DEVELOPMENT OF A SOFTWARE SYSTEM FOR TRAFFIC SIGN RECOGNITION

RAKIC, K., SEREMET, Z. & ANTONINI, A.

Abstract: *Artificial intelligence represents the theory and direction of the development of computer technologies that imitate human intelligence, senses, and behaviour, such as visual perception, speech recognition, decision making, and language translation. A neural network is a method in artificial intelligence that teaches computers to process data in a way inspired by the human brain using neurons and hidden layers. The developed software system for traffic sign recognition is based on a neural network. The programming languages used are Python and JavaScript. Vue.js and Vuetify were used for the development of the user interface, Flask for the server part of the application's functionality and communication, and TensorFlow for the creation of neural networks. This paper gives an overview of the basic concepts and history of artificial intelligence, neural networks, and a brief description of the implementation of the software system.*

Key words: *artificial intelligence, neural networks, image recognition, TensorFlow*

Authors´ data: Asst. Prof. Dr. Sc. Rakic, K[resimir]; Asst. Prof. Dr. Sc. Seremet, Z[eljko]; Antonini, A[nte], University of Mostar, Trg hrvatskih velikana 1, 88000, Mostar, Bosnia and Herzegovina, kresimir.rakic@fsre.sum.ba, zeljko.seremet@fsre.sum.ba, ante.antonini@student.fsre.ba

This Publication has to be referred as: Rakic, K[resimir]; Seremet, Z[eljko] & Antonini, A[nte] (2022). Development of a Software System for Traffic Sign Recognition, Chapter 12 in DAAAM International Scientific Book 2022, pp.143-154, B. Katalinic (Ed.), Published by DAAAM International, ISBN 978-3-902734-34-1, ISSN 1726-9687, Vienna, Austria
DOI: 10.2507/daaam.scibook.2022.12

1. Introduction

The concept of inanimate objects endowed with intelligence has existed since ancient times. The Greek god Hephaestus is depicted in myths as forging robot-like servants from gold. Engineers in ancient Egypt built statues of gods animated by priests. Throughout the centuries, thinkers from Aristotle to Ramon Llull the 13th century Spanish theologian, to René Descartes and Thomas Bayes used the tools and logic of their time to describe human thought processes as symbols, laying the foundation for the concepts of artificial intelligence.

The end of the 19th and the first half of the 20th century brought the fundamental work that would lead to the modern computer. In 1836, mathematician Charles Babbage of Cambridge University and Augusta Ada Byron, Countess of Lovelace, invented the first design of a programmable machine. In the 1940s, Princeton mathematician John Von Neumann came up with the stored-program computer architecture, the idea that a computer program and the data it processes can be stored in the computer's memory. Warren McCulloch and Walter Pitts laid the foundation for neural networks.

A summer symposium held at Dartmouth College in 1956 is often credited with launching the current discipline of artificial intelligence. On it, they presented the revolutionary Logic Theorist, which is often called the first AI program. It is a computer program that can prove certain mathematical theorems.

After a Dartmouth College symposium, pioneers in the development of artificial intelligence predicted that machine intelligence comparable to the human brain was inevitable. In the years that followed, important advances were made in artificial intelligence. McCarthy created Lisp, an artificial intelligence programming language that is still in use today. Newell and Simon published the General Problem Solver algorithm in the late 1950s. The natural language processing program ELIZA, created by MIT professor Joseph Weizenbaum in the mid-1960s, served as the inspiration for modern chatbots. In the late 1990s, a renaissance in artificial intelligence technology was launched, which has continued to this day, thanks to advances in computing power and the growth of data. Innovations in computer vision, robotics, machine learning, deep learning and more have come out of the recent focus on artificial intelligence. Garry Kasparov was defeated by IBM's Deep Blue in 1997, the first time a computer program had ever beaten a world chess champion (Burns, 2022). Artificial intelligence is cementing its place in popular culture and becoming increasingly tangible, driving cars, diagnosing diseases and more.

2. Artificial Intelligence

Often what we call artificial intelligence is only one of its elements. Learning, reasoning, and self-correction are the three cognitive abilities that artificial intelligence programming mostly focuses on. Learning process is the field of artificial intelligence which is concerned with collecting data and formulating rules that will enable the transformation of data into useful knowledge. Guidelines, also known as algorithms, give computer equipment detailed instructions on how to perform a specific activity.

Reasoning process represents the field of artificial intelligence that deals with choosing the best algorithm to achieve a certain result. Self-correcting process means constantly improving the algorithms and ensuring that they provide the most accurate results.

2.1. Categorization of Artificial Intelligence

Artificial intelligence can be classified as strong or weak. An artificial intelligence system that is created and trained to perform a specific task is called weak artificial intelligence, also known as narrow artificial intelligence. Weak artificial intelligence is used by industrial robots and virtual personal assistants like Apple's Siri. Strong artificial intelligence is a term used to describe computer behaviour that can mimic human cognitive functions. A strong artificial intelligence system can use fuzzy logic to transfer information from one area to another and figure out a solution on its own when faced with an unexpected job. Theoretically, powerful artificial intelligence software should be able to pass the Turing test (Burns, 2022.).

The Turing test, first called the imitation game by Alan Turing, determines the extent to which a machine is capable of displaying intelligent behaviour that is comparable or indistinguishable from that of a living person. According to Turing, a human judge would evaluate a natural language exchange between a person and a computer designed to give human-like responses. The outcome would not depend on the machine's ability to convert words into speech because the dialogue would be limited to a text-only channel, such as a computer keyboard and screen. A machine would pass the Turing test if the evaluator could not consistently distinguish a person from a machine (French, 2000).

2.2. Advantages and Disadvantages of Artificial Intelligence

Some of the artificial intelligence technologies, such as deep learning and neural networks, are developing rapidly, mainly because artificial intelligence can process huge amounts of data much faster and more accurately than a human can. While the vast amount of data generated every day would drown a human researcher, artificial intelligence technologies using machine learning can quickly transform that data into useful knowledge.

Some of the advantages of using artificial intelligence technologies are:

- Success in jobs that require attention to detail.
- Reduced time for performing tasks for data-intensive activities.
- Consistency of results.
- 24/7 availability of virtual agents with artificial intelligence capabilities.

Some of the problems that hinder the wider use of artificial intelligence tools and technologies are:

- The high cost of processing enormous amounts of data.
- The necessity of high technical competence of the staff.
- Limited availability of qualified experts to create artificial intelligence tools.

3. Neural Networks

A neural network is a set of algorithms whose goal is to identify underlying connections in a piece of data using a method that mimics the functioning of the human brain. In this context, neural networks are systems of neurons that can be of organic or synthetic origin. Because neural networks can adapt to changing inputs, the network can produce an optimal outcome without having to change the output criterion. The idea of neural networks based on artificial intelligence is rapidly gaining importance in the design of trading systems (Burns, 2021).

Warren McCulloch and Walter Pitts invented the first neural network in 1943. They published a fundamental study of how neurons might work and presented their theories by building a basic neural network out of electrical circuits. This paradigm opened the way to the research of neural networks in two directions:

- Biological processes in the brain
- Application of neural networks to artificial intelligence.

The initial goal of the neural network was to construct a computer system capable of solving problems in the same way that the human brain does. However, over time it evolved towards the use of neural networks for certain tasks, which resulted in a deviation from a strictly biological approach. Neural networks have since been used to serve a wide range of activities, including computer vision, voice recognition, machine translation, social network filtering, social and video games, and medical diagnosis (SAS Analytics Software and Solutions, n.d.).

3.1. How Neural Networks Work

There are three layers that make up the basic neural network: input, output, and hidden layer. The nodes that interconnect the layers create a "network" of interconnected nodes, called a neural network.

The data is delivered to the neural network via the input layer which is connected to the hidden layers. Processing takes place in hidden layers through a network of weighted connections. The nodes in the hidden layer then mix the data from the input layer with a set of coefficients and apply appropriate weights to the inputs. These products of weights and inputs are then added together. The total result is sent through the node's activation function, which determines how far the signal must travel through the network to affect the final output. Finally, the hidden layers are connected to the output layer, where the outputs are collected (IBM Cloud Education, 2020).

3.2. Types of Neural Networks

There are several types of neural networks. Some examples are (Gutai, et al., 2021):

- Convolutional neural networks consist of five layers: input, convolution, shrinkage, fully connected and output. Each layer has a specific function, such as compression, linking or activation. Convolutional neural networks have made image categorization and object recognition more popular. Convolutional neural networks are used in various fields such as natural language processing and prediction.

- Recurrent Neural Networks use sequential input, such as time-stamped data from a sensor device or spoken speech composed of a sequence of words. Unlike standard neural networks, the inputs to a recurrent neural network are not independent of each other, and the output for each element depends on the calculations of the elements before it. Forecasting and time series applications, sentiment analysis, and other text applications use recurrent neural networks.
- Feedforward neural networks connect each perceptron in one layer to each perceptron in the next layer. Information is sent from one layer to another in one direction only. There are no feedback loops.
- Autoencoder Neural Networks are used to construct abstractions known as encoders from a given set of inputs. Although autoencoders are comparable to more typical neural networks as they tend to represent the inputs themselves, the approach is considered unsupervised. Autoencoders work on the principle of desensitizing the irrelevant and sensitizing the relevant. Further abstractions are defined at higher levels as layers are added (layers closest to the point where the decoder layer is introduced). Linear or non-linear classifiers can then use these ideas.

3.3. Advantages and Disadvantages of Neural Networks

Neural networks represent one of the technological breakthroughs expected to drive the modernization of society to a whole new level, giving machine learning aspects of human-like behaviour. Along with the numerous advantages that neural networks have over other methods of data processing, there are also some disadvantages

Some of the advantages of neural networks are:

- Parallel processing means that the network can handle several tasks at once.
- Information is stored over the network, not just in a database.
- The ability to learn and model nonlinear, complicated interactions helps model input-output relationships in the real world.
- Fault tolerance indicates that damage to one or more cells of a neural network will not prevent the output from being produced.
- Gradual damage suggests that the network will gradually deteriorate over time, rather than the problem immediately damaging the network.
- Ability to produce results with imperfect knowledge, with performance loss proportional to the importance of the missing information.
- There are no restrictions on the input variables, such as the way they are distributed.

Some of the disadvantages of neural networks are:

- Since there are no criteria for identifying a suitable network topology, the appropriate design of an artificial neural network must be discovered through trial and error and experience.
- Since neural networks require processors with parallel processing capabilities, they are hardware dependent.
- Since the network operates on numerical data, all problems must be converted to numerical values before they can be submitted to the artificial neural network.

- One of the most significant disadvantages of neural networks is the lack of explanation behind the probing solution. The inability to articulate the why or how of solutions breeds mistrust in the network.

Neural networks are well-suited to helping humans solve complicated challenges in real-world scenarios. They can learn to model nonlinear and complicated interactions between inputs and outputs, make generalizations and inferences, discover hidden correlations, patterns, and predictions, and model the highly volatile data and variances needed to predict unusual events (such as fraud detection). As a result, neural networks have the potential to improve decision-making processes in areas such as (SAS Analytics Software and Solutions, n.d.):

- Credit card fraud detection
- Optimization of transport network logistics
- Natural language processing, commonly known as sign and speech recognition
- Medical identification of the disease
- Targeted marketing
- Stock price predictions, currency predictions, options predictions, futures predictions, bankruptcy predictions and bond rating predictions
- Robot management methods
- Quality assurance and process control
- Identification of chemical compounds
- Ecosystem assessment.

4. TensorFlow

TensorFlow is Google's open-source library designed primarily for deep learning applications. Traditional machine learning is also supported. Originally released in 2015, with the first version arriving on February 11, 2017, it has become one of the most widely used frameworks for deep and machine learning projects. TensorFlow was initially designed for large numerical computations, not for deep learning. However, it also proved very valuable for the development of deep learning, so Google made it open source. Tensors are multidimensional arrays of higher dimensions enabled by TensorFlow. Multidimensional arrays are especially useful when dealing with huge amounts of data. TensorFlow is based on data flow graphs with nodes and edges. Since the execution method is in the form of a graph, it is much easier to scale the TensorFlow code across a cluster of computers while using Graphics Processing Unit (Yegulalp, 2022).

4.1. How TensorFlow Works

TensorFlow allows developers to design data flow graphs, i.e. structures that define how data flows through a graph or set of nodes for processing. Each node in the graph represents a mathematical process, and each connection between the nodes is a tensor or multidimensional array of data. TensorFlow applications can be run on almost any suitable target: local system, cloud cluster, iOS and Android devices, and Central

or Graphic Processing Units. If Google's cloud is used, it is possible to accelerate TensorFlow by running it on Google's TensorFlow Processing Unit hardware. The models generated by TensorFlow, on the other hand, can be installed on almost any device and used to offer predictions.

TensorFlow 2.0, launched in October 2019, improved the framework in many areas based on user input, making it easier to use and more efficient. The new API makes it easier to deploy distributed training, and support for TensorFlow Lite enables models to be deployed on a wider range of systems.

The needs of TensorFlow can be divided into two categories: development (training the model) and execution (running the model on different platforms). The model can be trained and used on both CPUs and GPUs. After training the model, it is possible to test it on:

- Desktop computers with Linux, Windows or macOS operating systems)
- Mobile phones with iOS or Android operating systems
- Cloud computing as a service.

The TensorFlow Architecture works in three main steps:

- Data pre-processing is the process of structuring data and reducing it to a single threshold value.
- Model building creates a model for the data.
- Model training and evaluation involves using data to train the model and test it with unknown data.

4.2. Tensorflow Components

TensorFlow consists of two components: tensor and graphs. A tensor is a multidimensional extension of vectors and matrices. Tensors are arrays of data with variable dimensions and ranking that are given as input to a neural network. They form the core of the TensorFlow framework. Tensors are used in all TensorFlow calculations. The tensor can be generated by calculation, or it can be derived from the raw data. The graphs show all operations that take place during training. Each operation is called an operation node and is connected to the others. The graph shows operational nodes and connections between them, but no values are shown (Simplilearn, 2022).

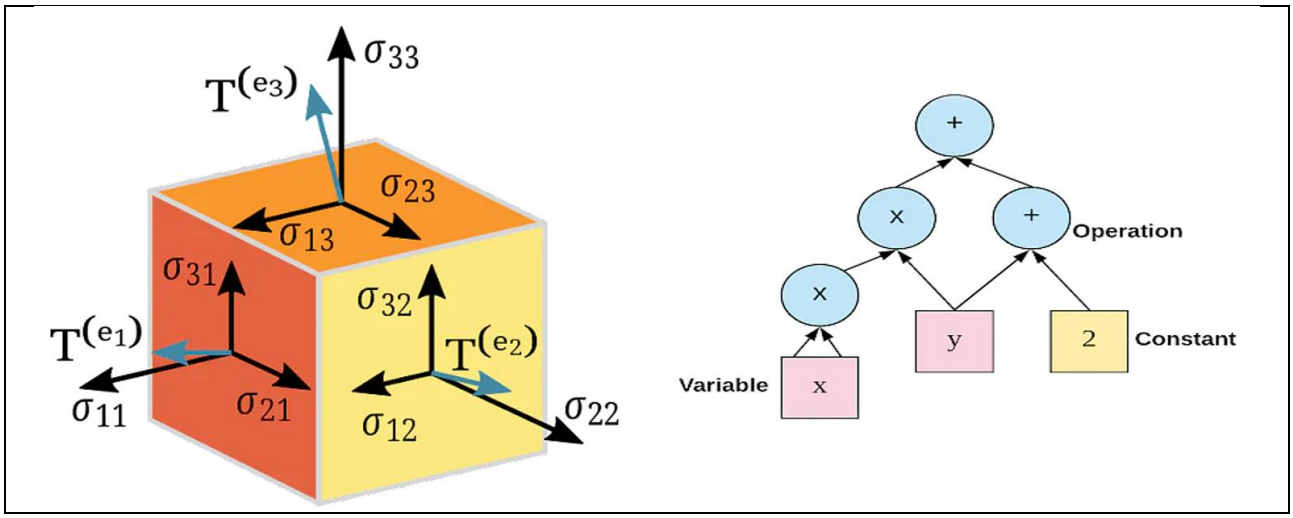

Fig. 1. Traffic sign and its prediction using the developed software solution

4.3. Programming Elements in TensorFlow

TensorFlow programs are based on two fundamental ideas:

- Creating a calculation chart
- Putting the computer chart into use.

To begin with, it is necessary to write the code for preparing the graph. After that, a session is established in which the chart is launched. TensorFlow programming is different from traditional programming. The way data is handled within a program is different from the way a standard programming language is generally handled. A variable must be established for anything that changes in traditional programming. However, with TensorFlow, data can be stored and processed using three different programming elements:

- Variables allow adding new parameters that can be trained on the graph. To declare the variable and initialize it, before starting the graph in the session, we use the command *tf.Variable()*.
- Constants are parameters with values that do not change. To define a constant, we use the command *tf.constant()*.
- Placeholders are used to enter data into a TensorFlow model from outside the model itself. They enable the subsequent assignment of values. The *tf.placeholder()* command is used to define a placeholder.

4.4. Why Use TensorFlow?

TensorFlow is compatible with many programming languages, including Python, JavaScript, C++, and Java. This adaptability is suitable for a wide range of applications in various industries.

Abstraction is the most important feature that TensorFlow provides for machine learning development. Instead of getting bogged down in the specifics of implementing algorithms or figuring out how to connect the output of one function to the input of another, the developer can focus on the overall logic of the application. TensorFlow handles the subtleties behind the scenes. TensorFlow provides additional benefits to developers who need to debug and gain insight into TensorFlow programs. Instead of generating the entire graph as a single opaque object and evaluating it all at once, each graph action can be evaluated and updated sequentially and openly. This so-called "fast execution mode", which was an option in previous versions of TensorFlow, is now the default.

Through an interactive web-based dashboard, the *TensorBoard* visualization package allows you to monitor and characterize graph behaviour. *Tensorboard.dev* (hosted by Google) is a service that allows you to host and share machine learning experiments based on TensorFlow. It is completely free to use, with storage of up to 100 million scalars, 1 GB of tensor data and 1 GB of binary object data.

TensorFlow also gets many benefits thanks to the support of an A-list commercial company at Google. Google has driven the rapid pace of development behind the project and created many significant offerings that make TensorFlow easier to implement and use. The aforementioned silicon TensorFlow Processing Unit for accelerated performance in Google's cloud is just one example.

5. Software System for Traffic Sign Recognition

For the development of a software system for traffic sign recognition, based on TensorFlow technology, programming languages Python and JavaScript were used. Vue.js and Vuetify technologies were also used to create the user interface, and Flask to set up the server for communication.

The project is divided into two phases. In the first phase, the MNIST (Modified National Institute of Standards and Technology Database) data set, i.e. a database of handwritten numbers, was used as a simple example for training different image processing systems.

Three ways of creating the convolution layer of the neural network were used: sequential, functional, and model class. Construction starts with an *Input layer* that takes the input form as an argument, in this case, an image. Since the images are 28x28 pixels in size and have only one channel, because they are in grayscale, this means that the shape must be 28x28x1. As the second layer, a convolutional layer was used using the *Conv2D* command, which accepts arguments such as filters and kernel size, and an activation function. The first parameter represents the number of filters, the second parameter the size of the 3x3 filter, and the rectified linear unit (ReLU) as an activation function. Convolution is a filter applied to an image to extract a feature from it.

Depending on the size of the convolutional layer, randomly generated filters will be applied, and these features are passed to the next layer as input. The rectified linear unit was used for the activation function, which is the most used activation function in deep learning. The function returns 0 if the input is negative, but for any positive input, it returns that value back.

After applying convolutions, pooling is used to reduce the size of the image. *MaxPooling* was used to select the maximum value from a matrix of a certain size, which is important for extracting salient features.

Then the *BatchNormalization layer* takes the output of the previous layer and normalizes the batch so that for each new batch it normalizes the data within that batch. It is used for faster and more stable training of neural networks.

After passing the image through all convolutional layers and pooling layers, the output will be passed to the *Dense layer*. A Dense layer is a simple layer of neurons in which each neuron receives input from all the neurons of the previous layer, hence it is called a Dense layer. The Dense layer is used to classify the image based on the output from the convolutional layers.

Each layer in a neural network contains neurons that calculate a weighted average of their input, and that weighted average is passed through a non-linear function, called an "activation function". The result of this activation function is treated as the output of that neuron. The output of the last layer is considered the output for that image.

In this case, the output layer has 10 neurons with a *softmax* activation function. A *softmax* activation function is used when we have two or more classes. If we have a total of 10 classes, then the number of neurons in the output layer will be 10. Each neuron represents one class. All 10 neurons will return the input image probabilities for the corresponding class. The class with the highest probability will be considered the output for that image.

Tab. 1 shows the impact of different types of models and the number of epochs on training duration and data accuracy. The greater the number of times we go through the data set for training, i.e. the number of epochs, the greater is the accuracy, but also the duration. Accuracy is the most important parameter. However, it is also necessary to pay attention to time because too long time can affect the overall development. The best ratio of processing time and accuracy is achieved by testing itself, and it is necessary to strive for the highest possible accuracy in the shortest possible processing time.

Number of epochs	Type of model	Time	Accuracy
1	Sequential	74 s	93.66 %
	Functional	80 s	93.43 %
	Model class	79 s	93.87 %
2	Sequential	155 s	96.23 %
	Functional	157 s	95.98 %
	Model class	159 s	96.13 %
3	Sequential	234 s	97.11 %
	Functional	239 s	97.00 %
	Model class	237 s	96.69 %

Tab. 1. Time and accuracy depending on model type and number of epochs

The *German Traffic Sign Recognition Benchmark* dataset was used for the second phase of the application development. The set of data was downloaded from the Kaggle website (Kaggle, 2018), and data analysis was performed. After data analysis, it is necessary to prepare sets for training, testing and validation.

Although the sequential method of creating the model in the table above gives the best ratio of time and accuracy, the functional method was used. The reason for this is that the functional mode allows stacking of multiple layers.

Before assembling the model, a data generator was created that takes an image from the data set. Then the built-in TensorFlow *ImageDataGenerator* function is used which will resize the images to basically divide the pixels by 255 which is the largest possible value in the RGB images which will be multiplied by the dataset. Three generators are needed for each part of the data set, and the *flow_from_directory* function is used, which will assume that all the images in the folder belong to the same class. Using the *target_size* function, all images will be resized to 60x60 pixels. The next parameter is *color_mode* which represents the colour of the image. A shuffle parameter which will shuffle the images between epochs was also used, so that the order of the images will not be the same. The generator is created for training, validation and test, the only difference is that a different set of data was used for each generator.

The next step is to prepare sets for training and validation, which will divide all images into two folders, one for training and the other for validation. The training map will use 90%, while the validation map will use 10% of all images. After compiling the model and dividing the data, the evaluation of the test and validation data set was implemented. An accuracy of 98.852% was achieved for the validation test, while the accuracy of the final model or test model is 98.75%, which is a very good accuracy.

This completes the complete logic and programming code needed to create a neural network for traffic sign recognition. The creation of the user interface was done in Vue.js, and HTML, CSS and Vuetify were also used. The main part of the software system is represented by background animations, a field for selecting an image, and a button for starting recognition. It is necessary for the user to first select an image with a specific traffic sign so that the software can recognize it.

After selecting an image, the user presses the recognition button to start the *sendImage* function, which sends the image to the server via a POST request. The server accepts the image and calls the *predict* function, which loads the models and makes a prediction over the image. The prediction is made on an already prepared and trained set of data, and after it is completed, it returns the data in the form of text. If the prediction was successful, it will return a text describing the traffic sign, and if not, it will return a text saying the prediction failed. The selected image, which represents a traffic sign, and its prediction using the developed software solution for traffic sign recognition are shown in Fig. 2.

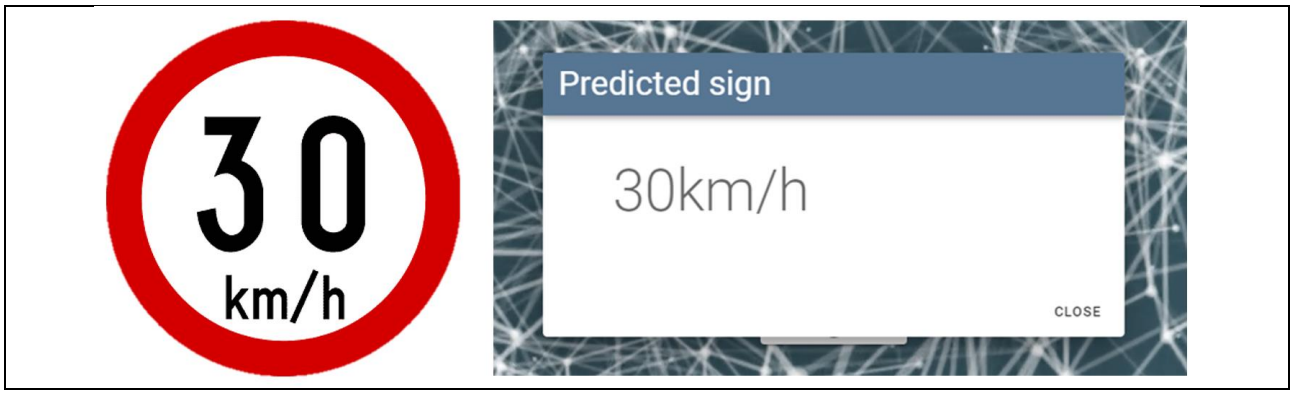

Fig. 2. Traffic sign and its prediction using the developed software solution

6. Conclusion

The possibilities of artificial intelligence have no limits, and they have the ability to shape the future of the world. It already serves as the basis behind the development of technologies such as image recognition, biometrics, and robots, and will continue to do so in the future. In a fraction of the time that the human needs for that task, computers using artificial intelligence can tap into vast amounts of data and use their intelligence to make the best decisions.

The main advantages of artificial intelligence are reducing human error, facilitating faster decision-making, and offering constant availability, but they are also the main cause of unemployment and require higher overall costs. The work of artificial intelligence based on image prediction has a very wide application in all aspects of the industry. All things considered, the human perspective will determine the pros and cons of artificial intelligence. Artificial intelligence will improve the way we think and explore the unknown. As is often said, necessity is the mother of all inventions, and this also applies to artificial intelligence. Things will move so fast in the near future that we will witness significant technological advances. For good or bad, artificial intelligence is here to stay.

7. References

Burns, E. (2021). What is a Neural Network? Explanation and Examples, Available from: https://www.techtarget.com/searchenterpriseai/definition/neural-network, Accessed on: 2021-03-02

Burns, E. (2022). What is Artificial Intelligence (AI)?, Available from: https://www.techtarget.com/searchenterpriseai/definition/AI-Artificial-Intelligence, Accessed on: 2022-02-10

French, R. M. (2000). The Turing Test: The First 50 Years. Trends in Cognitive Sciences, Vol. 4, No. 3, pp. 115-122.

Gutai, A; Havzi, S.; Stefanovic, D.; Anderla, A. & Sladojevic, S. (2021). Optical Character Recognition Methods For Number Plate Recognition: A Systematic Literature Review, Proceedings of the 32nd DAAAM International Symposium, pp. 0692-0700, B. Katalinic (Ed.), Published by DAAAM International, ISBN 978-3-902734-33-4, ISSN 1726-9679, Vienna, Austria

IBM Cloud Education, (2020). Neural Networks, Available from: https://www.ibm.com/cloud/learn/neural-networks, Accessed on: 2022-08-17

Kaggle, (2018). GTSRB - German Traffic Sign Recognition Benchmark: Multi-Class, Single-Image Classification Challenge, Available from: https://www.kaggle.com/datasets/meowmeowmeowmeowmeow/gtsrb-german-traffic-sign, Accessed on 2022-08-19

Simplilearn, (2022). What is Tensorflow, Available from: https://www.simplilearn.com/tutorials/deep-learning-tutorial/what-is-tensorflowc, Accessed on: 2022-07-24

SAS Analytics Software and Solutions, (n.d.). Artificial Neural Networks: What They Are & Why They Matter, Available from: https://www.sas.com/en_sa/insights/analytics/neural-networks.html, Accessed on: 2022-09-02

Yegulalp, S. (2022). What is TensorFlow? The Machine Learning Library Explained, Available from: https://www.infoworld.com/article/3278008/what-is-tensorflow-the-machine-learning-library-explained.html, Accessed on: 2022-06-03