GESTURE RECOGNITION USING A NEURAL NETWORK IN REAL TIME

Author: Anhelina Bohdanova
Advisors: Oleksandr Mazurets, Olena Sobko
Khmelnytskyi National University (Ukraine)

Abstract. *The aim of this work is to develop and implement a method for gesture recognition. The main functions of the application are capturing images from a web camera, memorizing and recognizing user gestures, and training the created model. Javascript programming language and TensorFlow.js, Jest, Bootstrap, Three.js libraries were used for development. The developed neural network is designed to recognize user gestures, namely recognizing digits formed by hand gestures. The practical use of the developed neural network is determined by capturing the current user's gestures in real time from the video stream, searching for the corresponding gesture of the digit and facilitating user interaction when working with the computer, which is especially important for people with limited abilities.*

Keywords: *Neural network, gesture recognition, digit recognition, real-time recognition.*

I. INTRODUCTION

Image recognition is the process of analyzing graphical data in real-time. That is, any device or system capable of recognizing graphical images can be programmed to automatically respond. Such devices/systems can perform actions without human involvement.

The ability to build such systems is one of the major achievements of the last decade. Nowadays, more and more human spheres are integrating such systems into their processes to increase efficiency or reduce time costs, e.g. medicine or gaming industry.

Techniques for recognizing human posture using sign language are tested for finger and hand position to understand problems in recognizing human gestures. In practice, segmentation of hands and distinguishing gestures from each other are important for achieving higher accuracy. Different models of alphabet and digit classification are used for recognizing sign language. For example, in (Hussain, 1999) an Adaptive Neuro-Fuzzy Inference System (ANFIS) is used for recognizing Arabic sign language. In this proposed technology, color gloves are used to avoid segmentation problems and this helps the system to work towards a good result. Handouyahia is an International Sign Language Recognition System (ISL). The developers used a Neural Network (NN) to study the alphabet. NN is used for such a purpose as recognition as it can easily be trained and trained with functions developed for sign language. Another approach includes the Fourier Descriptor (FD) which is used by Malassiotis & Strintzis for three-dimensional gesture recognition demonstrated by the user's hands. In their system, they used hand orientation and silhouettes for

object recognition. Similarly, Likar and Siroani in 2002 used Fourier coefficients to represent the shape of the hand in their system, which allowed them to analyze hand gestures for recognition. Freiman and Rot in 1994 used an orientation histogram to classify sign symbols, so large pedagogical data is used to solve the orientation problem and to avoid incorrect classification between symbols [1].

With the development of technologies, the medical field is rapidly evolving, and gesture recognition technologies are used in various medical fields, both as assistants to surgeons during complex operations requiring high precision, and for better interaction and creating a more comfortable life for people with disabilities.

Gesture recognition is especially necessary for people with vision and hearing impairments, as well as various diseases related to joints and muscles. Thus, people with poor vision have great difficulty working with a computer, people with hearing problems have problems with socialization and proper integration into society. People with joint problems often experience pain that complicates work with a computer, preparation for tests and exams, writing projects and theses.

The goal of this work is to develop a software application for real-time user gesture recognition using Javascript language, Jest library for testing, Tensorflow.js library, Three.js library, Bootstrap library, which performs the following functions:

- Capturing video images in real-time.
- Memorizing gestures.
- Recognizing gestures.
- Training a neural network model for further gesture guessing.

II. LITERATURE ANALYSIS

Working with images is an important area of Deep Learning technology application. All images from cameras around the world form a library of unstructured data. Using neural networks, machine learning and artificial intelligence, these data are structured and used for various tasks: household, social, professional and governmental, including security.

The basis of all video surveillance architectures is image recognition (object) as the first phase of analysis. Then, using machine learning, AI can recognize actions and classify them.

For image recognition, a neural network must be pre-trained on data. This is similar to the neural connections in the human brain – we have certain knowledge, see an object, analyze it and identify it.

Neural networks are demanding in terms of dataset size and quality on which they are trained. Datasets can be downloaded from open sources or collected independently. Several different AI neural network architectures are distinguished, including neural networks for image recognition:

- Multilayer Perceptron.
- Recurrent Neural Network.
- Recursive Neural Network.
- Long Short-Term Memory (LSTM).
- Convolutional Neural Network (CNN)

For the course project, a convolutional neural network (CNN) architecture was chosen to achieve the goal. CNN reduces the memory requirement for storing information simultaneously, making it better at recognizing high-quality images, repeating objects, edges, texture fragments, etc. This neural network is used in various fields to solve various tasks, such as:

- Image and signal classification. Widely used for MRI diagnostics, disease or symptom classification, in agribusiness neural network helps to recognize images obtained via satellite and further predict crop yield of a particular location.
- Object recognition. Used in unmanned vehicles, video surveillance, "smart home" systems.
- Recognition of a specific equipment brand.
- Three-dimensional facial reconstruction from photographs.
- Face recognition of people. This technology is relevant for the police work. Namely during the full-scale Russian invasion in Ukraine such technology helps to identify russians who have committed war crimes [2].

To train a ZNM model, the following sequence of steps is used:

1. Encode input data as OH-vectors.
2. Pass data to a layer to simplify 2-dimensional data into 1-dimensional.
3. Pass data to a layer with ReLU activation function.
4. Pass data to a layer with Softmax activation function.
5. Optimize algorithm in neural network training using Adam optimizer and optimize loss using categoricalCrossentropy [3].

Neural networks are widely used by companies such as Google, Amazon, Apple, as well as smaller, lesser-known ones. Each company or developer seeks the best ways to overcome and solve the task set before them.

In 2017, Korean company Macron presented its FingerTalk gesture recognition application. To launch the gesture recognition program, the user simply raises their arm in the air, which leads to the automatic appearance of a certain object on the user's arm. Then this object can be thrown or moved in other ways.

FingerTalk is equipped with not only traditional computer vision technology, but also deep learning technology. Based on optimization for mobile devices, it was developed to process gestures even faster on smartphones.

This application captures images in real time and displays them on the smartphone screen. The advantages of this application include the absence of the need for a 3D camera or additional sensors, as well as the ability to "interact" with other objects. Another advantage of the application is that gestures are recognized without the use of auxiliary devices, such as special gloves to control the device, making the application accessible to different segments of the population [4].

Another example is Terabee, a French electronics company, which created an app of the same name as an example of gesture recognition. It is not only a software product, but also a system of cameras, sensors and detectors for recognizing gestures and their practical application.

For people with certain limitations, who find it difficult to move constantly or

temporarily, or those who have difficulty manipulating small objects such as a smartphone or remote control, the app allows them to control various devices with minimal effort. Advantages include practical application of the product [5].

Recognition of gestures is widely used to create prostheses for people who have lost limbs. American scientists have applied AI to ensure high accuracy of control of a bionic hand.

When a patient with a bionic hand connected to peripheral nerves wants to move it, a signal from the brain reaches the same peripheral nerves, where it is decoded. The decoder's task is to correctly interpret the signal in real time and transmit it to the prosthesis as a command to perform the desired action. This "translation" of the signal from the nerves to the prosthesis requires processing a large amount of information, so it is not always done without errors, and AI was used to solve this problem.

Experiments have shown that the technology has certain drawbacks and works unstably. One of the participants in the experiment was even able to play a video game thanks to this development. However, the bionic hand responds to the movements of the fingers of the hand, and best of all – to the movements of the thumb and index finger. The technology will still go through stages of improvement and has great prospects [6].

Considering the great potential of convolutional neural networks for solving the task of image recognition, it was decided to use this architecture of neural network for real-time gesture recognition.

III. OBJECT, SUBJECT, AND METHODS OF RESEARCH

To recognize gestures in real time, this work decided to use such a neural network architecture as CNN – convolutional neural network, namely the MobileNet model presented by the TensorFlow.js library.

3.1. MobileNet model

MobileNet is a model designed for use in mobile applications, and it is the first mobile vision model from TensorFlow.

MobileNet uses depthwise separable convolutions, which significantly reduces the number of parameters compared to networks with regular convolutions of the same depth. This leads to the use of a simplified version of deep neural networks, yet equally effective in visual image recognition [7].

Figure 1 shows the Mobilenet model based on real-time neural network gesture recognition.

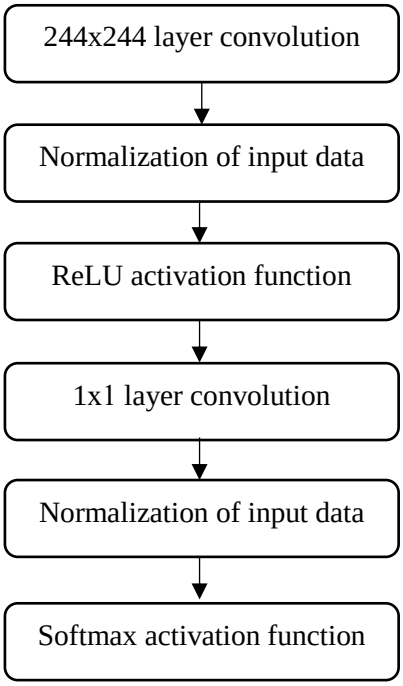

Fig.1. Mobilenet model based on real-time neural network gesture recognition

Folding with a separation depth consists of two operations:

– Deep folding.
– Point folding.

Depthwise convolution originated from the idea that depth and spatial dimension of the filter can be separated – hence the name separable. In this project, an image captured from a web camera in real-time of size 244x244 is fed into the model, thus there are 244x244 input channels. Correspondingly, the same number of output channels as input channels, i.e. 244 channels in this case.

Pointwise convolution is a convolution with channel size 1x1, thus it combines the layers created by depthwise convolution.

After receiving real-time data – images, it is necessary to normalize the obtained data so that the model can further work with them. For this, the image normalization function provided by the TensorFlow.js library is used.

After the model receives all the input data, a training stage is needed for it to be able to perform gesture recognition.

To do this, the normalized input data must first be encoded as an OH-vector. An OH-vector is a unitary code, a group of bits, where the only allowed combinations of values are those in which only one bit is set to (1) and all the others are turned off (0). These vectors are used as target labels during model training.

Then the input data must be sequentially passed through several internal layers, such as Flatten and Dense layers. The RELU function is used as the activation function of the Dense layer. This is one of the most popular functions for deep learning. The logic of the function is as follows: the RELU function returns 0 if it receives a negative argument and the same number if it receives a positive argument.

The RELU function can be represented by the following formula [8]:

$$f(z) = \max(0, z),$$

where z is the argument passed to the function – the input vector.

The next step is to pass the data back to the Dense layer, but with the SOFTMAX activation function, which transforms the vector z of dimension K into a vector σ of the same dimension, where each coordinate σi of the obtained vector is represented by a real number in the interval [0, 1] and the sum of the coordinates is equal to 1. Coordinates σi are calculated as follows [9]:

$$\sigma(z)_i = \frac{e^{z_i}}{\sum_{k=1}^{K} e^{z_k}},$$

Given vector z, with dimension K, and vector σ of the same dimension, the coordinates σ_i of the obtained vector are interpreted as the probability that the object belongs to class i. Vector-column z is calculated as follows [9]:

$$z = w^T x - \theta,$$

where the vector x is a column vector of object features of dimension M × 1, w^T is the transposed matrix of object feature weights of dimension K × M, θ is a column vector of threshold values of dimension K × 1, K is the number of object classes, and M is the number of object features.

Adam Optimizer [10] is used as the optimizer:

$$w_{t+1} = w_t - \alpha w_t,$$

$$m_t = \beta w_{t-1} + (1 - \beta)\left[\frac{\delta L}{\delta w_t}\right],$$

where m_t is the set of gradients, initially m_t = 0; m_{t-1} is the set of gradients at time t-1; w_t is the weights at time t; w_{t+1} is the weights at time t+1; α_t is the learning rate at time t; δL is the derivative of the loss function; δw_t is the derivative of the weights at time t; β is the parameter of the moving average (const, 0.9).

To determine the error between expected output data and actual data, the categoricalCrossentropy function is used, which returns a value in the form of an OH-vector.

3.2. The information structure of the system

There are two main objects: user and neural network with which the user interacts. To enable the neural network to recognize user gestures in real time, preparatory steps are first taken, namely – user camera settings, user permission to turn on the camera for further work.

Considering the work of neural networks and the need to train the model, the user must collect data, i.e. images captured by the camera, to enable model training. For this, the user must raise his hand and depict a gesture, in this case a number from 0 to 5, and press the corresponding button with the number of the gesture the user is demonstrating. After clicking the button, the model "remembers" the image and can create a data sample for further model training. The system takes 10 shots after clicking the corresponding button. Of course, the more shots the user takes, the better the neural network will be able to recognize user gestures.

If the user forgot to perform the above-described steps, he will see a message

that the model cannot be trained, as it did not receive the sample data it needs for training. After that, the user will still be able to perform the necessary steps for further work.

While capturing images, the model remembers and stores them. Then, when the user enters the required gestures, the model receives the sample, and it is possible to start training the model to recognize gestures. For this, the user needs to press the "Start training" button. After that, the user can show gestures to the camera, and the neural network will make its predictions, "guessing" the user's gestures.

To finish "predicting" gestures, press the "Stop predicting" button.

In order to clearly visualize the course of events and plan the main methods to ensure the uninterrupted operation of the program and its logic, a sequence diagram was created which is represented in Fig. 2.

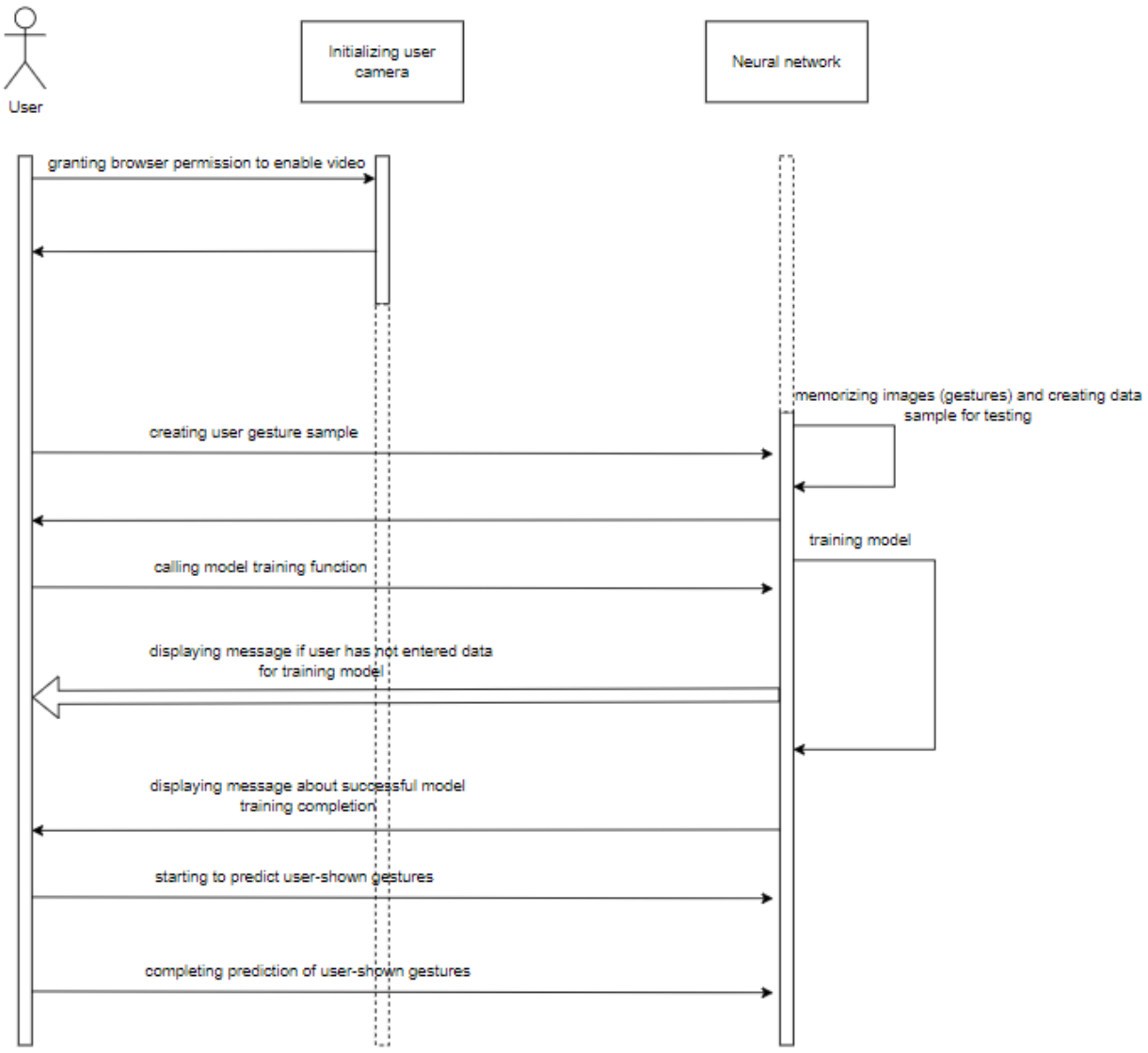

Fig. 2. Sequence diagram of the program

Understanding the application's overall information structure is one of the main requirements for designing a logical, sequential software product, avoiding unnecessary resource, time or redundant code usage which makes the software solution less understandable and requires more support for the written code.

IV. RESULTS

According to the set goal of the work, based on the previously created structure of the application, a function diagram was constructed (Figure 3).

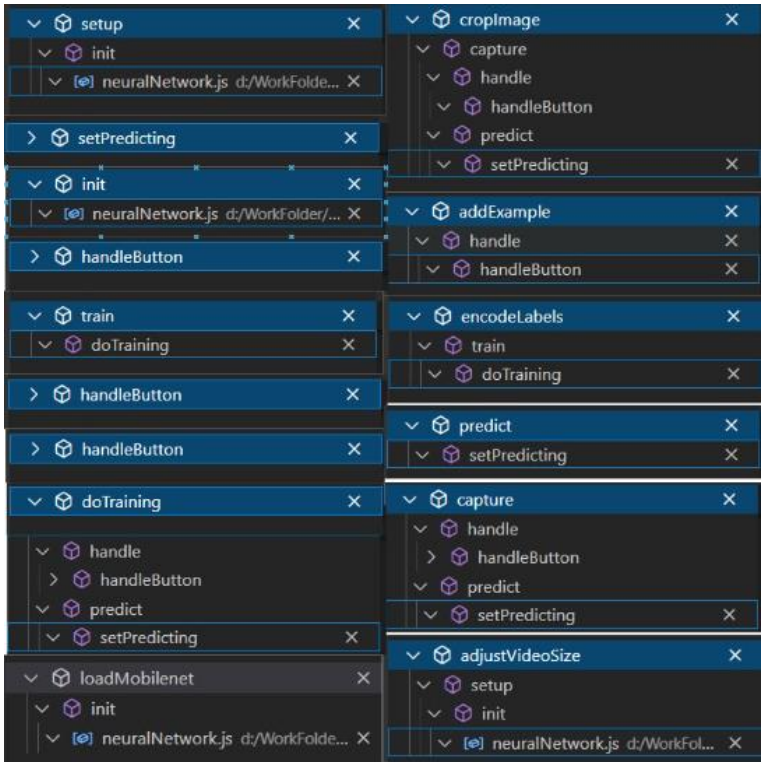

Fig. 3. Function diagram of the program

Each of the used functions has its own purpose and functionality. Some of them perform the logic of the neural network directly, and some are, so to speak, auxiliary functions that help to provide conditions for further work of the neural network.

To ensure proper functioning of the neural network, the user's camera needs to be configured and its size adjusted for correct display on the monitor screen. To do this, a function called adjustVideoSize is written, which takes the video's height and width as arguments. If the video width is greater than the height, the function allows adjusting its width relative to the height and vice versa.

To display the video stream in the corresponding window, the setup function is used. This function receives the video stream from the web camera and also adds a listener to the video element. This listener waits for the video to load and then calls the adjustVideoSize function described above.

The function loadMobilenet downloads the Mobilenet model to create a neural network and train it to recognize user gestures. The function cropImage receives an image from the webcam. It will then be called as a helper function to better process the image from the webcam and train the neural network.

The next function, called capture, gets an instant snapshot from the webcam and normalizes the data of the received image using the cropImage helper function mentioned above. When the network receives the necessary data to start training the

model, the first step is to convert each image into an OH-vector. The encodeLabels function performs this task using the deep learning library Tensorflow.js, which provides flexible tools and methods for working with neural networks.

The training of the neural network is helped by the train function. It consists of three layers: a Flatten layer that takes the input data and helps to flatten it into a one-dimensional vector, a Dense layer with the relu activation function, and a third Dense layer with the softmax activation function, which returns the probability that the gesture shown by the user matches the prediction of the neural network. The AdamOptimizer is used as an optimizer, and the categoricalCrossentropy function is used to determine the error between the expected output data and the real data, which returns a value in the form of an OH-vector. The addExample function receives an image that is passed as a video stream and adds it to the training set for further model training.

To start training the model, the handleButton function must be called, which receives the corresponding element on which the user clicked by the unique element identifier – the button. After the button is pressed, the model will receive 10 samples of the image and add them to the training set. The predict function is used to predict user gestures. It captures an image from the webcam, passes it through the MobileNet model, and then returns the result, which is then used in predicting the current user's gestures.

There are also several helper functions such as setPredicting, which changes the "prediction" state and is used in the predict function, as well as the main init function, which launches the setup and loadMobilenet functions.

For testing the application, several unit tests were created using the Jest library and several functional tests.

Every app must pass certain tests before finding its user. These tests determine the readiness of the created application for release or further testing with the participation of volunteer users. The testing of this program was successfully completed.

The created application is quite easy to use. There are text hints and even if the user performs an unnecessary action or forgets to do something, they will see a pop-up message about what they need to do.

To start using the application, it needs to be launched, then the user will see the program interface.

To turn on the camera, the browser needs to be allowed to use the camera by pressing the "Allow" button.

When the video is loaded, you can start demonstrating gestures. To do this, you need to show a certain gesture, in this case a number from 0 to 5, and press the corresponding button. This step can be done once to teach the model to recognize one number, or for each number. The number of created images will be displayed to the right of the button with the corresponding number (Figure 4).

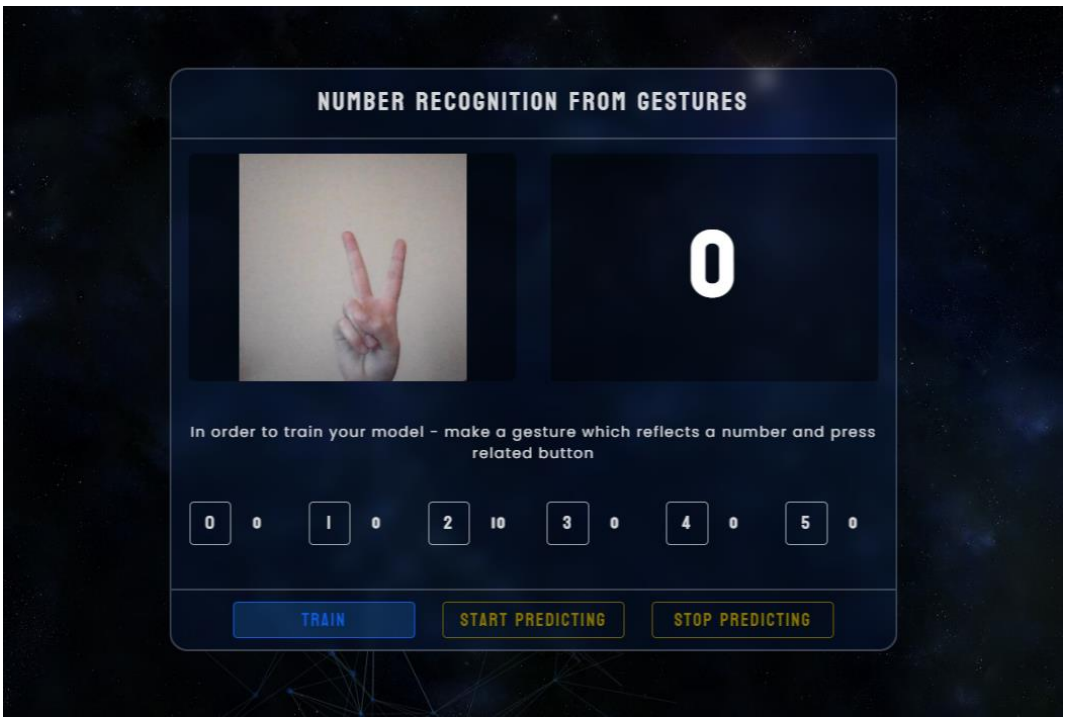

Fig. 4. Gesture demonstration for further modeling training

Once the user has finished demonstrating the gestures, they must press the "Train" button to start the model's training for recognizing the user's gestures in subsequent demonstrations and wait for the message that the model is ready to use.

Now you can start using the gesture prediction app. To do this, press the "Start predicting" button, show the gesture, and the neural network will "guess" the user's gesture, which will be displayed to the right of the user's camera (Figure 5).

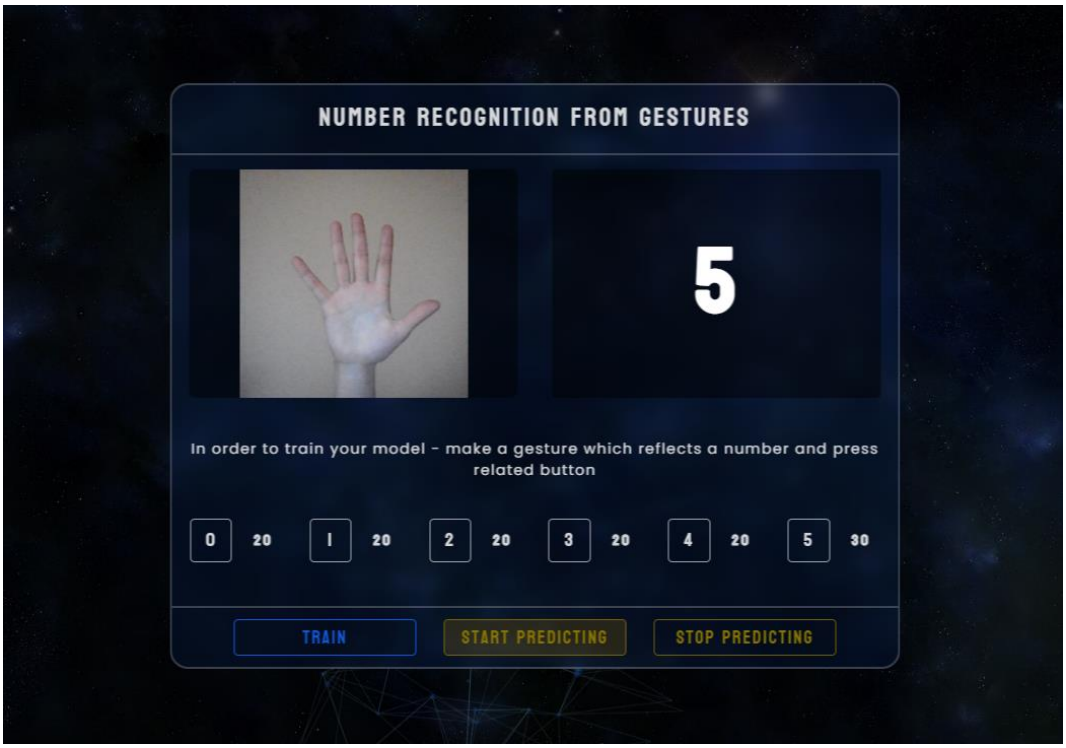

Fig. 5. Result of gesture recognition by a neural network.

To complete the program, press the "Stop predicting" button.

In this way, the real-time gesture recognition method fully fulfills its purpose. The implemented software application is easy to use and contains hints for users.

V. CONCLUSIONS

The purpose of the work was to develop and implement a method for real-time gesture recognition. For the development of the system, the programming language JavaScript was used, as well as the TensorFlow.js, Jest, Bootstrap, and Three.js libraries. As a result of the course project, the theory of artificial neural networks was studied, various types of architectures for solving various problems and various algorithms for creating models using artificial intelligence were considered. In practice, one of the types of artificial neural network architectures was used to solve the task of recognizing gestures in real time.

The use of artificial neural networks is gaining momentum, and this is not surprising, since they help to solve a variety of tasks and problems and make life easier for people. These approaches have been successfully implemented in various fields, such as cyber-policing, automotive industry, medicine.

Technology really has great importance not only in scientific terms, but also for practical application, since neural networks help, in particular, people with various diseases, such as Parkinson's disease, speech or hearing impairments, tunnel syndrome, to facilitate their interaction with the computer, to conduct effective training in schools and universities, and, of course, to socialize.

So, these are only the first steps to make life as easy as possible for people and make it bright, despite certain restrictions.

VI. REFERENCES

1. Zoran, J. (2020, December 5). «Я тебе (не) чую». Як в Україні живеться людям із порушеннями слуху. Lyuk. «Я тебе (не) чую». Як в Україні живеться людям із порушеннями слуху (lyuk.media)

2. Evergreen. (2020, February 1). https://evergreens.com.ua/ua/articles/cnn.html/

3. Youtube. Convolutional Neural Network. https://www.youtube.com/watch?v=HMcx-zY8JSg

4. Cision. (2017, October 24). Macron Releases Hand Gesture Recognition app "FingerTalk". https://www.prnewswire.com/news-releases/macron-releases-hand-gesture-recognition-app-fingertalk-300541991.html#:~:text=FingerTalk%20is%20a%20newly%20released,object%20in%20the%20user's%20palm/

5. Terabee. Custom designed gesture recognition applications. https://www.terabee.com/custom-designed-gesture-recognition-applications/

6. Catalina Markush. (2022, April 19). Штучний інтелект забезпечив високу точність керування біонічною рукою. Nauka.ua. https://nauka.ua/news/shtuchnij-intelekt-pidvishchiv-tochnist-keruvannya-bionichnoyu-rukoyu/

7. Pujara, A. (2020, July 4). Image Classification with MobileNet.

Medium. MobileNet Convolutional neural network Machine Learning Algorithms | Analytics Vidhya (medium.com)

8. Timchenko A., Skachkov V. Функція активації нейрона relu. Функція активації нейрона ReLU (ztu.edu.ua)

9. Wikipedia. Softmax. https://uk.wikipedia.org/wiki/Softmax/

10. GeeksforGeeks. (2020, October 24). Intuition of Adam Optimizer. https://www.geeksforgeeks.org/intuition-of-adam-optimizer/