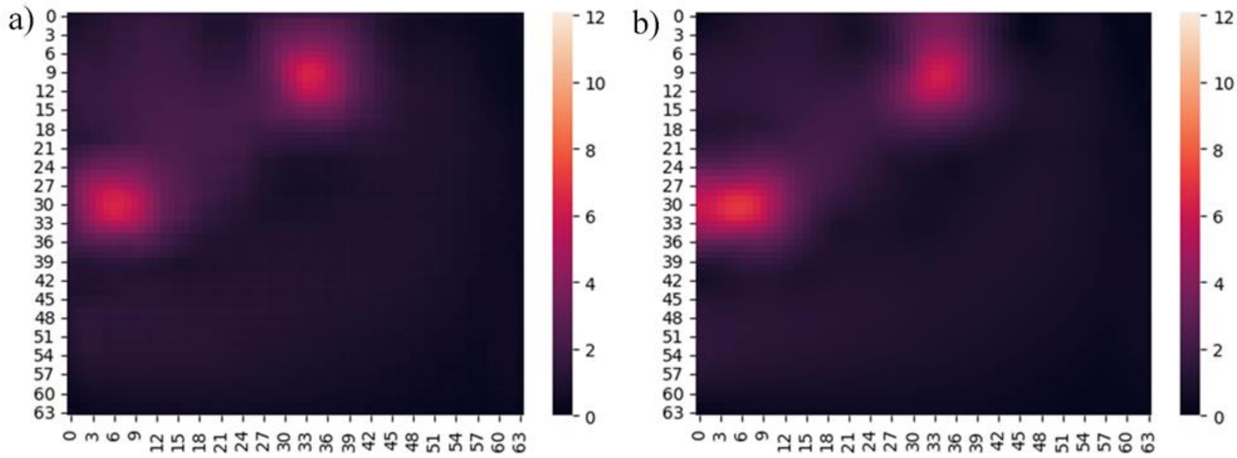

Figure 3 – Distribution of stresses predicted by model 1 (a) and training (b) in reinforced concrete slabs

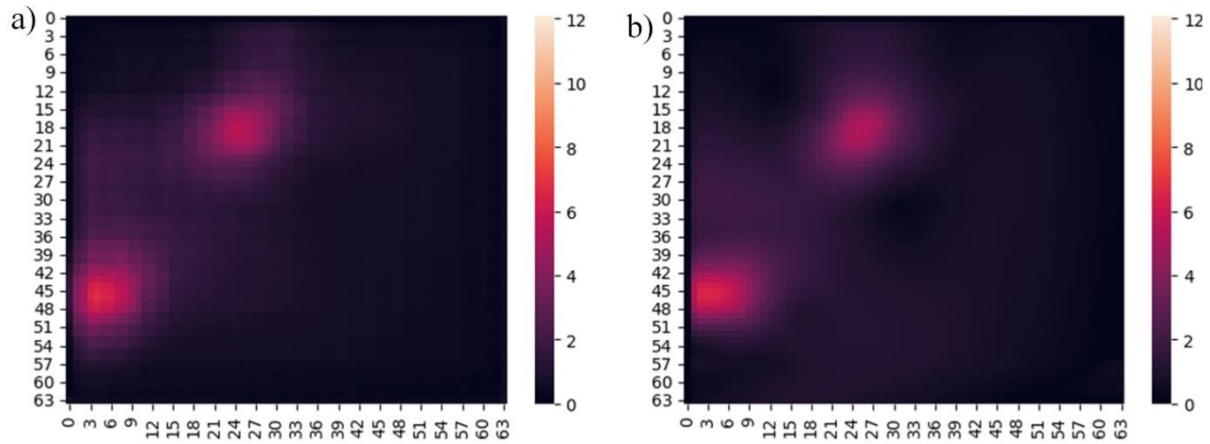

Figure 4 – Distribution of stresses predicted by model 2 (a) and training (b) in reinforced concrete slabs

https://doi.org/10.36773/1818-1112-2024-135-3-24-30

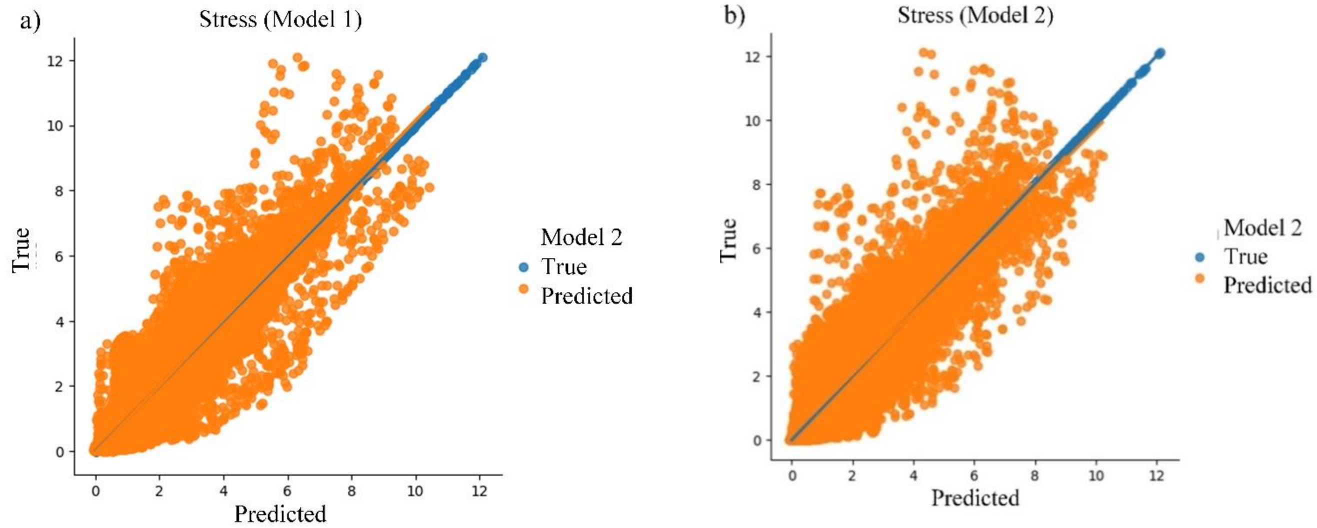

Figure 5 – Ratio of training and predicted stresses in reinforced concrete slabs using model 1 (a) and model 2 (b)

The predicted stresses in reinforced concrete slabs using the neural network have a relatively satisfactory coincidence with the training values in the whole range of values. On average, the predicted values of stresses exceed the training values by 7 % for model 1 and by 15 % for model 2, as evidenced by the value of the correction factor *b* (Figure 5, Table 1) for the mean deviation, determined by the expression:

$$b = \frac{\sum \sigma_a \sigma_p}{\sum \sigma_p^2}, \tag{5}$$

where σ_a, σ_p – are respectively the training (actual) and predicted stresses in the reinforced concrete slab [38].

Taking into account the small number of the training sample, this characterizes the predictive ability of the models as relatively high, but insufficiently safe. It is also impossible to speak about high density of distribution of predicted stress values along the line of training values. Most of the predicted values, which is about 85–90 %, deviate from it within 25 %. The maximum deviation, more typical for small stress values, is about 80 % (Figure 5).

The relatively low density of stress distribution is evidenced by the value of the mean absolute error, which for model 1 – *MAE* = 0,305, and for model 2 – *MAE* = 0,424. Relative to the mean value of stresses in reinforced concrete slabs, which is 0,938 MPa for the test sample, the mean absolute error (*MAE*) is 33 % and 45 %, respectively. The standard deviation for models 1 and 2 respectively are: *RMSE* = 0,478 and *RMSE* = 0,637.

It is accepted that computational models with a coefficient of determination above 0,8 and a correlation coefficient above 0,9 are considered good enough. When the coefficient of determination is equal to 1, there is a functional dependence between the compared values. In our study, when comparing training and predicted stresses, the values of correlation coefficient (for models 1 and 2 respectively: r = 0,924; 0,862) and coefficient of determination (for models 1 and 2 respectively: R^2 = 0,854; 0,733) meet the above criteria only for model 1 (Table 1).

The coefficient of variation of the error vector δ, equals to 0,015 and 0,146 for models 1 and 2, respectively, is less than the value of 0,33, which indicates a sufficiently high homogeneity of the studied data set [39].

Thus, it follows from the results of the statistical study that model 1 more accurately predicts the magnitude of stresses in the slab. The reason for this may lie in the input data for training, which in the first model are fed in the form of feature maps and form a common multilayer image of the object. Each feature map conveys the parameters of one layer of this image. All maps are united into a single image by the common geometric shape of the object. In the second model there is only one feature map, and the general geometry of the image is viewed only indirectly through the space coordinates, which are endowed with the necessary features.

7 Conclusions

Mathematical models of resistance of rigid reinforced concrete slabs do not allow to take into account a large number of variables simultaneously due to the complexity and labor-intensive nature of this approach. As a rule, such models take into account the behavior of each individual element of the structure, which in general for the structural system leads to the calculation of several equations, especially when taking into account the influence of more than one parameter on the resistance. This feature complicates the complexity and duration of the calculation and necessitates reliable alternative predictions.

As reported by many research papers, modelling the behavior of engineering structures using neural networks is much easier than using traditional mathematical models.

Neural networks can be used as an alternative to mathematical models or experimental tests at the initial design stage to obtain rapid prediction of the behavior of reinforced concrete slabs under load, determining the magnitude of resistance and deflections.

Neural networks are able to model the behavior of systems at limited design costs and provide fast and reasonably accurate solutions in complex, uncertain and individual situations.

Analyzing the results of the statistical study shows that model 1 is more accurate in predicting the magnitude of stresses in the slab due to a more efficient feedforward.

In general, despite the fact that most of the statistical parameters do not have the best values, the predictive ability of the models based on convolutional neural network with u-net architecture can be considered high enough.

The main reason for the error of the models is the small sample dataset size of training data, which requires replenishment of the sample, retraining of the neural network and subsequent assessment of its reliability.

References

1. Tully, S. H. A neural network approach for predicting the structural behavior of concrete slabs : a thesis submitted for the degree of master of engineering : 06.1997 / S. H. Tully. – Newfoundlan, 1997. – 126 p.
2. Betonnye i zhelezobetonnye konstrukcii = Betonnyya i zhalezabetonnyya kanstrukcyi : SP 5.03.01-2020. – Vved. 16.09.2020. – Minsk : RUP «Strojtekhnorm» : Ministerstvo arhitektury i stroitel'stva Respubliki Belarus', 2020. – 236 s.
3. Design of concrete structures – Part 1-1: General rules, rules for buildings, bridges and civil engineering structures : prEN 1992-1-1:2018 Eurocode 2. – Final draft of April 2018 by the Project Team SC2.T1 working on Phase 1 of the CEN/TC 250 work programme under Mandate M/515. – 293 p.
4. fib Model Code for Concrete Structures 2010. – Lausanne : International Federation for Structural Concrete (fib), Switzerland, 2013. – 432 p.

5. ACI. Building Code Requirements for Structural Concrete (ACI 318-08) and Commentary. In Aci-318-08; American Concrete Institute: Farmington Hills, MI, USA, 2008. – P. 1–471.
6. JSCE. Standard Specifications for Concrete Structures-2007. In JSCE Guideline for Concrete No. 15; JSCE: Tokyo, Japan, 2007. – P. 1–503.
7. CSA. Design of Concrete Structures. In A23.3-04; CSA: Mississauga, ON, Canada, 2004. – P. 1–232.
8. NZS. Concrete Structures Standard Part-1. In NZS 3101; New Zealand Standard. – Wellington, New Zealand, 2006. – Part 1. – P. 1–309.
9. KBCS. Korean Building Code-Structural (KBCS); Architectural Institute of Korea: Kimoondang, Korea, 2005.
10. Ahmad, A. Neural Network-Based Prediction: The Case of Reinforced Concrete Members under Simple and Complex Loading / A. Ahmad, N. D. Lagaros, D. M. Cotsovos // Applied Sciences. – 2021. – Vol. 11. – P. 4975. – DOI: 10.3390/app11114975.
11. Bamiyo, S. P. Prediction of Load deflection Behaviour of two way Rc Slab using Neural Network Approach / S. P. Bamiyo, O. A. Uche, M. Adamu // International journal of optimization in civil engineering. – 2017. – Vol. 7 (4). – P. 633–644.
12. Papadrakakis, M. Reliability-based structural optimization using neural networks and Monte Carlo simulation / M. Papadrakakis, N. D. Lagaros // Computer Methods in Applied Mechanics and Engineering. – 2002. – Vol. 191. – P. 3491–3507. – DOI: 10.1016/S0045-7825(02)00287-6.
13. Lagaros, N. D. Learning improvement of neural networks used in structural optimization / N. D. Lagaros, M. Papadrakakis // Advances in Engineering Software. – 2004. – Vol. 35, Iss. 1. – P. 9–25. – DOI: 10.1016/S0965-9978(03)00112-1.
14. Lagaros, N. D. An adaptive neural network strategy for improving the computational performance of evolutionary structural optimization / N. D. Lagaros, D. C. Charmpis, M. Papadrakakis // Computer Methods in Applied Mechanics and Engineering. – 2005. – Vol. 194, Iss. 33-33. – P. 3374–3393. – DOI: 10.1016/j.cma.2004.12.023.
15. Lagaros, N. D. Multi-objective design optimization using cascade evolutionary computations / N. D. Lagaros, V. Plevris, M. Papadrakakis // Computer Methods in Applied Mechanics and Engineering. – 2005. – Vol. 194. – P. 3496–3515.
16. Lagaros, N. D. Fragility assessment of steel frames using neural networks / N. D. Lagaros, M. Fragiadakis // Earthq. Spectra. – 2007. – Vol. 23. – P. 735–752. – DOI: 10.1193/1.2798241.
17. Ahmad, A. Prediction of Properties of FRP-Confined Concrete Cylinders Based on Artificial Neural Networks / A. Ahmad, V. Plevris, Q.-U.-Z. Khan // Crystals. – 2020. – Vol. 10. – 811 p. – DOI: 10.3390/cryst10090811.
18. Evaluating the Impacts of Pumping on Aquifer Depletion in Arid Regions Using MODFLOW, ANFIS and ANN / M. R. Almuhaylan, A. R. Ghumman, I. S. Al-Salamah [et al.] // Water. – 2020. – Vol. 12. – P. 2297. – DOI: 10.3390/w12082297.
19. Relating groundwater levels with meteorological parameters using ANN technique / M. Iqbal, U. A. Naeem, A. Ahmad [et al.] // Measurement. – 2020. – Vol. 166. – DOI: 10.1016/j.measurement.2020.108163.
20. Raza, A. Prediction of axial compressive strength for FRP-confined concrete compression members / A. Raza, Q. U. Z. Khan, A. Ahmad // KSCE Journal of Civil Engineering. – 2020. – Vol. 24. – P. 2099–2109. – DOI: 10.1007/s12205-020-1682-x.
21. The Role of Damage Extent in the Estimation of Direct Economic Losses of Existing RC Buildings / V. Manfredi, A. Masi, G. Nicodemo [et al.]. – Pisa, Italy : Pisa University Press, 2019. – P. 36–45.
22. Development of Integrated Techniques to Improve Seismic Performance and Energy Efficiency of Buildings: Preliminary Results of the Experimental Campaign on RC Portal Frames / D. Nigro, A. Digrisolo, G. Ventura [et al.]. – Pisa, Italy : Pisa University Press, 2019. – P. 128–137.
23. Structural health monitoring based on the hybrid ant colony algorithm by using Hooke–Jeeves pattern search / A. Shakya, M. Mishra, D. Maity, G. Santarsiero // SN Applied Sciences. – 2019. – Vol. 1. – P. 1–14. – DOI: 10.1007/s42452-019-0808-6.
24. Volna, E. Introduction to Soft Computing / E. Volna // 1st ed. ; Department of Computer Science. – Ostrava, Czechia : University of Ostrava, 2013. – P. 137.
25. Garrett, J. H. Neural Networks and their Applicability within Civil Engineering / J. H. Garrett // Proceedings, 8th Congress of Computing in Civil Engineering. ASCE. – New York, 1992.
26. Caudill, M. Naturally Intelligent Systems / M. Caudill, C. Butler. – Cambridge, Ma : MIT Press, 1990. – 314 p.
27. Moselhi, O. Neural networks for cost estimating of structural steel buildings / O. Moselhi, I. Siqueira // AACE Int. Transact. – 1998. – IT22.
28. Moselhi, O. Potential applications of neural networks in construction / O. Moselhi, T. Hegazy, P. Fazio // Canadian Journal of Civil Engineering. – 2002. – Vol. 19(3). – P. 521-529. – DOI: 10.1139/l92-061.
29. Betonnye i zhelezobetonnye konstrukcii = Betonnyya i zhalezabetonnyya kanstrukcyi : SP 5.03.01-2020. – Vved. 16.09.2020. – Minsk : RUP «Strojtekhnorm» : Min. arh. i str. Respubliki Belarus', 2020. – 244 s.
30. Golovko, V. A. Nejrosetevye tekhnologii obrabotki dannyh / V. A. Golovko, V. V. Krasnoproshin. – Minsk : Belorusskij gosudarstvennyj universitet, 2017. – 264 s.
31. Ronneberger, O. U-Net: Convolutional Networks for Biomedical Image Segmentation / O. Ronneberger, P. Fischer, T. Brox // University of Freiburg, Germany. – 2015. – arXiv: 1505.04597v1 [cs.CV]. – P. 8.
32. Ioffe, S. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift / S. Ioffe, Ch. Szegedy // International Conference on Machine Learning, Lille, France. – 2015. – Vol. 37. – DOI: 10.48550/arXiv.1502.03167.
33. Dropout i Batch normalization. Blog kompanii M. Video-El'dorado. – URL: https://habr.com/ru/companies/mvideo/articles/782360 (data obrashcheniya: 22.05.2024).
34. Long, J. Fully Convolutional Networks for Semantic Segmentation / J. Long, E. Shelhamer, T. Darrell // IEEE Conference on Computer Vision and Pattern Recognition (CVPR). – 2015. – DOI: 10.1109/CVPR.2015.7298965.
35. Barrodale, I. L1 Approximation and the Analysis of Data / I. Barrodale // Journal of the Royal Statistical Society Series C: Applied Statistics. – 1968. – Vol. 17, Iss. 1. – P. 51–57. – DOI: 10.2307/2985267.
36. Mean absolute error with L1 norm. – URL: https://montjoile.medium.com/l0-norm-l1-norm-l2-norm-l-infinity-norm-7a7d18a4f40c (date of access: 20.12.2023).
37. Python programming language and the Tensorflow framework. – URL: https://www.tensorflow.org/about/bib (date of access: 22.10.2023).
38. Osnovy proektirovaniya stroitel'nyh konstrukcij = Asnovy praektavannya budaunichyh kanstrukcyj : SN 2.01.01. – Vved. 16.19.2019. – Minsk : RUP «Strojtekhnorm»: Ministerstvo arhitektury i stroitel'stva Respubliki Belarus', 2020. – 89 s.
39. Koefficient variacii (Variation coefficient). – URL: https://wiki.loginom.ru/articles/variation-coefficient.html (data obrashcheniya: 28.02.2024).

Material received 10/11/2024, approved 11/11/2024, accepted for publication 11/11/2024