 https://doi.org/10.1051/e3sconf/202337603027

Application of artificial intelligence and neural network technologies in the construction industry

Nikolay Garyaev[1*] and *Andrey* Garyaev[1]

[1]Moscow State University of Civil Engineering, 26, Yaroslavskoe Shosse, 129337, Moscow, Russia

Abstract. One of the problems that may arise in the way of successful implementation of energy supply in urban areas is the difficulty of analyzing and interpreting a large amount of digital data received from various sensors. This problem may adversely affect the performance of energy organizations. The purpose of this study is to study modern tools to solve the problem of processing big data using technologies of simulation and artificial intelligence. This study is dedicated to the development of innovative digital models for the balanced distribution of energy consumption in urban areas.
Keywords: neural networks, artificial intelligence, construction, model, deep learning.

1 Introduction

The use of artificial intelligence to identify construction equipment using CCTV cameras is relevant as it has the potential to improve efficiency and safety at construction sites. By using AI to identify and classify machine types, operators can quickly determine their intended purpose, helping ensure they are used correctly and minimizing the risk of accidents. In addition, this technology can also help optimize construction processes, reduce downtime and increase overall productivity.

To solve the problem, the following stages were planned:

1. Study of foreign and Russian experience and practical development of methods for creating neural networks for the development of artificial intelligence systems.
2. Research and collection of data on the types of construction equipment and their purpose.
3. Development of an artificial intelligence model for recognition and classification of cars based on their visual characteristics.
4. Testing the system to evaluate its accuracy and effectiveness in determining the type of equipment.
5. Analysis and evaluation of the results in order to improve and adjust the AI model.

* Corresponding author: garyaev@mgsu.ru

1.1 Analysis of experience in the field of AI

The results of studying foreign and Russian experience [1,2,7] made it possible to group the following methods for creating neural networks for the development of artificial intelligence systems:

1. Supervised learning: training a neural network on a labeled dataset to learn the relationship between inputs and outputs and make predictions on unseen data.
2. Unsupervised Learning: Using a neural network to discover patterns and relationships in a dataset without predefined labels. This can be used for tasks such as clustering and dimensionality reduction.
3. Reinforcement learning: training a neural network through a series of trial and error where the network receives feedback on its actions and adjusts its behavior to maximize the reward signal.
4. Transfer learning: using a pre-trained neural network as a starting point for a new task, tuning the network on a smaller dataset to adapt it to the new task.
5. Generative Adversarial Networks (GAN): Training two neural networks, a generator and a discriminator, to work together to create new data samples similar to the training dataset.
6. Convolutional Neural Networks (CNN): Using convolutional layers to analyze image data and extract features that can be used for tasks like image classification and object detection.
7. Recurrent Neural Networks (RNN): The use of recurrent connections to allow the network to process sequential data such as time series data or natural language text.
8. Autoencoders: training a neural network to compress the input data into a smaller size, and then restore the data from the compressed representation.
9. Hybrid models: Combining different types of neural networks and methods to create a customized solution for a specific task or problem.

According to the results of studying the works of various authors [3,4,5,6], in addition to the above, the advantages of neural networks were highlighted:

- Ability to learn and generalize: Neural networks can learn complex relationships between inputs and outputs and generalize these relationships to unseen data.
- Resilience to noise and missing data: Neural networks can still make accurate predictions even if the input data is noisy or contains missing values.
- Processing of non-linear relationships: Neural networks can model non-linear relationships between inputs and outputs, which is difficult for traditional linear models.
- Ability to process large amounts of data: Neural networks can process large datasets efficiently, making them well suited for big data applications.
- Versatility: Neural networks can be used for a wide range of tasks, including classification, regression, and clustering, as well as for more specialized tasks such as image and speech recognition.

Disadvantages of neural networks:

- Complexity: Neural networks can be complex and difficult to interpret, making it difficult to understand how they make predictions.
- Overfitting: Neural networks can easily adapt to training data, resulting in poor performance on unseen data.
- Training Time: Training a neural network can be computationally intensive and time consuming, especially for large datasets or complex models.
- Lack of transparency: Neural networks can be considered a black box as it can be difficult to understand how they make decisions, which can be a problem in applications such as healthcare or finance where decisions have a significant impact.
- Vulnerability to hostile attacks: Neural networks can be vulnerable to hostile examples, where small changes in input data lead to incorrect predictions.

 https://doi.org/10.1051/e3sconf/202337603027

• Requirement of large volumes of labeled data. Neural networks often require large amounts of labeled data for efficient training, which can be difficult and time consuming to obtain.

To date, there are several popular Python libraries for creating and developing neural networks:

TensorFlow: An open-source software library for data flow and differentiated programming for a range of tasks.

PyTorch: An open-source machine learning library based on the Torch library used for applications such as computer vision and natural language processing.

Keras: An open-source software library that provides a Python interface to ANNs.

Theano: An open-source numerical computing library for Python that allows you to efficiently define, optimize, and evaluate mathematical expressions involving multidimensional arrays.

Caffe: A deep learning framework built with expressiveness, speed, and modularity in mind.

MATLAB Deep Learning Toolbox: Provides algorithms and tools for building, training, and visualizing deep neural networks.

Microsoft Cognitive Toolkit (CNTK): An open-source deep learning toolkit developed by Microsoft Research.

MXNet: An open-source deep learning platform designed for efficiency and flexibility.

Chainer: A flexible, intuitive and high-performance deep learning framework for Python.

DL4J: A deep learning library for Java designed to support the creation of a wide variety of deep neural networks.

1.2 Types of construction equipment and their purpose

The study of materials on this issue made it possible to form the main types of construction equipment and their purpose:

Excavators: used for digging, leveling and demolition.

Bulldozers: heavy equipment with a large metal plate used to push soil, sand and other materials.

Cranes: Used to lift and move heavy objects, including steel beams and concrete blocks.

Dump Trucks: Large open bed trucks for hauling and unloading heavy materials such as crushed stone, sand and gravel.

Concrete mixers: Vehicles that transport and mix concrete ingredients for pouring at a construction site.

Loaders: Heavy equipment used to load and move materials such as dirt and gravel onto trucks or onto conveyors.

Compactors: Used to compact soil and asphalt to create a solid surface for construction projects.

Asphalt Pavers: Used to lay asphalt and concrete on driveways, sidewalks, and other hard surfaces.

Graders: heavy bladed equipment used for leveling and leveling the surface of a construction site.

These are just some of the many types of construction equipment used in the industry. Each type of equipment has a specific purpose and is designed to perform construction tasks more efficiently and effectively.

At the next stage of work, a trial dataset of images of construction equipment was formed. An example of the main types of road construction machines is shown in Figure 1.

Fig. 1. The main types of construction machines.

2 Software development of artificial intelligence model

The neural network model is implemented using convolutional layers that allow you to analyze image data.

A convolutional layer is a type of layer in a neural network that works by moving a small matrix (known as a filter or kernel) over an input image and computing element-wise multiplications between the filter elements and the input, creating a new feature map. This process helps the network to learn local features of the image, such as edges, shapes, and textures.

Python example using Keras library:

```
from keras.layers import Conv2D
from keras.models import Sequential
model = Sequential ()
model.add (Conv2D (32, kernel_size= (3,3), activation='relu', input_shape= (28, 28, 1)))
```

This is the creation of a convolutional layer with 32 filters, each one 3x3 in size, followed by a ReLU activation function. The input shape is listed as 28x28 with 1 channel (grayscale).

Additional convolutional layers can be added to the network to increase its complexity and ability to learn more complex features in the image data. You can also add pooling layers between convolutional layers to reduce the spatial dimensions of the feature maps, which helps reduce the computation required and prevent overfitting.

An example with two additional convolutional layers and a maximum pooling layer:

```
model.add(Conv2D(64, kernel_size=(3,3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Conv2D(128, kernel_size=(3,3), activation='relu'))
```

This adds two more convolutional layers with 64 and 128 filters respectively, each with a 3x3 kernel. A maximum pooling layer is also added between the first and second convolutional layers, with a pool size of 2x2.

You can also add fully connected layers to the end of the network to make predictions based on the features learned by the convolutional layers. Dropdown layers can also be added to prevent overfitting. Here is an example:

```
model.add(Flatten())
model.add(Dense(128, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(10, activation='softmax'))
```

This concatenates the feature maps into a one-dimensional array, adds a dense layer with 128 units and ReLU activation, followed by an exclusion layer with a factor of 0.5. The last layer is a dense layer with 10 units and "softmax" activation, which is used for multi-class classification problems.

The practical implementation of the model was carried out in Google Colab, a free cloud platform for data processing and the development of machine learning systems. The platform provides access to the Jupyter notebook environment, which supports multiple programming languages, including Python. The platform also provides access to free GPUs and TPUs, making it easier for users to perform complex and resource-intensive computing.

3 Training and testing an artificial intelligence model

Practical experience has shown that neural network training results depend on various factors, including the quality of training data, network architecture, the optimization algorithm used, the amount of training data, the choice of hyperparameters, and the amount of computing power available.

The goal was to train the network to minimize the difference between its predictions and the actual output, known as the loss or cost function. During training, the weights and biases of the network are updated to reduce the loss of training data. If the network overfits, which means it performs well on the training data but poorly on new data, the model can be streamlined, such as adding dropout or weight reduction.

Ultimately, the performance of a trained neural network can be quantified using metrics such as accuracy, reproducibility, and recall. The best results depend on the specific problem being solved and the desired performance trade-off.

4 Conclusions

When implementing AI algorithms, the following problems arise:

1. The quality and quantity of data available to train AI algorithms can have a significant impact on their performance. Poor quality data can lead to biased models.
2. Overfitting occurs when the AI model is too complex and matches the training data too closely.
3. Training some AI algorithms can be computationally expensive, requiring large amounts of memory and processing power. This may limit their practical use in systems with limited resources.
4. AI models can sometimes produce results that are difficult to interpret or explain.
5. Safety and reliability. This can be a problem as AI models can sometimes behave in unexpected ways.

In conclusion, it should be noted that the use of artificial intelligence to determine the type of construction equipment using CCTV cameras has shown great potential. With the development of AI technology, surveillance cameras equipped with AI algorithms can accurately identify and classify various types of construction vehicles in real time. This information can be used to improve site safety, improve efficiency and reduce costs. The use of AI in video surveillance also opens up new opportunities for advanced data analysis and decision making in the construction industry.

5 Directions for further research

By integrating AI technology with construction machines, the system can automatically detect the type of machine being used, its current status, and its location on the construction site. This information can then be used to improve construction efficiency in the following ways:

Vehicle Distribution: The AI system can determine which type of vehicle is best for each task and distribute equipment accordingly. This ensures that the correct machine is used for the job, resulting in faster lead times and lower costs.

Performance Monitoring: The AI system can monitor the performance of each machine and provide real-time feedback to the operator. This allows operators to make adjustments to the machine to optimize performance and avoid breakdowns.

Safety enhancements: The AI system can detect potential safety hazards such as collisions or rollovers and alert operators to take corrective action. This improves safety on the construction site and reduces the risk of accidents.

References

1. M.V. Akinin, M.B. Nikiforov, A.I. Taganov, *Neural network systems of artificial intelligence in image processing tasks* (RiS, M., 2016)
2. D. Barret, *The last invention of mankind: artificial intelligence and the end of the era of Homo sapien*s (Alpina non-fiction, M., 2015)
3. E.V. Borovskaya, N.A. Davydova, *Fundamentals of artificial intelligence: textbook, manual* (BINOM. Knowledge Laboratories, M., 2014)
4. V.I. Demkin, D.K. Lukov, Bulletin of Modern Research **21**, 366-368 (2018)
5. M.T. Jones, Programming artificial intelligence in applications **311** (2018)
6. Yu.A. Zagorulko, G.B. Zagorulko, Artificial intelligence **93** (2018)
7. Yu.A. Isakov, Artifical intelligence, 25-27 (2018)
8. S.L. Grankin, N.A. Garyaev, Scientific review **13**, 159-162 (2015)