UDC 004.032.26 **doi:10.31650/2707-3068-2025-29-60-67**

ARTIFICIAL INTELLIGENCE IN STRUCTURAL DESIGN, AN INTRODUCTION TO NEURAL NETWORKS

Wojnar M., MSc Eng.,
mwojnar@agh.edu.pl, ORCID: 0000-0003-4510-5976
AGH University of Krakow
al. Adama Mickiewicza 30, Krakow, 30-059, Poland
Wojnar A., PhD, Assistant Professor
awojnar@prz.edu.pl, ORCID no. 0000-0002-0537-3864
Rzeszow University of Technology
Aleja Powstańców Warszawy 12, Rzeszow, 35-959, Poland

Abstract. This article presents the potential application of artificial intelligence, particularly artificial neural networks (ANNs), in the design of engineering structures. The subject of the paper is the automatic recognition of the geometric shape of an arc as circular, elliptical or parabolic. Correct identification of the arc shape is fundamental to the creation of the static scheme and computational model of the structure, which is necessary for strength analysis of the structure. This paper analyzes two identification methods based on ANN: multilayer perceptron (MLP) and convolutional neural network (CNN). The MLP network classifies the type of arc based on the geometric features of selected points lying on the arc, while the CNN network makes recognition based on the graphical representation of the arc as a black and white image. The prospects for AI applications in civil engineering are also discussed, with a focus on generative models and their potential use in the design, simulation and automation of construction processes.

Keywords: machine learning, pattern recognition, artificial neural networks, engineering structures, parabolic arc, elliptic arc, circular arc

Introduction.

The dynamic development of artificial intelligence (AI) is opening up new opportunities in engineering, including civil engineering. This paper discusses the application of AI in the design of engineering structures, particularly in the area of geometry and shape of structures. An example is the recognition by AI algorithms of arc shapes: parabolic, elliptical and circular. This issue is relevant, for example, in the context of arch bridge structures (Figure 1), where arches serve as the main load-bearing element that defines the characteristics of the structure. Determining the geometry of the arch is crucial at the early design stage, as the type of arch influences the static scheme of the structure, from which the computational model for strength analysis is created. This analysis is necessary to determine the internal forces in the members and nodal displacements. Therefore, the correct identification of the arch geometry is crucial for the safety and optimisation of the structure.

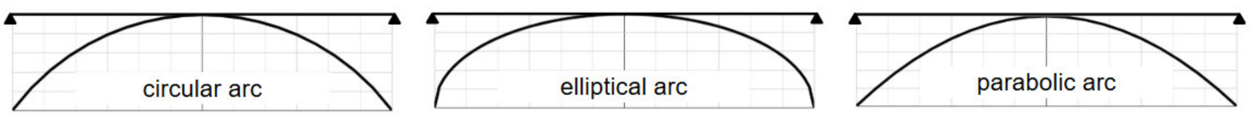


Fig. 1. Examples of static schemes for arched bridges

This paper investigates the ability of artificial neural networks (ANNs) to automatically recognise structural curve types. The focus was on two architectures:

1. Multilayer perceptron (MLP) - classification is based on geometric features of points on an arc. Mathematical relations describing each arc type are learned for recognition.
2. Convolutional neural network (CNN) - a 64x64 pixel black and white image of an arc is analysed, identifying visual patterns specific to each arc class.

The aim of this paper is to evaluate the effectiveness of both methods in identifying arch geometries in engineering structures. The paper also presents potential applications of artificial intelligence in civil engineering, such as AI-assisted programming and the use of generative models in the design and automation of engineering processes. An experiment with a diffusion model to generate images of arches is presented.

The beginnings of neural networks.

Neural networks take their inspiration from the structure and operation of the human brain. Biological neurons communicate using electrical and chemical impulses to form complex networks of connections that enable information processing. This concept was the inspiration for the creation of artificial neural networks. ANNs were invented in the 1940s by Warren McCulloch and Walter Pitts [1]. One of the first network models was the perceptron, developed by Frank Rosenblatt in the 1950s [2]. This was the simplest form of network capable of learning linear classification. The perceptron consisted of a single layer of neurons and was able to discriminate data according to simple rules (Figure 2). On the basis of simple networks such as the perceptron, machine learning and deep learning techniques have evolved and now dominate artificial intelligence applications.

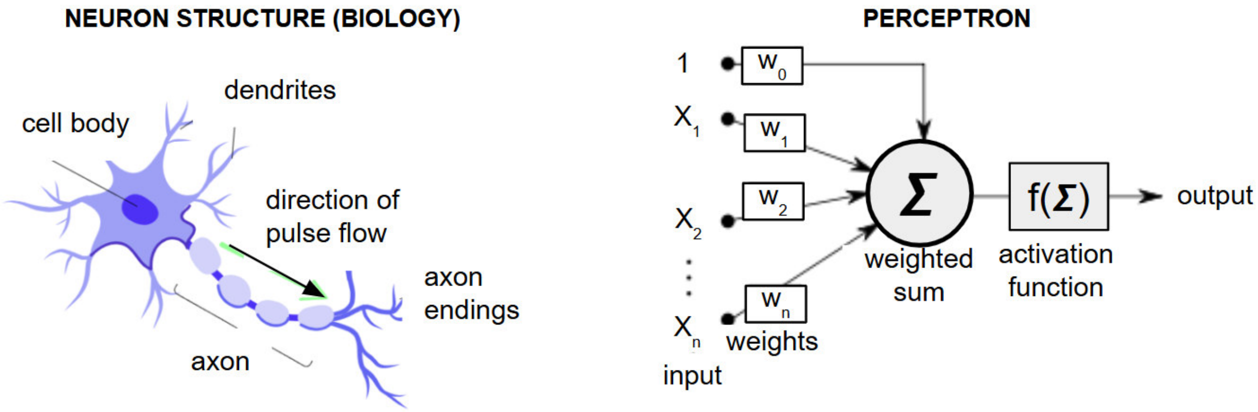


Fig. 2. Comparison of the structure of a biological neuron and a perceptron

Machine learning and deep learning.

Machine learning (ML) is a branch of artificial intelligence that focuses on developing algorithms capable of learning from data. By analysing data and identifying patterns, these algorithms can make decisions or generate predictions based on previously observed trends [3].

A particularly interesting subset of machine learning is deep learning, which is based on multi-layer neural networks - hence the term “deep”. These networks, inspired by the structure and functioning of the human brain, are highly effective at recognising complex patterns in huge data sets. Deep learning has revolutionised areas such as image and speech recognition, natural language processing and autonomous systems, achieving human-like performance in some tasks.

Machine learning algorithms can be divided into three main types [3]:

1. supervised learning (supervised learning) - the algorithm is trained using input data paired with appropriate labels, allowing it to learn mappings from input to output for future predictions, e.g. classification tasks,
2. unsupervised learning - deals with unlabelled data to discover hidden structures or patterns without predefined output, e.g. clustering and dimensionality reduction,
3. reinforcement learning - involves an agent interacting with the environment and learning from reward feedback, e.g. robot control, autonomous vehicles.

Neural network architectures.

Neural networks are models that process input data x through a series of parameterised transformations. The parameters, namely weights (w) and bias (b), are trainable - that is, they are subject to change during training. Each neuron in the network applies an activation function (f) to its input, creating intermediate feature representations, called hidden layers (h). The final layer of the network uses an output activation function (g) to produce the desired output o, which can be a classification label, a numerical prediction or a structured output such as a sequence or image element. A general schematic of neural network processing, along with the relevant formulas and labels, can be found in Figure 5. Training is performed using iterative optimisation techniques such as stochastic gradient descent (SGD) and backpropagation, which adjust the network parameters to minimise the predefined loss function [4].

Neural networks consist of layers tailored to specific tasks. MLPs are the simplest architecture, consisting of linear layers in which each neuron is connected to all neurons in the previous layer. MLPs are widely used for classification, regression and pattern recognition, making them an important building block in the design of neural networks.

CNNs specialise in processing structured data, particularly images, by using spline filters to detect spatial features such as edges, textures or objects [4]. CNNs are key in image recognition, medical imaging or autonomous cars.

Recurrent neural networks (RNNs) introduce feedback loops that allow them to retain memory of previous input data, making them well suited to sequence modelling tasks such as natural language processing (NLP) and time series prediction [4]. The introduction of attention has further improved sequence processing by allowing models to selectively focus on the most relevant parts of the input data [5]. Attention-based architectures, such as transformers, have played a key role in the development of modern NLP techniques, including machine translation, text generation and sentiment analysis.

By combining different types of layers, researchers have developed advanced architectures capable of solving increasingly complex problems. For example, integrating an attention engine with MLP layers led to the development of the transformer architecture, which serves as the basis for large language models (LLM) [5]. Similarly, combining an attention engine with convolutional layers has improved generative vision models, enabling the synthesis of highly realistic images and videos [6].

Neural networks can also be combined into larger systems. Among the most interesting architectures are autoencoders, which consist of an encoder that compresses input data into a lower-dimensional representation and a decoder that reconstructs it. This structure is widely used for dimensionality reduction, anomaly detection, and generative modeling [4]. Generative adversarial networks (GANs) introduce competition between a generator that creates synthetic samples and a discriminator that tries to distinguish real data from generated samples [7]. Adversarial training has led to breakthroughs in image synthesis, deepfake technology, and data augmentation. Combining multiple neural networks into complex processing pipelines enables a wide range of applications, from autonomous systems to creative content generation.

Case study: static diagram – recognizing arc geometry.

As an illustration of the capabilities of neural networks in civil engineering (e.g. structures, theoretical mechanics, geometry and engineering graphics), the problem of recognizing the shape of arcs (i.e. the algorithm recognizes the type of arc) was chosen. This can be an introduction to the static diagram of a structure. In the presented example, arcs are classified as one of three selected curves: parabola, ellipse or circle. Solutions to this problem are presented using two methods:

1. based on the analysis of the coordinates of selected control points on the arc,
2. based on the analysis of the shape of the arc.

Classification based on a feature vector.

In the problem under consideration, a feature vector is used to train the ANN – an ordered set of values describing a given type of arc. These features should be as informative as possible in the context of the decision being made. It is also possible to use algorithms to select the most important features and eliminate the less significant ones. When creating a data set for training the model, the features can be selected by experts or result from measurements that can be carried out in the analyzed environment.

The first dataset contains feature vectors consisting of five elements, which are the tangent of the angle between the plane and the segment connecting the point (-10, 0) and the selected points on the arc. The dataset was generated according to the following assumptions:

- the arcs always intersect the X axis at points −10 and 10 (we are only interested in the shape in the area of positive Y axis values),
- the curves are symmetrical about the Y axis,
- each arc is described by one parameter “a”, which defines its point of intersection with the Y axis,
- the measurement of features is performed at points with coordinates −9, −8, −7, −6, −5 on the X axis, as illustrated in Figure 3,
- the value of parameter “a” was drawn from a uniform distribution in the interval [1, 10], while the type of arc was randomly selected from three available variants: parabola – ellipse – circle.

A total of 10,000 examples were generated, each with five measurements (features) and a label specifying the arc type.

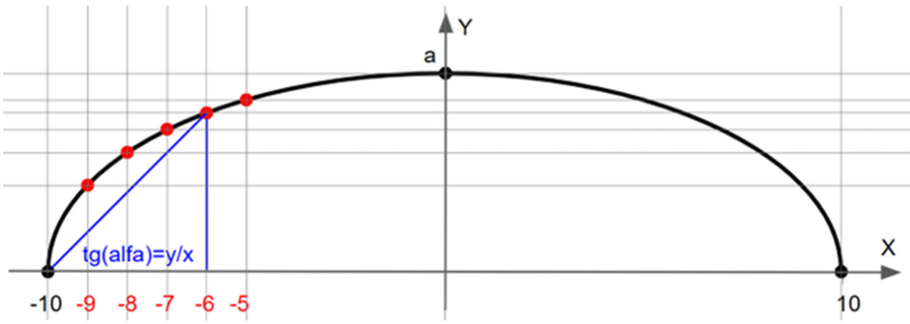


Fig. 3. Illustration of an example arc in the coordinate system, where the black points indicate the intersections of the curve with the axes, the red points indicate the points on the X-axis where measurements were made and represent points on the curve corresponding to the selected points, while the method of measuring the features is marked in blue

Image based classification.

The second dataset contains arc images in 64x64 pixel resolution, in which the arc curve is shown as a white line on a black background, Figure 4. The images represent a fragment of the coordinate system covering the range from -10 to 10 on the X-axis and from 0 to 20 on the Y-axis. The arcs were generated according to the same assumptions as in the previously described dataset, and a total of 10,000 examples were generated.

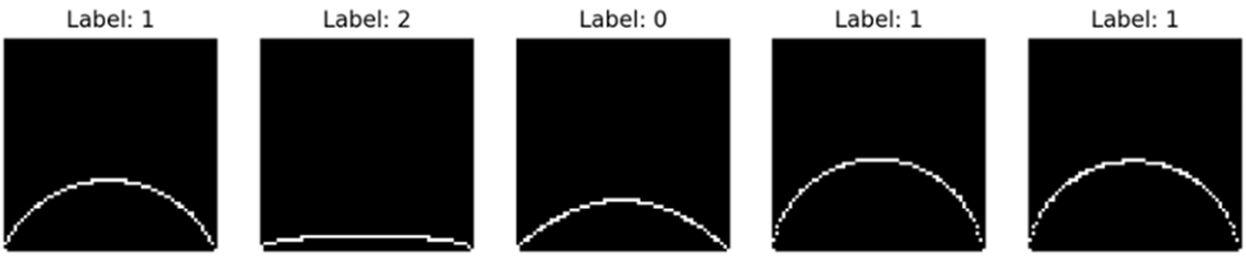


Fig. 4. Example images with arcs from the dataset. Labels: 0 – parabola, 1 – ellipse, 2 – circle

Both datasets were divided into three subsets: 70% for training, 20% for validation, and 10% for testing. The training sets were used to optimize the models, while the validation sets allowed monitoring performance during training and guiding the selection of hyperparameters such as the number of optimizer steps. The test sets, which remained unseen during training, were used to evaluate the generalization ability of the models.

MLP Network for Feature-Based Classification – Results.

The MLP model was implemented using the Keras[1] framework and the TensorFlow[2] library. The neural network consisted of an input layer, three linear layers with 32 neurons and ReLU (Rectified Linear Unit) activations, and an output layer with softmax activation. The network was trained using the Adam algorithm with a learning rate of 0.001. The output was the classification probability of the arc curve (Figure 5).

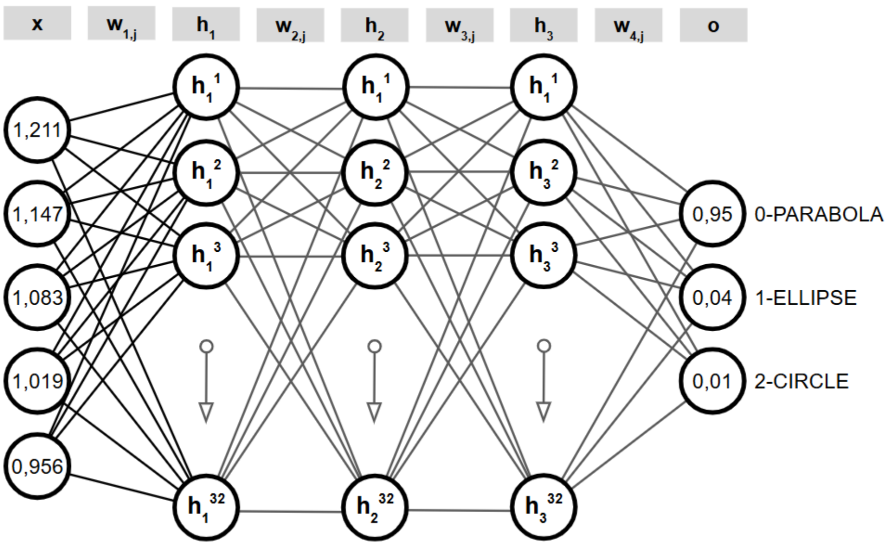
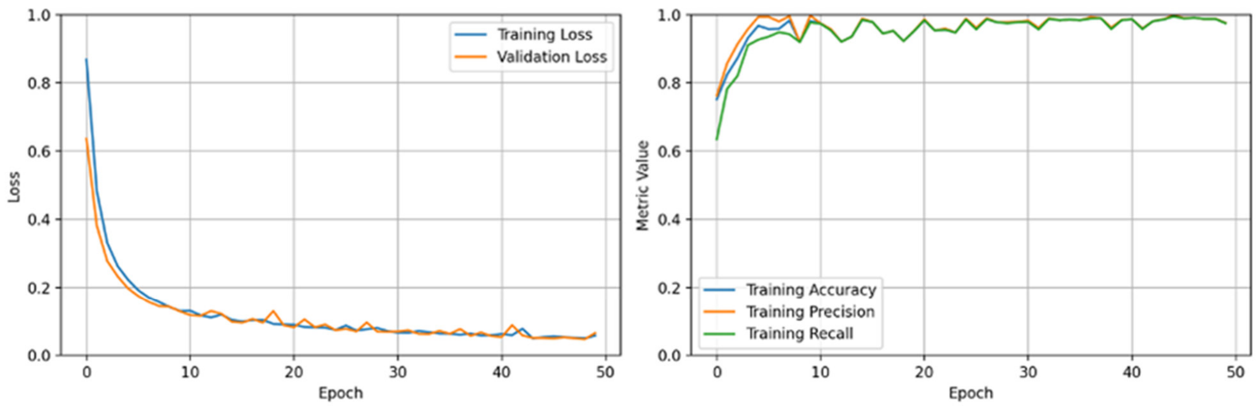


Fig. 5. Schematic representation of a neural network in the MLP architecture. On the left is an example input vector x, while on the right are the output probabilities for each class

The training process was carried out for 50 epochs with a batch size of 64. The model achieved an accuracy of 99% on the test set. Figure 6 illustrates the loss function curves and metrics during training, showing fast convergence and good model fit.



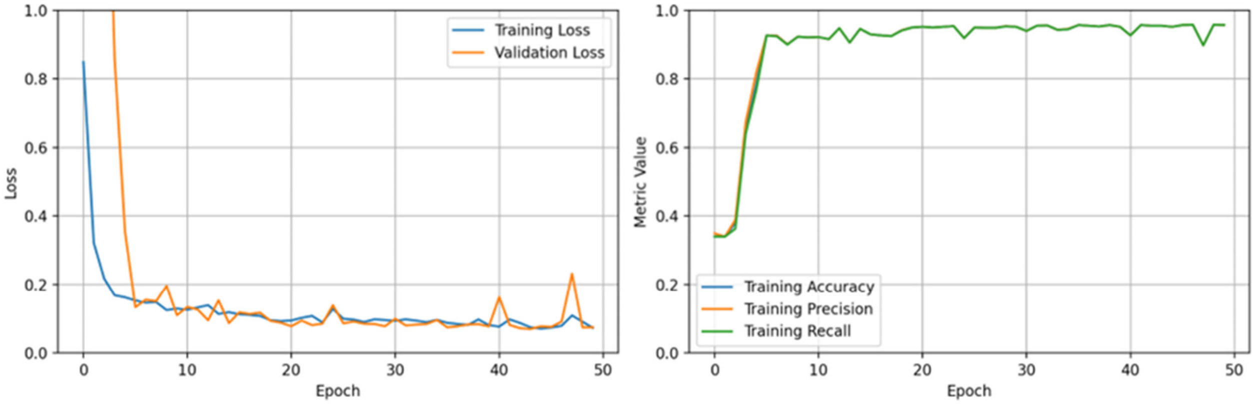

```
Epoch 50/50
110/110 ━━━━━━━━━━━━━━━━━━━━ 0s 934us/step - categorical_accuracy: 0.9817 - loss: 0.0530 - precision: 0.9817 - recall: 0.9815 -
63/63 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - categorical_accuracy: 0.9921 - loss: 0.0382 - precision: 0.9921 - recall: 0.9921

Test results:
Accuracy: 0.9915
Precision: 0.9915
Recall: 0.9915
```

Fig. 6. The value of the loss function and metrics on the training and validation sets during MLP model training

[1] https://keras.io
[2] https://www.tensorflow.org

CNN for Image-Based Classification – Results.

For image-based classification, a CNN model consisting of five convolutional blocks, each containing a 2D convolutional layer (32 filters, 3x3 kernel size, ReLU activation), batch normalization, and max pooling, was used. The extracted features were then flattened into a vector and passed through two linear layers with 128 and 64 neurons, respectively, both using ReLU activation. Dropout layers with a deactivation factor of 0.5 were introduced to mitigate the overfitting effect, and the output layer used softmax activation. The CNN model was trained with the same optimizer and cost function as the MLP model. The final accuracy estimate on the test set was 96%. The loss function and metric curves are shown in Figure 8. It is worth noting that the CNN model achieves slightly lower performance than the MLP, but it is used to solve a much more difficult task (Figure 8) involving the classification of arcs where the data is based on images of the arc geometry and not on features determined from the measured arc characteristics.




```
Epoch 50/50
110/110 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - categorical_accuracy: 0.9607 - loss: 0.0748 - precision_1: 0.9611 - recall_1: 0.9605
63/63 ━━━━━━━━━━━━━━━━━━━━ 1s 7ms/step - categorical_accuracy: 0.9612 - loss: 0.0693 - precision_1: 0.9612 - recall_1: 0.9612

Test results:
Accuracy: 0.9635
Precision: 0.9635
Recall: 0.9635
```

Fig. 7. The value of the loss function and metrics on the training and validation sets during CNN model training

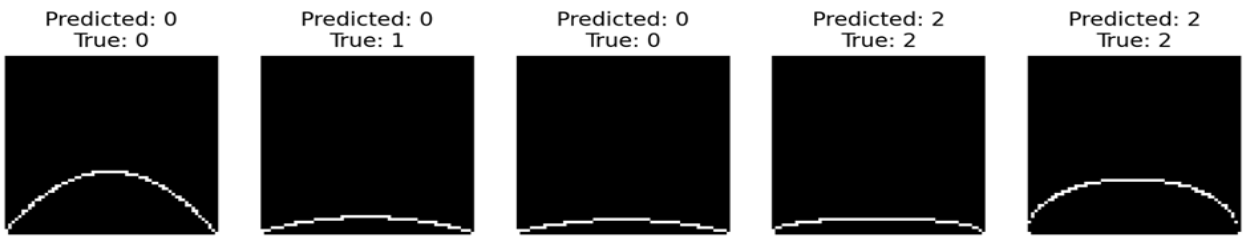


Fig. 8. Example of a CNN model “mistake” – second image from the left: predicted arc curve – parabola, real – ellipse

AI-Assisted Programming and Design.

AI is not limited to simple problem solving and classification, but also encompasses advanced generative models and programming support systems, also in engineering and scientific contexts. The vast majority of the code used in the experiments described in this paper was generated by ChatGPT[3] or GitHub Copilot[4] . ChatGPT, as a general-purpose LLM, was used to create the main parts of the code, in particular arc generation, dataset generation (both feature-based and image-based), and model implementation and neural network training. GitHub Copilot, as a tool directly supporting the user during programming, allowed for the coherent connection of the code and the addition of important functionalities, such as graph generation or further processing of the dataset[5].

[3] https://chatgpt.com/
[4] https://github.com/features/copilot
[5] https://chatgpt.com/share/6786b9ab-7ea4-800c-a62b-1e0bfe09b133

This means that even an inexperienced user is able to train AI systems relatively quickly and with little effort, especially small or medium-scale specialized models. In combination with interesting, good-quality data sets, an AI model can be a useful tool in a scientist's workshop. LLM can also act as an assistant generating code documentation or supporting the process of finding a solution to a given problem. The basic role of the user is therefore the ability to verify the LLM response and at least basic knowledge of the domain of the query (prompt) directed to the AI system. This is necessary to use AI tools effectively, i.e. to distinguish verified information from hallucinations and to be able to direct the model to the expected answer.

Potential Applications of Generative Models in Civil Engineering

Generative models can be used in civil engineering, supporting the process of designing structures at various stages. One of the key areas of their use may be supporting architectural design by generating various proposals, which allows for the rapid exploration of alternative structural solutions. Diffusion models enable the creation of many design variants and their adaptation to functional and aesthetic requirements. Another example may be the optimization of existing structures - AI could analyze and modify designs to improve aerodynamics, effectively distribute stresses or save materials while maintaining the required strength.

AI is used in fast physical simulations, enabling prediction of the behavior of structures under different environmental conditions. Special neural network architectures, such as Physics-Informed Neural Network (PINN) [9], allow solving differential equations describing the physical properties of structures. Generative AI models can additionally support testing designs in extreme conditions, creating realistic data for simulation.

Beyond design and optimization, AI models can also support the automation of construction processes. Construction robots could perform repetitive and dangerous tasks, increasing work efficiency and reducing risk to humans. AI can also improve the logistics of material deliveries, optimizing their scheduling and transportation, leading to reduced losses and delays on the construction site.

AI could also improve safety on construction sites. Drones equipped with AI systems could monitor progress and conduct inspections, identifying potential hazards. Advanced sensor data analysis systems could detect anomalies and predict structural failures before serious damage occurs. Additionally, intelligent warning systems could continuously monitor the environment and immediately inform workers of potential dangers.

Conclusions.

This paper presents a case study of the use of ANN to recognize the types of arcs: circular, elliptical and parabolic. This problem is important in the context of civil engineering, where correct geometry identification is a prerequisite to creating static schemes, e.g. arch bridges. Two classification methods were investigated: analytical, using the MLP network, and graphical, based on the CNN network.

The MLP model analysed a vector of five geometric features (tangents of angles) determined for points on the arc. The CNN model processed black-and-white images of arcs with a resolution of 64x64 pixels. Data sets containing 10,000 examples each, generated according to specific geometric assumptions, were used for training and testing the network.

The analysis of the results showed very high efficiency of both approaches. The MLP network achieved a classification accuracy of 99% on the test set. The CNN network achieved an accuracy of 96%. The lower result of the CNN was due to the higher complexity of the image processing task compared to the analysis of predefined features. Both results confirm the great potential of ANNs in automating geometric pattern recognition tasks in engineering.

Artificial intelligence is becoming a versatile tool that can significantly improve various stages of a design engineer's work. However, effective use of these tools requires the ability to verify results and basic knowledge of the field.

References

[1] W.S. McCulloch, W. Pitts. “A logical calculus of the ideas immanent in nervous activity”. Bulletin of Mathematical Biophysics 5, 115–133 (1943).
[2] F. Rosenblatt. “The perceptron: A perceiving and recognizing automaton”. Report, Project PARA, Cornell Aeronautical Laboratory, 85-460-1 (1957).
[3] C. Bishop. “Pattern Recognition and Machine Learning (Information Science and Statistics)”. Springer-Verlag (2006).
[4] I. Goodfellow, Y. Bengio, A. Courville. “Deep learning”. MIT Press (2016).
[5] A. Vaswani et al. “Attention is All you Need”. Advances in Neural Information Processing Systems 30 (2017).
[6] W. Peebles, S. Xie. “Scalable Diffusion Models with Transformers”. arXiv cs.CV 2212.09748 (2023).
[7] I. Goodfellow et al. “Generative Adversarial Networks”. arXiv stat.ML 1406.2661 (2014).
[8] J. Ho, A. Jain, P. Abbeel. “Denoising diffusion probabilistic models”. Proceedings of the 34th International Conference on Neural Information Processing Systems 574 (2020).
[9] M. Raissi, P. Perdikaris, G. Karniadakis. “Physics Informed Deep Learning (Part I): Data-driven Solutions of Nonlinear Partial Differential Equations”. arXiv cs.AI 1711.10561 (2017).

ШТУЧНИЙ ІНТЕЛЕКТ У СТРУКТУРНОМУ ПРОЄКТУВАННІ, ВСТУП В НЕЙРОНІ МЕРЕЖІ

Вожнар М., магістр,
mwojnar@agh.edu.pl, ORCID: 0000-0003-4510-5976
Краковський университет AGH
Вожнар А., к.т.н.,
awojnar@prz.edu.pl, ORCID: 0000-0002-0537-3864
Жешувський технологічний університет

Анотація. У статті розглядається потенційне застосування штучного інтелекту, зокрема штучних нейронних мереж (ШНМ), під час проєктування інженерних споруд. Предметом статті є автоматичне розпізнавання геометричної форми дуги як кругової, еліптичної чи параболічної. Правильне визначення форми дуги має основне значення для створення статичної схеми та обчислювальної моделі конструкції, що необхідно для аналізу міцності конструкції. У цій статті аналізуються два методи ідентифікації на основі ШНМ: багатошаровий персептрон (БШП) та нейронна мережа (СНМ). Мережа БШП класифікує тип дуги на основі геометричних особливостей обраних точок, що лежать на дузі, в той час як мережа СНМ здійснює розпізнавання на основі графічного представлення дуги у вигляді чорно-білого зображення. Також обговорюються перспективи застосування ШІ у цивільному будівництві з акцентом на генеративні моделі та їхнє потенційне використання при проектуванні, моделюванні та автоматизації будівельних процесів.

Аналіз результатів показав дуже високу ефективність обох підходів. Мережа БШП досягла точності класифікації 99% на тестовому наборі. Мережа СНМ досягла точності 96%. Нижчий результат СНМ був обумовлений більшою складністю завдання обробки зображень у порівнянні з аналізом заздалегідь визначених ознак. Обидва результати підтверджують великий потенціал ШНМ в автоматизації завдань розпізнавання геометричних візерунків в інженерії. Штучний інтелект стає універсальним інструментом, який може значно поліпшити різні етапи роботи інженера-конструктора. Однак ефективне використання цих інструментів вимагає вміння перевіряти результати та базових знань у цій галузі.

Ключові слова: машинне навчання, розпізнавання образів, штучні нейронні мережі, інженерні конструкції, параболічна дуга, еліптична дуга, дуга кола.